The 30th Web Conference (WWW 2021)

Graph Contrastive Learning with Adaptive Augmentation

Presented by **Yanqiao ZHU**

yanqiao.zhu@cripac.ia.ac.cn

@ https://SXKDZ.github.io

Center for Research on Intelligent Perception and Computing
National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences

Joint work with Yichen XU, Feng YU, Qiang LIU, Shu WU, and Liang WANG

Outline

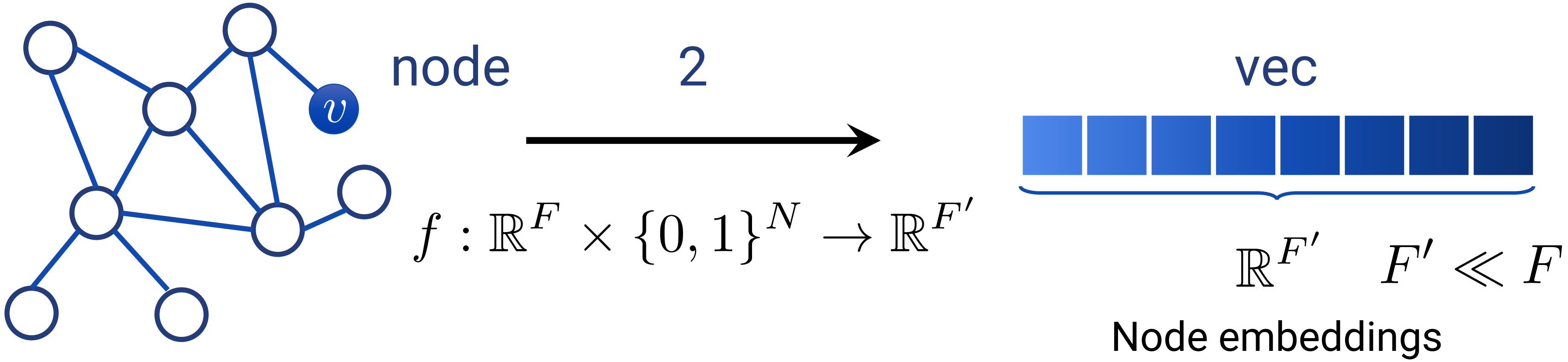
1. Preamble
2. The Proposed Method
3. Experiments
4. Concluding Remarks

Outline

1. **Preamble**
2. The Proposed Method
3. Experiments
4. Concluding Remarks

Representation Learning on Graphs

- Goal: efficient feature learning for machine learning on graphs
- Low-dimensional node embeddings encode both structural and attributive information.



Self-supervised learning comes to rescue!

- Most GNN models are established in a supervised manner.
  - It is often expensive to obtain high-quality labels at scale in real world.
  - Supervised models learn the inductive bias encoded in labels, instead of reusable, task-invariant knowledge.

> *"Labels are the opium of the machine learning researcher."*
>
> --- Jitendra Malik
>
> *"The future is self-supervised!"*
>
> --- Yann LeCun

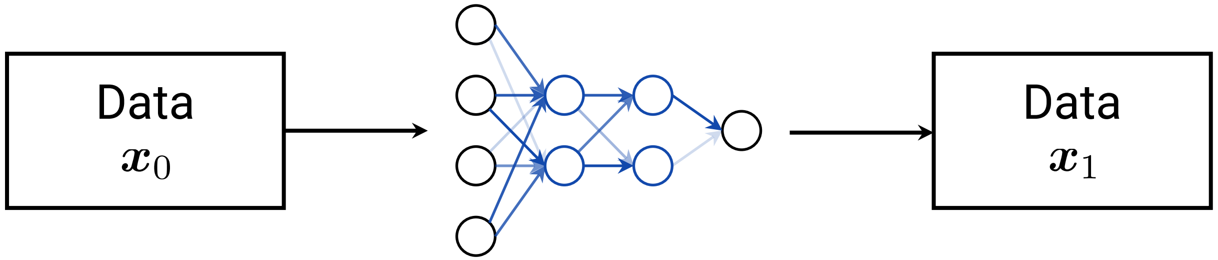
Self-supervised learning comes to rescue!

- Most GNN models are established in a supervised manner.
  - It is often expensive to obtain high-quality labels at scale in real world.
  - Supervised models learn the inductive bias encoded in labels, instead of reusable, task-invariant knowledge.
- Self-supervised methods employ **proxy tasks** to guide learning the representations.
  - The proxy task is designed to predict any part of the input from any other observed part.
  - Typical proxy tasks for visual data include corrupted image restoration, rotation angle prediction, reorganization of shuffled patches, etc.

