
Appendix: Information-Aware Graph Contrastive Learning

Anonymous Author(s)

Affiliation

Address

email

1 Comparison of Graph Contrastive Learning Approaches

2 We summarize the published approaches of graph contrastive learning in Table 1. Specifically, the
 3 tasks include graph-level and node-level tasks. The GNNs are used to encode the graph data and
 4 they are quite flexible. A typical augmented view is the graph itself, but it focuses on different
 5 structure-level of the graph. The two views adopted by mvgrl are the graph itself and the diffused
 6 graph, respectively. The four views used by GraphCL are discussed in the text. GCA applies edges
 7 removing and attribute masking to generate two views. In CSSLs, edge deletion/insertion and node
 8 deletion/insertion are used to generate different views. Similar to GCA, SelfTask also applies edges
 9 removing and attribute masking to generate views. The contrastive modes are discussed in the text,
 10 which include global-global, local-global, local-local, multi-scale mode, and hybrid modes. mvgrl
 11 studies the effect of various contrastive modes. SelfTask uses both local-local and global-global.
 12 The additional comparison of sampling strategies and objective functions between published graph
 13 contrastive learning approaches is shown in Table-2.

Approach	Task	GNN	Augmented Views	Contrastive Mode
DGI [6]	Node	GCN	Graph itself	Global-local
InfoGraph [4]	Both	GIN	Graph itself	Multi-scale
mvgrl [1]	Both	GCN	Two views	Various
GCC [3]	Both	GIN	Graph itself	Local-local
GRACE [9]	Node	GCN, GraphSAGE	Graph itself	Local-local
GraphCL [7]	Both	GCN, GIN, GAT	Four views	Global-global
GCA [10]	Node	GCN	Two views	Local-local
CSSLs [8]	Graph	HGP-SL	Four views	Global-global
SelfTask [2]	Node	GCN	Two views	Two modes

Table 1: A comparison of published approaches for graph contrastive learning.

Approach	Sampling Strategy	Obj. Function
DGI [6]	Randomly sampled graphs (or a graph transformation)	JSD
InfoGraph [4]	Global and local patch across all graph instances in a batch	JSD
mvgrl [1]	Joint distribution for positive, product of marginals for negative	JSD, InfoNCE
GCC [3]	Randomly sampled graphs	InfoNCE
GRACE [9]	Negative samples are all other nodes in the two views	NT-xent
GraphCL [7]	N-1 augmented graphs within the same minibatch	NT-xent
GCA [10]	Negative samples are all other nodes in the two views	InfoNCE
CSSLs [8]	Randomly sampled graphs	NT-xent
SelfTask [2]	Randomly sampled edges/attributes	Cross Entropy

Table 2: Different sampling strategies and objective functions of published approaches for graph contrastive learning.

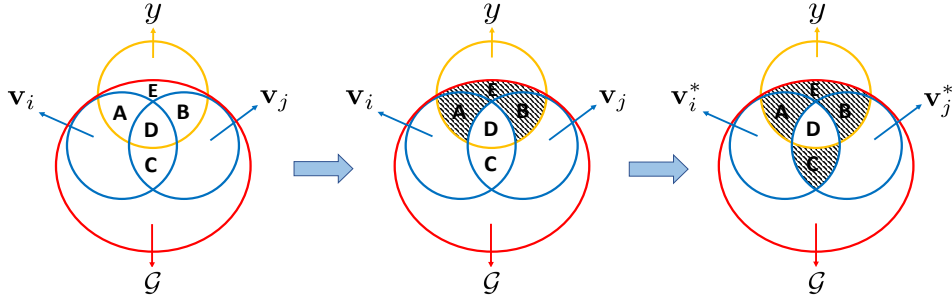


Figure 1: Illustration of optimal views.

14 2 Proofs of Corollaries

15 Basically, we follow the proofs in [5] to provide the proofs of our corollaries. The main differences
 16 are two-fold: 1) the input is graph \mathcal{G} for our framework, and 2) the concept of optimal contrastive
 17 mode is unique for graph contrastive learning.

