ENGAGE: Explanation Guided Data Augmentation for Graph Representation Learning

Yucheng Shi[1], Kaixiong Zhou[2], and Ninghao Liu[1]✉

[1] University of Georgia, Athens, GA 30602, USA
{yucheng.shi, ninghao.liu}@uga.edu
[2] Rice University, Houston, TX 77005, USA
kaixiong.zhou@rice.edu

A Theoretical Justification

Properties of Mutual Information. We will first go through some important properties of mutual information. For any random variables x, y and z:

(1) Positivity:

$$I(x;y) \geq 0, \;\; I(x;y|z) \geq 0.$$

(2) Symmetry:

$$I(x;y) = I(y;x).$$

(3) Chain rule:

$$I(y;z) = I(xy;z) - I(x;z|y).$$

(4) Interaction Information:

$$I(y;z) = I(x;y;z) + I(y;z|x).$$

Proposition A.1. Let z be a representation of X, we can have $I(X;z) = I(X;z|y) + I(y;z)$ (Equation 9).

Proof:

Since z is a representation of X, we can have $I(y;z|X) = 0$ according to Proposition B.1 in [2]. So,

$$\begin{aligned} I(Xy;z) &= I(y;z|X) + I(X;z) \\ &= I(X;z). \end{aligned}$$

Also, $I(Xy;z)$ can be further divided, so we have

$$\begin{aligned} I(X;z) &= I(Xy;z) \\ &= I(X;z|y) + I(y;z). \end{aligned}$$

Proposition A.2. Let z be a sufficient representation of X, then we have $I(X;y|z) = 0$.

Proof:

Since z is a sufficient representation of X, we have $I(X;y) = I(z;y)$ according to Definition 1. Thus,

$$\begin{aligned} 0 &= I(X;y) - I(z;y) \\ &= I(X;y) - I(z;y) - I(y;z|X) \\ &= I(X;y) - I(X;y;z) \\ &= I(X;y|z). \end{aligned}$$

Proposition A.3. Given the learned representations z_1 of view U_1, which is minimal sufficient of U_1 for y, then $I(z_1;U_1) = I(z_1;y)$.

Proof:

Since z_1 is a minimal sufficient representation, for $\forall z_1'$ there exists $I(z_1';U_1) \geq I(z_1;U_1)$, where z_1' is a sufficient representation. Then, according to Proposition A.1, we have $I(z_1;U_1) = I(U_1;z_1|y) + I(z_1;y)$, $I(z_1';U_1) = I(U_1;z_1'|y) + I(z_1';y)$. Since both representations are sufficient, according to Definition 1, we have $I(U_1;y) = I(z_1;y) = I(z_1';y)$. So,

$$\begin{aligned} I(U_1;z_1'|y) - I(U_1;z_1|y) &= I(U_1;z_1') - I(U_1;z_1) + I(z_1;y) - I(z_1';y) \\ &= I(U_1;z_1') - I(U_1;z_1) + I(U_1;y) - I(U_1;y) \\ &= I(U_1;z_1') - I(U_1;z_1) \geq 0. \end{aligned}$$

Thus, we have $I(U_1;z_1'|y) \geq I(U_1;z_1|y), \forall z_1'$. Since there always exists a z_1' that meets $I(U_1;z_1'|y) = 0$, so we have $I(U_1;z_1|y) \leq 0$. As mutual information is non-negative, we have $I(U_1;z_1|y) = 0$. So,

$$\begin{aligned} I(z_1;U_1) &= I(U_1;z_1|y) + I(z_1;y) \\ &= I(z_1;y). \end{aligned}$$

B Algorithm

We provide the algorithm of SAM-based data augmentation method in Algorithm 1. The code for implementation can be found here [3].

C Ablation Study

We conduct detailed ablation studies on 13 datasets. The result is shown in Figure 6∼ 14. In the ablation study, grid search is performed for λ_e and λ_f within the range of {-3,-2,-1,0,1,2,3}, which means the views are set to be from less perturbed to more perturbed. The results show that the optimal λ_e and λ_f vary across different datasets. To achieve optimal downstream task performance, the original view needs to be perturbed at different levels for different datasets. Thus, a general random perturbation scheme may hurt task-relevant information, which

[3] Github:https://github.com/sycny/ENGAGE.

