Predictive Masking for Semi-Supervised Graph Contrastive Learning

Yufei Jin
Dept. of Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, FL 33431, USA
yjin2021@fau.edu

Xingquan Zhu
Dept. of Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, FL 33431, USA
xzhu3@fau.edu

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020970

***Abstract*—Graph Contrastive Learning (GCL) has recently emerged to leverage contrastive loss as a pseudo-supervision signal for self-supervised learning. In order to introduce contrastive learning loss to graphs, existing GCL methods mostly focus on leveraging network topology or node similarity to classify a pair of nodes as same/different node pairs or close/distant node pairs. In this paper, we propose a semi-supervised graph contrastive learning framework, pmGCL, leveraging GCL to augment the performance of a classifier through a predictive masking approach. Specifically, a classifier is trained using a small number of labeled nodes to predict node labels. The label prediction results are then transformed into a binary prediction of whether two nodes have the same label or not for all node pairs. The converted result, serving as a binary masking matrix, will help the succeeding GCL learning to learn to pull nodes likely belonging to the same class to be closer and push the ones belonging to different classes to be further away from each other. Experiments and comparisons, with respect to different benchmark networks and label percentages, show that pmGCL consistently outperforms rival graph convolution neural network (GCN) and GCL baseline with a simple constraint posed on the problem.**

***Index Terms*—Graph contrastive learning, graph convolution networks, predictive masking, semi-supervised Learning**

I. Introduction

Graph Contrastive Learning (GCL) [1] [2], a self-supervised graph learning approach, has recently drawn significant attention, due to its superiority in not requiring supervised label information in the network for learning. The strength of GCL stems from its unique design of creating contrastive learning tasks without requiring supervision. Following the first GCL method, Deep Graph Infomax (DGI), which focuses on contrasting local *vs.* global views [3] [4], many researches now propose to contrast same *vs.* different nodes by creating noisy versions of original data points. The main idea is to learn representations that pull similar data closer and push different data away. After learning the encoder, a downstream task such as a node classification model can be trained using learned representations.

One of the essential challenges in GCL learning is that the number of negative node pairs is far more than positive pairs. Early methods, such as GRACE [5], use all node pairs created from different nodes as negative pairs. The idea is simple but purely heuristic because many negative pairs are unnecessary. Later research proposes to extract semantic information by leveraging similarity scores between nodes to estimate the probability of whether two nodes may have the same label or not. As a result, the estimated scores can help select a subset of hard negative pairs and use scores for constructing soft contrastive loss. [6] [7] [8] [9].

For semi-supervised learning scenarios, when labeled nodes are provided, for graph data, graph convolution neural network can be used with cross-entropy loss to train the model [10]. In [11], a method is proposed to use provided labels with label propagation [12] along with GCN [13] [11]. In [14], an uncertainty or probability-based GNN method is proposed to utilize the labeled nodes for estimating multidimensional uncertainty and leverage them for the node classification. Another work that has a similar idea to ours is [15] which applies an unsupervised learning scheme to the deep learning architecture layers to simultaneously train both goals, where our works transform the prediction result to an input that can be utilized by unsupervised learning for further training. Another important family of semi-supervised learning methods includes cluster kernels [16], TSVM [17], Laplacian SVM [18] that utilize the manifold assumption of the unlabeled data and impose regularization together with supervised loss.

To date, most of the existing semi-supervised methods use clustering or manifold regularization along with a supervised loss to jointly train the model. Our method is different from these methods in that we intend to use the results of the classifier trained from the labeled node to guide or enhance graph contrastive learning instead of directly adding the contrastive learning goal together with the training architecture.

The existing semi-supervised GCL method directly uses given labels to create node pairs. [19] uses labeled nodes to create positive pairs and modify contrastive loss using the mean of positive pairs for each node. This creates a way to unify contrastive learning with supervised labeling knowledge but provides only nodes with labels. In most cases, the number of labeled nodes is very few. As a result, it will limit the number of positive pairs for effective contrastive learning.