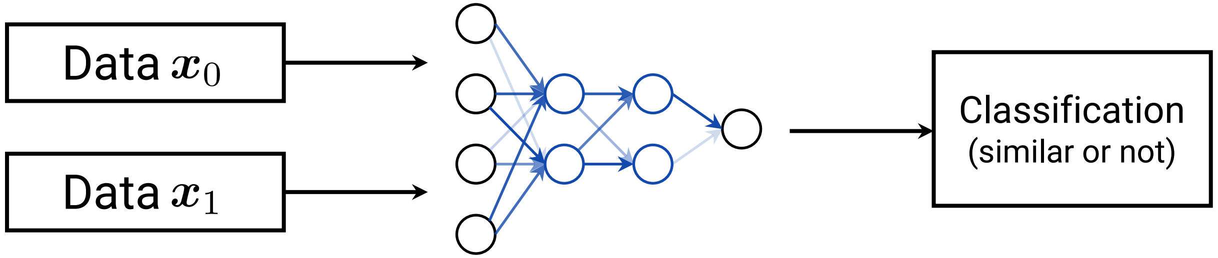
[Jing and Tian, 2020] L. Jing and Y. Tian, Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey, *TPAMI*, 2020.

Taxonomy of Self-Supervised Learning

- (a) Generative/predictive: loss measured in the output space



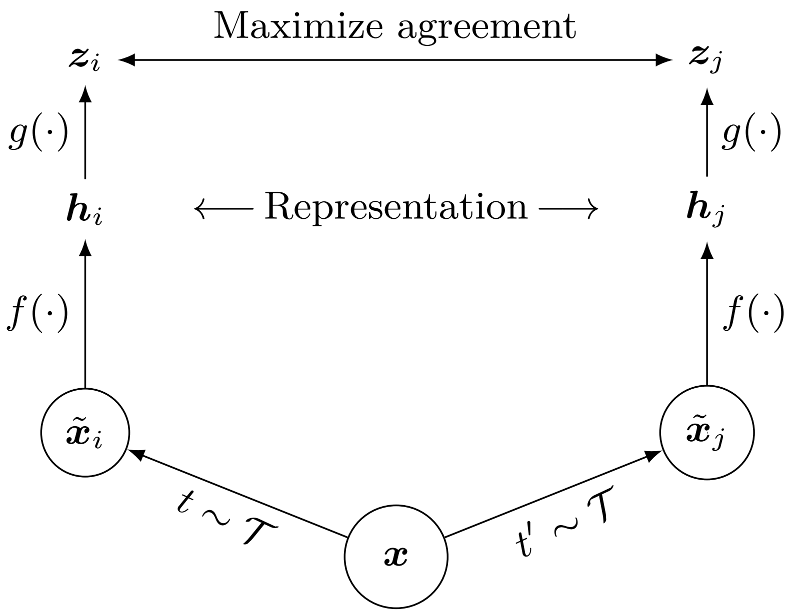
- (b) Contrastive: loss measured in the latent space



The Contrastive Learning Paradigm

- Contrastive learning aims to maximize the agreement of latent representations under stochastic data augmentation.

- Three main components:
  - Data augmentation pipeline $\mathcal{T}$
  - Encoder f and representation extractor g
  - Contrastive mode and objective $\mathcal{L}$



[Chen et al., 2020] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, A Simple Framework for Contrastive Learning of Visual Representations, in *ICML*, 2020.

Contrastive Learning Objectives

- A common pattern:

$$s(f(\boldsymbol{x}), f(\boldsymbol{x}^+)) \gg s(f(\boldsymbol{x}), f(\boldsymbol{x}^-))$$

  - $f(\cdot)$ is the encoder, e.g., CNN and GNN.
  - $s(\cdot, \cdot)$ measures similarity between two embeddings.
- Usually implemented with an N-way softmax function:

$$\mathcal{L} = -\mathbb{E}_X \left[\log \frac{\exp(s(\boldsymbol{x}, \boldsymbol{x}^+))}{\exp(s(\boldsymbol{x}, \boldsymbol{x}^+)) + \sum_{j=1}^{N-1} \exp(s(\boldsymbol{x}, \boldsymbol{x}_j))} \right]$$

  - Commonly referred to as **the InfoNCE loss**.
  - The critic function can be simply implemented as $s(\boldsymbol{x}, \boldsymbol{y}) = g(\boldsymbol{x})^\top g(\boldsymbol{y})$.

[Oord et al., 2018] A. van den Oord, Y. Li, and O. Vinyals, Representation Learning with Contrastive Predictive Coding, arXiv.org, vol. cs.LG. 2018.