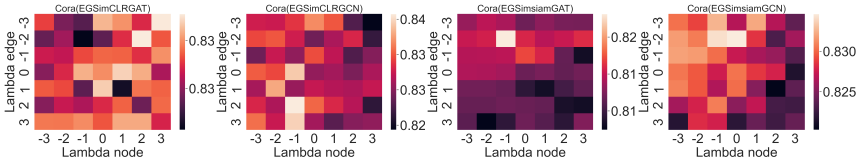
Algorithm 1 SAM-based data augmentation for graph-level representation learning.

Input: Encoder $f(\cdot)$, the target graph $G_n = \{\mathcal{V}, \mathcal{E}, \boldsymbol{A}, \boldsymbol{X}\} \in \mathcal{G}$, total iteration time T, start guiding time T', random perturbing possibility r_e and r_f, nearest neighbors size m.

for $t = 0$ *to* T **do**
  if $t \leq T'$ **then** // Conduct random perturbation
    $\boldsymbol{M}^{feat,1}, \boldsymbol{M}^{feat,2} \leftarrow \text{Bernoulli}(r_f)$; $\boldsymbol{M}^{edge,1}, \boldsymbol{M}^{edge,2} \leftarrow \text{Bernoulli}(r_e)$;
    $\boldsymbol{A}_1, \boldsymbol{A}_2 \leftarrow \boldsymbol{A} \odot \boldsymbol{M}^{edge,1}, \boldsymbol{A} \odot \boldsymbol{M}^{edge,1}$;
    $\boldsymbol{X}_1, \boldsymbol{X}_2 \leftarrow \boldsymbol{X} \odot \boldsymbol{M}^{feat,1}, \boldsymbol{X} \odot \boldsymbol{M}^{feat,1}$;
  else
    $\boldsymbol{Z} \leftarrow f(\boldsymbol{X}, \boldsymbol{A})$; $\boldsymbol{z} \leftarrow Pooling(\boldsymbol{Z})$; // $z \in \mathbb{R}^K$, the graph-level embedding of G_n
    For G_n, find its m nearest-neighbor graphs $\tilde{\mathcal{N}}_n$ in the latent space by quantization;
    $\tilde{w}_k^{graph} \leftarrow normalize(\boldsymbol{z} + \sum_{n' \in \tilde{\mathcal{N}}_n}^{m} \boldsymbol{z}_{n'})[k]$; //smoothed importance score of channel k
    for $v_i \in G_n$ **do**
      $F_{k,i}^L \leftarrow \boldsymbol{Z}[i,k]$;
      $\psi_i \leftarrow \text{ReLU}(\sum_k \tilde{w}_k^{graph} \odot F_{k,i}^L)$; // node importance score
      Conduct perturbation in Equation 5∼8;

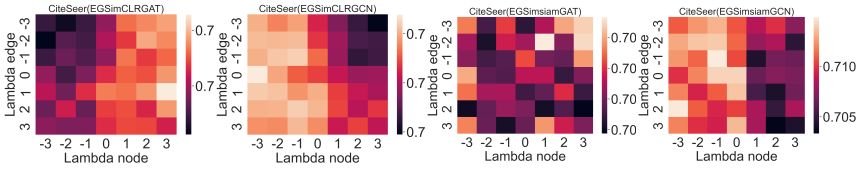
Output: Augmented views $\boldsymbol{A}_1, \boldsymbol{A}_2, \boldsymbol{X}_1, \boldsymbol{X}_2$.

will lead to performance degradation. We can also observe that especially in graph-level datasets, the optimal combination of λ_e and λ_f obtained in EGSimCLR and EGSimsiam are similar for each dataset. This indicates that for each dataset even in different contrastive learning models, the importance threshold is generally consistent. We assume the threshold is mostly determined by dataset properties.