To further utilize the label information, we propose, pmGCL, a predictive masking-based Graph Contrastive Learn-

TABLE I
COMPARISON OF DIFFERENT MASKING METHODS

Masking Methods	Learning Type	Node identity	Label	Constraints
Contrastive Masking [5]	Self Supervised	✓	None	None
Supervised Contrastive Masking [19]	Semi-Supervised	✓	✓	None
Predictive Masking	Semi-Supervised	✓	✓	✓

ing method. Our theme is to estimate node labels and create binary relation masks to support contrastive learning. Another perspective of looking at our scheme is to enhance the performance of a classifier with a contrastive learning model.

Table I summarizes the main differences between our proposed masking method and existing solutions. Compared with existing methods like supervised contrastive learning that directly use given labels, our method further extracts information from given labels by training a classifier and using it to predict all the node pair relations. Our method then uses the relationship between each pair of nodes instead of directly using the predicted node label.

Compared to existing research in the field, our research brings the following three main contributions:

- We propose a new masking scheme for semi-supervised graph contrastive learning;
- We propose to decompose a multi-class prediction task as a binary mask prediction task (where the latter often has a higher prediction accuracy), and therefore can boost graph contrastive learning;
- Experiments and analysis show that our method can extract useful information from a small number of labeled nodes to boost node embedding learning.

II. Problem Definition & Preliminary

$G = (V, E, X, Y_l)$ represents a graph, where $V = \{v_i\}_{i=1,\cdots,n}$ is the set of vertices representing nodes in the graph and $e_{i,j} = (v_i, v_j) \in E$ is an edge capturing relationship between node v_i and v_j. An adjacency matrix A represents topological structures of a graph G, with $A_{i,j} = 1$ if $(v_i, v_j) \in E$ or $A_{i,j} = 0$ otherwise. The feature matrix $X \in \mathbb{R}^{n\times m}$ represent features of all nodes, with $\mathbf{x_i} \in X$ denoting feature vector of node v_i, and each node has m features, *i.e.* $\mathbf{x}_i \in \mathbb{R}^{1\times m}$. The label space of each node is denoted by $\mathcal{Y} \equiv \{c_1, \cdots, c_K\}$, so there are $|\mathcal{Y}| = K$ unique labels/categories for the node label space.

For each node v_i, its true label is denoted by $y_i \in \mathcal{Y}$. For ease of calculation, we denote v_i's label in a one-hot encoded vector as $\mathbf{y}_i \in \mathbb{Z}^{1\times K}$, with j^{th} element corresponding to the class c_j, $y_{i,j} = 1$ if node label is c_j and 0 otherwise. For example, if a node v_i's label is c_2, $\mathbf{y}_i = [0, 1, 0, \cdots, 0]$, with $y_{i,2} = 1$ and $y_{i,j} = 0, \forall j \neq 2$. The predicted label of node v_i is denoted by $\hat{y}_i$ and the one-hot encoded prediction is represented as $\hat{\mathbf{y}}_i \in \mathbb{Z}^{1\times K}$. The predicted probability of a node v_i belonging to class c_j is denoted by $\hat{p}_{i,j}$. A small subset of nodes V_l has labels and the label set is denoted by $Y_l \in \mathbb{Z}^{|V_l|\times K}$. The subset of nodes without labels is denoted by V_u and the ground truth of the label set for V_u is $Y_u \in \mathbb{Z}^{|V_u|\times K}$.

Given a graph G, and a portion of labeled nodes V_l (including their labels Y_l), our goal is to learn models to accurately predict labels Y_u of remaining nodes V_u in the network.

A. GCN and GCL Preliminary

Our method involves both GCN and GCL components. Here we introduce their core concepts and differences. For both GCN classifier and GCL representation learning, a key concept is a graph convolution neural network that involves the following convolution operation [10] for each layer i:

$$H^{i+1} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^i W^i) \tag{1}$$

where H^{i+1} is the output embedding from layer i, σ is an activation function, $\tilde{A} = A + I$,where I is the identity matrix with same size as A and $\tilde{D}$ is the diagonal matrix of $\tilde{A}$. H^1 is feature matrix X, and W^i is the weight matrix of layer i. The classifier GCN_c has an additional dropout layer after $Eq.$ (1) and its output is a predicted label $\in \mathcal{Y}$. The encoder GCN_e has an additional stage to augment the original graph and outputs the latent representation of the node.

