Supplementary Material for "Generative Subgraph Contrast for Self-Supervised Graph Representation Learning

Yuehui Han, Le Hui, Haobo Jiang, Jianjun Qian*, and Jin Xie*

Key Lab of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education
Jiangsu Key Lab of Image and Video Understanding for Social Security
PCA Lab, School of Computer Science and Engineering
Nanjing University of Science and Technology
{hanyh, le.hui, jiang.hao.bo, csjqian, csjxie}@njust.edu.cn

1 BFS Based Subgraph Sampling

As shown in the Fig. 1, BFS based sampling method samples subgraph by spreading outward from the center node. Specifically, considering the various size of different neighbors, we first set a target $k(k > 1)$ for the subgraph sampler S. Then, for a specific node i, the subgraph sampler S first samples nodes in the first-order neighborhood. If the target is not reached, it moves to the next-order neighborhood for subsequent sampling until the target is reached. When the sampling goal is achieved, we can obtain the nodes of subgraph around node i. The index of chosen nodes and the nodes embeddings can be denoted as:

$$idx = S(\boldsymbol{A}, k) \tag{1}$$

Next, derive the connection relationship of the subgraph nodes from the adjacency matrix $\boldsymbol{A}$.

$$\boldsymbol{A}_i = \boldsymbol{A}_{idx,idx} \tag{2}$$

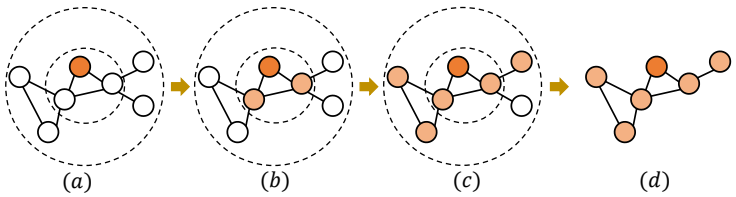


Fig. 1. BFS based subgraph sampling. For a specific node and its neighborhood nodes (a), we first sample nodes in the first-order neighborhood (b). If the target is not reached, move to the next-order neighborhood for subsequent sampling until the target is reached (c). Finally, we obtain the sampled subgraph (d).

* Corresponding authors

2 Datasets Details

The description of different datasets can be seen in the Table. 1.

Table 1. Description of different datasets.

Dataset	Task	Nodes	Edges	Features	Classes	Train/Val/Test
Cora	Transductive	2,708	5,429	1,433	7	0.05 / 0.18 / 0.37
Citeseer	Transductive	3,327	4,732	3,703	6	0.04 / 0.15 / 0.30
Pubmed	Transductive	19,717	44,338	500	3	0.003 / 0.03 / 0.05
Reddit	Inductive	231,443	11,606,919	602	41	0.66 / 0.10 / 0.24
PPI	Inductive	56,944	818,716	50	121	0.79 / 0.11 / 0.10

Transductive learning. We utilize three standard citation networks, Cora, Citeseer, and Pubmed [10], to predict article subject categories. In all of these datasets, graphs are constructed by the nodes and edges that nodes correspond to articles and edges to (undirected) citations. Every node has a bag-of-words representation and a class label.

Inductive learning on large graphs. For the inductive learning on large graphs, we use a large scale social network, named Reddit. This dataset contains 231443 nodes and 11606919 edges, which is preprocessed by [3]. In the dataset, nodes correspond to the posts, and the edge exists if the same user has commend on the both posts. Node features are composed of the post title, content, and connect, along with other metrics such as post score and the number of comments. Following the inductive setup of [3,11], we use the posts made in the first 20 days for training, while the remaining for testing.

Inductive learning on multiple graphs. For the inductive learning on multiple graphs, we use the protein-protein interaction (PPI) networks [13] to evaluate the effectiveness of the proposed method. The dataset contains multiple graphs that 20 graphs for training, 2 for validation and 2 for testing. Each node has 50 features that include positional gene sets, motif gene sets, and immunological signatures and 121 labels that represent gene ontology.

3 Encoder for Different Datasets

Transductive learning for small graph. We take a two-layer GCN [7] as the encoder for the transductive learning tasks (Cora, Citeseer and Pubmed). The architecture of encoder is defined as:

$$GCN_i(\boldsymbol{X}, \boldsymbol{A}) = \sigma(\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X}\boldsymbol{W}_i) \tag{3}$$

$$f(\boldsymbol{X}, \boldsymbol{A}) = GCN_2(GCN_1(\boldsymbol{X}, \boldsymbol{A}), \boldsymbol{A}) \tag{4}$$

where $\boldsymbol{X}$ is the features of nodes in graph, $\hat{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ represrnts the adjacency matrix with self-loops and $\boldsymbol{I}$ is identity matrix, $\hat{\boldsymbol{D}} = \sum_i \hat{\boldsymbol{A}}_i$ denotes the degree

matrix, σ is a nonlinear activation function (e.g., relu, prelu) [4] and $\boldsymbol{W}_i$ is the trainable parameter matrix.

