Applications and Future Directions

Kaize Ding, Arizona State University/Northwestern University

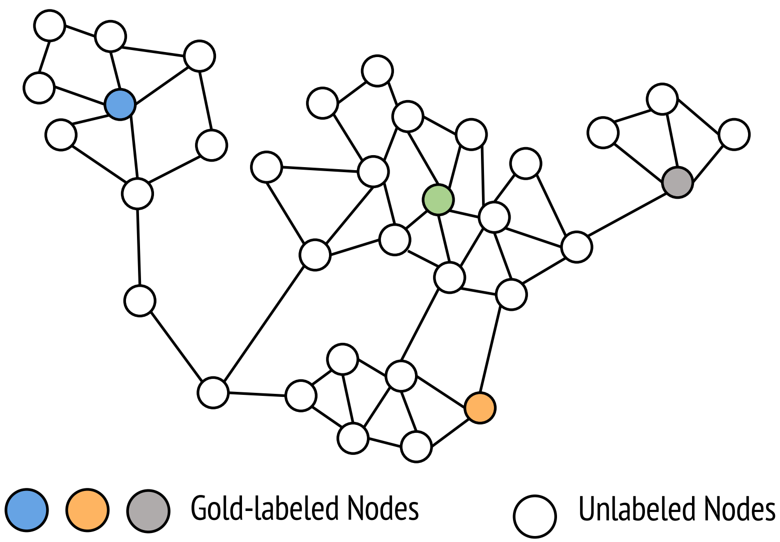
Applications of Graph Data Augmentation

- Graph Data Augmentation methods can be used for improving different Graph ML learning problems
  - Reliable Graph Learning
    - Graph Structure Learning
    - Graph Adversarial Defense
    - Graph Feature Denoising
    - …
  - Data-Efficient Graph Learning
    - Graph Semi/Weakly-Supervised Learning
    - Graph Self-Supervised Learning
    - …

Ding et al. **Data Augmentation for Deep Graph Learning: A Survey.** SIGKDD Explorations'22

Graph Semi/Weakly-Supervised Learning

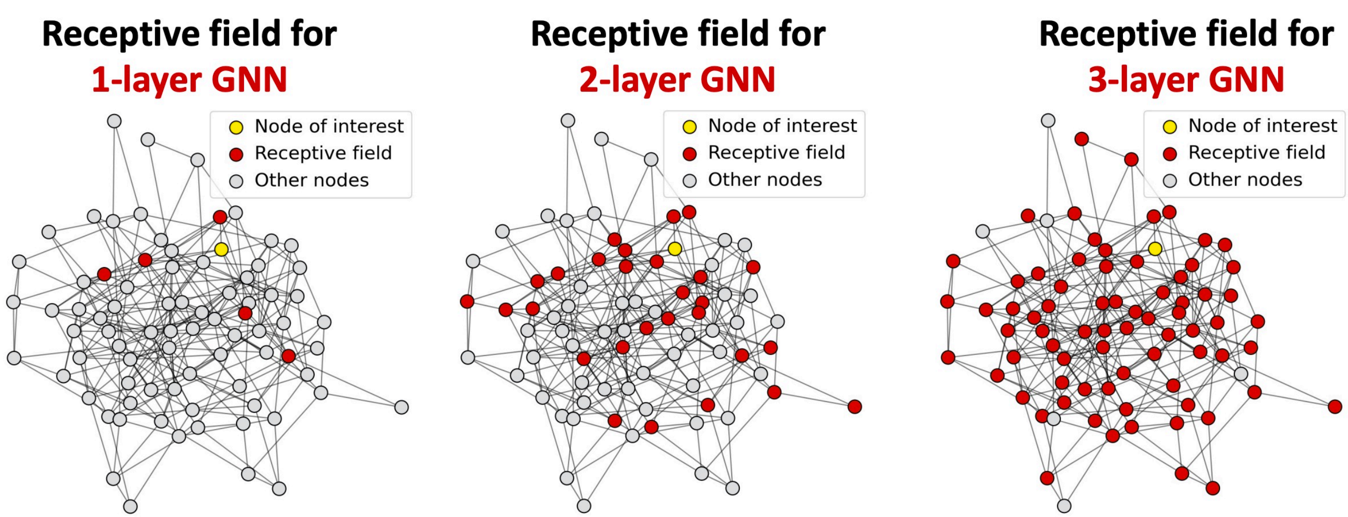
- Graph Learning with Incomplete Supervision
  - [AAAI'22] Graph few-shot semi-supervised learning
  - [WSDM'23] Graph few-shot learning with weak supervision
  - [WSDM'19] Interactive graph anomaly detection



Ding et al. **Meta Propagation Networks for Graph Few-shot Semi-supervised Learning.** AAAI'22

Learning Data-Efficient GNNs with Incomplete Supervision

- **Potential solution**: enlarge GNN receptive fields to better propagate the feature patterns of labeled nodes throughout the graph



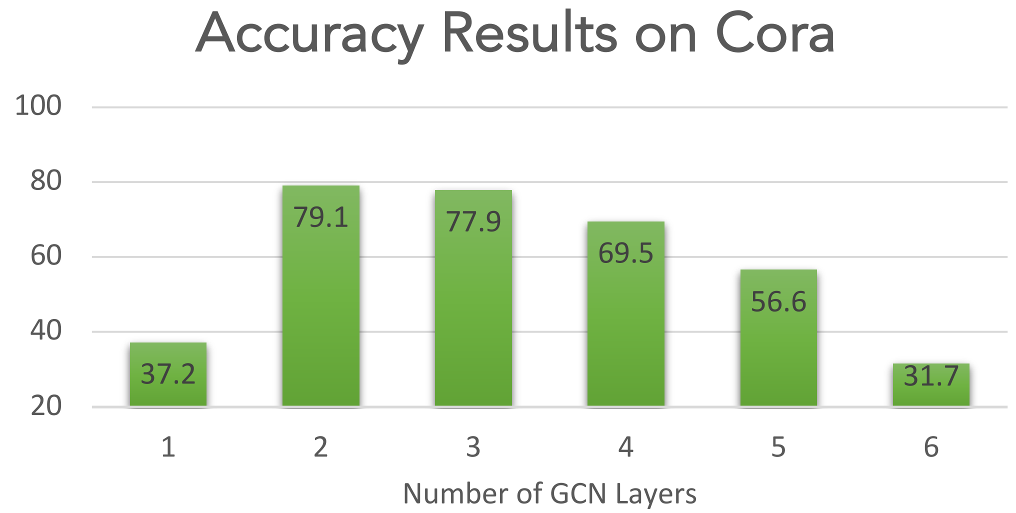
[Leskovec Stanford CS224W]

Make your GNN deeper!

When GNNs go deeper …

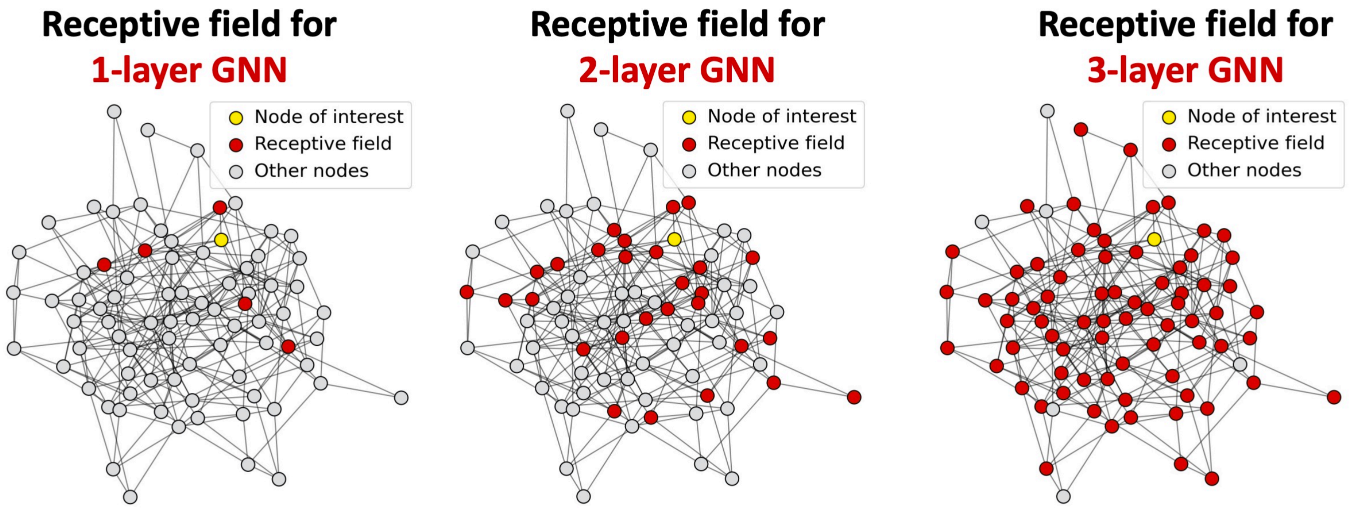
- Even though having larger receptive fields, conventional GNNs will have a catastrophic performance decrease when going deeper

Question: why larger receptive field is not working as expected?



