Graph Contrastive Learning with Augmentations

Yuning You[1*], Tianlong Chen[2*], Yongduo Sui[3], Ting Chen[4], Zhangyang Wang[2], Yang Shen[1]

[1]Texas A&M University, [2]University of Texas at Austin,

[3]University of Science and Technology of China, [4]Google Research, Brain Team

NEURAL INFORMATION PROCESSING SYSTEMS

arXiv

GitHub

➢ Motivations

- Graphs are abstracted representations of raw data with **diverse nature**, thus it is difficult to design a GNN pre-training scheme **generically beneficial** to down-stream tasks [1,2].
- **Contrastive learning** exploiting data- or task-specific **augmentations** to inject the desired feature invariance can mitigate the challenge.
- Recently, contrastive learning has renewed a surge of interest in visual representation learning [3,4], while it is **not straightforward** to be directly applied to graph representation learning.

➢ Method. Data Augmentation for Graphs

- Data augmentation aims at creating novel and realistically rational data through applying certain transformation without affecting the semantics label, remaining **under-explored for graphs**.
- We propose four general data augmentations for graphsto introduce different **prior knowledge**.
- We demonstrate that for different categories of graph datasets some data augmentations might be more desired than others due to the diversity challenge.

Table 1: Overview of data augmentations for graphs.

Data augmentation	Type	Underlying Prior
Node dropping	Nodes, edges	Vertex missing does not alter semantics.
Edge perturbation	Edges	Semantic robustness against connectivity variations.
Attribute masking	Nodes	Semantic robustness against losing partial attributes per node.
Subgraph	Nodes, edges	Local structure can hint the full semantics.

➢ Method. Graph Contrastive Learning

- Motivated by recent contrastive learning developments in visual representation learning [3,4], we propose a graph contrastive learning framework (**GraphCL**) for (self-supervised) pre-training of GNNs.
- GraphCL consists of the four major components:
  (1) Graph data augmentation to obtain two correlated views of a graph as a positive pair;
  (2) GNN-based encoder extracting graph-level representation vectors for augmented graphs;
  (3) Projection head to map augmented representations to another latent space where the contrastive loss is calculated;
  (4) Contrastive loss function enforcing maximizing the consistency between positive pairs compare with negative pairs.
- We show that GraphCL can be viewed as a kind of **mutual information maximization** between the latent representations of two kinds of augmented graphs, and furthermore can be rewrited as **a general framework** unifying a broad family of contrastive learning methods on graph-structured data.

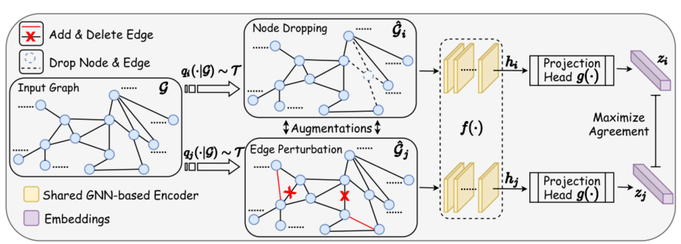

Figure 1: A framework of graph contrastive learning. Two graph augmentations $q_i(\cdot|\mathcal{G})$ and $q_j(\cdot|\mathcal{G})$ are sampled from an augmentation pool $\mathcal{T}$ and applied to input graph $\mathcal{G}$. A shared GNN-based encoder $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize the agreement between representations z_i and z_j via a contrastive loss.

➢ References

[1] Weihua Hu et al. "Strategies for Pre-Training Graph Neural Networks", ICLR 2019.
[2] Yuning You*, Tianlong Chen*, Zhangyang Wang, Yang Shen. "When Does Self-Supervision Help Graph Convolutional Networks?", ICML 2020.
[3] Kaiming He et al. "Momentum Contrast for Unsupervised Visual Representation Learning", CVPR 2020.
[4] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations", ICML 2020.

