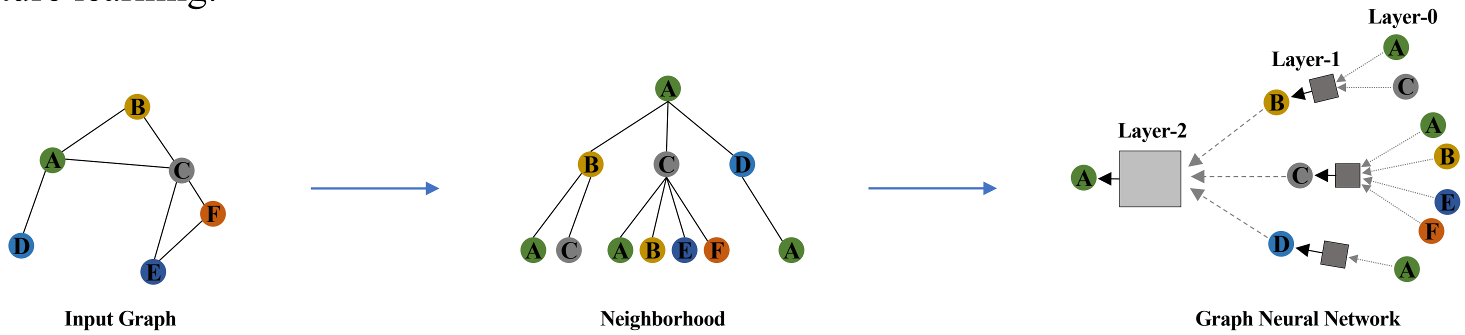
Grace: Graph Self-Distillation and Completion to Mitigate Degree-Related Biases

Presenter: Hui Xu,
Joint work with: Liyao Xiang, Femke Huang, Yuting Weng, Ruijie Xu, Xinbing Wang, Chenghu Zhou

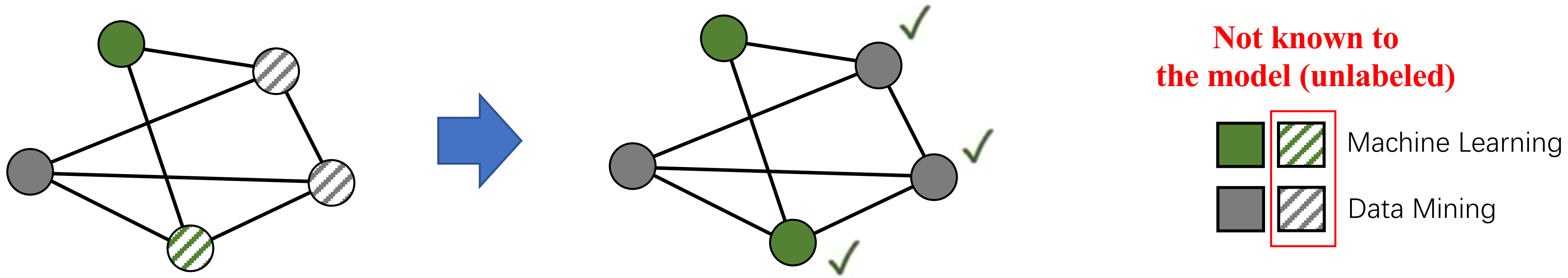
Shanghai Jiao Tong University

Background

- Extensive studies for Graph Neural Networks (GNNs) have arisen in recent years showing the great power of graph structure learning.



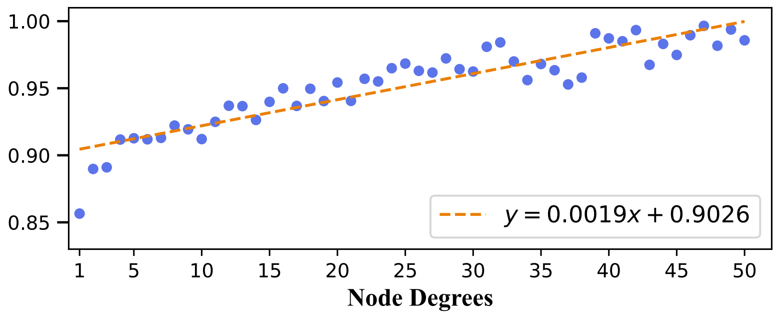
- Graph Neural Networks (GNNs) have already played a crucial role in node classification task.