Problem1: Oversmoothing

- The shared neighbors quickly grows when we increase the number of hops (num of GNN layers) and will cause the oversmoothing issue

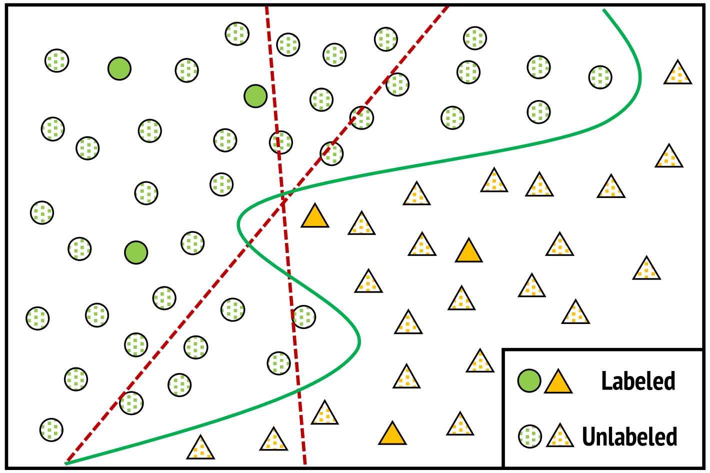


[Leskovec Stanford CS224W]

The computation graphs for different nodes are almost the same

Problem2: Overfitting

- An over-parametric deep GNN model tends to overfit limited training data, and generalizes poorly to the testing data
  - Prediction errors will be propagated along the graph structure
  - Gradient vanishing makes the training even harder

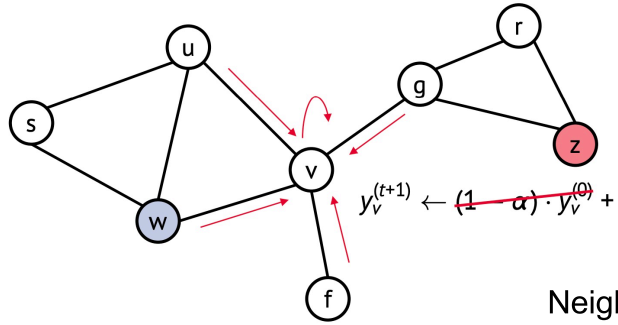


Cannot well characterize the real data distribution

Graph Data Augmentation – Label Propagation

- Label Propagation (LP) propagate labels from labeled nodes to unlabeled nodes
  - It allows large propagation steps without oversmoothing
  - It generates more labels that can mitigate overfitting

$$\hat{\mathbf{Y}} = \arg\min_{\hat{\mathbf{Y}}} \Big(\underbrace{\mathbf{Tr}(\hat{\mathbf{Y}}^{\mathrm{T}}(\mathbf{I}-\mathbf{S})\hat{\mathbf{Y}})}_{\text{smoothness constraint}} + \underbrace{\mu \|\hat{\mathbf{Y}} - \mathbf{Y}\|_2^2}_{\text{fitting constraint}} \Big)$$



$$y_v^{(t+1)} \leftarrow \cancel{(1-\alpha)\cdot y_v^{(0)}} + \alpha \cdot \frac{1}{4}\left(y_u^{(t)} + y_w^{(t)} + y_f^{(t)} + y_g^{(t)}\right)$$

Neighbor Averaging

Graph Data Augmentation – Label Propagation

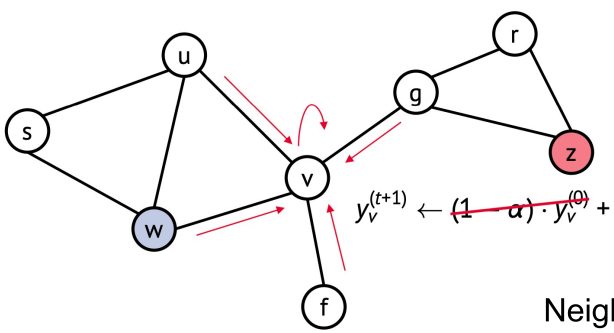
- Label Propagation (LP) propagate labels from labeled nodes to unlabeled nodes
  - It allows large propagation steps without oversmoothing
  - It generates more labels that can mitigate overfitting

Propagation matrix derived from adjacency

$$\hat{\mathbf{Y}} = \arg\min_{\hat{\mathbf{Y}}} \Big(\underbrace{\mathbf{Tr}(\hat{\mathbf{Y}}^{\mathrm{T}}(\mathbf{I}-\mathbf{S})\hat{\mathbf{Y}})}_{\text{smoothness constraint}} + \underbrace{\mu\|\hat{\mathbf{Y}}-\mathbf{Y}\|_2^2}_{\text{fitting constraint}} \Big) \Longrightarrow \mathbf{Y}^{(t+1)} = \alpha\mathbf{S}\mathbf{Y}^{(t)} + (1-\alpha)\mathbf{Y}^{(0)}$$

Personalized PageRank (PPR)

By setting appropriate **teleport probability**, LP can enable large propagation steps without losing focus on the local neighborhoods

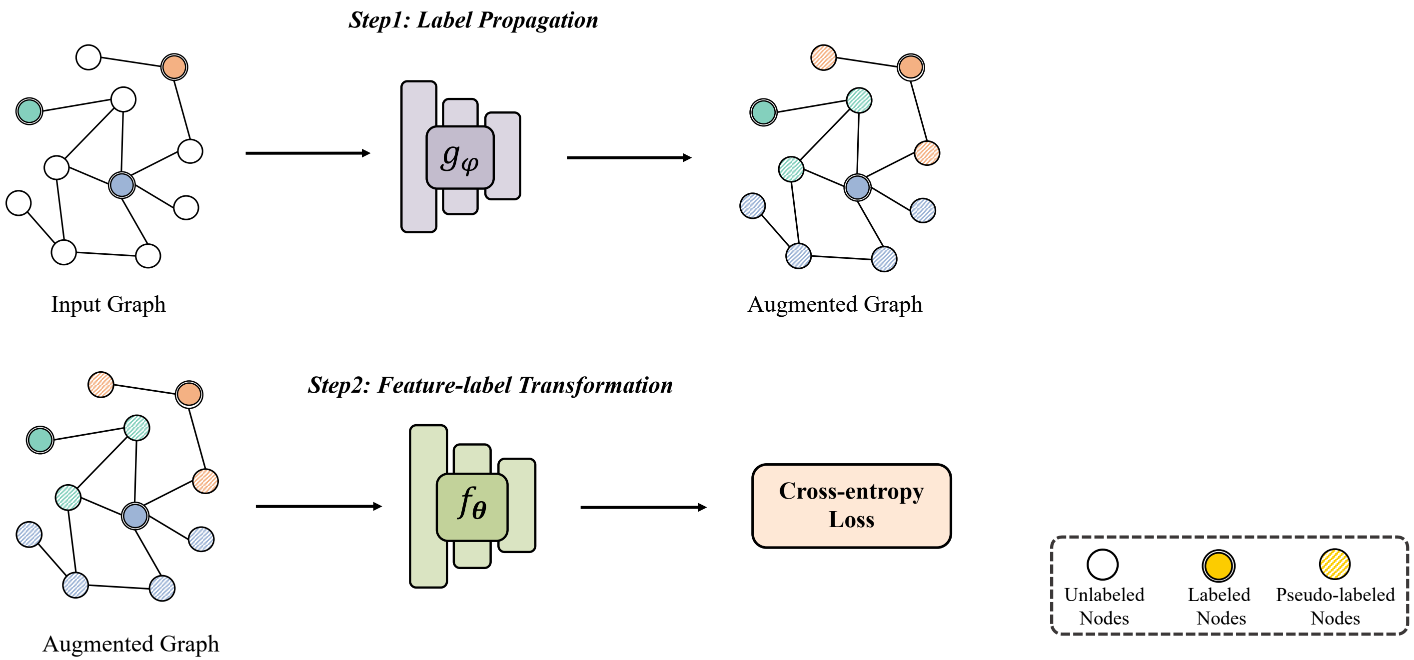


$$y_v^{(t+1)} \leftarrow \cancel{(1-\alpha)\cdot y_v^{(0)}} + \alpha \cdot \tfrac{1}{4}\left(y_u^{(t)} + y_w^{(t)} + y_f^{(t)} + y_g^{(t)}\right)$$

Neighbor Averaging

Decoupling Propagation and Transformation

- Our base model: decoupling the propagation and transformation operations in each GNN layer to two steps
  - Label Propagation (LP) + Feature Transformation (FT) is a special case of GNN

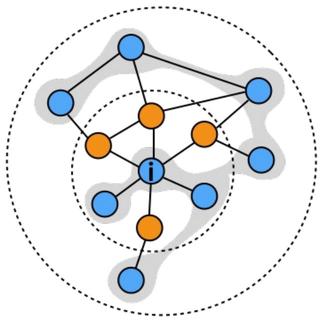


Ding et al. **Meta Propagation Networks for Graph Few-shot Semi-supervised Learning.** AAAI'22

The Blindness of Label Propagation

- LP uses fixed label propagation strategy and follows the Homophily principle, which cannot always generate meaningful pseudo labels for the target task
  - LP and FT have two disjoint objectives
  - Cannot handle graphs with different properties, different label ratios

Question: how to adaptively adjust the label propagation strategy?