Contrastive Learning Objectives

- A common pattern:

$$s(f(\boldsymbol{x}), f(\boldsymbol{x}^+)) \gg s(f(\boldsymbol{x}), f(\boldsymbol{x}^-))$$

  - $f(\cdot)$ is the encoder, e.g., CNN and GNN.
  - $s(\cdot, \cdot)$ measures similarity between two embeddings.
- Usually implemented with an N-way softmax function:

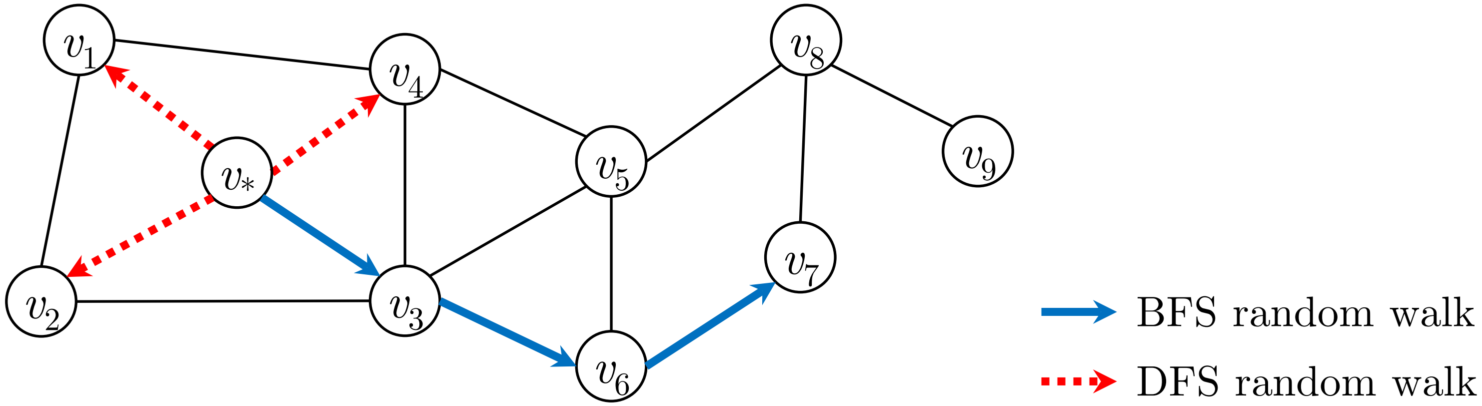
$$\mathcal{L} = -\mathbb{E}_X \left[\log \frac{\exp(s(\boldsymbol{x}, \boldsymbol{x}^+))}{\exp(s(\boldsymbol{x}, \boldsymbol{x}^+)) + \sum_{j=1}^{N-1} \exp(s(\boldsymbol{x}, \boldsymbol{x}_j))} \right]$$

Distinguish a pair of representations from two augmentations of the same sample (positives) apart from $(N-1)$ pairs of representations from different samples (negatives).

[Oord et al., 2018] A. van den Oord, Y. Li, and O. Vinyals, Representation Learning with Contrastive Predictive Coding, arXiv.org, vol. cs.LG. 2018.

Traditional Graph Contrastive Learning

- Traditional work of network embedding inherently follows a contrastive paradigm originated in the skip-gram model.
  - Nodes appearing on the same random walk are considered as positive samples and are encouraged to share similar embeddings.
  - Network embedding schemes could be regarded as reconstructing a preset graph proximity matrix, having difficulty in leveraging attributes.



[Grover and Leskovec, 2016] A. Grover and J. Leskovec, node2vec: Scalable Feature Learning for Networks, in *KDD*, 2016.

Deep Graph Contrastive Learning

- GNNs employ more powerful encoders for learning representations by aggregating information from neighborhood.

- GNN-based contrastive learning studies are in their infancy. Existing work primarily differs in **contrastive modes** and **data augmentation** techniques.
  - Contrastive mode: defines which embeddings to pull together or push apart.
  - Data augmentation: transforms the original graphs to congruent counterparts.

Graph Contrastive Modes

- Global-local contrastive learning:
  - DGI [Veličković et al., 2019] and MVGRL [Hassani and Khasahmadi, 2020] maximize the agreement between node- and graph-level representations.
  - The graph readout function should be injective [Xu et al., 2019], which is hard to fulfill. Otherwise, it is not guaranteed to distill enough information from node-level embeddings.
- Local-local contrastive learning:
  - Follow-up work GCC [Qiu et al., 2020], GRACE [Zhu et al., 2020], and GraphCL [You et al., 2020] eschew the need of an injective readout function and directly maximize the agreement of node embeddings across two augmented views.