18 **Corollary 1.** (*Optimal Augmented Views*) For a downstream task T whose goal is to predict a
 19 semantic label y , the optimal views, \mathbf{v}_i^* , \mathbf{v}_j^* , generated from the input graph \mathcal{G} are the solutions to the
 20 following optimization problem :

$$(\mathbf{v}_i^*, \mathbf{v}_j^*) = \arg \min_{\mathbf{v}_i, \mathbf{v}_j} I(\mathbf{v}_i; \mathbf{v}_j) \quad (1)$$

$$s.t. \quad I(\mathbf{v}_i; y) = I(\mathbf{v}_j; y) \quad (2)$$

$$I(\mathbf{v}_i; y) = I(\mathcal{G}; y) \quad (3)$$

21 *Proof.* Because Equation (3) holds and v_i, v_j are functions of \mathcal{G} , it is natural to know $I(\mathcal{G}; y)$
 22 $= I(\mathbf{v}_i, \mathbf{v}_j; y)$ holds. Because $I(\mathbf{v}_i, \mathbf{v}_j; y) = I(\mathbf{v}_i; y) + I(\mathbf{v}_j; y|\mathbf{v}_i)$ and Equation (3) holds, we
 23 know $I(\mathbf{v}_i, \mathbf{v}_j; y) = I(\mathcal{G}; y) + I(\mathbf{v}_j; y|\mathbf{v}_i)$. Because $I(\mathcal{G}; y) = I(\mathbf{v}_i, \mathbf{v}_j; y)$ already holds and the
 24 nonnegativity of mutual information, we know

$$I(\mathbf{v}_j; y|\mathbf{v}_i) = 0. \quad (4)$$

25 Thus, we know $I(\mathbf{v}_i; \mathbf{v}_j) = I(\mathbf{v}_i; \mathbf{v}_j) + I(\mathbf{v}_j; y|\mathbf{v}_i) = I(\mathbf{v}_i, y; \mathbf{v}_j)$. In addition, we know $I(\mathbf{v}_i, y; \mathbf{v}_j)$
 26 $= I(\mathbf{v}_j; y) + I(\mathbf{v}_i; \mathbf{v}_j|y)$. Thus, $I(\mathbf{v}_i; \mathbf{v}_j) = I(\mathbf{v}_j; y) + I(\mathbf{v}_i; \mathbf{v}_j|y) \geq I(\mathbf{v}_j; y) = I(\mathcal{G}; y)$. Therefore,
 27 $I(\mathbf{v}_i; \mathbf{v}_j)$ can reach the minimum, which is $I(\mathcal{G}; y)$ And when $I(\mathbf{v}_i; \mathbf{v}_j)$ can reach the minimum, the
 28 optimal views are conditionally independent, which is described as

$$I(\mathbf{v}_i^*; \mathbf{v}_j^*|y) = 0. \quad (5)$$

29 \square

30 It is noticed that: 1) when Equation (4) holds, the area of B in Figure 1 becomes null. 2) when
 31 Equation (5) holds, the area of C in Figure 1 becomes null.

32 **Corollary 2.** (*Optimal View Encoder*) Given the optimal views, \mathbf{v}_i^* , \mathbf{v}_j^* , for a downstream task T
 33 whose goal is to predict a semantic label y , the optimal view encoder for view \mathbf{v}_i^* is the solution to
 34 the following optimization problem :

$$f_i^* = \arg \min_{f_i} I(f_i(\mathbf{v}_i^*); \mathbf{v}_i^*) \quad (6)$$

$$s.t. \quad I(f_i(\mathbf{v}_i^*); \mathbf{v}_j^*) = I(\mathbf{v}_i^*; \mathbf{v}_j^*) \quad (7)$$

35 We can follow the basic idea of Proposition A.2 and Proposition A.3 in Appendix of [5] to get the
 36 proof of Corollary 2. The main difference is that we focus on the optimal view encoders, however,
 37 [5] focuses on the optimal representations.

38 **Corollary 3.** (*Optimal Contrastive Mode*) Given the latent representations, \mathbf{z}_i^* , \mathbf{z}_j^* , extracted by the
 39 optimal view encoders, i.e., $\mathbf{z}_i^* = f_i^*(\mathbf{v}_i^*)$, $\mathbf{z}_j^* = f_j^*(\mathbf{v}_j^*)$, and a downstream task T with label y , the
 40 optimal contrastive mode is the solution to the following optimization problem, where c_i, c_j are the
 41 aggregation operations applied to the latent representations:

$$(c_i^*, c_j^*) = \arg \min_{(c_i, c_j)} -I(c_i(\mathbf{z}_i^*); c_j(\mathbf{z}_j^*)). \quad (8)$$

42 *Proof.* Recall the objective function of graph contrastive learning is to minimize the contrastive loss,
 43 e.g., \mathcal{L}_{NCE} . Minimizing the loss equivalently maximizes the mutual information between latent
 44 representations based on contrastive modes. For example, minimizing \mathcal{L}_{NCE} equivalently maximizes
 45 $I(\mathbf{z}_i, \mathbf{z}_j)$, which is because $I(\mathbf{z}_i, \mathbf{z}_j) \geq \log(N) - \mathcal{L}_{NCE}$. Thus, we just need to iterate through all the
 46 modes and choose the mode with the largest mutual information value.