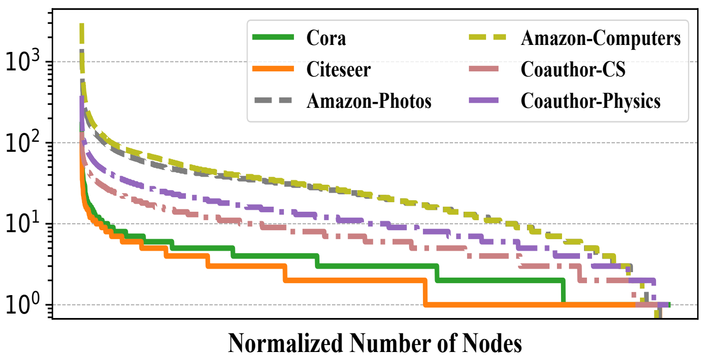


Background

Degree-related Bias: The prediction accuracy of graph neural networks increases with the increase in node degrees on homophily graphs

This phenomenon significantly affects applications of GNNs in *recommendation systems*, *e-commerce services*, and *social networks*.





Graph data in the real world often follows the long-tailed distribution, where the majority of nodes belong to low-degree and isolated nodes.

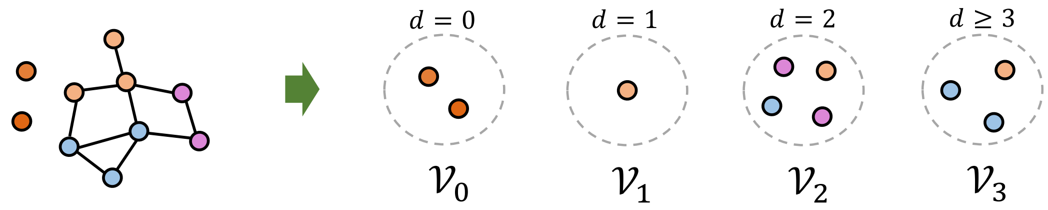
Challenges for low-degree nodes:

- ✓ **Challenge 1**: insufficient neighborhood information
- ✓ **Challenge 2**: GNNs may overlook the learning of intrinsic features of low-degree nodes
- ✓ …

Degree-related bias can highly limit the node classification performance of GNNs on dataset following long-tail degree distribution !!!

Problem Definition

- We split the node set $\mathcal{V} = \bigcup_i^{deg_{max}} \mathcal{V}_i$ to a maximal deg_{max} groups and each $\mathcal{V}_i$ refers to the set of nodes whose degrees are i. For $i = deg_{max}$, $\mathcal{V}_i$ refers to the set of nodes whose degrees are no less than deg_{max}



Set $deg_{max} = 3$

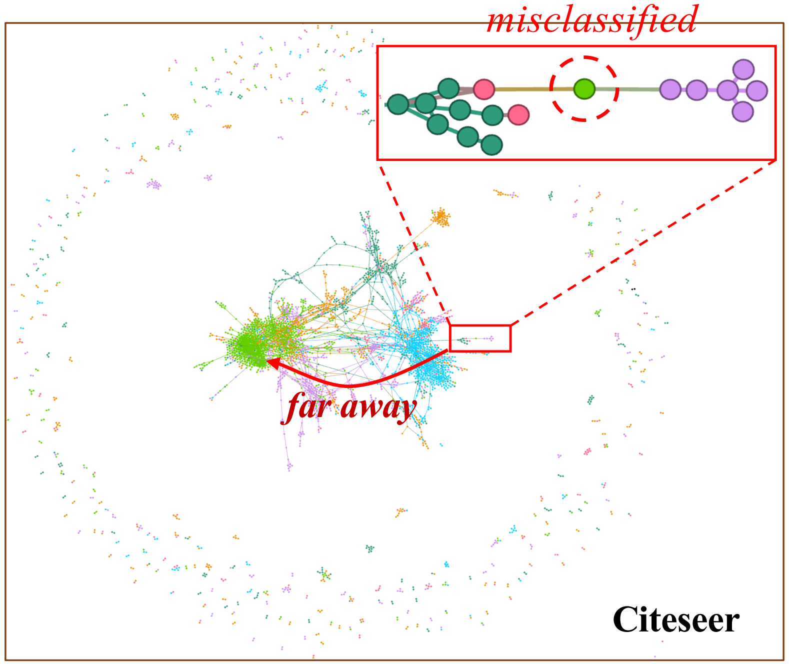
- Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$ and labels Y for the labeled set, we aim to learn a GNN-based model to *maintain* ***overall classification performance*** and achieve a ***balanced performance*** *for all degree groups*

Data Analysis

***Question 1**: Is the small number of neighbors for low-degree nodes the main reason for the degree-related bias?*

Through experiments, we made two observations:

- ✓ the majority of misclassified low-degree nodes often have a very small proportion of same-class neighboring nodes.
- ✓ Low-degree nodes with a higher proportion of neighboring nodes belonging to the same class tend to be correctly classified.