Augmentation for Graph CL

- Existing studies mostly adopt a bi-level augmentation scheme:
  - Attribute-level augmentation
    - Dropping / masking features [You et al., 2020; Zhu et al., 2020]
    - Adding Gaussian noise
    - …
  - Structure-level augmentation
    - Shuffling the adjacency matrix [Veličković et al., 2019]
    - Adding / dropping edges [You et al., 2020; Zhu et al., 2020]
    - Sampling subgraphs [Hassani and Khasahmadi, 2020; Qiu et al., 2020; You et al., 2020]
    - Generating global view via diffusion kernels [Hassani and Khasahmadi, 2020]
    - …

Augmentation for Graph CL (cont.)

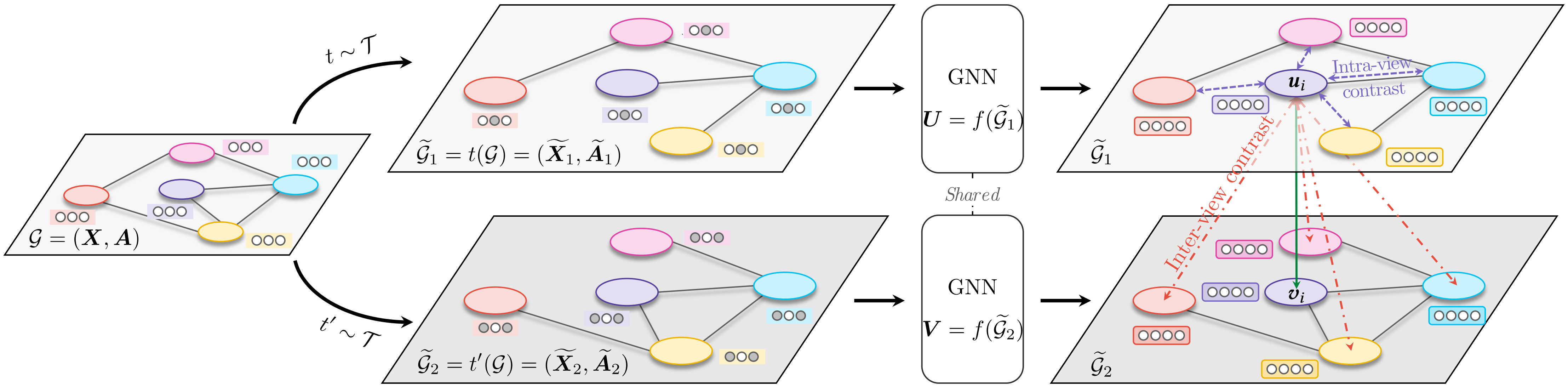
- How to integrate augmentation schemes into graph CL is still an empirical choice.
- In essence, CL seeks to learn representations that are insensitive to perturbation induced by augmentation schemes.
  - In the graph domain, there is discrepancy in the impact of nodes and edges exists.
  - Augmentation should thereby preserve important structural and attribute information of graphs.

[Wu et al., 2020] M. Wu, C. Zhuang, M. Mosse, D. Yamins, and N. Goodman, On Mutual Information in Contrastive Learning for Visual Representations, arXiv.org, vol. cs.LG. 27-May-2020.
[Xiao et al., 2020] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, What Should Not Be Contrastive in Contrastive Learning, arXiv.org, vol. cs.CV. 13-Aug-2020.

Outline

1. Preamble
2. **The Proposed Method**
3. Experiments
4. Concluding Remarks

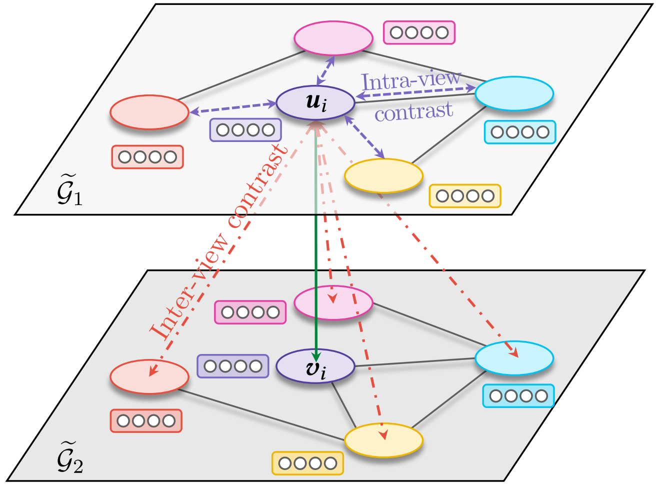
The Proposed Approach: GCA



Graph Contrastive Learning Across Views

- Firstly, we generate two correlated graph views by randomly augmenting the structure and features.
- Then, we train the model using a contrastive loss to maximize the agreement between node embeddings in the latent space.