III. Proposed Method

A. Motivation

As shown in Table I, existing contrastive masking methods mainly rely on whether a pair of nodes belong to the same node to create contrastive term. This does not allow label information to be accommodated in the contrastive learning process. On the other hand, supervised contrastive masking uses labels to create contrastive loss, leaving the majority of unlabeled nodes not able to be considered in the contrastive learning process. The main motivation of our predictive masking is to leverage limited label information in the network to create masks, such that unlabeled nodes can be modeled during the contrastive learning process.

To explicitly capture node relationships for contrastive learning, we use a Boolean matrix $M \in \mathbb{R}^{n\times n}$ to represent the node relationship between each node pair in terms of their labels. A mask matrix value $M_{i,j} = 1$ if node v_i and node v_j does not belong to the same class, $i.e., y_i \neq y_j$, otherwise, $M_{i,j} = 0$. In self-supervised learning, the label information is not available, and therefore a rough estimation is needed to fill the matrix M. Because mask M provides essential guidance on how the final representation will be learned, a more accurate estimation should be used to substitute rough estimation. Therefore, we propose to use a classifier to predict and generate mask matrix M, using two major steps: (1) create the mask M based on prediction from a base classifier; (2) train encoder to learn node representation Z using mask matrix M, where $Z \in \mathbb{R}^{n\times d}$ is defined as learned representations from encoder with latent space dimension d for all n nodes. $\mathbf{z}_i \in \mathbb{R}^{1\times d}$ is a representation of node v_i.

B. Predictive Masking

GCN classifier, denoted by GCN_c, is chosen as our base classifier due to its great performance in a supervised manner. GCN_c is trained using G and Y_l with cross-entropy loss. Then, $\hat{Y}_u$ is predicted and concatenated with Y_l to form a label matrix of all nodes, which is denoted by $\hat{Y}$.

$$\hat{Y} = \begin{bmatrix} Y_l \\ \hat{Y}_u \end{bmatrix} \tag{2}$$

We store $\hat{Y}$ using one-hot encoding form for later computation of mask M with dimension, *i.e.*, $\hat{Y} \in \mathbb{R}^{n\times|\mathcal{Y}|}$.

After that, mask matrix M is constructed using $\hat{Y}$ by using the formula:

$$M = |\hat{Y} \times \hat{Y}^\top - \mathbf{J}| \tag{3}$$

where $\mathbf{J} \in \mathbb{R}^{n\times n}$ is an all-ones matrix. $\mathbf{M}_i$ represents relationship between node v_i and all other nodes, with $M_{i,j} = 1$ if $\hat{y}_i \neq \hat{y}_j$, or $M_{i,j} = 0$ otherwise.

Please note that $\hat{Y}$ in Eq. (2) concatenates both ground truth labels (for labeled nodes) and predicted labels (for unlabeled nodes). If a node v_i is labeled, its label in $\hat{Y}$ would be its ground truth label, meaning $\hat{y}_i = y_i$. In this way, we can make mask matrix M as accurate as possible to capture pairwise node label relationship.

C. Representation Learning

After obtaining the new mask matrix M, the contrastive learning will use M to enhance the representation learning, through a contrastive learning process. To create different graph views for contrastive learning, two augmented versions of graph views are created by randomly dropping edges and masking features. A shared GCN encoder GCN_e is trained to learn the representation $\mathbf{z}$ of each node for the two views and use a projection header to further project the representations. The learned representation Z is used to compute the loss and further improve GCN_e.

For a specific node v_i, we use ΔV_i^+ to denote the set of nodes having the same label as v_i. Likewise, we use ΔV_i^- to denote the set of nodes whose labels are different from v_i. In the following, we outline the loss functions used to regularize representation learning.