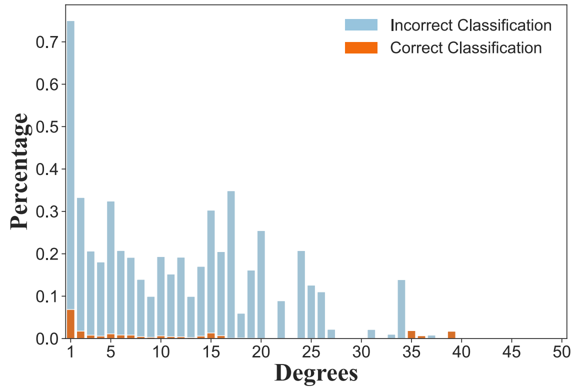
∗ Low-degree nodes with a green label, circled in red, do not have any neighboring green nodes of the same class and are also far away from all other green nodes.

Data Analysis

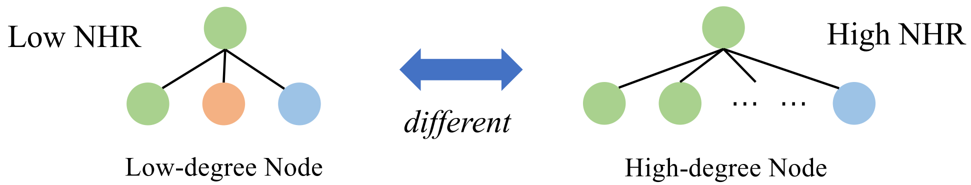
- **Neighborhood Homophily Ratio (NHR):**

$$NHR(v) = \frac{1}{|\mathcal{N}_v|} \sum_{u \in \mathcal{N}_v} \mathbb{1}(y_v = y_u)$$

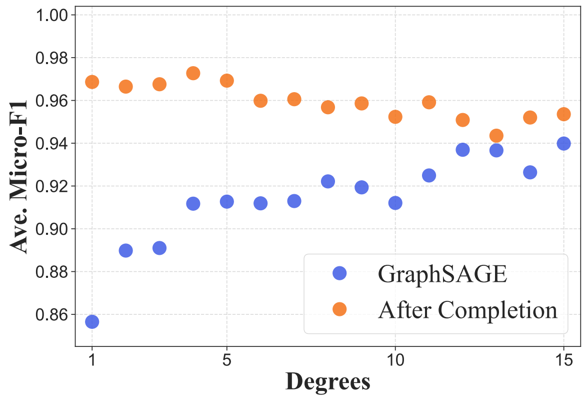
where y_v is the label of node v, and $1(\cdot)$ is the indicator function.



- **Discrepancy of Neighborhood Distribution**



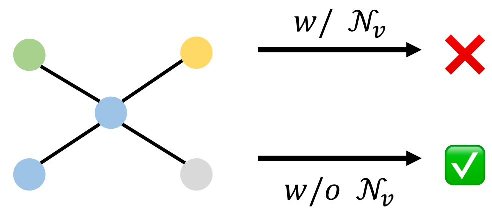
The aforementioned differences make it difficult for GNNs to effectively utilize the neighborhood distribution of low-degree nodes for accurate node classification.



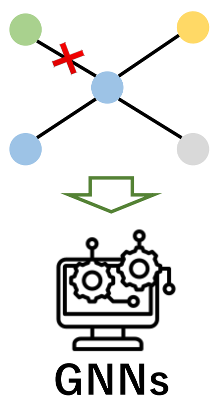
Motivation 1: Increasing the NHR of low-degree nodes can help mitigate the degree-related bias in GNNs for node classification task.