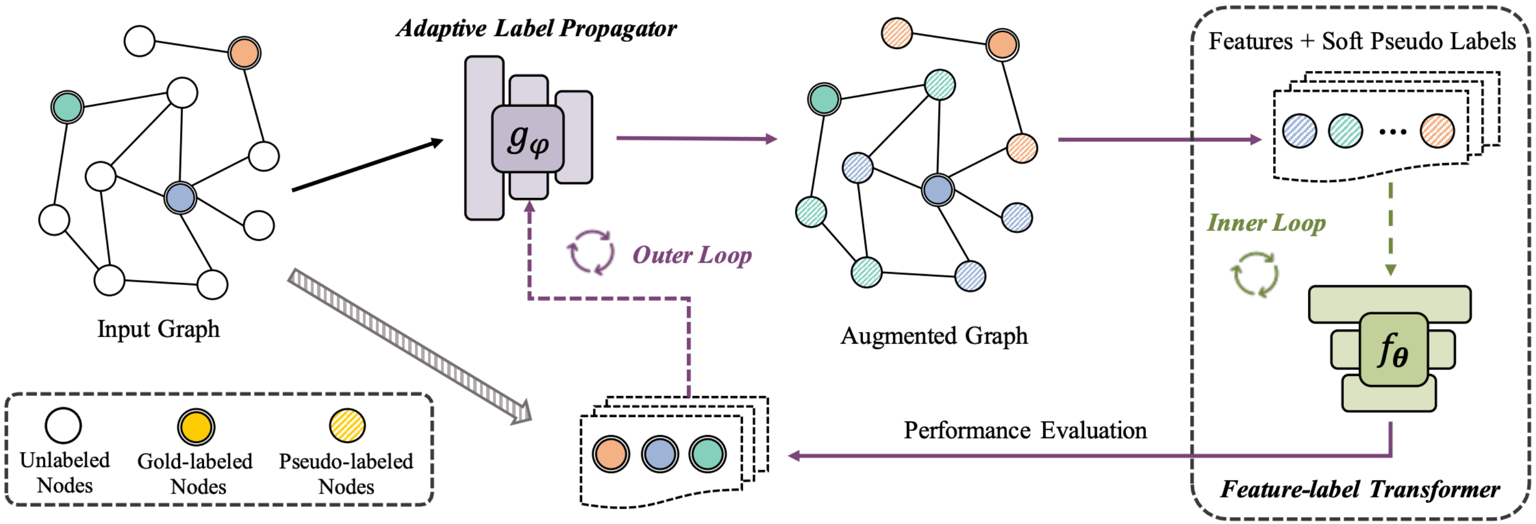


Non-homophilic Graph

Ding et al. Meta Propagation Networks for Graph Few-shot Semi-supervised Learning. AAAI'22

Learning to Propagate

- We design a meta-learning algorithm to connect the two disjoint learning objectives
  - Pseudo labels generated by the *Adaptive Label Propagator* (meta-learner) should maximize the performance of the *Feature-label Transformer* (target model) on the small set of labeled training nodes



Ding et al. **Meta Propagation Networks for Graph Few-shot Semi-supervised Learning.** AAAI'22

Meta Propagation Networks (Meta-PN)

- Adaptive Label Propagator $\boldsymbol{g_\phi}$(meta-learner)
  - Use weighting parameters (attention scores) to adjust the contribution of different propagation steps

$$\hat{\mathbf{Y}}_{i,:} = \sum_{k=0}^{K} \gamma_{ik} \mathbf{Y}_{i,:}^{(k)}, \mathbf{Y}^{(k+1)} = \mathbf{T}\mathbf{Y}^{(k)}$$

Transition Matrix

$$\gamma_{ik} = \frac{\exp\left(\mathbf{a}^{\mathrm{T}}\mathrm{ReLU}\left(\mathbf{W}\mathbf{Y}_{i,:}^{(k)}\right)\right)}{\sum_{k'=0}^{K} \exp\left(\mathbf{a}^{\mathrm{T}}\mathrm{ReLU}\left(\mathbf{W}\mathbf{Y}_{i,:}^{(k')}\right)\right)}$$

- Feature-label Transformer $\boldsymbol{f_\theta}$(target model)
  - Transform the node features to the propagated pseudo labels via MLP

$$\mathbf{P}_{i,:} = f_{\boldsymbol{\theta}}(\mathbf{X}_{i,:})$$

Predicted Labels

Cross-entropy Loss

$$\hat{\mathbf{Y}}_{i,:} = \sum_{k=0}^{K} \gamma_{ik} \mathbf{Y}_{i,:}^{(k)}, \mathbf{Y}^{(k+1)} = \mathbf{T}\mathbf{Y}^{(k)}$$

Propagated Labels

Ding et al. **Meta Propagation Networks for Graph Few-shot Semi-supervised Learning.** AAAI'22

Model Optimization

- Learning to Propagate as bi-level optimization

Meta-learner parameters ⟵ $$\phi^* = \arg\min_{\boldsymbol{\phi}} \mathbb{E}_{v_i \in \mathcal{V}^L}[\mathcal{L}(f_{\boldsymbol{\theta}^*(\boldsymbol{\phi})}(\mathbf{X}_{i,:}), \mathbf{Y}_{i,:})]$$

Target model parameters ⟵ $$\boldsymbol{\theta}^*(\boldsymbol{\phi}) = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{v_i \in \mathcal{V}^U}[\mathcal{L}(f_{\boldsymbol{\theta}}(\mathbf{X}_{i,:}), g_{\boldsymbol{\phi}}(\mathbf{Y}, \mathbf{A})_{i,:})]$$

- Lower-level update:
  - Update the target model using the pseudo labels generated by the adaptive label propagator

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \eta_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}} J_{\text{pseudo}}(\boldsymbol{\theta}, \boldsymbol{\phi})$$

- Upper-level update:
  - Update the meta-learner according to the performance of the target model on the labeled training set

$$\boldsymbol{\phi}' = \boldsymbol{\phi} - \eta_{\boldsymbol{\phi}} \nabla_{\boldsymbol{\phi}} J_{\text{gold}}(\boldsymbol{\theta}'(\boldsymbol{\phi}))$$

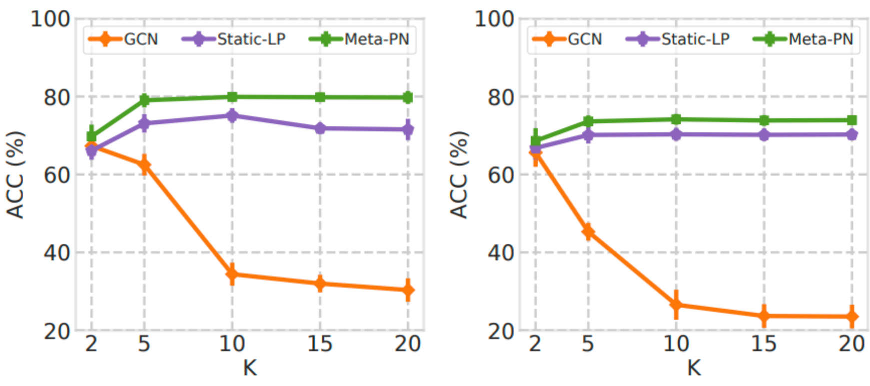
Ding et al. **Meta Propagation Networks for Graph Few-shot Semi-supervised Learning.** AAAI'22

Overall Comparison

- Our approach Meta-PN outperforms SOTA methods on few-shot semi-supervised node classification
  - It inherits the merit of both graph self-training and deep GNNs
  - It captures long-range node interactions without suffering the oversmoothing issue

