AutoGCL: Automated Graph Contrastive Learning via Learnable View Generators

Nanyang Technological University[1], Baidu Research[2], Harvard University[3], The Pennsylvania State University[4]

Yihang Yin[1], Qingzhong Wang[2], Siyu Huang[3], Haoyi Xiong[2], Xiang Zhang[4]

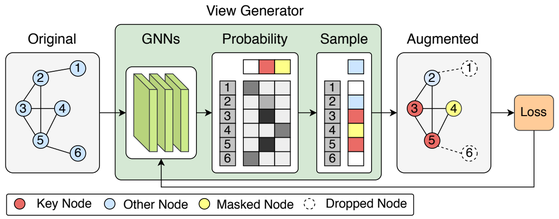
AAAI-22

Source Code: https://github.com/Somedaywilldo/AutoGCL

Motivation

- Data augmentation matters to contrastive learning, both the combination and the intensity of data augmentation will affect the final performance.
- The InfoMin Principle: Make two view generator to keep label-related information while having less mutual information. (Tian et al. 2020)
- To introduce the InfoMin Principle into graph contrastive learning, we design a framework with node-wise learnable data augmentation mechanism, namely AutoGCL.

Learnable View Generator



- For each node, we use the embedded node feature to predict the prob- ability of selecting a certain augment operation.
- We use gumbel-softmax, which employs the reparameterization trick to make the sampling process differentiable.
- Formally, we have:

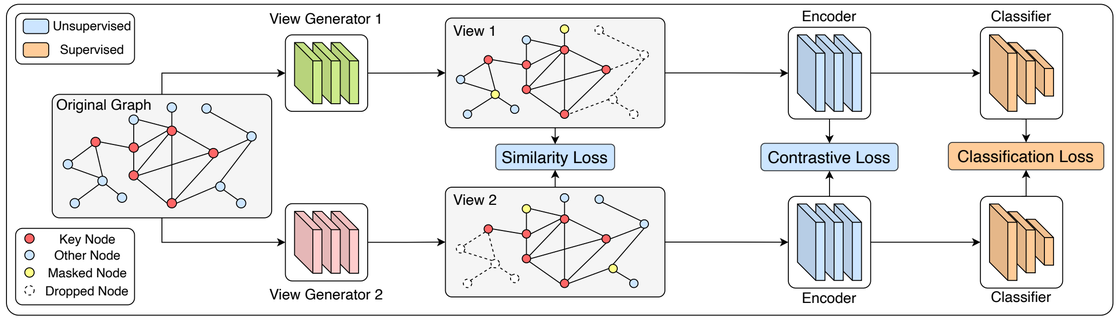
$$\boldsymbol{h}_v^{(k-1)} = \textbf{COMBINE}^{(k)}(\boldsymbol{h}_v^{(k-2)}, \boldsymbol{a}_v^{(k-1)}) \qquad f_v = \textbf{GumbelSoftmax}(\boldsymbol{a}_v^{(k)})$$

$$\boldsymbol{a}_v^{(k)} = \textbf{AGGREGATE}^{(k)}(\{\boldsymbol{h}_u^{(k-1)} : u \in \mathcal{N}(v)\}) \qquad \boldsymbol{x}_v^{'} = \textbf{Aug}(\boldsymbol{x}_v, f_v)$$

Property	CMRLG	GRACE	GraphCL	GCA	JOAO	AD-GCL	Ours
Topological	✓	✓	✓	✓	✓	✓	✓
Node Feature	-	✓	✓	✓	✓	-	✓
Label-preserving	-	-	-	-	-	-	✓
Adaptive	-	-	-	✓	✓	✓	✓
Variance	-	✓	✓	✓	✓	✓	✓
Differentiable	-	-	-	-	-	✓	✓
Efficient BP	-	-	-	-	-	✓	✓

- Our learnable view generator has the following advantages:
  - It supports both the augmentations of the graph topology and node feature.
  - It is label-preserving, retaining the semantic information in the original graph.
  - It is adaptive to different data distributions and scalable to large graphs.
  - It provides sufficient variances for contrastive multi-view pre-training.
  - It is end-to-end differentiable and efficient enough for fast gradient computation via back-propagation (BP).

Framework



Training Strategy

Algorithm 1: Naive training strategy (naive-strategy).