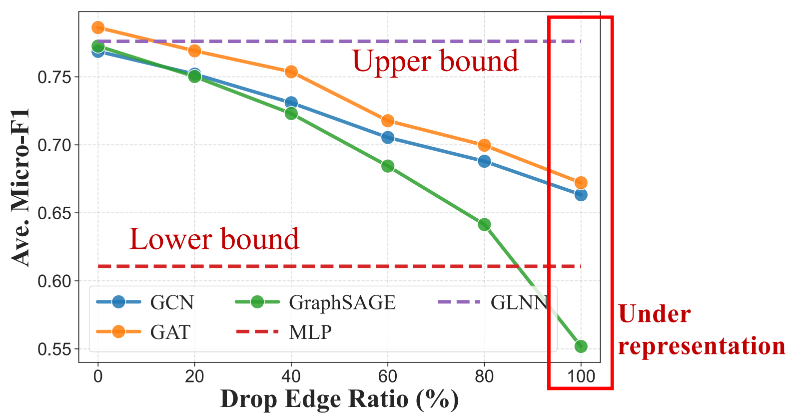
Data Analysis

Question 2: *Can GNNs effectively utilize the node's own features for node classification in the case of insufficient neighborhood information?*

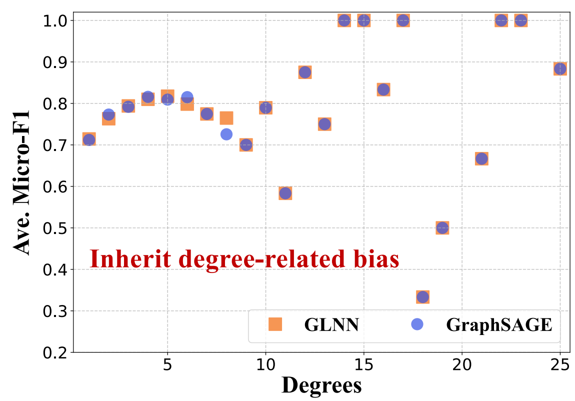


- Random Drop Edge



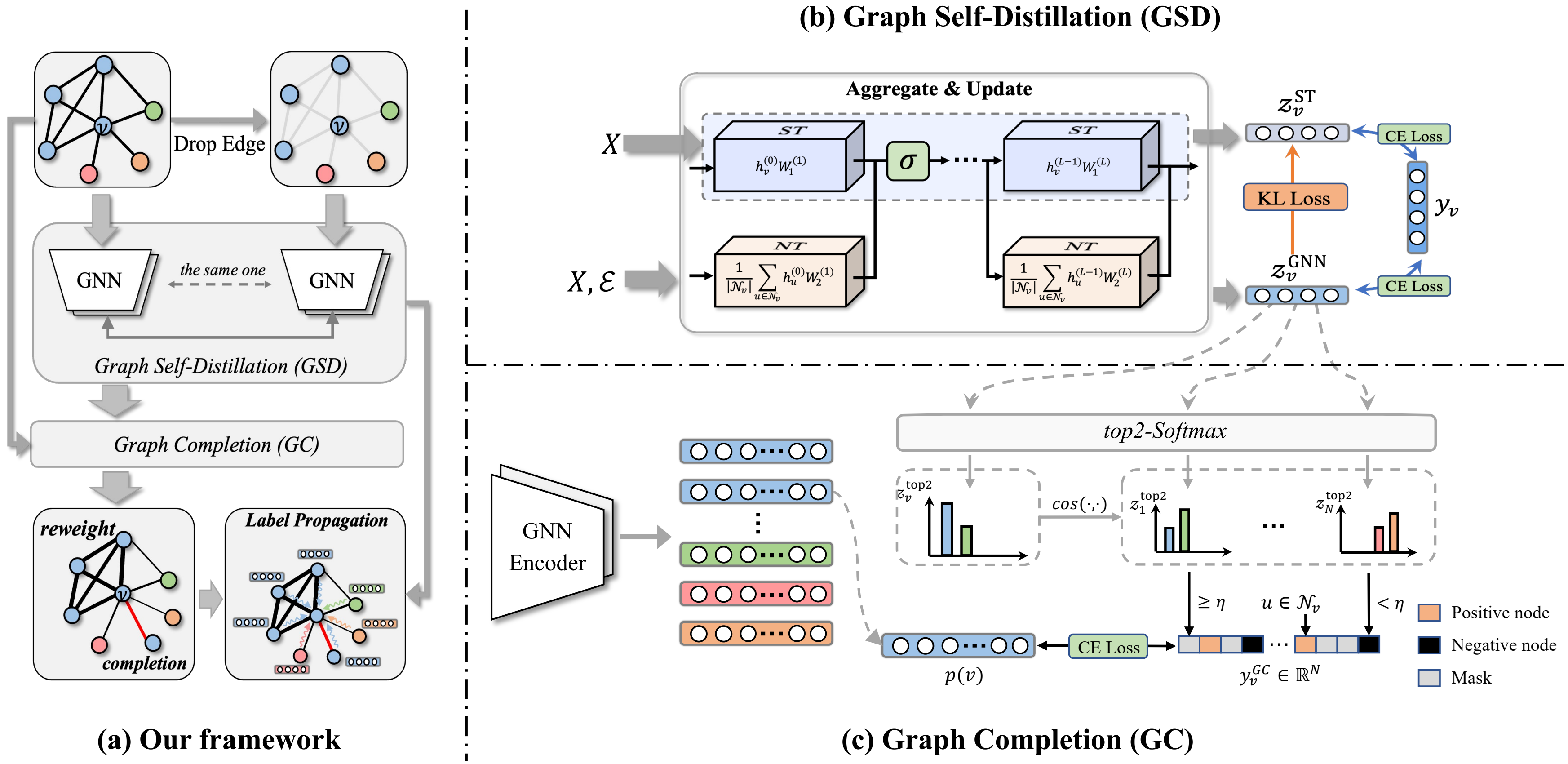


- Graph Distillation



Motivation 2: To alleviate the degree-related bias in node classification tasks, we need to enhance the representation capability of nodes own features in GNNs

Proposed Framework



Proposed Framework

Graph Self-Distillation

- Aggregate layer

$$h_v^{(l)} = \sigma(\underbrace{h_v^{(l-1)} \cdot W_1^{(l)}}_{\substack{self-transformation \\ \textbf{ST}}} + \underbrace{\mathrm{MEAN}(\{h_u^{(l-1)} \cdot W_2^{(l)}, \forall u \in \mathcal{N}_v\})}_{\substack{neighborhood\ transformation \\ \textbf{NT}}})$$