$$\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) = \log \frac{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}{\underbrace{e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_i)/\tau}}_{\text{positives}} + \underbrace{\sum_{k \neq i} e^{\theta(\boldsymbol{u}_i, \boldsymbol{v}_k)/\tau}}_{\text{inter-view negatives}} + \underbrace{\sum_{k \neq i} e^{\theta(\boldsymbol{u}_i, \boldsymbol{u}_k)/\tau}}_{\text{intra-view negatives}}}$$



Adaptive Augmentation on Graphs

- Bi-level augmentation to jointly consider topology and attributes:
  - Remove edges at the topology level
  - Mask features at the attribute level

- Data augmentation should be adaptive to the given graph.
  - We propose to keep important structures and attributes unchanged by setting the removal probability inversely proportional to importance scores of edges/attributes.
  - From an amortized perspective, we emphasize important structures and attributes over randomly corrupted views.

Topology-level Augmentation

- We sample a modified edge subset $\widetilde{\mathcal{E}}$ with probability

$$P\{(u,v) \in \widetilde{\mathcal{E}}\} = 1 - p^e_{uv}.$$

- In network science, **node centrality** $\varphi_c(\cdot)$ is an often used measure that quantifies the influence of a node in the graph.
- The **edge importance** w^e_{uv} for edge (u,v) can be defined based on the centrality of two connected nodes.
  - Directed graphs: $w^e_{uv} = \varphi_c(v)$
  - Undirected graphs: $w^e_{uv} = (\varphi_c(u) + \varphi_c(v))/2$

[Newman, 2018] M. E. J. Newman, Networks: An Introduction (Second Edition), Oxford University Press, 2018.

Topology-level Augmentation (cont.)

- Alleviate the nodes with heavily dense connections:

$$s^e_{uv} = \log w^e_{uv}.$$

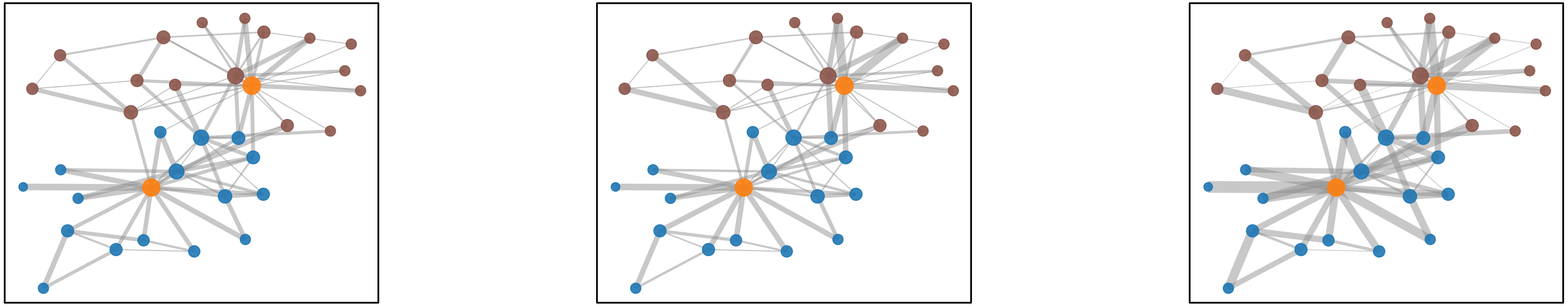
- Normalize to avoid overly high removal probabilities:

$$p^e_{uv} = \min\left(\frac{s^e_{\max} - s^e_{uv}}{s^e_{\max} - \mu^e_s} \cdot p_e,\ \ p_\tau\right),$$

  - p_e is a hyperparameter that controls the overall removing probability.
  - $s^e_{\max}$ and μ^e_s is the maximum and average of s^e_{uv} .
  - $p_\tau < 1$ is a cut-off probability.

Centrality Measures

- We consider three well-known centrality measures:



(a) Degree (b) Eigenvector (c) PageRank

- Visualized on the Karate club dataset.
- The three measures all highlight connection around central nodes (the two coaches) and exhibit negligible performance difference.

Attribute-level Augmentation

- We add noise to node attributes via randomly masking a fraction of dimensions with zeros in node features:

$$\widetilde{m}_i \sim \mathrm{Bern}(1 - p_i^f), \quad \forall i,$$
$$\widetilde{\boldsymbol{X}} = [\boldsymbol{x}_1 \circ \widetilde{\boldsymbol{m}}; \;\; \boldsymbol{x}_2 \circ \widetilde{\boldsymbol{m}}; \;\; \cdots; \;\; \boldsymbol{x}_N \circ \widetilde{\boldsymbol{m}}]^\top.$$

- The importance for each dimension of node features can be derived from node centrality scores.
  - Assumption: feature dimensions frequently appearing in influential nodes should be important.

$$w_i^f = \sum_{u \in \mathcal{V}} x_{ui} \cdot \varphi_c(u),$$

  - $x_{ui} \in \{0, 1\}$ indicate the occurrence of dimension i in node u.