1) Loss Functions: For supervised loss, the cross entropy loss is used and given as:

$$\ell_{CE}(Z) = -\sum_{i=1, v_i\in V_l}^{|V_l|} \sum_{j=1}^{|\mathcal{Y}|} y_{i,j} \cdot \log \hat{p}_{i,j} \tag{4}$$

For each node v_i, its true label is denoted as y_i, we denote its form in one hot encoder for class j as $y_{i,j}$. The predicted probability of class j is denoted by $\hat{p}_{i,j}$.

For self-supervised Learning, the contrastive loss is to consider the same node as positive and different nodes as negative pairs. Methods like hard negative sampling [8] proposed to estimate the hardness of the negative sample pairs and tried to use those true and hard negative samples for the negative loss. We consider this view as a negative mining scheme and the formula is given:

$$\ell_{CL}(Z) = -\frac{1}{2n}\sum_{i=1}^{2n} \log \frac{\mathrm{e}^{\frac{\mathbf{z}_i\cdot\mathbf{z}^+}{\tau}}}{\mathrm{e}^{\frac{\mathbf{z}_i\cdot\mathbf{z}^+}{\tau}} + \frac{q}{h}\sum_{j=1}^{h} \mathrm{e}^{\frac{\mathbf{z}_i\cdot\mathbf{z}_j^-}{\tau}}} \tag{5}$$

where h is the number of hard negative examples to be selected and q is a hyperparameter to control the balance between the hardness and correctness of the selected samples. $\mathbf{z}^+$ is the representation of a node that has a different view from node i and $\mathbf{z}_j^-$ is the representation of node j who has different labels from node i. The intuition of a hard negative example is similar to the idea of support vectors in the Support Vector Machine (SVM) model.

Other works like Supervised Contrastive Learning [19] suggest an SC loss that focuses on positive masks. For each node, it gets the average of all the positive nodes. We consider it a positive mining scheme and the formula is given:

$$\ell_{SC}(Z) = \frac{1}{2n}\sum_{i=1}^{2n} \frac{-1}{|\Delta(V_i^+)|} \sum_{j\in\Delta V_i^+} \log \frac{\mathrm{e}^{\frac{\mathbf{z}_i\cdot\mathbf{z}_j}{\tau}}}{\sum_{a=1,a\neq i}^{a=2n} \mathrm{e}^{\frac{\mathbf{z}_i\cdot\mathbf{z}_a}{\tau}}} \tag{6}$$

where ΔV_i^+ is the set of positive nodes with node i and a is the index of the node. The labeled information is utilized as a more accurate positive set ΔV_i^+.

Using Boolean mask matrix M generated from the previous step, we can create contrastive loss to select the negative pairs and consider the following negative contrastive losses:

$$\ell_{PM}(Z) = \frac{1}{2n}\sum_{i=1}^{2n} -\log \frac{\mathrm{e}^{\frac{\mathbf{z}_i\cdot\mathbf{z}^+}{\tau}}}{\mathrm{e}^{\frac{\mathbf{z}_i\cdot\mathbf{z}^+}{\tau}} + \sum_{a=1,a\neq i}^{a=2n} \mathrm{e}^{\frac{\mathbf{z}_i\cdot\mathbf{z}_a\cdot M_{i,a}}{\tau}}} \tag{7}$$

where $\mathbf{z}^+$ is the representation of the node which has the same index as node v_i, but from a different graph view. $M_{i,a}$ is element of matrix M, specifying node v_i and v_a relationship. Compared to contrastive loss in Eq. (5), our loss is not controlled with hyperparameter q as we use the baseline predictor to produce the whole relationship mask directly. Our loss Eq. (7) is a smooth version of Eq. (5) by putting the mask inside the exponential term and giving 1 to the predicted positive terms or considering it as adding a count of predicted positive terms. We compute the joint loss of our proposed contrastive loss and cross-entropy loss with a hyperparameter λ to regulate. The final loss is computed as:

$$\ell(Z) = \ell_{PM}(Z) + \lambda \cdot \ell_{CE}(Z) \tag{8}$$

Algorithm 1: pmGCL: Predictive Masking GCL Learning

Require:
(1) Graph: $G = (V, E, X, Y_l)$.
(2) L: The numbers of GCN layers.
(3) $Aug(.)$: Augmentation function to create noisy graph views.
(4) GCN architecture: A graph convolution neural network architecture with cross-entropy loss with a set of parameters denoted by W_{GCN}.
(5) GCL architecture: a graph convolution neural network architecture with the contrastive loss with a set of parameters denoted by W_{GCL}.