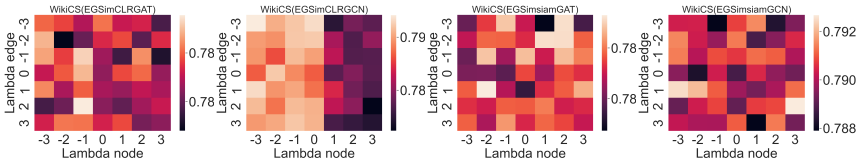
(a) EGSimCLRGAT (b) EGSimCLRGCN (c) EGSimsiamGAT (d) EGSimsiamGCN

Fig. 6: Ablation result of Cora.



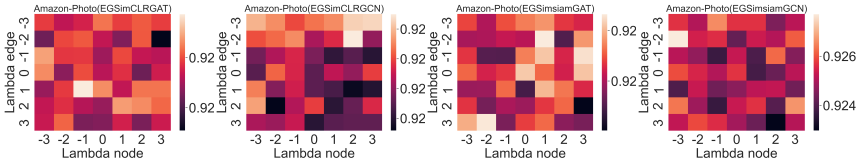
(a) EGSimCLRGAT (b) EGSimCLRGCN (c) EGSimsiamGAT (d) EGSimsiamGCN

Fig. 7: Ablation result of CiteSeer.



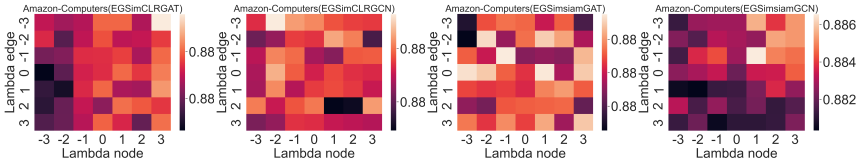
(a) EGSimCLRGAT (b) EGSimCLRGCN (c) EGSimsiamGAT (d) EGSimsiamGCN

Fig. 8: Ablation result of WikiCS.



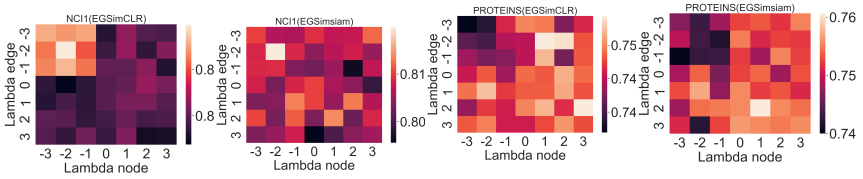
(a) EGSimCLRGAT (b) EGSimCLRGCN (c) EGSimsiamGAT (d) EGSimsiamGCN

Fig. 9: Ablation result of Amazon-Photo.



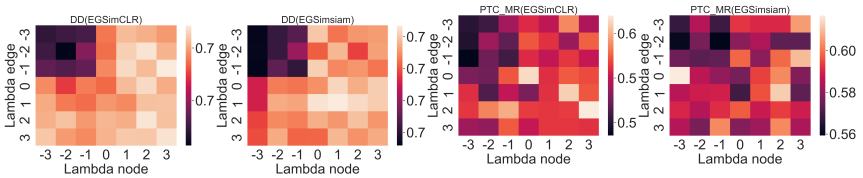
(a) EGSimCLRGAT (b) EGSimCLRGCN (c) EGSimsiamGAT (d) EGSimsiamGCN

Fig. 10: Ablation result of Amazon-Computers.



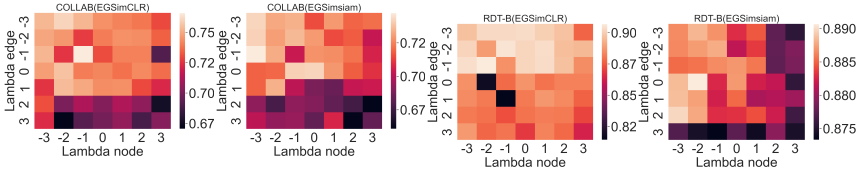
(a) EGSimCLR-NCI1 (b) EGSimsiam-NCI1 (c) EGSimCLR-PROTEINS (d) EGSimsiam-PROTEINS

Fig. 11: Ablation result of NCI1 and PROTEINS.