Theoretical Groundings

Definition 1. Mutual Information.

- Mutual information (MI) $I(X;Y)$ is a measure of the mutual dependence between the two random variables X and Y, determining how different the joint distribution of the pair $P(X,Y)$ is to the marginal $P(X)P(Y)$.

Definition 2. InfoMax Principle.

- A function that maps a set of input values I to a set of output values O should be learned so as to maximize the MI between I and O.

[Linsker, 1998] R. Linsker, Self-Organization in a Perceptual Network, *IEEE Computer*, 1988.

Theoretical Groundings (cont.)

Theorem 1. Connections to MI maximization.

- Let $\boldsymbol{X}_i = \{\boldsymbol{x}_k\}_{k \in \mathcal{N}(i)}$ be the neighborhood of node v_i that collectively maps to its output embedding, where $\mathcal{N}(i)$ denotes the set of neighbors of node v_i specified by GNN architectures, and $\boldsymbol{X}$ be the corresponding random variable with a uniform distribution $p(\boldsymbol{X}) = {}^1\!/_N$.
- Given two random variables $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{F'}$ being the embedding in the two views, with their joint distribution denoted as $P(\boldsymbol{U}, \boldsymbol{V})$, our objective $\mathcal{J}$ is a lower bound of MI between input $\boldsymbol{X}$ and node representations in two graph views $\boldsymbol{U}, \boldsymbol{V}$:

$$\mathcal{J} \leq I(\boldsymbol{X}; \boldsymbol{U}, \boldsymbol{V}).$$

Theoretical Groundings (cont.)

Theorem 2. Connections to the triplet loss.

- When the projection function g is the identity function, and we measure embedding similarity by simply taking the inner product, i.e. $s(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v}$, and further assuming that positive pairs are far more aligned than negative pairs, i.e. $\boldsymbol{u}_i^\top \boldsymbol{v}_k \ll \boldsymbol{u}_i^\top \boldsymbol{v}_i$ and $\boldsymbol{u}_i^\top \boldsymbol{u}_k \ll \boldsymbol{u}_i^\top \boldsymbol{v}_i$, minimizing the pairwise objective $\ell(\boldsymbol{u}_i, \boldsymbol{v}_i)$ coincides with maximizing the triplet loss, as given in the sequel

$$-\ell(\boldsymbol{u}_i, \boldsymbol{v}_i) \propto 4\tau + \sum_{j \neq i} \left(\|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{v}_j\|^2 + \|\boldsymbol{u}_i - \boldsymbol{v}_i\|^2 - \|\boldsymbol{u}_i - \boldsymbol{u}_j\|^2 \right).$$

Outline

1. Preamble
2. The Proposed Method
3. **Experiments**
4. Concluding Remarks

Datasets

Dataset	#Nodes	#Edges	#Features	#Classes
Wiki-CS	11,701	216,123	300	10
Amazon-Computers	13,752	245,861	767	10
Amazon-Photo	7,650	119,081	745	8
Coauthor-CS	18,333	81,894	6,805	15
Coauthor-Physics	34,493	247,962	8,415	5

Baselines

- Network embedding methods:
  - DeepWalk [Perozzi et al., 2014] and node2vec [Grover and Leskovec, 2016]
- Unsupervised GNNs:
  - Recontraction-based methods: GAE, VGAE [Kipf and Welling, 2016], and GraphSAGE [Hamilton et al., 2017]
  - Contrastive learning methods: DGI [Veličković et al., 2019], GMI [Peng et al., 2020], and MVGRL [Hassani and Khasahmadi, 2020]
- Supervised GNNs:
  - GCN [Kipf and Welling, 2017] and GAT [Veličković et al., 2018]

Experimental Configurations

- Linear evaluation: unsupervised training and then employing a simple ℓ_2-regualrlized logistic regression model on the learned node embeddings.

- Evaluation metrics: node classification accuracy.