Ensure:
$[\hat{Y}_u]$: predicted result for all unlabeled nodes

1: $A \leftarrow V \times E$. Create an adjacency matrix from V and E.
2: **while** not convergence **do**
3:   **for** i=1 to L **do**
4:     **if** $i == 1$ **then**
5:       $Z^{(i-1)} \leftarrow X$
6:     **end if**
7:     $Z^i \leftarrow Conv^{(i)}\left(Z^{(i-1)}, A\right)$ using Eq. (1)
8:   **end for**
9:   Back-propagate loss gradient from Z^{L-1},Z^L and Y_l using Eq. (4)
10:   Update predictor parameters W_{GCN} with loss
11: **end while**
12: $\hat{Y}_u \leftarrow \hbar_1(Z^{L-1}, Y_l)$ where result is in one hot encoded form, $\hbar_1()$ is a linear regression model trained with embedding Z^{L-1} and label Y_l
13: $\hat{Y} \leftarrow$ concatenate $(\hat{Y}_u, Y_l)$ using Eq. (2)
14: M $\leftarrow$ mask matrix using $Eq.$ (3) and $\hat{Y}$
15: $G_1 \leftarrow Aug(G)$
16: $G_2 \leftarrow Aug(G)$
17: **while** not convergence **do**
18:   **for** i=1 to L **do**
19:     **if** $i == 1$ **then**
20:       $Z_1^{(i-1)} \leftarrow X_1$
21:       $Z_2^{(i-1)} \leftarrow X_2$
22:     **end if**
23:     $Z_1^i \leftarrow Conv^{(i)}\left(Z_1^{(i-1)}\right)$ using Eq. (1)
24:     $Z_2^i \leftarrow Conv^{(i)}\left(Z_2^{(i-1)}\right)$ using Eq. (1)
25:   **end for**
26:   Back-propagate loss gradient from Z_1^{L-1}, Z_2^{L-1},Z_1^L,Z_2^L, M and Y_l using Eq. (8)
27:   Update encoder parameters W_{GCL} with loss
28: **end while**
29: $\hbar_2 \leftarrow$ Linear regression model trained with embedding Z^{L-1} and label Z_l
30: $\hat{Y}_u \leftarrow \hbar_2(Z^{L-1})$ where result is in one hot encoded form.

D. Algorithm

Algorithm 1 lists the pseudo code of the proposed predictive masking graph contrastive learning process. The algorithm consists of three main phases.

The first phase (from Step 1 to Step 10) is to first learn a graph neural network GCN_c, using input graph $G = (V, E, X, Y_l)$. The result of this phase will learn the embedding feature for each node, and graph G is encoded as a dataset Z^{L-1} with each instance representing a node in G.

The second phase (from Step 11 to Step 14) is the mask generation phase. We first use Z^{L-1} and node labels Y_l to train a classifier $\hbar_1(Z^{L-1}, Y_l)$. The classifier is applied to predict node labels for unlabeled nodes. Given the predicted one-hot encoding result $V \in \mathbb{R}^{n\times|\mathcal{Y}|}$, we can compute our mask M using Eq. (3).

The third phase (from step 15 till the end) is the graph contrastive learning process. We first generate graph augmentations G_1 and G_2 and then learn two embedding features (Z_1^i and Z_2^i) for each node in the network. Step 26 will use graph contrastive loss defined in Eq. (8) to back-propagate gradient for network training. Once the graph neural network is suitably trained, the embedding vectors Z^{L-1} are then used to train another classifier $\hbar_2$ to predict labels of unlabeled nodes in the network.

