Adversarial Graph Augmentation to Improve Graph Contrastive Learning

Susheel Suresh[1], Pan Li[1], Cong Hao[2], Jennifer Neville[1,3] | [1] Purdue University, [2] Georgia Tech, [3] Microsoft Research

suresh43@purdue.edu

github.com/susheels/adgcl

NEURAL INFORMATION PROCESSING SYSTEMS

Introduction

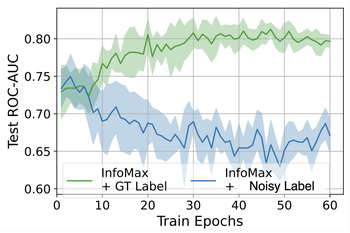
- Perturbing the attributes or structures of a graph may keep the identity of the graph. Graph contrastive learning (GCL)[2] helps to learn GNNs by maximizing the correspondence between the representations of the same graph in its different augmented forms. Current GCL approaches may be risky as they can encode information that is irrelevant to the downstream task.
- We propose adversarial-GCL (AD-GCL) that's aims to capture minimal sufficient information. This is theoretically motivated from the graph information bottleneck principle.

Noisy information suffices InfoMax principle

- Graph Contrastive Learning (GCL) leverages the InfoMax principle [1, 2]. Goal of InfoMax is not to capture the full information from the input but to let the encoder f be strong enough to keep the identity of the graph.

$$\text{InfoMax:} \quad \max_f I(G; f(G)), \quad \text{where } G \sim \mathbb{P}_{\mathcal{G}}.$$

- InfoMax principle may be risky because it may push encoders to capture redundant information that is irrelevant to the downstream tasks.
- Redundant information suffices to achieve InfoMax. Encoding it yields brittle representations with sub-optimal downstream task performance.



Two GNNs maintain InfoMax. Simultaneously with supervision from ground-truth labels (green) and noisy labels (blue) respectively. The curves show their testing performance on predicting ground-truth labels.

Capturing useful information: Graph Information Bottleneck viewpoint

$$\text{GIB:} \quad \max_f I(f(G); Y) - \beta I(G; f(G)),$$
$$\text{where } (G, Y) \sim \mathbb{P}_{\mathcal{G}\times\mathcal{Y}}, \beta \text{ is a positive constant.}$$

We can be clever if downstream labels Y are known. Easily avoid above problem !

- But, traditional GIB requires knowledge of Y, and thus is not self-supervised.
- GCL methods use graph data augmentation (GDA) to perturb the original graphs and thereby control the amount of info the representations encode.

$$\text{GDA-GCL:} \max_f I(f(t_1(G)); f(t_2(G))),$$
$$\text{where } G \sim \mathbb{P}_{\mathcal{G}}, t_i(G) \sim T_i(G), i \in \{1, 2\}.$$

Various ways to perturb viz. node dropping, edge dropping, subgraph sampling, attribute masking.

- GDAs are hand designed, domain knowledge required, dataset specific and require extensive evaluation.

Our AD-GCL principle: Learnable graph data augmentation

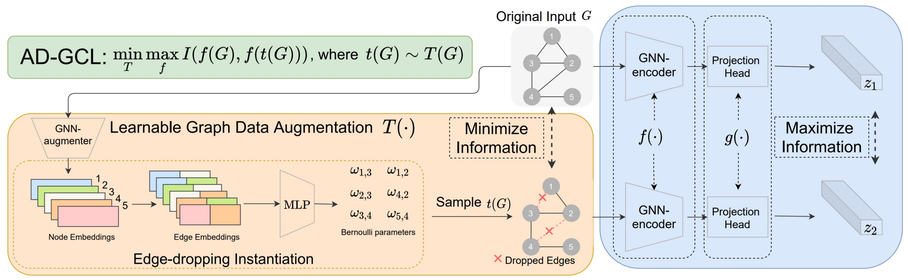
AD-GCL: We optimize the following objective, over a GDA family $\mathcal{T}$ (defined below).

$$\text{AD-GCL:} \quad \min_{T\in\mathcal{T}} \max_f I(f(G); f(t(G))), \quad \text{where } G \sim \mathbb{P}_{\mathcal{G}}, t(G) \sim T(G),$$

Definition (Graph Data Augmentation Family). *Let $\mathcal{T}$ denote a family of different GDAs $T_\Phi(\cdot)$, where Φ is the parameter in some universe. A $T_\Phi(\cdot) \in \mathcal{T}$ is a specific GDA with parameter Φ.*

- **Insight**: "Learn" the graph augmentation (GDA) process (over a parameterized family) so that the encoder can capture the **minimal information that is sufficient to identify each graph**.
- Minimal information ⇒ largest randomness/perturbation
- Even with a very aggressive GDA i.e., where *t(G)* is very different from *G*, the encoder maintains high correspondence between the perturbed graph and the original graph.
- We show we can recover a form of the GIB principle using our AD-GCL while being self-supervised.
- We give an upper bound on the irrelevant information captured by the encoder following AD-GCL and a lower bound guarantee on the mutual information between learnt representations and downstream task labels.