- Base model: we employ a two-layer GCN as the encoder for all baselines.

$$\mathrm{GC}_i(\boldsymbol{X}, \boldsymbol{A}) = \sigma\left(\hat{\boldsymbol{D}}^{-\frac{1}{2}} \hat{\boldsymbol{A}} \hat{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{W}_i\right),$$

$$f(\boldsymbol{X}, \boldsymbol{A}) = \mathrm{GC}_2(\mathrm{GC}_1(\boldsymbol{X}, \boldsymbol{A}), \boldsymbol{A}).$$

Overall Performance

Method	Training Data	Wiki-CS	Computers	Photo	CS	Physics
Raw features	$\boldsymbol{X}$	71.98	73.81	78.53	90.37	93.58
node2vec	$\boldsymbol{A}$	71.79	84.39	89.67	85.08	91.19
DeepWalk	$\boldsymbol{A}$	74.35	85.68	89.44	84.61	91.77
DeepWalk + features	$\boldsymbol{X}, \boldsymbol{A}$	77.21	86.28	90.05	87.70	94.90
GAE	$\boldsymbol{X}, \boldsymbol{A}$	70.15	85.27	91.62	90.01	94.92
VGAE	$\boldsymbol{X}, \boldsymbol{A}$	75.63	86.37	92.20	92.11	94.52
DGI	$\boldsymbol{X}, \boldsymbol{A}$	75.35	83.95	91.61	92.15	94.51
GMI	$\boldsymbol{X}, \boldsymbol{A}$	74.85	82.21	90.68	OOM	OOM
MVGRL	$\boldsymbol{X}, \boldsymbol{A}$	77.52	87.52	91.74	92.11	95.33
GCA-DE	$\boldsymbol{X}, \boldsymbol{A}$	78.30	**87.85**	92.49	**93.10**	95.68
GCA-PR	$\boldsymbol{X}, \boldsymbol{A}$	**78.35**	87.80	**92.53**	93.06	95.72
GCA-EV	$\boldsymbol{X}, \boldsymbol{A}$	78.23	87.54	92.24	92.95	**95.73**
GCN	$\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Y}$	77.19	86.51	92.42	<u>93.03</u>	<u>95.65</u>
GAT	$\boldsymbol{X}, \boldsymbol{A}, \boldsymbol{Y}$	<u>77.65</u>	<u>86.93</u>	<u>92.56</u>	92.31	95.47

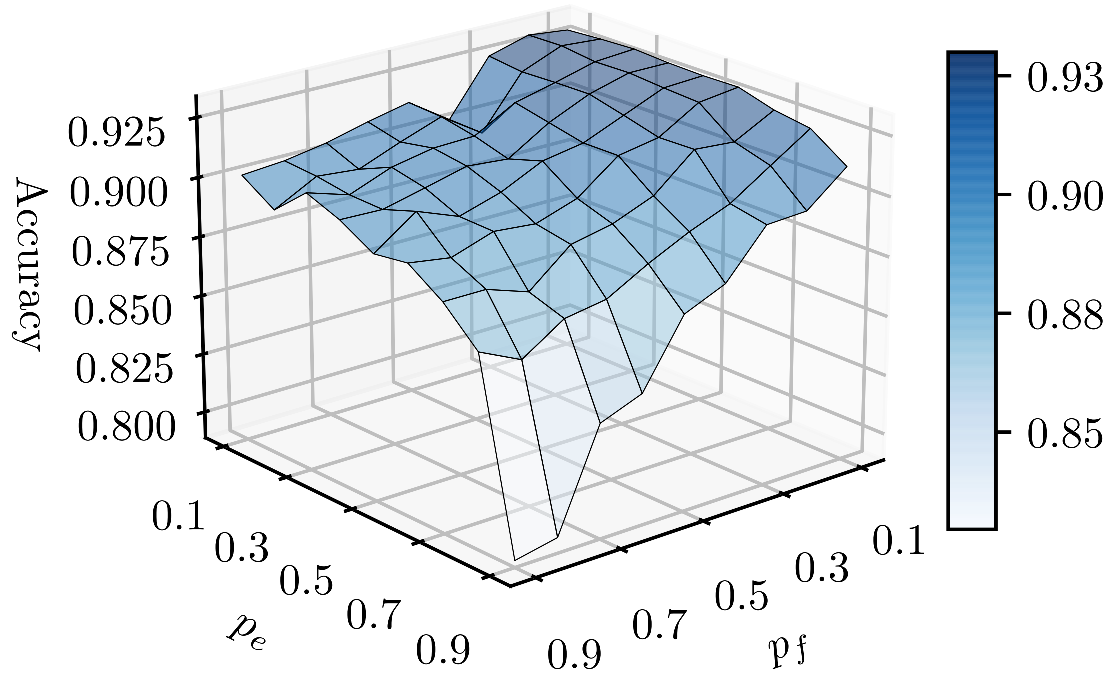
Ablation Studies

- GCA–T–A (GRACE): uniform augmentation.
- GCA–T and GCA–A: substitute the topology and the attribute augmentation scheme with uniform sampling respectively.