Method	Cora-ML		CiteSeer	
	3-shot	5-shot	3-shot	5-shot
MLP	41.07 ± 0.76	51.12 ± 0.61	43.34 ± 0.56	44.90 ± 0.60
LP	62.07 ± 0.71	68.01 ± 0.62	54.07 ± 0.59	55.73 ± 1.19
GCN	48.02 ± 0.89	67.32 ± 1.02	53.60 ± 0.86	62.60 ± 0.58
SGC	49.60 ± 0.55	67.24 ± 0.86	57.37 ± 0.98	61.55 ± 0.53
GLP	65.57 ± 0.26	71.26 ± 0.31	65.76 ± 0.49	71.36 ± 0.18
IGCN	66.60 ± 0.29	72.50 ± 0.20	67.47 ± 0.29	72.92 ± 0.10
M3S	64.66 ± 0.31	69.64 ± 0.18	65.12 ± 0.20	68.18 ± 0.18
APPNP	72.39 ± 0.98	78.32 ± 0.58	67.55 ± 0.77	71.08 ± 0.61
DAGNN	71.86 ± 0.75	77.20 ± 0.69	66.62 ± 0.27	70.55 ± 0.12
C&S	68.93 ± 0.68	73.37 ± 0.24	63.02 ± 0.72	64.72 ± 0.53
GPR-GNN	70.98 ± 0.84	75.18 ± 0.52	64.32 ± 0.81	65.28 ± 0.52
Meta-PN	**74.94 ± 0.25**	**79.88 ± 0.15**	**70.48 ± 0.34**	**74.14 ± 0.50**

Propagation steps K



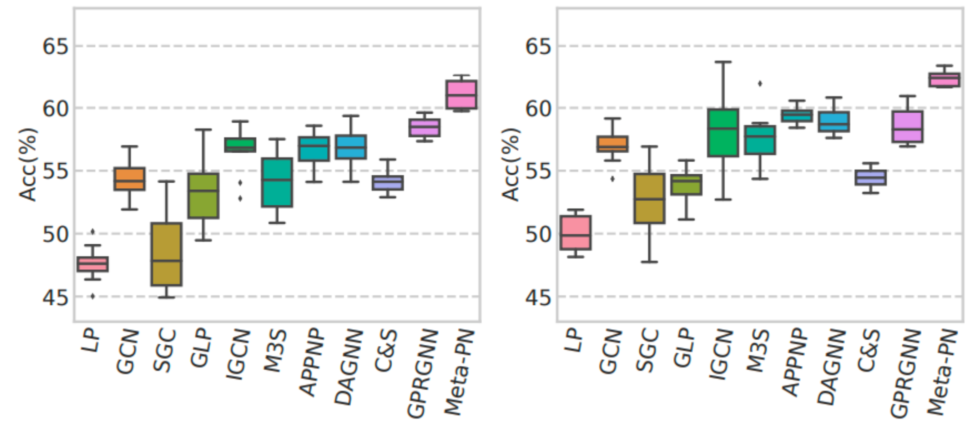
(a) Cora-ML (b) CiteSeer

Ding et al. Meta Propagation Networks for Graph Few-shot Semi-supervised Learning. AAAI'22

Effectiveness of Learning to Propagate

- Meta-PN is also effective on the standard semi-supervised node classification setting or large-scale datasets (e.g., Open Graph Benchmark)
  - Meta-PN can adaptively adjust its label propagation strategy for different learning settings

Method	Cora-ML	CiteSeer	PubMed	MS-CS
MLP	68.42 ± .34	63.98 ± .44	69.47 ± .47	88.30 ± .13
LP	75.74 ± .27	65.62 ± .43	69.82 ± .70	72.03 ± .25
GCN	82.70 ± .37	73.62 ± .39	76.84 ± .44	91.10 ± .20
SGC	75.97 ± .72	75.57 ± .28	71.24 ± .86	90.56 ± .14
GLP	81.67 ± .14	75.21 ± .14	78.95 ± .09	91.85 ± .04
IGCN	82.11 ± .09	75.22 ± .10	79.06 ± .07	91.60 ± .03
M3S	82.72 ± .13	73.73 ± .32	77.62 ± .11	91.08 ± .09
APPNP	85.09 ± .25	75.73 ± .30	79.73 ± .31	91.74 ± .16
DAGNN	85.65 ± .23	74.53 ± .17	79.59 ± .37	92.80 ± .17
C&S	83.18 ± .31	70.51 ± .24	77.10 ± .34	92.49 ± .19
GPR-GNN	83.53 ± .31	71.18 ± .25	79.62 ± .46	92.57 ± .21
Meta-PN	**86.33 ± .36**	**77.13 ± .31**	**80.39 ± .53**	**93.92 ± .17**



(a) label ratio = 1.0% (b) label ratio = 1.5%

ogbn-arxiv

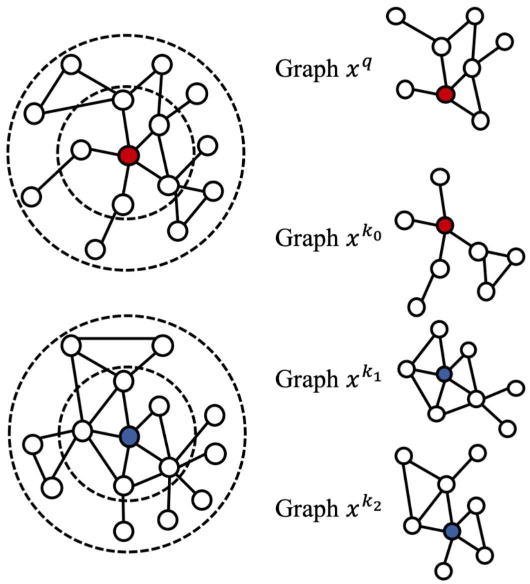
Ding et al. Meta Propagation Networks for Graph Few-shot Semi-supervised Learning. AAAI'22

Graph Self-Supervised Learning

- Graph Self-Supervised Learning aims to learn generalizable node/edge/graph representations without using any human-annotated labels
  - **Graph Generative Modeling**
    - ➢ Learn generalizable representations by reconstructing the node features or/and graph structure

Graph Self-Supervised Learning (Graph SSL)

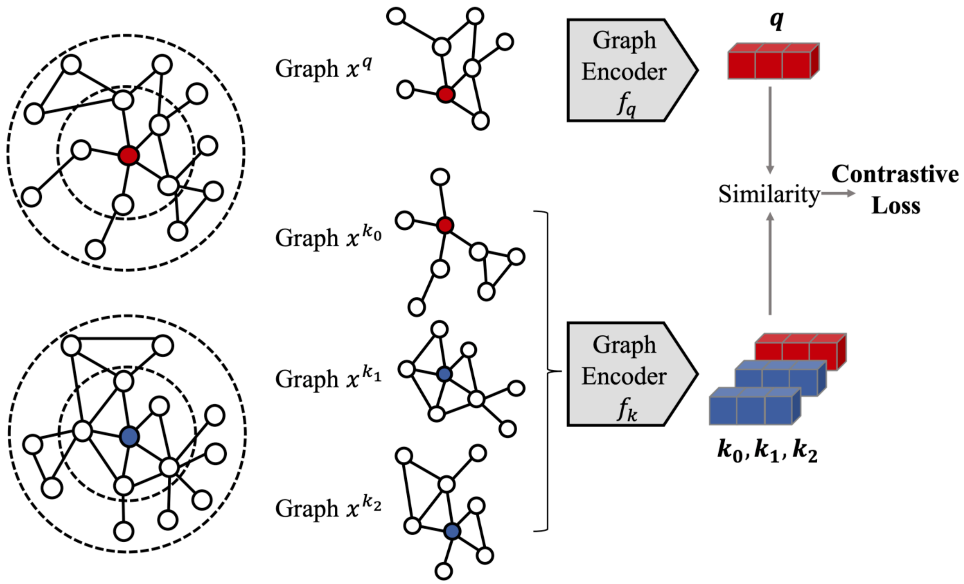
- Graph Self-Supervised Learning aims to learn generalizable node/edge/graph representations without using any human-annotated labels
  - **Graph Generative Modeling**
    - ➢Learn generalizable representations by reconstructing the node features or/and graph structure
  - **Graph Contrastive Learning (GCL)**
    - ➢Create different views from the unlabeled input graph via data augmentation



[Qiu et al. KDD'20]

Graph Self-Supervised Learning (Graph SSL)

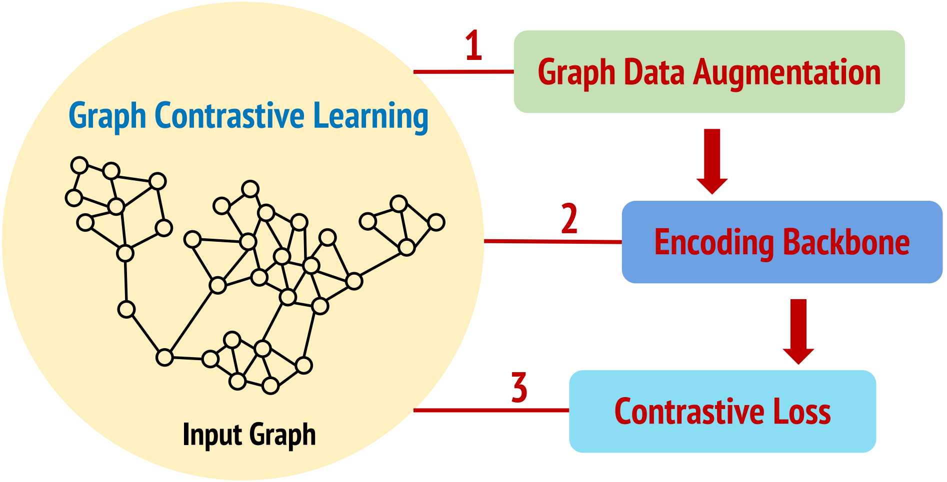
- Graph Self-Supervised Learning aims to learn generalizable node/edge/graph representations without using any human-annotated labels
  - **Graph Generative Modeling**
    - ➢ Learn generalizable representations by reconstructing the node features or/and graph structure
  - **Graph Contrastive Learning (GCL)**
    - ➢ Create different views from the unlabeled input graph via data augmentation
    - ➢ Maximize the agreement between representations of different augmented views of the same instance