Inductive learning for large graph. Since the large scale of Reddit dataset, we may no longer directly encode the graph by transductive manner that rely on the fixed and known adjacency matrix. Instead, we employ a three-layer mean-pooling GraphSAGE [3] with residual connections [5] as encoder, which is defined as:

$$MP_i(\boldsymbol{X}, \boldsymbol{A}) = \sigma([\hat{\boldsymbol{D}}^{-1}\hat{\boldsymbol{A}}\boldsymbol{X} \| \boldsymbol{X}]\boldsymbol{W}_i) \tag{5}$$

$$f(\boldsymbol{X}, \boldsymbol{A}) = MP_3(MP_2(MP_1(\boldsymbol{X}, \boldsymbol{A}), \boldsymbol{A}), \boldsymbol{A}) \tag{6}$$

where $\hat{\boldsymbol{D}}^{-1}$ performs mean-pooling propagation operation, $\|$ represents feature concatenation (center nodes and its neighborhood).

Following [3], we perform neighborhood sampling for central nodes. What should be noted is that we add a tiny change. After random sampling to determine the central nodes, we sample the subgraphs for each center nodes. We then treat all nodes contained in sampled subgraphs as the new center nodes. Then sampling a new neighborhood for each center nodes, which is used for get the representations of the center nodes. Specifically, 10, 10 and 25 nodes at the first, second and third level are sampled, respectively.

Inductive learning for multiple graphs. For PPI dataset, inspired by the previous successful supervised network architectures, we employ a three-layer GAT with residual connections [5] as encoder, which is defined as:

$$\widehat{MP}_i = \sigma([\boldsymbol{X}\boldsymbol{W}_i^{'}; \hat{\boldsymbol{D}}^{-1}\hat{\boldsymbol{A}}\boldsymbol{X}\boldsymbol{W}_i]) \tag{7}$$

$$f(\boldsymbol{X}, \boldsymbol{A}) = \widehat{MP}_3(\widehat{MP}_2(\widehat{MP}_1(\boldsymbol{X}, \boldsymbol{A}), \boldsymbol{A}), \boldsymbol{A}) \tag{8}$$

where $\boldsymbol{W}_i^{'}$ and $\boldsymbol{W}_i$ are trainable parameter matrices.

4 Parameter Settings

The specific network parameter settings for different datasets are shown in the Table. 2.

Table 2. Parameter settings for different datasets. lr, σ, dim and k stand for learning rate, nonlinear activation function, dimension of node representations and size of subgraph. 256×4: dimension of node representations in each heads and the number of heads.

Dataset	lr	σ	dim	k
Cora	0.0001	relu	1024	15
Citeseer	0.0001	prelu	1024	7
Pubmed	0.0005	prelu	1024	30
PPI	0.0001	relu	256×4	10
Reddit	0.0001	relu	1024	15

Table 3. Comparative experiments with GraphCL

	Cora	Citeseer	Pubmed
GraphCL-node	81.4	72.7	78.8
GraphCL-egde	81.3	72.6	78.4
GraphCL-subgraph	81.4	72.7	78.9
GraphCL-mask	81.3	72.6	78.6
BFS (ours)	84.6	73.7	82.1

Table 4. Performance of different contrastive sample proportions on Cora and Citeseer datasets.

Algorithm	Cora					Citeseer				
	10%	30%	50%	70%	100%	10%	30%	50%	70%	100%
DGI [11]	75.2	78.4	80.5	81.8	82.3	67.2	69.3	70.6	71.4	71.8
GMI [9]	78.2	79.5	81.0	82.1	83.0	68.6	70.2	70.9	72.2	73.0
GraphCL [2]	80.1	81.6	82.5	83.2	83.6	70.1	70.9	71.4	72.0	72.5
Subg-Con [6]	79.6	81.2	82.3	83.0	83.5	70.3	71.2	72.1	72.9	73.2
GSC (ours)	82.9	83.6	84.1	84.6	84.6	71.8	72.5	73.4	73.6	73.7

5 More Comparative

GraphCL. Here, we give comparison results with another GraphCL [12] proposed by You et al, as can be seen in Table. 3, "node, egde, subgraph and mask" represent different perturbation methods, i.e., node dropping, edge perturbation, attribute masking and subgraph based perturbation.