IV. Experiments

A. Experimental Settings

We use three benchmark graph datasets, as shown in Table IV, in the experiments. To evaluate the performance of the algorithms, we use different portions of labeled samples, and evaluate the algorithm performance on the rest samples. More specifically, for each network, we label $1\%, 2\%, 3\%, 10\%, 20\%$ and 30% of nodes respectively, and test the algorithm performance on the remaining nodes. To examine the effectiveness of our method, we compare it with different baselines and test our experiment with different label rates. Each label rate is repeated 10 times with a random split.

1) Parameter Settings: For each set of labeled portions, we separate 70% of labeled samples for training, and the rest 30% is used for validation. For hyperparameter selection, we follow the same setting used in the original papers. The training epoch number is selected based on the validation set. The added hyperparameter λ is empirically examined from $[0, 0.2, 0.5, 1, 3, 5]$ and we found that λ should be chosen small when the label percentage is small and relatively larger when the label percentage increases. $\lambda = 0.2$ is best for lower labeling rate and up to 1 for a higher labeling rate. For both GCL and GCN architecture, we have two GNN layers with GCN adding an additional dropout layer with 0.5 dropout rate and GCL contains additional parameters including drop edge and drop feature ratio to control augmentation level and temperature τ used in the loss function. The Relu function is used for the activation function. The other parameters for both GCL and GCN include the number of hidden units, decay weight, and initial learning rate. We follow the recommended setting used in the original paper for the selection of these parameters. The early stopping technique is used together with the validation set for early convergence.

2) Baseline Methods: Table II lists the baselines and our proposed method and compares their differences from two perspectives: loss function and label utilization. SCL is a supervised contrastive learning proposed in [19], which is a baseline for comparing whether our method extracts more information from labels. For the GCL method, we choose GRACE [5] as our baseline, because it uses rough estimation without any label information, making it a good baseline for comparison. GCN [10] is selected for baseline to examine

TABLE II
A SUMMARY OF BASELINE AND THE PROPOSED METHODS WITH RESPECT TO LEARNING TYPES, MASK GENERATION, AND LOSS FUNCTION

Method	Learning Type	Mask Generation	Loss Function
GCN [10]	Semi Supervised	NA	CE
GCL [5]	Self Supervised	consider same or different nodes	CL
GCL(with correction)	Semi Supervised	consider the original label information	CL
pmGCL	Semi Supervised	predict relation + label information	CL+CE
pmGCL	Semi Supervised	predict relation + label information	SC
SCL [19]	Semi Supervised	directly leverage label information	SC

TABLE III
DATASETS DESCRIPTION

Datasets	# of nodes	# of edges	# of classes
Cora	2,708	10,556	7
Citeseer	3,327	9,228	6
Pubmed	19,717	88,651	3

whether we can further improve the results from GCN through contrastive learning. For loss functions, CE represents cross-entropy loss, SC represents supervised contrastive loss, and CL is contrastive loss.

- GCN [10] a semi-supervised framework that applies Graph Convolution Neural Network to graph data with cross-entropy loss.
- GCL [5] a self-supervised framework applied Graph Convolution Neural Network with contrastive loss to encode node representation for downstream tasks.
- GCL (with correction) a semi-supervised framework applies and Graph Convolution Neural Network with contrastive loss upon given labels to encode node representation for downstream tasks.
- pmGCL(SC loss) our proposed framework with substitution of supervised contrastive loss.
- SCL [19] a semi-supervised framework applies and Graph Convolution Neural Network with supervised contrastive loss upon given labels to encode node representation for downstream tasks.

B. Results and Analysis

For fairness of comparison, we extract the representation learned from all models and follow the standard procedure to use the latent features with a simple linear regression model for evaluation.