- Objective function

Teacher

$$\mathcal{L}_t = \sum_{v \in \mathcal{V}_{train}} CE(z_v^{\mathrm{GNN}}, y_v) + \lambda ||\Theta_t||_2^2$$

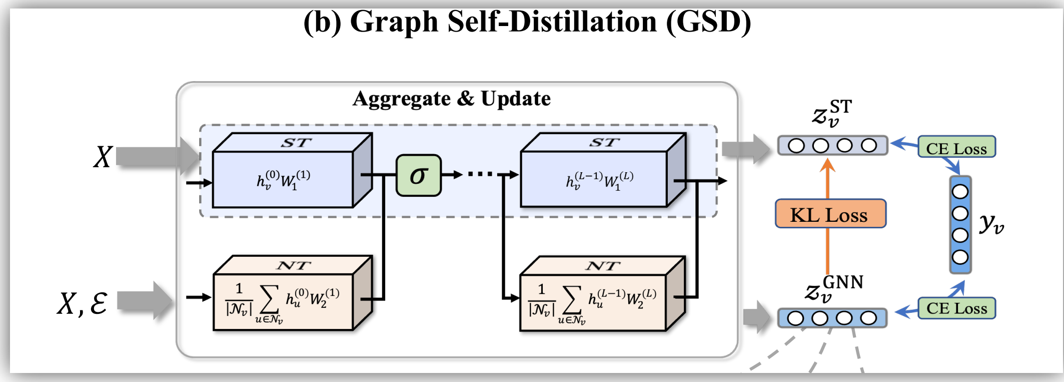
distillation

Student

$$\mathcal{L}_s = \gamma \sum_{v \in \mathcal{V}} \mathrm{KL}(z_v^{\mathrm{ST}} || z_v^{\mathrm{GNN}}) + (1-\gamma) \sum_{v \in \mathcal{V}_{train}} CE(z_v^{\mathrm{ST}}, y_v)$$

Joint Learning

$$\mathcal{L}_{SD} = \mathcal{L}_t + \mathcal{L}_s$$



REMARK 1. *Without the non-linear activation in Eq. 3, the self-distillation guides the self-representation weights* $\{W_1^{(l)} | l = 1, ..., L\}$ *to learn a neighborhood translation of node features.*

Self-distillation does not introduce any additional parameters and inherits the efficiency of GraphSAGE.