47

□

48 3 Experimental Settings

49 The algorithm of InfoGCL for both graph and node classification is described in Algorithm 1. The
 50 view augmentation methods include node dropping, edge perturbation, attribute masking, subgraph
 51 sampling, and graph diffusion. The GNN backbones of view encoders include GCN, GAT, and GIN.
 52 The aggregation operations include identical transformation and taking average, which construct the
 53 global-global, local-global, local-local, multi-scale mode, and hybrid modes. It is noticed that the
 54 two views share the same view encoder (Proposition 2), which is because the domains of the two
 views are the same.

Algorithm 1: Training algorithm of InfoGCL for both graph and node classification

```

1: Input: The input graph  $\mathcal{G}$ , a task  $T$  with label  $y$ , a bunch of graph view augmentation methods,
    $\{q_1(\cdot), q_2(\cdot), \dots\}$ , a set of view encoders,  $\{f^1(\cdot), f^2(\cdot), \dots\}$ , a set of aggregation operations,
    $\{c_1(\cdot), c_2(\cdot), \dots\}$ .
2: Output: The recommended augmentation methods,  $q_i^*(\cdot), q_j^*(\cdot)$ , the recommended view
   encoder,  $f^*(\cdot)$ , the recommended contrastive mode,  $(c_i^*(\cdot), c_j^*(\cdot))$ .
3: # Proposition 1: optimal view augmentations
4:  $L_a \leftarrow []$ 
5: for view augmentation  $q_i(\cdot) \in \{q_1(\cdot), q_2(\cdot), \dots\}$  do
6:   for view augmentation  $q_j(\cdot) \in \{q_1(\cdot), q_2(\cdot), \dots\}$  do
7:      $\mathbf{v}_i \leftarrow q_i(\mathcal{G})$ 
8:      $\mathbf{v}_j \leftarrow q_j(\mathcal{G})$ 
9:      $\mathbf{a} \leftarrow I(\mathbf{v}_i; y) + I(\mathbf{v}_j; y) - I(\mathbf{v}_i; \mathbf{v}_j)$ .
10:    Add  $\mathbf{a}$  to  $L_a$ .
11:   end for
12: end for
13: Choose the augmentations with the maximum value in  $L_a$  as  $q_i^*(\cdot)$  and  $q_j^*(\cdot)$ , respectively.
14: # Proposition 2: optimal view encoder
15:  $L_b \leftarrow []$ 
16: for view encoder  $f(\cdot) \in \{f^1(\cdot), f^2(\cdot), \dots\}$  do
17:    $\mathbf{z}_i \leftarrow f(\mathbf{v}_i)$ 
18:    $\mathbf{z}_j \leftarrow f(\mathbf{v}_j)$ 
19:    $\mathbf{b} \leftarrow I(\mathbf{v}_i; \mathbf{z}_i; y) + I(\mathbf{v}_j; \mathbf{z}_j; y)$ .
20:   Add  $\mathbf{b}$  to  $L_b$ .
21: end for
22: Choose the encoder with the maximum value in  $L_b$  as  $f^*(\cdot)$ .
23: # Proposition 3: optimal contrastive mode
24:  $L_c \leftarrow []$ 
25: for aggregation operation  $c_i(\cdot) \in \{c_1(\cdot), c_2(\cdot), \dots\}$  do
26:   for aggregation operation  $c_j(\cdot) \in \{c_1(\cdot), c_2(\cdot), \dots\}$  do
27:      $\mathbf{c} \leftarrow I(c_i(\mathbf{z}_i); c_j(\mathbf{z}_j); y)$ .
28:     Add  $\mathbf{c}$  to  $L_c$ .
29:   end for
30: end for
31: Choose the operations with the maximum value in  $L_c$  as  $c_i^*(\cdot)$  and  $c_j^*(\cdot)$ , respectively.

```

55

56 **4 Quantitative Evaluation of Proposition**

57 To test the validity of our proposed propositions, we conduct the ablation studies on the Cora and
 58 Citeseer datasets. Specifically, we use the GraphCL [7] as the backbone and test it on Proposition 1.

59 **Proposition 1.** For a task T with label y , given a bunch of graph view augmentation methods, $\{q_1(\cdot),$
 60 $q_2(\cdot), \dots\}$, that create two views $\mathbf{v}_i, \mathbf{v}_j$, the recommended augmentation methods are the ones, $q_i(\cdot),$
 61 $q_j(\cdot)$, that maximize $I(\mathbf{v}_i; y) + I(\mathbf{v}_j; y) - I(\mathbf{v}_i; \mathbf{v}_j)$, i.e., the area of $A+B+D$ in Figure 1.