[Qiu et al. KDD'20]

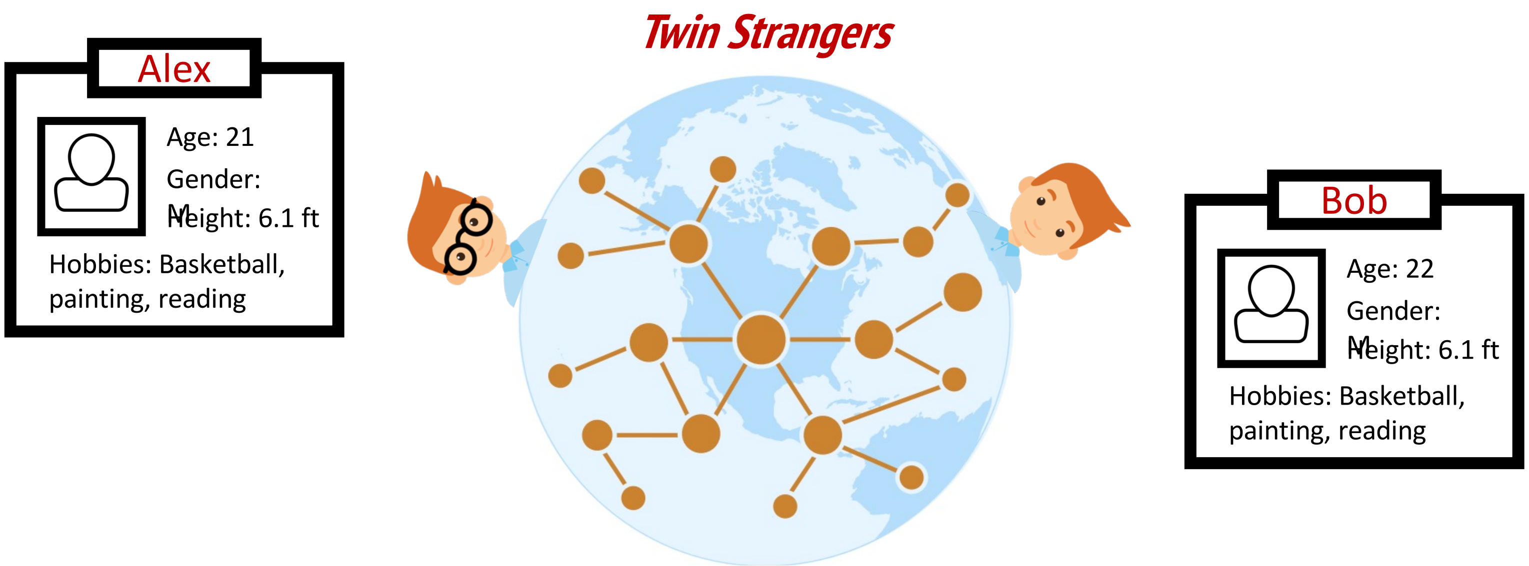
Design of Graph Contrastive Learning

- The current recipe of unsupervised graph contrastive learning
  - **Graph Data Augmentation**
    - ➢Create different views of each instance (e.g., node)
    - ➢Arbitrary graph data augmentation (e.g., edge dropping, feature masking)
  - **Encoding Backbone**
    - ➢Encode different augmented views
    - ➢Shallow GNNs (e.g., 2-layer GCN)
  - **Contrastive Loss**
    - ➢Maximize the agreement between representations learned from different augmented views
    - ➢Instance-level contrastive learning



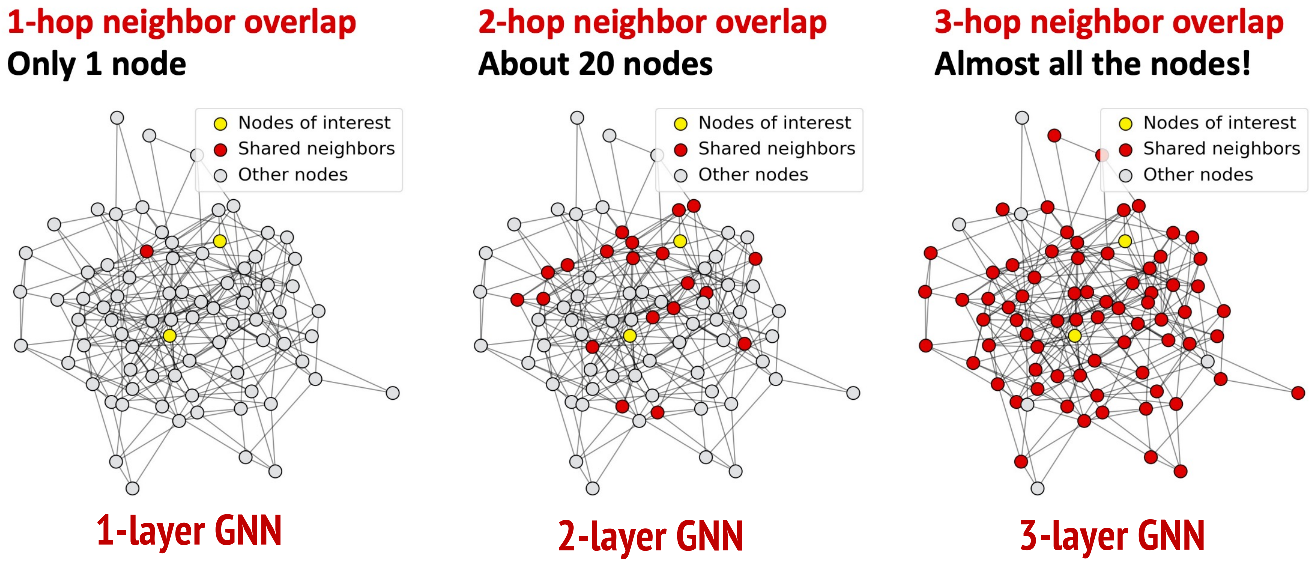
Current GCL methods are shortsighted

- Most of existing methods only consider the information within local neighborhoods
  - **Problem:** nodes sharing similar properties may not always be closely connected on a graph, while existing unsupervised GCL methods cannot attain such global awareness



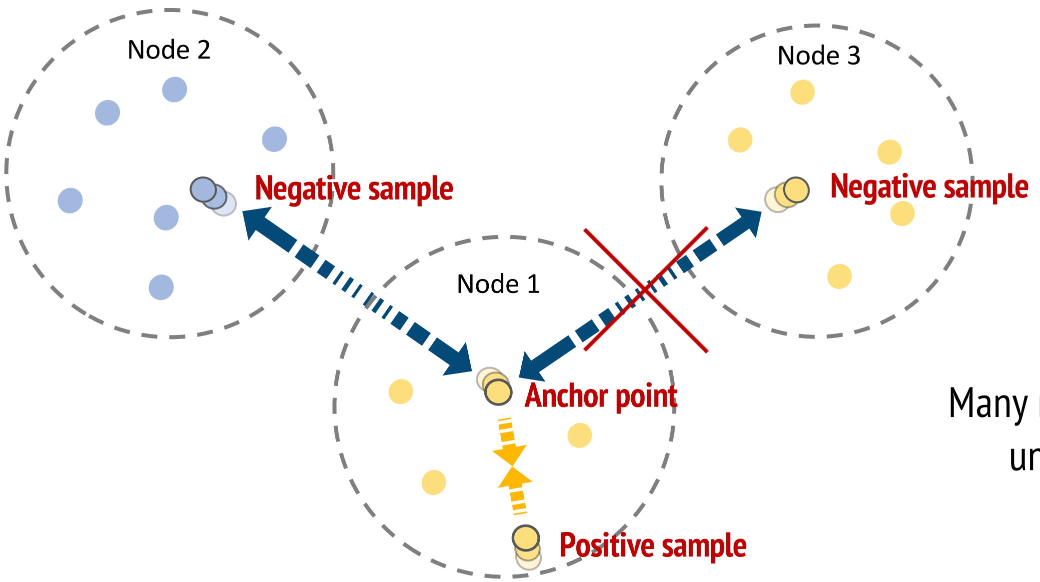
The shallow properties of current GCL

- Structural Perspective
  - Shallow GNN encoders cannot capture long-range node interactions
  - Directly stacking multiple layers of GNNs will inevitably lead to the oversmoothing issue



The shallow properties of current GCL (cont.)