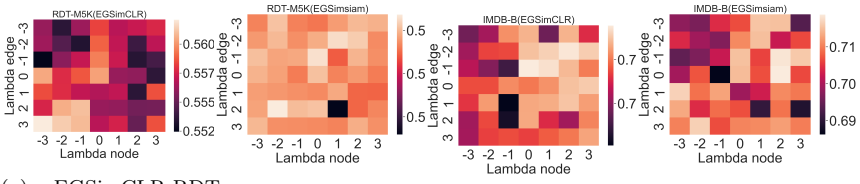
(a) EGSimCLR-DD (b) EGSimsiam-DD (c) EGSimCLR-PTC-MR (d) EGSimsiam-PTC-MR

Fig. 12: Ablation result of DD and PTC-MR.



(a) EGSimCLR-COLLAB (b) EGSimsiam-COLLAB (c) EGSimCLR-RDT-B (d) EGSimsiam-RDT-B

Fig. 13: Ablation result of COLLAB and RDT-B.



(a) EGSimCLR-RDT-M5K (b) EGSimsiam-RDT-M5K (c) EGSimCLR-IMDB-B (d) EGSimsiam-IMDB-B

Fig. 14: Ablation result of RDT-M5K and IMDB-B.

D Comparison of Guided Data Augmentation

Previous research [16] suggests that the downstream task performance is upper-bounded by a reverse-U shaped curve of mutual information. It points out that keeping the information intact while reducing the mutual information between views will improve image representation learning performance. Peng et al. [12] have found that the representation from unsupervised learning contains certain semantic information that could be leveraged to improve contrastive learning.

In domains like graphs, to tackle the issue, Zhu et al. [26] propose to use node centrality as an importance indicator and perturb nodes with lower node centrality. However, the centrality is a static value and may not adapt to the training process. You et al. [23] design an end-to-end framework named JOAO to automatically optimize data augmentation for specific datasets. LG2AR [6] is also an end-to-end framework that aims to select optimal data augmentation methods and parameters by learning their distributions. However, both JOAO and LG2AR require much greater computation expenses. The comparison between different guided augmentation methods is listed in Table 4.

Table 4: Comparison of guided data augmentation.

Approach	Task	Importance score	Strategy	Obj. Function
InfoMin [16]	Images	—	Min-Max optimization	InfoNCE loss
CAST [13]	Images	Saliency map	Object localization&random cropping	CAST Loss
ContrastiveCrop [12]	Images	Heatmap	Object localization&center-suppressed cropping	Multiple loss
GCA [26]	Node	Node centrality	Preserving high centrality part	InfoNCE loss
JOAO [23]	Node and graph	—	Min-Max optimization	JOAO/JOAOv2 loss
LG2AR [6]	Node and graph	—	Learning data augmentation and parameters distribution	InfoNCE loss

E Dataset Description

To demonstrate the effectiveness of our proposed ENGAGE model, we select 13 representative benchmark datasets for both node-level and graph-level classification task. They have been widely applied in the previous research [5,24,25,26].

Datasets for Node-level Classification Task. For the node-level classification task, we select Cora, CiteSeer, WikiCS, Amazon-Computers, and Amazon-Photo datasets as benchmarks. Cora and CiteSeer are citation network datasets from [21], where nodes stand for articles and edges stand for citation links. WikiCS dataset is extracted from Wikipedia [8]. Its nodes represent CS area articles, while its edges represent hyperlinks. Amazon-Computers and Amazon-Photo are derived from purchase history [14], where nodes features are corresponding to product reviews. For fair comparison, we follow dataset split settings in [25,26]. The dataset statistics are shown in Table 5.

Table 5: Dataset statistics for node-level classification task.

	Cora	CiteSeer	WikiCS	Amazon-Computers	Amazon-Photo
Classes	7	6	10	10	8
Features	1433	3703	300	767	745
Nodes	2708	3327	11701	13381	7487
Edges	5429	4732	216123	245778	119043

Datasets for Graph-level Classification Task. For the graph-level classification task, we select NCI1, PROTEINS, DD, PTC-MR, COLLAB, RDT-B, RDT-M5K, and IMDB-B from TUDataset [9] as benchmarks. The first four datasets are mainly biochemical molecules and chemical compounds, while the latter four datasets are derived from social networks. In the experiments, we follow dataset split settings in [24]. The dataset statistics are shown in Table 6.