Augmented Views	Classification Accuracy		$I(\mathbf{v}_i; y) + I(\mathbf{v}_j; y) - I(\mathbf{v}_i; \mathbf{v}_j)$	
	Cora	Citeseer	Cora	Citeseer
EdgePert vs. Identical	82.5 ± 0.1	72.2 ± 0.2	4.5671 ± 0.1	812.5062 ± 0.6
EdgePert vs. EdgePert	82.3 ± 0.2	73.1 ± 0.2	4.1921 ± 0.4	821.3120 ± 1.0

Table 3: The comparison between classification accuracy and $I(\mathbf{v}_i; y)+I(\mathbf{v}_j; y)-I(\mathbf{v}_i; \mathbf{v}_j)$ for different augmented views. $I(\mathbf{v}_i; y) + I(\mathbf{v}_j; y) - I(\mathbf{v}_i; \mathbf{v}_j)$ is approximated by $-(CE_i+CE_j-(CE'_i+CE'_j))$.

62 To calculate $I(\mathbf{v}_i; y) + I(\mathbf{v}_j; y) - I(\mathbf{v}_i; \mathbf{v}_j)$, we use cross-entropy to approximate the mutual informa-
 63 tion. Specifically, we feed \mathbf{v}_i into GNNs and the outputs are further fed into a MLP to generate logits.
 64 We calculate the cross-entropy, CE_i , based on the logits and y . Similarly, we are able to calculate the
 65 cross-entropy, CE_j , based on \mathbf{v}_j and y . To approximate $I(\mathbf{v}_i; \mathbf{v}_j)$, we first feed $\mathbf{v}_i, \mathbf{v}_j$ into another
 66 GNNs and MLP. Then, we use the outputs of \mathbf{v}_i as logits and the outputs of \mathbf{v}_j as labels to get CE'_i .
 67 Symmetrically, we are able to get CE'_j . Thus, we use $-(CE_i + CE_j - (CE'_i + CE'_j))$ to approximate
 68 $I(\mathbf{v}_i; y) + I(\mathbf{v}_j; y) - I(\mathbf{v}_i; \mathbf{v}_j)$. We use two-layer GCNs as the GNNs and $CE_i + CE_j - (CE'_i + CE'_j)$
 69 as the loss function. We adopt the default settings and train the model until it converges. The results
 70 are reported in Table 3. It is observed that, on the Cora dataset, the view pair of edge perturbation
 71 and the graph itself shows a little higher performance than the pair of edge perturbation and edge
 72 perturbation. According to the right part of Table 3, we see the view pair of edge perturbation and the
 73 graph itself has a greater value of $I(\mathbf{v}_i; y) + I(\mathbf{v}_j; y) - I(\mathbf{v}_i; \mathbf{v}_j)$ than the pair of edge perturbation
 74 and edge perturbation. Similar observations can be observed on the Citeseer dataset. Thus, the
 75 validity of Proposition 1 is verified.

76 **References**

- 77 [1] Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning
78 on graphs. In *ICML*, pages 3451–3461. 2020.
- 79 [2] Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang.
80 Self-supervised learning on graphs: Deep insights and new direction. *arXiv preprint*
81 *arXiv:2006.10141*, 2020.
- 82 [3] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan
83 Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training.
84 *KDD*, 2020.
- 85 [4] Fan-Yun Sun, Jordan Hoffman, Vikas Verma, and Jian Tang. Infograph: Unsupervised and
86 semi-supervised graph-level representation learning via mutual information maximization. In
87 *ICLR*, 2019.
- 88 [5] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What
89 makes for good views for contrastive learning. In *NeurIPS*, volume 33, 2020.
- 90 [6] Petar Veličković, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R Devon
91 Hjelm. Deep Graph Infomax. In *ICLR*, 2019.
- 92 [7] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen.
93 Graph contrastive learning with augmentations. In *NeurIPS*, volume 33, 2020.
- 94 [8] Jiaqi Zeng and Pengtao Xie. Contrastive self-supervised learning for graph classification. *arXiv*
95 *preprint arXiv:2009.05923*, 2020.
- 96 [9] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep Graph
97 Contrastive Representation Learning. In *ICML Workshop*, 2020.
- 98 [10] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive
99 learning with adaptive augmentation. *arXiv preprint arXiv:2010.14945*, 2020.