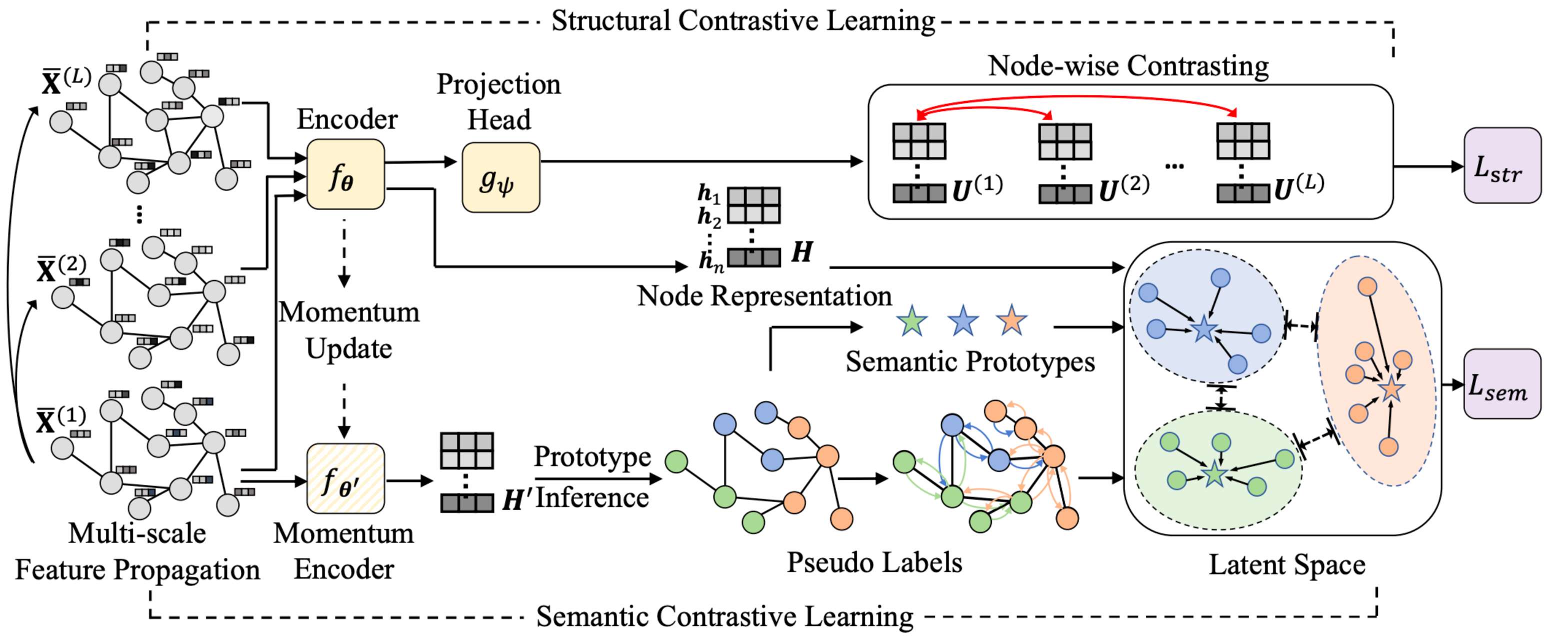
- **Semantic Perspective**
  - Existing GCL methods mainly focus on instance-wise discrimination, while cannot consolidate semantic structure in the latent space from a global view



Many negative pairs that share similar semantics are undesirably pushed away in the latent space

S³-CL: Simple Neural Networks with Structural and Semantic Contrastive Learning

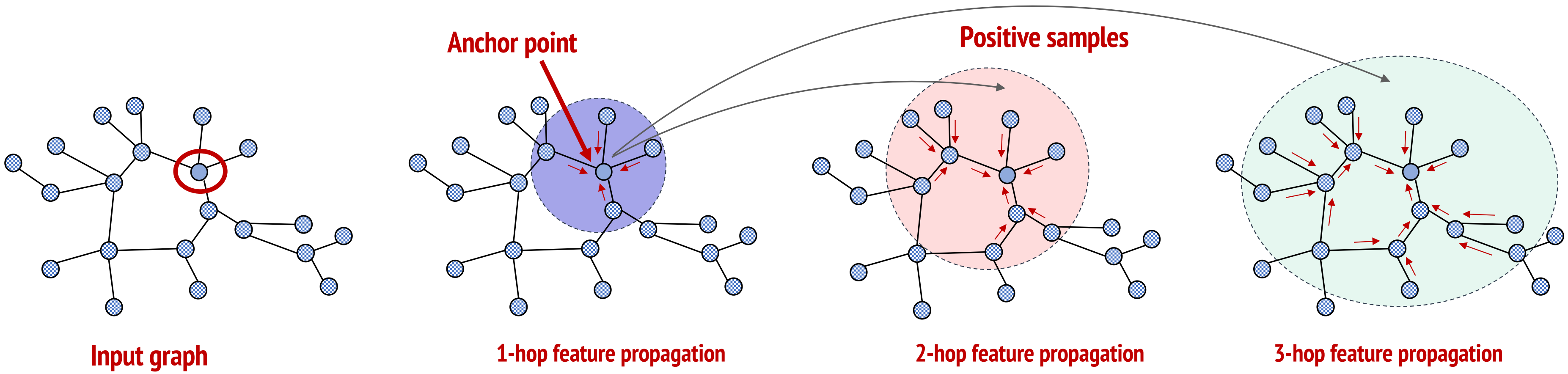
- **Graph Data Augmentation:** Multi-Scale Feature Propagation
- **Encoding Backbone:** Simple Neural Networks (e.g., 1-layer NN)
- **Contrastive Loss:** Structural + Semantic Contrastive Learning



Ding et al. **Eliciting Structural and Semantic Global Knowledge in Unsupervised Graph Contrastive Learning.** AAAI'23

Structural Contrastive Learning

- Structural Contrastive Learning exploits both local and global structural knowledge for learning expressive node representations
  - It maximizes the agreement between the representations of each node learned from its local view and different high-order views



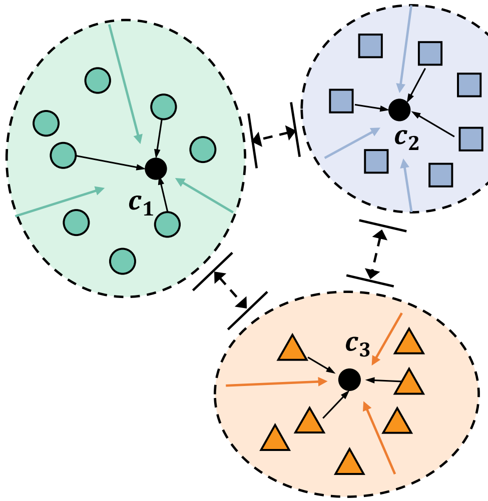
$$\mathcal{L}_{str} = -\sum_{i=1}^{N}\sum_{l=2}^{L} \log \frac{\exp(\mathbf{u}_i^{(1)} \cdot \mathbf{u}_i^{(l)} / \tau_1)}{\sum_{j=1}^{M+L-1} \exp(\mathbf{u}_i^{(1)} \cdot \mathbf{u}_j^{(l)} / \tau_1)}$$

Ding et al. Eliciting Structural and Semantic Global Knowledge in Unsupervised Graph Contrastive Learning. AAAI'23

Semantic Contrastive Learning

- Semantic Contrastive Learning captures global semantic knowledge by enforcing intra-cluster compactness and inter-cluster separability
  - A cluster prototype inference algorithm to infer the clusters and cluster prototypes
  - Perform contrastive learning between each node and its corresponding cluster prototypes

$$\mathcal{L}_{sem} = -\sum_{i=1}^{N} \log \frac{\exp(\mathbf{h}_i \cdot \mathbf{c}_{z_i}/\tau_2)}{\sum_{k=1}^{K} \exp(\mathbf{h}_i \cdot \mathbf{c}_k/\tau_2)}$$



Ding et al. **Eliciting Structural and Semantic Global Knowledge in Unsupervised Graph Contrastive Learning.** AAAI'23

Cluster Prototype Inference

- We use Bayesian nonparametric clustering to automatically infer suitable number of clusters and cluster assignments in the latent representation space
  - Cluster prototypes can be computed based on the cluster assignments

Repeat until convergence:

- For each node: re-assign each node to the cluster corresponding to its closest prototype. If the distance to the closest prototype is greater than ξ, we initialize a new cluster:

$$z_i = \arg\min_k \{d_{ik}\}, \quad d_{ik} = \begin{cases} ||\mathbf{h}_i - \mathbf{c}_k||^2 & \text{for } k = 1, ..., \tilde{K} \\ \xi & \text{for } k = \tilde{K} + 1, \end{cases}$$