Practical AD-GCL Instantiation using learnable edge dropping



- Choose edge dropping as the way to perform graph augmentation. The dropping probability can be learnt using the Gumbel-Max reparameterization technique [3].
- Regularize the space of possible augmentations for control over what information is captured.

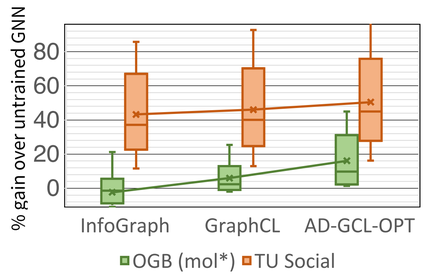
$$\min_\Phi \max_\Theta I(f_\Theta(G); f_\Theta(t(G))) + \lambda_{\text{reg}} \mathbb{E}_G\Big[\sum_{e\in E} \omega_e/|E|\Big], \text{where } G \sim \mathbb{P}_{\mathcal{G}}, t(G) \sim T_\Phi(G).$$

- Minimize the averaged edge dropping probabilities to avoid being too aggressive.

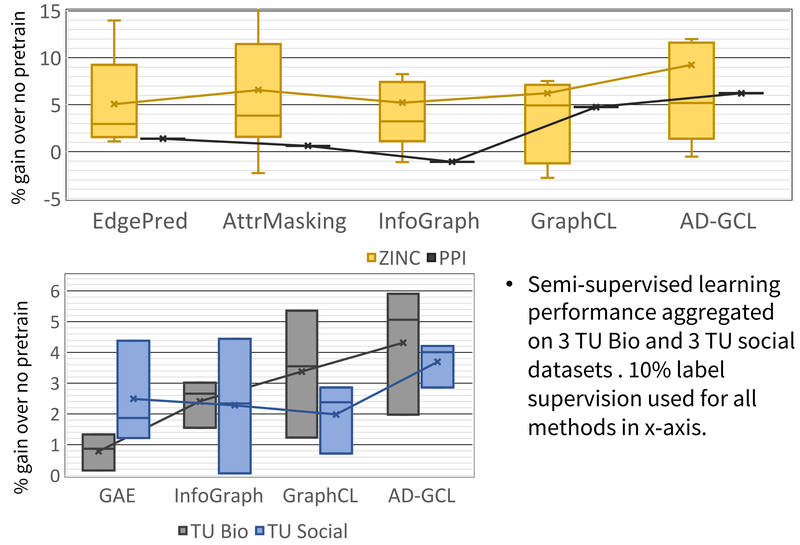
References

[1] Linsker. Computer 1988 [2] You et al. NeurIPS 2020 [3] Maddison et al. ICLR 2017
[4] Tian et al. NeurIPS 2020. [5] Tschannen et al. ICLR 2020.

Improved performance in varied learning settings

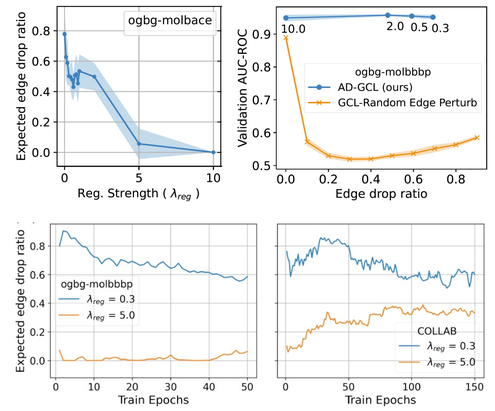


- Unsupervised learning performance aggregated on 8 OGB chemical datasets and 5 TU Social datasets for downstream graph classification task.
- Transfer learning performance aggregated fine tune tasks.
- Pretraining done using methods on x-axis.



- Semi-supervised learning performance aggregated on 3 TU Bio and 3 TU social datasets . 10% label supervision used for all methods in x-axis.
- All three learning settings showcase the superior performance of AD-GCL with clear aggregated gains. More detailed results in paper.

Sensitivity analysis of AD-GCL



- AD-GCL robust to different λ reg strength values.
- AD-GCL allows for non-uniform edge-dropping probability.
- AD-GCL pushes high drop probability on redundant edges while low drop probability on critical edges
- Training dynamics plays a vital role in our principle.