➢ Experiments. The Role of Data Augmentation in Graph Contrastive Learning

- We **systematically assess** the role of data augmentation for graph-structured data in our GraphCL framework.
- We achieve the following **observations**:
  (1) Data augmentations are crucial in graph contrastive learning: without any data augmentation graph contrastive learning is not helpful and often worse compared with training from scratch
  (2) Composing different augmentations benefits more: composing augmentation pairs of a graph rather than the graph and its augmentation further improves the performance.
  (3) Edge perturbation benefits social networks but hurts some biochemical molecules.
  (4) Applying attribute masking achieves better performance in denser graphs.
  (5) Node dropping and subgraph are generally beneficial across datasets.
  (6) Unlike "harder" ones, overly simple contrastive tasks do not help.
- In total, we decide the **augmentation pools** as: node dropping and subgraph for biochemical molecules; all for dense social networks; and all except attribute masking for sparse social networks.

Table 2: Datasets statistics.

Datasets	Category	Graph Num.	Avg. Node	Avg. Degree
NCI1	Biochemical Molecules	4110	29.87	1.08
PROTEINS	Biochemical Molecules	1113	39.06	1.86
COLLAB	Social Networks	5000	74.49	32.99
RDT-B	Social Networks	2000	429.63	1.15

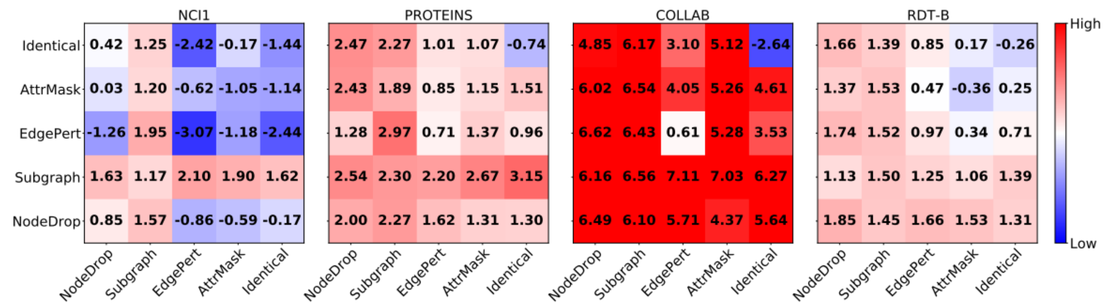

Figure 2: Semi-supervised learning accuracy gain (%) when contrasting different augmentation pairs, compared to training from scratch, under four datasets: NCI1, PROTEINS, COLLAB, and RDT-B. Pairing "Identical" stands for a no-augmentation baseline for contrastive learning, where the positive pair diminishes and the negative pair consists of two non-augmented graphs. Warmer colors indicate better performance gains. The baseline training-from-scratch accuracies are 60.72%, 70.40%, 57.46%, 86.63% for the four datasets respectively.

➢ Experiments. Comparison with the State-of-the-art Methods

- We compare our proposed GraphCL, with state-of-the-art methods in the settings of semi-supervised, unsupervised, transfer learning and adversarial robustness on graph classification.
- Experiment results verify **the state-of-the-art performance** of our proposed framework in both generalizability and robustness.

Table 3: Semi-supervised learning with pre-training & finetuning. Red numbers indicate the best performance and the number that overlap with the standard deviation of the best performance (comparable ones). 1% or 10% is label rate; baseline and Aug. represents training from scratch without and with augmentations, respectively.