- For each cluster: update the cluster mean (prototype) based on the cluster assignments:

$$\mathbf{c}_k = \frac{1}{\sum_{z_i = k} 1} \sum_{z_i = k} \mathbf{h}_i$$

Ding et al. **Eliciting Structural and Semantic Global Knowledge in Unsupervised Graph Contrastive Learning**. AAAI'23

Model Learning as Expectation-Maximization

- The optimization of S[3]-CL
  - Observed data: $G = (\mathbf{X}, \mathbf{A})$
  - Model parameter: Encoder $\boldsymbol{\theta}$
  - Latent Variable: prototype assignment $\boldsymbol{Z}$, and prototype representation $\boldsymbol{C}$
  - Objective: maximize the log-likelihood of observed data via the online EM algorithm

E-step: we fix the network parameter $\boldsymbol{\theta}$ and estimate the prototype assignment $\boldsymbol{Z}$ and the prototypes $\boldsymbol{C}$ with our proposed cluster prototype inference algorithm

M-step: we aim to maximize the expectation of loglikelihood of observed data, by directly optimizing the contrastive loss functions w.r.t. the network parameter $\boldsymbol{\theta}$

We optimize the structural loss and semantic loss together in M-step

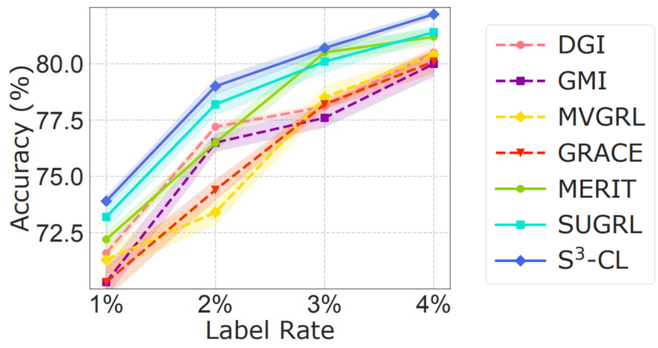
$$\mathcal{L} = \gamma \mathcal{L}_{str} + (1-\gamma)\mathcal{L}_{sem}$$

Ding et al. **Eliciting Structural and Semantic Global Knowledge in Unsupervised Graph Contrastive Learning**. AAAI'23

Node Classification

- S^3-CL outperforms other unsupervised GCL, even achieved better performance than semi-supervised GNN
  - Node representations learned from S^3-CL encode more global knowledge
  - Works better under the extreme low-data scenarios

Methods	Cora	Citeseer	Pubmed	Amazon-P	Coauthor CS	ogbn-arxiv
SEMI-SUPERVISED						
MLP	55.2 ± 0.4	46.5 ± 0.5	71.4 ± 0.3	78.5 ± 0.2	76.5 ± 0.3	55.5 ± 0.2
LP	68.0 ± 0.5	45.3 ± 0.6	63.0 ± 0.3	75.4 ± 0.0	74.3 ± 0.0	68.3 ± 0.0
GCN	81.7 ± 0.4	70.5 ± 0.3	79.4 ± 0.4	87.3 ± 1.0	91.8 ± 0.1	71.7 ± 0.3
GAT	83.0 ± 0.7	72.5 ± 0.7	79.0 ± 0.3	86.2 ± 1.5	90.5 ± 0.7	$\mathbf{73.2 \pm 0.2}$
SGC	81.5 ± 0.2	73.1 ± 0.1	79.7 ± 0.4	88.3 ± 1.1	91.5 ± 0.3	69.8 ± 0.2
SELF-SUPERVISED + FINE-TUNING						
DGI	81.7 ± 0.6	71.5 ± 0.7	77.3 ± 0.6	83.1 ± 0.3	90.0 ± 0.3	67.1 ± 0.4
GMI	82.7 ± 0.2	73.0 ± 0.3	80.1 ± 0.2	85.1 ± 0.0	91.0 ± 0.0	69.6 ± 0.3
MVGRL	82.9 ± 0.7	72.6 ± 0.7	79.4 ± 0.3	87.3 ± 0.1	91.3 ± 0.1	71.3 ± 0.2
GRACE	80.0 ± 0.4	71.7 ± 0.6	79.5 ± 1.1	81.8 ± 0.8	90.1 ± 0.8	71.1 ± 0.2
MERIT	83.1 ± 0.6	$\underline{74.0 \pm 0.7}$	80.1 ± 0.4	$\underline{88.8 \pm 0.4}$	$\underline{92.4 \pm 0.4}$	71.7 ± 0.1
SUGRL	83.4 ± 0.5	73.0 ± 0.5	$\mathbf{81.9 \pm 0.5}$	88.5 ± 0.2	92.2 ± 0.5	69.3 ± 0.2
S^3-CL	$\mathbf{84.5 \pm 0.4}$	$\mathbf{74.6 \pm 0.4}$	80.8 ± 0.3	$\mathbf{89.0 \pm 0.5}$	$\mathbf{93.1 \pm 0.4}$	72.8 ± 0.3



Cora

Ding et al. **Eliciting Structural and Semantic Global Knowledge in Unsupervised Graph Contrastive Learning.** AAAI'23

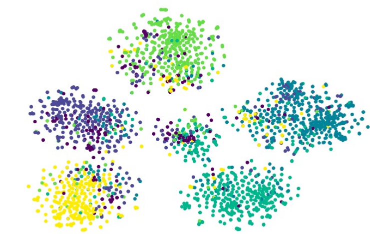
Node Clustering

- The learned node representations from S^3-CL achieves SOTA clustering performance
  - The proposed semantic contrastive learning brings better intra-cluster compactness and inter-cluster separability for the learned node representations

Methods	Cora			Citeseer			Pubmed		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-Means	49.2	32.1	22.9	54.0	30.5	27.8	59.5	31.5	28.1
GAE (Kipf and Welling 2016)	59.6	42.9	34.7	40.8	17.6	12.4	67.2	27.7	27.9
ARGA (Pan et al. 2018)	64.0	44.9	35.2	57.3	35.0	34.1	66.8	30.5	29.5
ARVGA (Pan et al. 2018)	64.0	45.0	37.4	54.4	26.1	24.5	69.0	29.0	30.6
GALA (Park et al. 2019)	74.5	57.6	53.1	69.3	44.1	44.6	69.3	32.7	32.1
DGI (Veličković et al. 2019)	55.4	41.1	32.7	51.4	31.5	32.6	58.9	27.7	31.5
DBGAN (Zheng et al. 2020)	<u>74.8</u>	56.0	<u>54.0</u>	67.0	40.7	41.4	69.4	32.4	32.7
MVGRL (Hassani and Khasahmadi 2020)	73.2	56.2	51.9	68.1	43.2	43.4	69.3	34.4	32.3
MERIT (Jin et al. 2021)	73.6	57.1	52.8	68.9	43.9	44.1	69.5	34.7	32.8
SUGRL (Mo et al. 2022)	73.9	<u>58.5</u>	53.0	<u>70.5</u>	<u>45.8</u>	<u>47.0</u>	<u>69.5</u>	<u>35.0</u>	<u>33.4</u>
S^3-CL (ours)	**75.1**	**60.7**	**56.6**	**71.2**	**46.3**	**48.5**	**71.3**	**36.0**	**34.7**

Citeseer

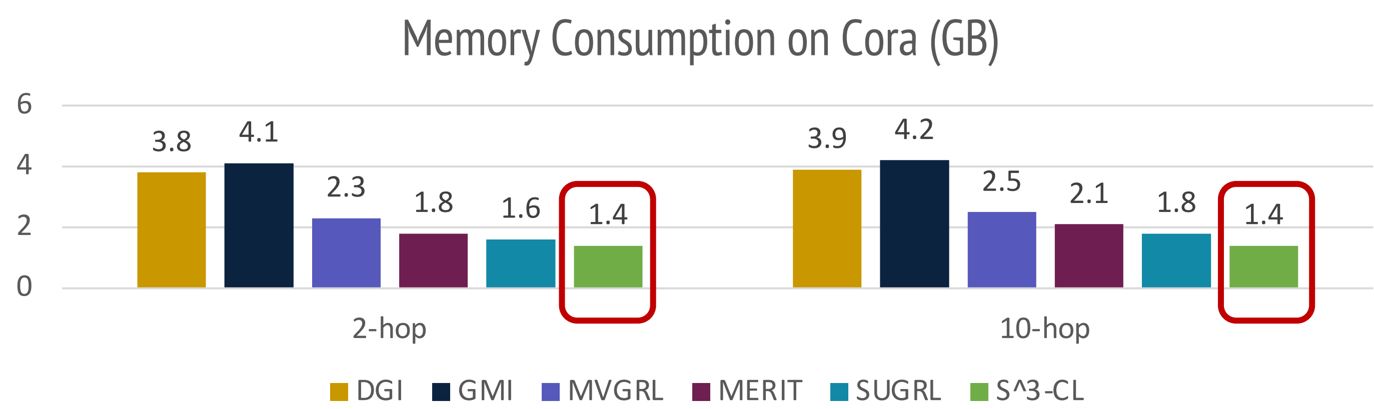
SUGRL(up) vs. S^3-CL (down)