Table 6: Dataset statistics for graph-level classification task.

	NCI1	PROTEINS	DD	PTC-MR	COLLAB	RDT-B	RDT-M5K	IMDB-B
Types	Molecules and Compounds				Social Networks			
Graph Number	4110	1113	1178	344	5000	2000	2000	1000
Average Node	29.87	39.06	284.32	14.29	74.49	429.63	429.63	19.77
Average Degree	1.08	1.86	715.66	14.69	32.99	1.15	497.75	96.53

F Brief Description of Baseline Methods

For the graph-level self-supervised methods, there are mainly two kinds of models: topological context learning and contrastive learning. The topological context learning model builds on the idea that nodes with similar contexts tend to have similar representations, which includes node2vec [4], sub2vec [1], and graph2vec [10] models. On the other hand, the contrastive learning model aims

to maximize the mutual information of positive views. The contrastive learning models can be further separated into three types: (1) Graph contrastive learning with random data augmentation, including GraphCL [24] and InfoGraph [15]. This type is simple yet effective compared with the topological context model. (2) JOAO [23], JOAOv2 [23], which are proposed to prevent semantic information damage from random sampling, including JOAO, JOAOv2. These models have their data augmentation process automated to optimize the performance on downstream tasks. (3) MVGRL [5] and SimGRACE [19]. The former introduces multiple augmented views instead of two positive views into contrastive learning to improve the representation quality. The latter one gets rid of data augmentation and positive views, but chooses to perturbate the encoder model to implement the contrastive learning.

For the node-level tasks, the self-supervised learning baseline models also contain two types: (1) The first type applies random data augmentation on contrastive learning, which includes GRACE [25], DGI [17], MVGRL. (2) The second type, on the other hand, improves data augmentation by introducing domain knowledge. GCA model uses node centrality to help determine which part should be perturbed.

It can be observed that the contrastive learning models consistently outperform the previous models (node2vec, sub2vec, graph2vec). This is because contrastive learning has better architecture design and more effective learning objectives.

And between different contrastive learning methods: (1) Our model shows the best overall performance, thanks to the carefully designed data augmentation strategy guided by the SAM explanation. SAM identifies graph components that are important in preserving the properties of graphs. Our model can keep these important components intact during data augmentation, while removing the superfluous information to learn more generalizable representations. (2) JOAO and JOAOv2 outperform InfoGraph, while GCA [23] outperforms DGI. The reason is their better data augmentation strategy, which is either optimized by extra learning objectives or guided by graph inherent features, compared with random augmentation. (3) MVGRL works better than DGI in node-level tasks, and better than GraphCL in graph-level tasks. The success of MVGRL is because of the extra view introduced in the training process which helps contrastive learning better capture the key information.

G Implemention Details

Missing Feature Matrix in Social Networks. For social networks including COLLAB, RDT-B, RDT-M5K, and IMDB-B, the feature matrix is missing, which is not consistent with the GIN model structure [20]. To tackle this challenge, in previous research [15,24,19], a replacement matrix with feature dimension set to one is created, and each node is assigned the same value (1.) equally regardless of their difference. In our proposed model, we argue that for different nodes, feature values should be different, therefore, we create the replacement matrix using the node explanation result. The importance score obtained from the last epoch is

its new feature for each node. It can be observed in the experiment that our proposed replacement matrix shows better performance.

Hyperparameter Tunning. In the experiments, we observe that the selection of learning rate and training epochs of contrastive learning are vital to the downstream performance. To be specific, for learning rate, we choose {0.01,0.001,0.0001,0.00001} as the candidate set. For training epochs, we set the candidate set as {20,50,100,200,300}. We optimize these two hyperparameters on the validation dataset. The detailed configurations for different datasets are logged in the yaml file along with code.

Running Environment. All experiments are conducted on a workstation with a CPU of Intel(R) Core(TM) i9-10900X and a GPU of NVIDIA RTX3090 (24GB memory). The GNN models and contrastive learning models are implemented in PyTorch [11], which is under BSD license. Additionally, the GNN models are implemented with PyG (PyTorch Geometric) [3], which is under MIT license. And we apply faiss [7] in m-nearest embedding search, which is under MIT license.