Initialize weights of the two view generator G_1, G_2
Initialize weights of the classifer F
while not reached maximum epochs **do**
  for mini-batch x from unlabeled data **do**
    Get augmentation $x_1 = G_1(x), x_2 = G_2(x)$
    Sample two views v_1, v_2 from $\{x, x_1, x_2\}$
    $\mathcal{L} = \mathcal{L}_{\text{cl}}(v_1, v_2)$
    Update the weights of G_1, G_2, F to minimize $\mathcal{L}$
while not reached maximum epochs **do**
  for mini-batch x from labeled data **do**
    $\mathcal{L} = \mathcal{L}_{\text{cls}}(x)$
    Update the weights of F to minimize $\mathcal{L}$

Similarity Loss

$$\text{sim}(\boldsymbol{z}_1, \boldsymbol{z}_2) = \frac{\boldsymbol{z}_1 \cdot \boldsymbol{z}_2}{\|\boldsymbol{z}_1\|_2 \cdot \|\boldsymbol{z}_2\|_2}$$

$$\mathcal{L}_{\text{sim}} = \text{sim}(A_1, A_2)$$

Contrastive Loss

$$\ell_{(i,j)} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

$$\mathcal{L}_{\text{cl}} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$$

Classification Loss

$$\mathcal{L}_{\text{cls}} = \ell_{\text{cls}}(F(g), y) + \ell_{\text{cls}}(F(g_1), y) + \ell_{\text{cls}}(F(g_2), y)$$

Algorithm 2: Joint training strategy (joint-strategy).

Initialize weights of G_1, G_2, F.
while not reached maximum epochs **do**
  for mini-batch x from unlabeled data **do**
    Fix the weights of G_1, G_2
    Get augmentation $x_1 = G_1(x), x_2 = G_2(x)$
    Sample two views v_1, v_2 from $\{x, x_1, x_2\}$
    $\mathcal{L} = \mathcal{L}_{\text{cl}}(v_1, v_2)$
    Update the weights of F to minimize $\mathcal{L}$
  for mini-batch x from labeled data **do**
    Get augmentation $x_1 = G_1(x), x_2 = G_2(x)$
    $\mathcal{L} = \mathcal{L}_{\text{cls}}(x, x_1, x_2) + \lambda \cdot \mathcal{L}_{\text{sim}}(x_1, x_2)$
    Update the weights of G_1, G_2, F to minimize $\mathcal{L}$

Adversarial

- Our Joint training strategy is designed to meet the InfoMin (Tian et al. 2020) Principle. We want the two view generators to generate topologically different yet semantically similar training pairs, while keep the task-related information.
- We alternatively optimize three objective function, in a naturally adversarial training style.
- Only use $\mathcal{L}_{\text{cl}}$ with learnable view generators could leads to weak/no augmentation.
- $\mathcal{L}_{\text{sim}}$ prevents too weak augmentation.
- $\mathcal{L}_{\text{cls}}$ encourages label-preserving augmentation.

Our Contribution

- We propose a graph contrastive learning framework with learnable graph view generators embedded into an auto augmentation strategy. To the best of our knowledge, this is the first work to build learnable generative node-wise augmentation policies for graph contrastive learning.
- We propose a joint training strategy for training the graph view generators, the graph encoder, and the graph classifier under the context of graph contrastive learning in an end-to-end manner.
- We extensively evaluate the proposed method on a variety of graph classification datasets with semi-supervised, unsupervised, and transfer learning settings. The t-SNE and view visualization results also demonstrate the effectiveness of our method.

Experiments

Table 3: Comparison with the existing methods for transfer learning. The **bold** numbers denote the best performance and the numbers in blue denote the second best performance.

Model	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE
No Pretrain	65.8±4.5	74.0±0.8	63.4±0.6	57.3±1.6	58.0±4.4	71.8±2.5	75.3±1.9	70.1±5.4
Infomax	68.8±0.8	75.3±0.5	62.7±0.4	58.4±0.8	69.9±3.0	75.3±2.5	76.0±0.7	75.9±1.6
EdgePred	67.3±2.4	76.0±0.6	64.1±0.6	60.4±0.7	64.1±3.7	74.1±2.1	76.3±1.0	79.9±0.9
AttrMasking	64.3±2.8	**76.7±0.4**	**64.2±0.5**	61.0±0.7	71.8±4.1	74.7±1.4	77.2±1.1	79.3±1.6
ContextPred	68.0±2.0	75.7±0.7	63.9±0.6	60.9±0.6	65.9±3.8	75.8±1.7	77.3±1.0	79.6±1.2
GraphCL	69.68±0.67	73.87±0.66	62.40±0.57	60.53±0.88	75.99±2.65	69.80±2.66	**78.47±1.22**	75.38±1.44
JOAOv2	71.39±0.92	74.27±0.62	63.16±0.45	60.49±0.74	80.97±1.64	73.67±1.00	77.51±1.17	75.49±1.27
AD-GCL	70.01±1.07	76.54±0.82	63.07±0.72	**63.28±0.79**	79.78±3.52	72.30±1.61	78.28±0.97	78.51±0.80
Ours	**73.36±0.77**	75.69±0.29	63.47±0.38	62.51±0.63	**80.99±3.38**	**75.83±1.30**	78.35±0.64	**83.26±1.13**

Table 2: Comparison with the existing methods for unsupervised learning. The **bold** numbers denote the best performance and the the numbers in blue represent the second best performance.