Dataset	NCI1	PROTEINS	DD	COLLAB	RDT-B	RDT-M5K	GITHUB	MNIST	CIFAR10
1% baseline	60.72±0.45	-	-	57.46±0.25	-	-	54.25±0.22	60.39±1.95	27.36±0.75
1% Aug.	60.49±0.46	-	-	58.40±0.97	-	-	56.36±0.42	67.43±0.36	27.39±0.44
1% GAE	61.63±0.84	-	-	63.20±0.67	-	-	59.44±0.44	57.58±2.07	21.09±0.53
1% Infomax	62.72±0.65	-	-	61.70±0.77	-	-	58.99±0.50	63.24±0.78	27.86±0.43
1% GraphCL	62.55±0.86	-	-	64.57±1.15	-	-	58.56±0.59	83.41±0.33	30.01±0.84
10% baseline	73.72±0.24	70.40±1.54	73.56±0.41	73.71±0.27	86.63±0.27	51.33±0.44	60.87±0.17	79.71±0.65	35.78±0.81
10% Aug.	73.59±0.32	70.29±0.64	74.30±0.81	74.19±0.13	87.74±0.39	52.01±0.20	60.91±0.32	83.99±2.19	34.24±2.62
10% GAE	74.36±0.24	70.51±0.17	74.54±0.68	75.09±0.19	87.69±0.40	53.58±0.13	63.89±0.52	86.67±0.93	36.35±1.04
10% Infomax	74.86±0.26	72.27±0.40	75.78±0.34	73.76±0.29	88.66±0.95	53.61±0.31	65.21±0.88	83.34±0.24	41.07±0.48
10% GraphCL	74.63±0.25	74.17±0.34	76.17±1.37	74.23±0.21	89.11±0.19	52.55±0.45	65.81±0.79	93.11±0.17	43.87±0.77

Table 4: Comparing classification accuracy on top of graph representations learned from graph kernels, SOTA representation learning methods, and GIN pre-trained with GraphCL. The compared numbers are from the corresponding papers under the same experiment setting.

Dataset	NCI1	PROTEINS	DD	MUTAG	COLLAB	RDT-B	RDT-M5K	IMDB-B
GL	-	-	-	81.66±2.11	-	77.34±0.18	41.01±0.17	65.87±0.98
WL	80.01±0.50	72.92±0.56	-	80.72±3.00	-	68.82±0.41	46.06±0.21	72.30±3.44
DGK	80.31±0.46	73.30±0.82	-	87.44±2.72	-	78.04±0.39	41.27±0.18	66.96±0.56
node2vec	54.89±1.61	57.49±3.57	-	72.63±10.20	-	-	-	-
sub2vec	52.84±1.47	53.03±5.55	-	61.05±15.80	-	71.48±0.41	36.68±0.42	55.26±1.54
graph2vec	73.22±1.81	73.30±2.05	-	83.15±9.25	-	75.78±1.03	47.86±0.26	71.10±0.54
InfoGraph	76.20±1.06	74.44±0.31	72.85±1.78	89.01±1.13	70.65±1.13	82.50±1.42	53.46±1.03	73.03±0.87
GraphCL	77.87±0.41	74.39±0.45	78.62±0.40	86.80±1.34	71.36±1.15	89.53±0.84	55.99±0.28	71.14±0.44

Table 5: Transfer learning comparison with different manually designed pre-training schemes, where the compared numbers are from [9].

Dataset	BBBP	Tox21	ToxCast	SIDER	ClinTox	MUV	HIV	BACE	PPI
No Pre-Train	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	64.8±1.0
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	64.1±1.5
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	65.7±1.3
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	65.2±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	64.4±1.3
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	67.88±0.85

Table 6: Adversarial performance under three adversarial attacks for GNN with different depth (following the protocol in [60]). Red numbers indicate the best performance.

Methods	Two-Layer		Three-Layer		Four-Layer	
	No Pre-Train	GraphCL	No Pre-Train	GraphCL	No Pre-Train	GraphCL
Unattack	93.20	94.73	98.20	98.33	98.87	99.00
RandSampling	78.73	80.68	92.27	92.60	95.13	97.40
GradArgmax	69.47	69.26	64.60	89.33	95.80	97.00
RL-S2V	42.93	42.20	41.93	61.66	70.20	84.86

*Equal contribution. Contact: {yuning.you,yshen}@tamu.edu, {tianlong.chen,atlaswang}@utexas.edu, syd2019@mail.ustc.edu.cn, iamtingchen@google.com