The results show that, in most cases, pmGCL outperforms all other baselines. Substituting SC loss with our predictive mask performs worse than our own loss function. This shows that our proposed loss function is better suitable to our method. Our method with SC loss performs better than direct label information with SC loss when a reasonable number of labels is given, indicating that our predictive mask gets more information from labels. Our method works better than GCN

TABLE IV
MEAN ACCURACY ON THE CORA DATASET

Label Rate	1%	3%	5%	10%	20%	30%
GCN [10]	0.5231	0.7155	0.757	0.8187	**0.8456**	0.8528
GCL [5]	0.6165	0.6464	0.6596	0.7738	0.7624	0.8446
GCL(with correction)	**0.625**	0.6344	0.6436	0.6942	0.732	0.7417
pmGCL	0.5454	**0.7199**	**0.7625**	**0.8196**	0.8426	**0.8578**
pmGCL(SC loss)	0.5287	0.7062	0.7552	0.8183	0.8437	0.8536
SCL [19]	0.5996	0.6356	0.6524	0.7072	0.7234	0.83

TABLE V
MEAN ACCURACY ON THE CITESEER DATASET

Label Rate	1%	3%	5%	10%	20%	30%
GCN [10]	0.4442	0.631	0.6634	0.7061	0.7294	0.7415
GCL [5]	**0.6042**	0.6251	0.6152	0.7077	0.705	0.7145
GCL(with correction)	0.5855	0.6004	0.6142	0.6555	0.675	0.684
pmGCL	0.5123	**0.6529**	**0.68**	**0.7174**	**0.7399**	**0.7492**
pmGCL(SC loss)	0.45	0.622	0.67	0.7119	0.7341	0.742
SCL [19]	0.5702	0.6145	0.6155	0.6525	0.6661	0.73

TABLE VI
MEAN ACCURACY ON THE PUBMED DATASET

Label Rate	1%	3%	5%	10%	20%	30%
GCN [10]	0.8047	0.8305	**0.844**	**0.8464**	**0.8518**	0.8528
GCL [5]	0.8054	0.8102	0.8175	0.8239	0.8333	0.8354
GCL(with correction)	0.8124	0.812	0.8197	0.8258	0.8348	0.8391
pmGCL	**0.8198**	**0.8351**	0.8396	0.8234	0.8421	**0.8663**
SCL [19]	0.8123	0.811	0.8217	0.8283	0.836	0.8406

showing that further improvement can be made by leveraging supervised results from the representation learning perspective.

In Table VI, GCN performs better in many cases. This is because Pubmed set only has $K = 3$ classes, suggesting that the converted mask may be corrupted with higher model prediction probability. For those cases, noise becomes a more influencing factor and reduces the amount of information brought by the prediction. Grace performs better in the extreme fewer label rate showing that rough estimation can work when no label or only fewer labels are given but can be improved greatly with our methods when more labels are given.

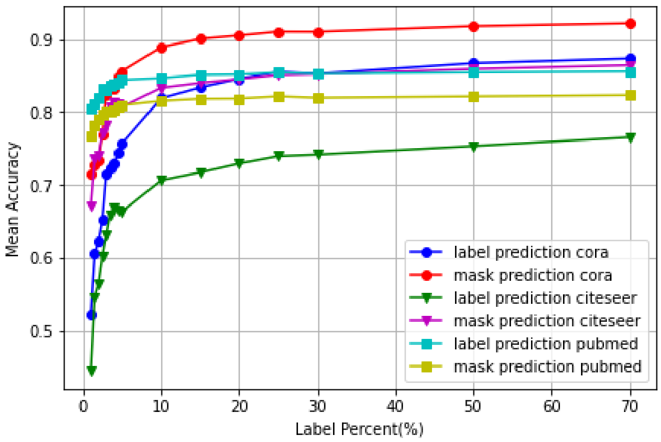

Fig. 1. Mean accuracy comparison between binary relation prediction and label prediction on three datasets. x-axis is the percentage of labeled nodes and y-axis is the mean accuracy of the prediction results. Label prediction means prediction for a multi-class classification problem and mask prediction means prediction for a binary mask problem.

C. Predictive Masking Rationality

In order to explain why predictive masking-based contrastive graph learning can improve learning accuracy, we explicitly compare label prediction accuracy $vs.$ pairwise label mask accuracy. For example, for a network with 10 nodes $v_1, \cdots, v_10$, if we correctly predict labels for 6 nodes, the label prediction accuracy is 60%. Then we use predicted labels to create a mask, check the predicted pairwise label relationships, and report mask prediction accuracy.