Model	MUTAG	PROTEINS	DD	NCI1	COLLAB	IMDB-B	REDDIT-B	REDDIT-M-5K
GL	81.66±2.11	-	-	-	-	65.87±0.98	77.34±0.18	41.01±0.17
WL	80.72±3.00	72.92±0.56	-	80.01±0.50	-	72.30±3.44	68.82±0.41	46.06±0.21
DGK	87.44±2.72	73.30±0.82	-	80.31±0.46	-	66.96±0.56	78.04±0.39	41.27±0.18
node2vec	72.63±10.20	57.49±3.57	-	54.89±1.61	-	-	-	-
sub2vec	61.05±15.80	53.03±5.55	-	52.84±1.47	-	55.26±1.54	71.48±0.41	36.68±0.42
graph2vec	83.15±9.25	73.30±2.05	-	73.22±1.81	-	71.10±0.54	75.78±1.03	47.86±0.26
InfoGraph	**89.01±1.13**	74.44±0.31	72.85±1.78	76.20±1.06	70.65±1.13	73.03±0.87	82.50±1.42	53.46±1.03
GraphCL	86.80±1.34	74.39±0.45	**78.62±0.40**	77.87±0.41	71.36±1.15	71.14±0.44	**89.53±0.84**	55.99±0.28
JOAOv2	-	71.25±0.85	66.91±1.75	72.99±0.75	70.40±2.21	71.60±0.86	78.35±1.38	45.57±2.86
AD-GCL	-	73.59±0.65	74.49±0.52	69.67±0.51	**73.32±0.61**	71.57±1.01	85.52±0.79	53.00±0.82
Ours	88.64±1.08	**75.80±0.36**	77.57±0.60	**82.00±0.29**	70.12±0.68	**73.30±0.40**	88.58±1.49	**56.75±0.18**

Table 4: Comparison with existing methods and different strategies for semi-supervised learning. **Bold** numbers denote the best performance and the numbers in blue denote the second best performance. Red is our default setting for joint training strategy.

Model	PROTEINS	DD	NCI1	COLLAB	GITHUB	IMDB-B	REDDIT-B	REDDIT-M-5K
Full Data	78.25±1.61	80.73±3.78	83.65±1.16	83.44±0.77	66.89±1.04	76.60±4.20	89.95±2.06	55.59±2.24
10% Data	69.72±6.71	74.36±5.86	75.16±2.07	74.34±2.00	61.05±1.57	64.80±4.92	76.75±5.60	49.71±3.20
10% GCA	73.85±5.56	76.74±4.09	68.73±2.36	74.32±2.30	59.24±3.21	**73.70±4.88**	77.15±6.96	32.95±10.89
10% GraphCL Aug Only	70.71±5.63	76.48±4.12	70.97±2.08	73.56±2.52	59.80±1.94	71.10±5.11	76.45±4.83	47.33±4.02
10% GraphCL CL	74.21±4.50	76.65±5.12	73.16±2.90	75.50±2.15	63.51±1.02	68.10±5.15	78.05±2.65	48.09±1.74
10% JOAOv2	73.31±0.48	75.81±0.73	74.86±0.39	75.53±0.18	**66.66±0.60**	-	88.79±0.65	52.71±0.28
10% AD-GCL	73.96±0.47	**77.91±0.73**	**75.18±0.31**	75.82±0.26	-	-	**90.10±0.15**	**53.49±0.28**
10% Our Aug Only	75.49±5.15	77.16±4.53	73.33±2.86	75.92±1.93	60.65±1.04	71.90±2.88	79.65±2.84	47.97±2.22
10% Our CL Naive	74.57±3.29	75.55±4.76	73.22±2.48	76.60±2.15	60.95±1.32	71.00±2.91	79.10±4.38	46.71±2.64
10% Our CL Joint ($\mathcal{L}_{\text{cls}}$)	74.66±2.58	76.57±5.08	71.78±1.61	75.38±2.15	60.39±1.50	70.60±4.17	78.90±3.11	46.89±3.13
10% Our CL Joint ($\mathcal{L}_{\text{cls}}$+$\mathcal{L}_{\text{sim}}$)	75.12±3.35	76.23±3.57	72.55±2.72	75.60±2.08	60.18±1.75	71.70±3.86	79.25±2.88	47.51±2.51
10% Our CL Joint ($\mathcal{L}_{\text{cl}}$ +$\mathcal{L}_{\text{cls}}$)	74.75±3.35	76.82±3.85	73.07±2.31	76.18±2.46	61.75±1.30	71.50±5.32	78.35±4.21	47.73±2.69
10% Our CL Joint ($\mathcal{L}_{\text{cl}}$ +$\mathcal{L}_{\text{cls}}$+$\mathcal{L}_{\text{sim}}$)	**75.65±2.40**	77.50±4.41	73.75±2.25	**77.16±1.48**	62.46±1.51	71.90±4.79	79.80±3.47	49.91±2.70