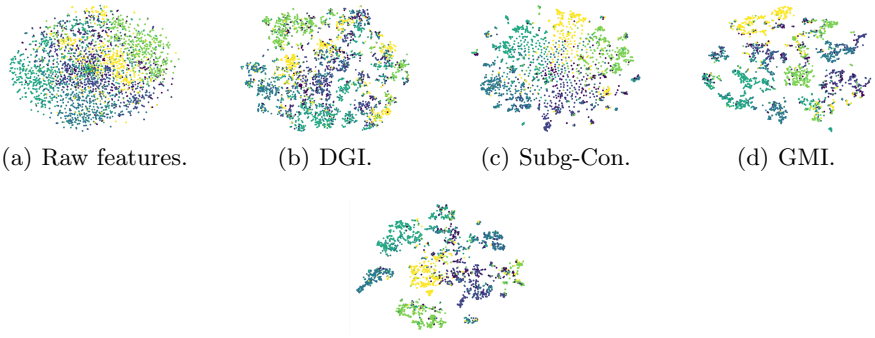
GOT. The OT is just a regularization term for supervised alignments loss in GOT [1]. While in our method, OT is firstly employed as the similarity metric for subgraphs based self-supervised graph contrastive learning.

OT-GNN. The differences with OT-GNN are: (1) OT-GNN [1] treats the graph as a set of node features only. Our method considers the nodes features and structures of graph at the same time. (2) OT-GNN relies on the shared parametric prototypes, while our method generates the unique contrastive sample for each subgraph.

Insufficient contrastive samples. As is shown in Table. 4, we give the evaluation results with insufficient contrastive samples on Cora and Citeseer datasets.

6 Visualization Results

As shown in Fig. 2, we also visualize the raw features and learned embeddings of Citeseer with the t-SNE [8] plot for different graph contrastive learning methods, including DGI, Subg-Con, GMI and GSC (ours).



(a) Raw features. (b) DGI. (c) Subg-Con. (d) GMI.

(e) GSC (ours).

Fig. 2. Visualization of t-SNE embeddings from raw features, DGI, Sub-Con, GMI, and GSC (ours) on Citeseer.

Table 5. Ablation studies on different negative sample settings. N and 2N represent the number of negative samples.

	Cora	Citeseer	Pubmed
Only Sampled (N)	84	71.3	73.8
Only Generated (N)	83.4	72.7	73.4
Sampled + Generated (N)	83.2	72.3	73.5
Sampled + Generated (2N) (ours)	84.6	73.7	82.1

7 Abalation Studies

Different negative sample settings. We compare different negative sample settings and list the experiment results in Table 5. "Only Sampled (N)" and "Only Generated (N)" represent the N negative samples only from the sampled or generated subgrpahs. "Sampled + Generated (N)" represents the N negative samples come from the sampled or generated subgrpahs. "Sampled + Generated (2N)" means the $2N$ negative samples come from both the sampled and generated subgrpahs. As can be seen, the settings in our method achieves the best results. This indicates that we need to push away both the sampled and generated negative samples to ensure the effectiveness of encoder and generator.

Ablation studies on different modules of inductive datasets. We also set up ablation experiments on different modules of inductive datasets (e.g., PPI), the nodes classification results of Gene + Readout, Perturbation + OT and Gene + OT (ours) are 63.5, 68.3 and 69.1 respectively. This indicates that different modules also play an effective role on inductive dataset.

8 Training Time

Based on the sampling and parallel computing of subgraphs, the average epoch time of GSC is only 0.2 seconds on Cora dataset where OT based contrastive regularization is 0.12 seconds. Although OT distance consumes a part of the training time, it can explore more discriminative information from nodes and structures of graph and accelerate convergence speed.

References

1. Chen, L., Gan, Z., Cheng, Y., Li, L., Carin, L., Liu, J.: Graph optimal transport for cross-domain alignment. In: International Conference on Machine Learning. pp. 1542–1553. PMLR (2020) 3
2. Hafidi, H., Ghogho, M., Ciblat, P., Swami, A.: Graphcl: Contrastive self-supervised learning of graph representations. arXiv preprint arXiv:2007.08025 (2020) 5
3. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 1025–1035 (2017) 2, 3
4. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015) 3
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 3
6. Jiao, Y., Xiong, Y., Zhang, J., Zhang, Y., Zhang, T., Zhu, Y.: Sub-graph contrast for scalable self-supervised graph representation learning. arXiv preprint arXiv:2009.10273 (2020) 5
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016) 2
8. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008) 4
9. Peng, Z., Huang, W., Luo, M., Zheng, Q., Rong, Y., Xu, T., Huang, J.: Graph representation learning via graphical mutual information maximization. In: Proceedings of The Web Conference 2020. pp. 259–270 (2020) 5
10. Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., Eliassi-Rad, T.: Collective classification in network data. AI magazine **29**(3), 93–93 (2008) 2
11. Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. arXiv preprint arXiv:1809.10341 (2018) 2, 5
12. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems **33** (2020) 3
13. Zitnik, M., Leskovec, J.: Predicting multicellular function through multi-layer tissue networks. Bioinformatics **33**(14), i190–i198 (2017) 2