Figure 1 shows the empirical study comparing binary masking accuracy *vs.* multi-class prediction accuracy, which confirms that binary masks always have higher accuracy for all tasks except when $K = 3$. This shows that instead of directly relying on given labels to train graph neural networks, one can predict labels for unlabelled nodes and use their pairwise label relationships to boost graph learning.

V. Conclusion

In this paper, we propose a predictive masking contrastive learning framework that uses a classifier to predict node labels and leverages labels to create a binary mask for contrastive learning. The niche of our predictive masking stems from its rationality that instead of direct predicting a multi-class task, we can create a more accurate binary task classifier, and use the results to help improve the embedding learning. Another way of looking at this scheme is to consider the framework as an enhancement for a classifier using a self-supervised objective. By using predictive masks to regulate contrastive loss, experiments and comparisons demonstrate that pmGCL consistently outperforms all common GCN and GCL baseline for semi-supervised graph learning. Noted that in our theorem, we didn't constrain the specific model architecture and data type, it is worthwhile for the future study to explore other prediction models and data types to see if the framework can be generalized.

Acknowledgment

This research is sponsored by US National Science Foundation under grant No. IIS-1763452.

References

[1] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[2] M. Wu, S. Pan, and X. Zhu, "Attraction and repulsion: Unsupervised domain adaptive graph contrastive learning network," *Transactions on Emerging Topics in Computational Intelligence*, vol. 6, no. 5, 2022.

[3] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," 2018. [Online]. Available: https://arxiv.org/abs/1808.06670

[4] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," 2018. [Online]. Available: https://arxiv.org/abs/1809.10341

[5] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, "Deep Graph Contrastive Representation Learning," in *ICML Workshop on Graph Representation Learning and Beyond*, 2020. [Online]. Available: http://arxiv.org/abs/2006.04131

[6] C.-Y. Chuang, J. Robinson, Y.-C. Lin, A. Torralba, and S. Jegelka, "Debiased contrastive learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 8765–8775.

[7] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, "Hard negative mixing for contrastive learning," 2020. [Online]. Available: https://arxiv.org/abs/2010.01028

[8] J. Robinson, C.-Y. Chuang, S. Sra, and S. Jegelka, "Contrastive learning with hard negative samples," 2020. [Online]. Available: https://arxiv.org/abs/2010.04592

[9] Z. Yang, M. Ding, C. Zhou, H. Yang, J. Zhou, and J. Tang, "Understanding negative sampling in graph representation learning," 2020. [Online]. Available: https://arxiv.org/abs/2005.09863

[10] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[11] H. Wang and J. Leskovec, "Combining graph convolutional neural networks and label propagation," *ACM Trans. Inf. Syst.*, vol. 40, no. 4, nov 2021. [Online]. Available: https://doi.org/10.1145/3490478

[12] X. Zhu and Z. Ghahramani, "Learning from labeled and unlabeled data with label propagation," *CMU CALD tech report CMU-CALD-02-107*, 07 2003.

[13] Q. Huang, H. He, A. Singh, S.-N. Lim, and A. R. Benson, "Combining label propagation and simple models out-performs graph neural networks," 2020. [Online]. Available: https://arxiv.org/abs/2010.13993

[14] X. Zhao, F. Chen, S. Hu, and J.-H. Cho, "Uncertainty aware semi-supervised learning on graph data," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 12 827–12 836.

[15] J. Weston, F. Ratle, and R. Collobert, "Deep learning via semi-supervised embedding," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 1168–1175. [Online]. Available: https://doi.org/10.1145/1390156.1390303

[16] O. Chapelle, J. Weston, and B. Schölkopf, "Cluster kernels for semi-supervised learning," in *NIPS*, 2002.

[17] T. Joachims, "Transductive inference for text classification using support vector machines," in *ICML*, 1999.

[18] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, p. 2399–2434, dec 2006.

[19] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," 2020. [Online]. Available: https://arxiv.org/abs/2004.11362

[20] J. Xia, L. Wu, J. Chen, G. Wang, and S. Z. Li, "Debiased graph contrastive learning," 2021. [Online]. Available: https://arxiv.org/abs/2110.02027