Proposed Framework

Graph Completion

- Label construction

$$z_v^{\text{top2}} = \text{softmax}(\text{top2}(z_v^{\text{GNN}})), \forall v \in \mathcal{V}$$

$$y_{v,u}^{GC} = \begin{cases} 0, & \cos(z_v^{\text{top2}}, z_u^{\text{top2}}) < \eta, \\ 1, & u \in \mathcal{N}_v, \\ \text{not selected}, & otherwise, \end{cases}$$

- Objective function

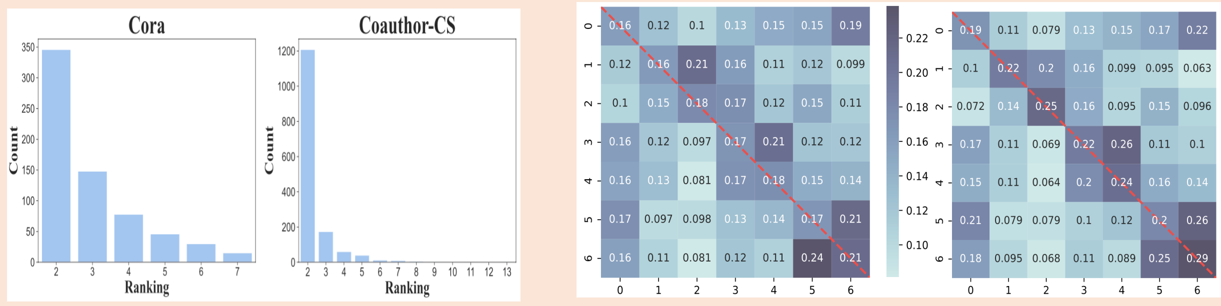
$$\mathcal{L}_{GC} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} CE(p(v), y_v^{GC})$$

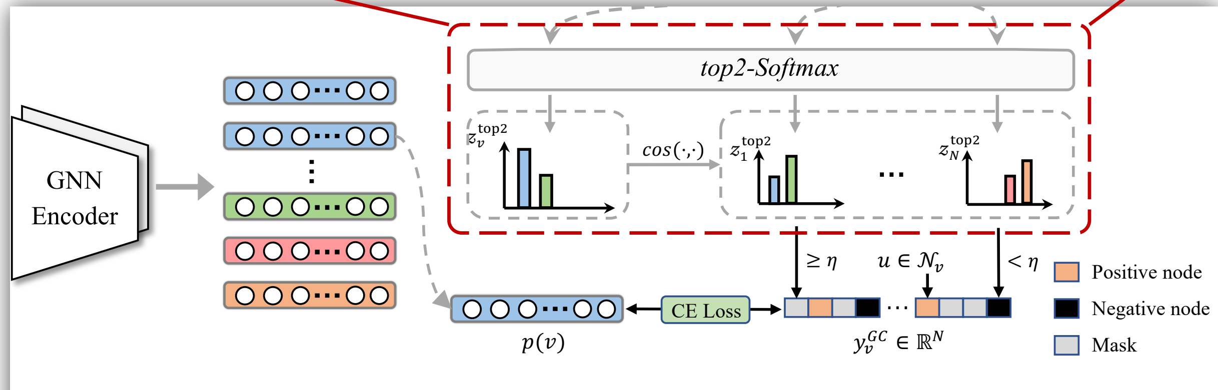
- Avoid Error Propagation

$$\mathcal{G}' = (\mathcal{V}, \bar{\mathcal{E}}', X)$$

$$\hat{y}_v = \frac{1}{|\mathcal{N}_v'| + 1} \sum_{u \in \{v\} \cup \mathcal{N}_v'} z_v^{\text{GNN}}$$

Reduce erroneous negative samples in the training data



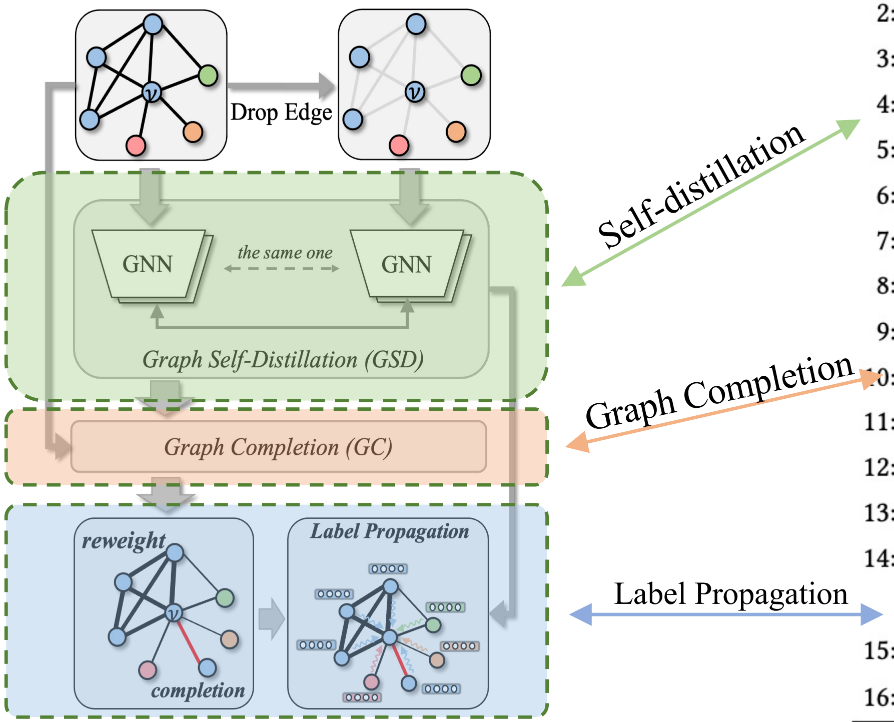


(c) Graph Completion (GC)