Variant	Topology	Attribute	Wiki-CS	Computers	Photo	CS	Physics
GCA–T–A	Uniform	Uniform	78.19	86.25	92.15	92.93	95.26
GCA–T	Uniform	Adaptive	78.23	86.72	92.20	93.07	95.59
GCA–A	Adaptive	Uniform	78.25	87.66	92.23	93.02	95.54
GCA	Adaptive	Adaptive	**78.30**	**87.85**	**92.49**	**93.10**	**95.68**

Sensitivity Analysis

- Vary the removal and masking probabilities from 0.1 to 0.9 to see the robustness under different magnitudes of perturbation.



Outline

1. Preamble
2. The Proposed Method
3. Experiments
4. **Concluding Remarks**

Wrapping Up

1. We have developed a novel graph CL framework GCA with adaptive augmentation.
2. We argue that important nodes/attributes should be preserved during augmentation to force the model learn intrinsic patterns of graphs. Specifically, we set the removal probability inversely proportional to centrality scores of the edges/attributes to reflect their importance.
3. Our proposed method achieves SOTA performance and bridges the gap between unsupervised and supervised learning.

Graph SSL: Retrospect and Prospect

- Graph self-supervised learning (SSL) is a promising way to learn graph embeddings without human annotations.
- Stemming from traditional network embedding approaches, graph CL has established a new paradigm for unsupervised representation learning on graphs.
- However, the development of graph CL remains nascent, yet calls for a principled understanding of it.
  - Utilization of both topology and attribute spaces
  - Data augmentation and positive/negative sampling on graphs
  - Contrastive objectives
  - …

Useful Resources

- A curated list of must-read papers, surveys, and talks
  - `http://bit.ly/GraphSSL`

- Graph contrastive learning library for PyTorch
  - To be released in late March
  - `http://bit.ly/GraphCL`

Bibliographies (1/2)

[Chen et al., 2020] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, A Simple Framework for Contrastive Learning of Visual Representations, in *ICML*, 2020.

[Grover and Leskovec, 2016] A. Grover and J. Leskovec, node2vec: Scalable Feature Learning for Networks, in *KDD*, 2016.

[Hamilton et al., 2017] W. L. Hamilton, Z. Ying, and J. Leskovec, Inductive Representation Learning on Large Graphs, in *NIPS*, 2017.

[Hassani and Khasahmadi, 2020] K. Hassani and A. H. Khasahmadi, Contrastive Multi-View Representation Learning on Graphs, in *ICML*, 2020.

[Jing and Tian, 2020] L. Jing and Y. Tian, Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey, *TPAMI*, 2020.

[Kipf and Welling, 2016] T. N. Kipf and M. Welling, Variational Graph Auto-Encoders, in *BDL@NIPS*, 2016.

[Kipf and Welling, 2017] T. N. Kipf and M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, in *ICLR*, 2017.

[Linsker, 1998] R. Linsker, Self-Organization in a Perceptual Network, *IEEE Computer*, 1988.

[Newman, 2018] M. E. J. Newman, Networks: An Introduction (Second Edition), Oxford University Press, 2018.

[Oord et al., 2018] A. van den Oord, Y. Li, and O. Vinyals, Representation Learning with Contrastive Predictive Coding, arXiv.org, vol. cs.LG. 2018.

Bibliographies (2/2)

[Peng et al., 2020] Z. Peng, W. Huang, M. Luo, Q. Zheng, Y. Rong, T. Xu, and J. Huang, Graph Representation Learning via Graphical Mutual Information Maximization, in *WWW*, 2020.

[Perozzi et al., 2014] B. Perozzi, R. Al-Rfou, and S. Skiena, DeepWalk: Online Learning of Social Representations, in *KDD*, 2014.

[Qiu et al., 2020] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training, in *KDD*, 2020.

[Veličković et al., 2018] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, Graph Attention Networks, in *ICLR*, 2018.

[Veličković et al., 2019] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, Deep Graph Infomax, in *ICLR*, 2019.

[Wu et al., 2020] M. Wu, C. Zhuang, M. Mosse, D. Yamins, and N. Goodman, On Mutual Information in Contrastive Learning for Visual Representations, arXiv.org, vol. cs.LG. 27-May-2020.

[Xiao et al., 2020] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, What Should Not Be Contrastive in Contrastive Learning, arXiv.org, vol. cs.CV. 13-Aug-2020.

[Xu et al., 2019] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, How Powerful are Graph Neural Networks?, in *ICLR*, 2019.

[You et al., 2020] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, Graph Contrastive Learning with Augmentations, in *NeurIPS*, 2020.

[Zhu et al., 2020] Y. Zhu, Y. Xu, F. Yu, Q. Liu, S. Wu, and L. Wang, Deep Graph Contrastive Representation Learning, in *GRL+@ICML*, 2020.

THANKS

Code

Paper

Slides