Dependencies. We list the main packages needed to implement our code in the following list.

- torch 1.10.1+cu113
- torch-cluster 1.5.9
- torch-geometric 2.0.3
- torch-scatter 2.0.9
- torch-sparse 0.6.12
- faiss-cpu 1.7.2

H Visualization Results on SAM

We have taken the synthetic dataset BA-shapes [22] as input to visualize the explanation result. The explanation results of Node 405 and 575 at different epochs are shown in Figure 15 and Figure 16.

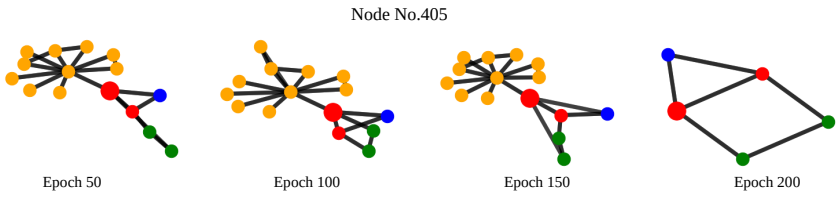


Fig. 15: Visualization result of BA-shapes node 405.

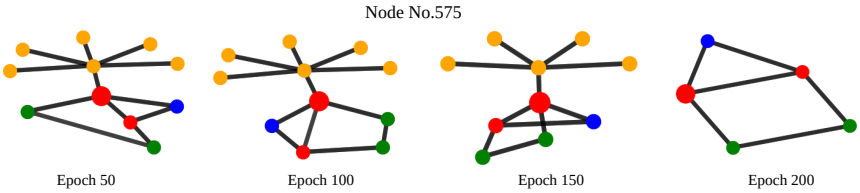


Fig. 16: Visualization results of BA-shapes node 575.

I Quantitative Results on SAM

Graph representation learning maps nodes/graphs into latent space, where the embeddings of similar nodes/graphs will be close to each other [18], i.e., they will have a smaller $L2$ distance. Under the above assumption, if the embedding of a subgraph and the original graph has a similar set of neighbors, then we can infer that this subgraph is important for the original graph, since it preserves the key properties of the original graph.

Based on the above discussion, we propose a metric called Overlapped-Neighbors (OP) to quantify the explanation effectiveness:

$$OP = \frac{1}{|G|}\sum_{i}^{|G|} \frac{|H_i \cap \tilde{H}_i|}{N} \times 100\%, \tag{1}$$

where $|G|$ stands for the total graph number in the dataset, N means the number of neighbors. H, $\tilde{H}$ are the set of neighbors for the subgraph embedding and original graph embedding, respectively. A higher OP value indicates that the selected subgraph retains more discriminative information from the original graph, which means it is a better explanation.

To validate the effectiveness of our SAM explanations, we design three strategies to build the subgraphs: (1) sample nodes randomly to build the subgraph, (2) sample nodes with high importance scores assigned by SAM, (3) sample nodes with low importance scores assigned by SAM. The sampling rate is 10%. The representation of the subgraph and graph are obtained using a freeze-parameter encoder from a trained EG-SimCLR model. The OP values are reported in Table 7.

Table 7: OP results on three kinds of sample strategies.

Drop type	NCI1	PROTEINS	DD	PTC-MR	COLLAB	RDT-B	RDT-M5K	IMDB-B
Sample Nodes Randomly	10.35 ± 0.03	12.15 ± 0.13	13.59 ± 0.20	13.30 ± 0.11	11.32 ± 0.25	13.48 ± 0.11	10.71 ± 0.03	6.88 ± 0.54
Sample Nodes with High Importance	21.11 ± 0.80	20.05 ± 1.00	26.63 ± 0.29	27.19 ± 0.60	33.77 ± 2.86	30.60 ± 1.05	17.25 ± 0.76	11.12 ± 0.52
Sample Nodes with Low Importance	10.45 ± 0.30	9.55 ± 0.26	15.20 ± 0.25	15.19 ± 0.30	16.80 ± 0.62	9.80 ± 1.29	9.80 ± 0.56	5.25 ± 0.22

It can be observed that: when building the subgraphs with important nodes, the OP value is consistently higher than the random subgraphs or subgraphs with unimportant nodes. It means SAM could capture the discriminative information

in graphs, which can be viewed as the explanation of the contrastive learning model.