Ding et al. **Eliciting Structural and Semantic Global Knowledge in Unsupervised Graph Contrastive Learning.** AAAI'23

Parameter Efficiency

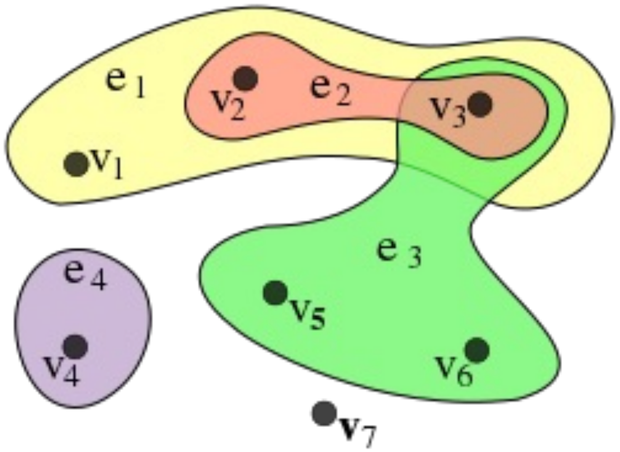
- S^3-CL has huge advantage in saving computation memory
  - By virtue of the proposed structural and semantic contrastive learning, S^3-CL allows the network encoder to be built with a simple neural network, which can significantly improve the parameter/computation efficiency

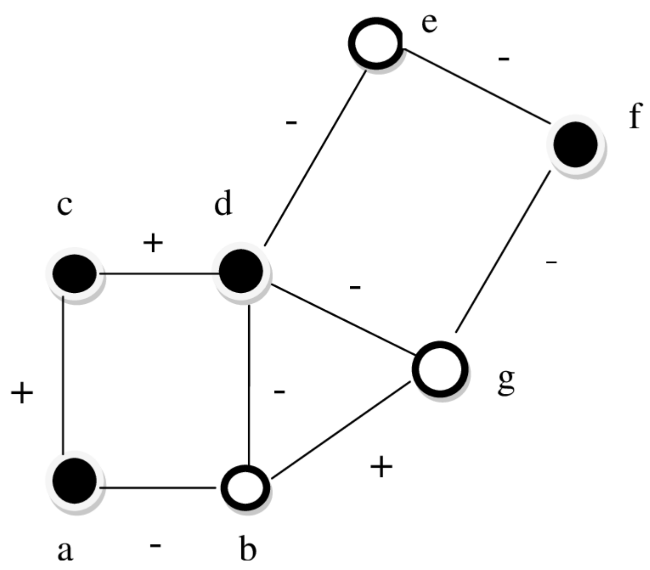


Ding et al. **Eliciting Structural and Semantic Global Knowledge in Unsupervised Graph Contrastive Learning**. AAAI'23

Future Direction

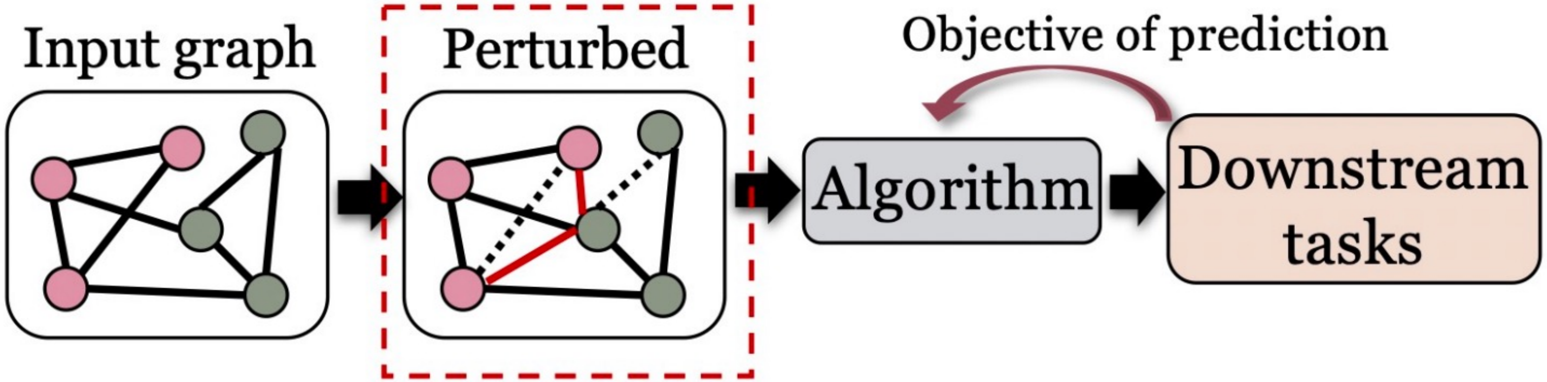
- Data Augmentation for Complex Graphs
  - Real-world graphs are of different types with different properties
  - Heterogeneous graphs, hypergraphs, signed graphs, heterophilic graphs





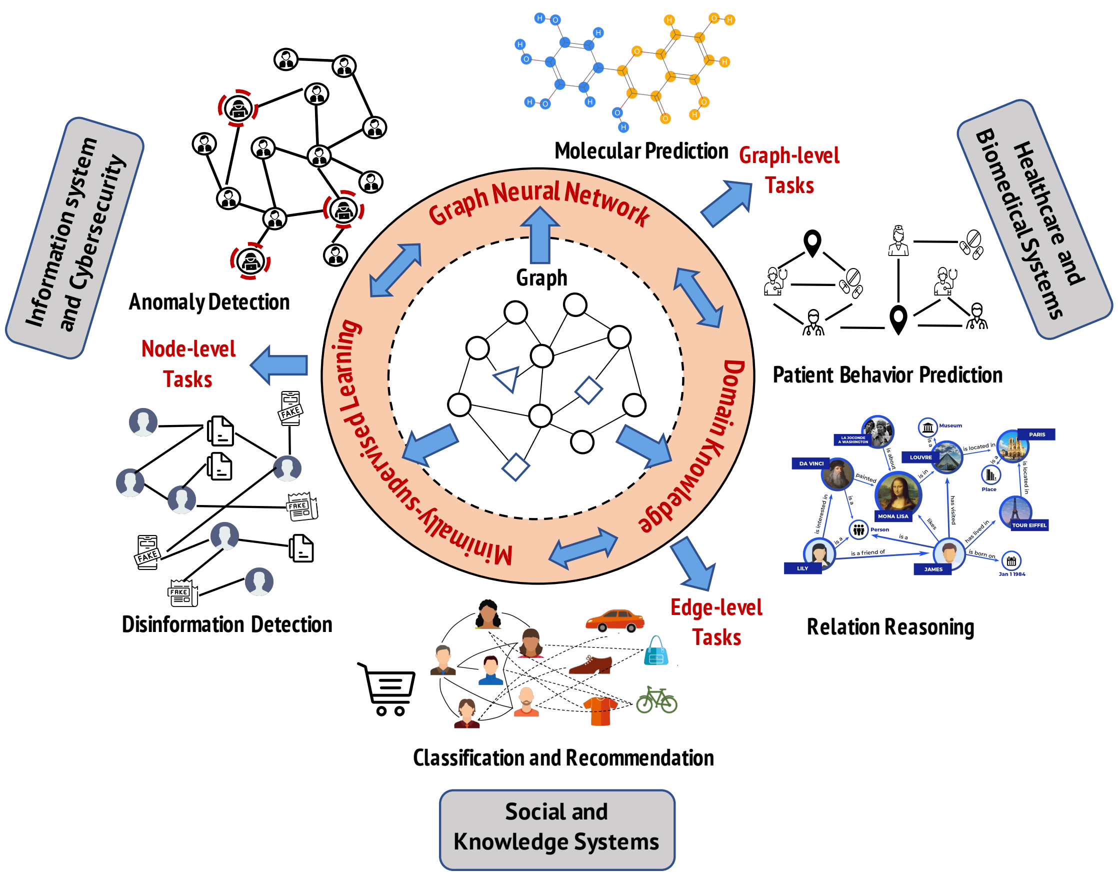
Future Direction

- Data Augmentation for Trustworthy Graph Learning
  - Fairness
  - Causality
  - Explainability



Future Direction

- Knowledge-Guided Graph Data Augmentation
  - GDA for Social Good
  - GDA for Cybersecurity
  - GDA for NLP
  - GDA for Drug Discovery



Thank you!

- Tutorial website: https://github.com/zhao-tong/SDM2023_Graph_Data_Augmentation_Tutorial
- Contacts:
  - Tong Zhao, tzhao@snap.com
  - Kaize Ding, kaize.ding@asu.edu
  - Wei Jin, jinwei2@msu.edu
  - Gang Liu, gliu7@nd.edu
  - Meng Jiang, mjiang2@nd.edu
  - Neil Shah, nshah@snap.com