Proposed Framework

Training Step

- We perform a two-stage training in Grace



Algorithm 1 GRACE

Input: Edge set $\mathcal{E}$, attribute matrix X, training labels Y_t, hyperparameters γ, η, k and K.

Output: Classification results $\hat{Y}$.

```
# First stage of Training: Graph Self-Distillation.
Randomly initialize Θ = {Θ_s, Θ_t};
while Θ not converged do
    Calculate Z^GNN and Z^ST according to Eq.2 and Eq.3;
    Compute L_SD and update parameter Θ.
end while
# Second stage of Training: Graph Completion.
Randomly initialize θ of GNN Encoder;
Compute Z^GNN and construct Y^GC according to Eq. 10.
while θ not converged do
    Compute Loss L_GC and update parameter θ.
end while
# Inference stage: Label Propagation.
Complete G to G': for each non-isolated node v with degree no greater than K, link it with its top-k neighbors by p(v).
Perform label propagation to acquire Ŷ via Eq. 12.
return Classification results Ŷ
```

Experiments

- **Datasets**
  - We evaluate Grace on six benchmark datasets:

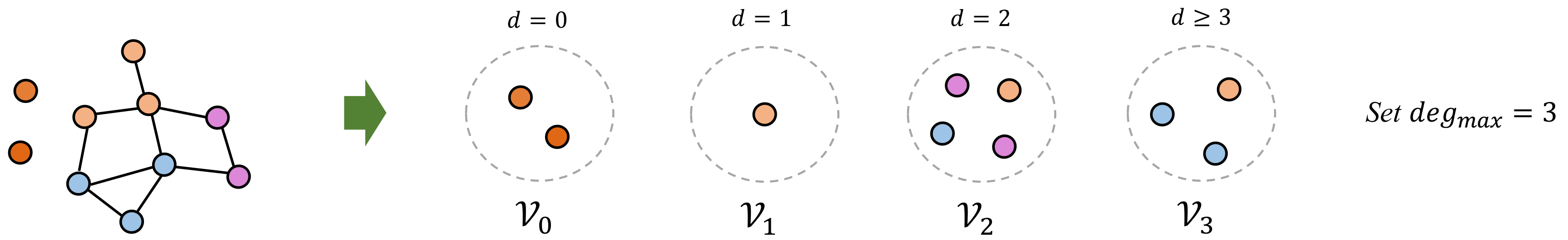
Dataset	Nodes	Edges	Features	Classes
Cora	2,485	5,069	1,433	7
Citeseer	3,327	9,104	3703	6
Amazon-Photo	7,650	238,162	745	8
Amazon-Computers	13,752	491,722	767	10
Coauthor-CS	18,333	163,788	6,805	15
Coauthor-Physics	34,493	495,924	8,415	5

- **Baselines**
  - General GNNs: *GCN, GraphSAGE, GAT*
  - Enhanced GNNs: *AKGNN, Order GNN*
  - Degree specific GNNs: *Demo-Net*
  - Missing neighbors-aware GNNs: *Tail-GCN, ColdBrew*
  - *Biased gradient-aware GNNs: RawlsGCN*

Experiments

- **Metrics**
  - We use micro-F1 score to evaluate the overall performance.
  - For degree group performance, we define the following metrics by the node degree:

$$\text{MicroF1}(k) = \text{MicroF1}(\{u, \forall \text{node } u \text{ such that } d(u) = k\}),$$
$$G.Mean = \mathbb{E}[\{\text{MicroF1}(k), \forall \text{node degree } k\}],$$
$$G.bias = \text{Std}(\{\text{MicroF1}(k), \forall \text{node degree } k\}),$$



Experiments

- Node classification performance of different methods on three different metrics