J Limitation

Currently, we conduct the N-nearest embeddings search with a naive strategy of quantization. This step may negatively affect the training speed especially when the graph is large. So there is still room for our proposed data augmentation to improve speed. In our future work, we consider using product quantization (PQ) in [7] to improve sorting speed.

References

1. Adhikari, B., Zhang, Y., Ramakrishnan, N., Prakash, B.A.: Sub2vec: Feature learning for subgraphs. In: PAKDD (2018)
2. Federici, M., Dutta, A., Forré, P., Kushman, N., Akata, Z.: Learning robust representations via multi-view information bottleneck. arXiv preprint arXiv:2002.07017 (2020)
3. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. In: ICLR Workshop on Representation Learning on Graphs and Manifolds (2019)
4. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
5. Hassani, K., Khasahmadi, A.H.: Contrastive multi-view representation learning on graphs. In: International Conference on Machine Learning. pp. 4116–4126. PMLR (2020)
6. Hassani, K., Khasahmadi, A.H.: Learning graph augmentations to learn graph representations. arXiv preprint arXiv:2201.09830 (2022)
7. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with gpus. IEEE Transactions on Big Data **7**(3), 535–547 (2019)
8. Mernyei, P., Cangea, C.: Wiki-cs: A wikipedia-based benchmark for graph neural networks. arXiv preprint arXiv:2007.02901 (2020)
9. Morris, C., Kriege, N.M., Bause, F., Kersting, K., Mutzel, P., Neumann, M.: Tudataset: A collection of benchmark datasets for learning with graphs. arXiv preprint arXiv:2007.08663 (2020)
10. Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., Jaiswal, S.: graph2vec: Learning distributed representations of graphs. ArXiv **abs/1707.05005** (2017)
11. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`

12. Peng, X., Wang, K., Zhu, Z., You, Y.: Crafting better contrastive views for siamese representation learning. arXiv preprint arXiv:2202.03278 (2022)
13. Selvaraju, R.R., Desai, K., Johnson, J., Naik, N.: Casting your model: Learning to localize improves self-supervised representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11058–11067 (2021)
14. Shchur, O., Mumme, M., Bojchevski, A., Günnemann, S.: Pitfalls of graph neural network evaluation. arXiv preprint arXiv:1811.05868 (2018)
15. Sun, F.Y., Hoffmann, J., Verma, V., Tang, J.: Infograph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. arXiv preprint arXiv:1908.01000 (2019)
16. Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? Advances in Neural Information Processing Systems **33**, 6827–6839 (2020)
17. Veličković, P., Fedus, W., Hamilton, W.L., Liò, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. arXiv preprint arXiv:1809.10341 (2018)
18. Wang, T., Isola, P.: Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere. arXiv e-prints arXiv:2005.10242 (May 2020)
19. Xia, J., Wu, L., Chen, J., Hu, B., Li, S.Z.: Simgrace: A simple framework for graph contrastive learning without data augmentation. arXiv preprint arXiv:2202.03104 (2022)
20. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? arXiv preprint arXiv:1810.00826 (2018)
21. Yang, Z., Cohen, W., Salakhudinov, R.: Revisiting semi-supervised learning with graph embeddings. In: International conference on machine learning. pp. 40–48. PMLR (2016)
22. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. Advances in neural information processing systems **32** (2019)
23. You, Y., Chen, T., Shen, Y., Wang, Z.: Graph contrastive learning automated. In: International Conference on Machine Learning. pp. 12121–12132. PMLR (2021)
24. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Advances in Neural Information Processing Systems **33**, 5812–5823 (2020)
25. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Deep graph contrastive representation learning. arXiv preprint arXiv:2006.04131 (2020)
26. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference 2021. pp. 2069–2080 (2021)