Method	Cora			Citeseer			Amazon-Photo		
	Micro-F1↑	G.Mean↑	G.Bias↓	Micro-F1↑	G.Mean↑	G.Bias↓	Micro-F1↑	G.Mean↑	G.Bias↓
GCN	78.74±1.65%	80.53±2.52%	8.11±1.58%	68.54±1.46%	76.20±2.19%	**12.25±1.19%**	82.85±2.49%	84.41±2.28%	7.87±1.49%
GraphSAGE	76.50±1.77%	79.15±2.25%	8.40±1.61%	67.62±1.57%	75.52±2.11%	16.30±2.73%	87.56±1.85%	87.99±1.87%	8.84±1.95%
GAT	78.30±2.15%	80.02±2.61%	8.05±1.89%	66.57±1.87%	74.97±2.56%	12.47±1.42%	82.90±3.55%	83.75±3.45%	8.32±1.54%
AKGNN	79.45±1.47%	**82.13±1.87%**	7.77±1.29%	68.16±1.60%	77.32±1.83%	12.65±0.88%	86.76±3.19%	87.29±3.19%	7.77±1.02%
Ordered GNN	77.85±1.80%	80.24±2.15%	7.95±1.50%	65.77±1.67%	74.26±2.17%	13.66±1.53%	88.18±1.92%	88.90±1.87%	6.39±0.54%
Demo-Net	76.39±2.06%	78.52±2.47%	8.70±1.65%	65.07±2.27%	74.34±2.62%	13.31±1.29%	70.17±4.90%	69.39±5.29%	14.24±1.98%
Tail-GCN	77.21±1.91%	80.02±2.27%	8.18±1.45%	65.97±2.45%	76.18±2.20%	13.54±1.41%	83.36±3.93%	84.15±3.77%	8.14±1.37%
ColdBrew-S	55.46±2.13%	56.86±2.63%	9.58±2.76%	54.04±2.13%	61.79±3.94%	13.01±2.76%	76.26±1.91%	77.66±1.92%	6.69±0.61%
ColdBrew-T	79.04±1.30%	80.41±1.66%	8.61±1.10%	68.04±1.51%	76.59±1.84%	12.83±1.04%	86.70±1.09%	87.18±1.11%	7.65±0.64%
RawlsGCN	75.67±2.04%	78.63±2.16%	8.76±1.37%	67.02±1.99%	76.18±2.50%	12.56±1.42%	87.33±1.93%	87.75±2.00%	6.13±0.44%
GRACE	**80.40±2.11%**	81.59±2.23%	**7.61±1.36%**	**69.24±2.14%**	**77.41±2.25%**	12.97±1.43%	**89.23±1.73%**	**89.75±1.75%**	**5.96±0.52%**

Method	Amazon-Computers			Coauthor-CS			Coauthor-Physics		
	Micro-F1↑	G.Mean↑	G.Bias↓	Micro-F1↑	G.Mean↑	G.Bias↓	Micro-F1↑	G.Mean↑	G.Bias↓
GCN	68.08±3.44%	69.48±3.30%	10.12±1.70%	91.21±0.58%	91.47±1.26%	4.22±0.63%	93.23±0.91%	95.53±0.87%	2.92±0.24%
GraphSAGE	76.81±2.45%	76.89±2.41%	10.01±1.72%	91.72±0.63%	93.07±0.73%	3.73±0.33%	92.77±1.01%	95.22±0.90%	3.16±0.25%
GAT	73.26±4.70%	74.17±4.39%	9.71±1.40%	88.25±1.34%	88.10±1.81%	4.92±0.39%	90.70±1.52%	93.17±1.70%	3.40±0.50%
AKGNN	75.71±3.87%	75.84±3.92%	9.70±0.84%	88.85±0.76%	89.97±0.96%	4.77±0.56%	92.29±1.17%	94.21±2.53%	3.22±0.26%
Ordered GNN	76.99±2.69%	77.12±2.60%	8.75±0.57%	92.44±0.58%	93.46±0.59%	3.70±0.36%	93.13±0.92%	95.42±0.81%	2.92±0.24%
Demo-Net	53.23±3.55%	50.60±3.73%	15.54±1.44%	89.22±0.89%	90.72±0.94%	5.11±0.48%	92.14±1.22%	95.07±0.86%	3.99±0.69%
Tail-GCN	73.34±4.47%	73.42±4.35%	9.47±1.08%	-	-	-	-	-	-
ColdBrew-S	63.39±3.24%	63.53±2.99%	**7.27±0.66%**	88.29±1.15%	89.28±1.21%	4.28±0.60%	89.83±2.27%	92.93±1.90%	3.73±0.59%
ColdBrew-T	70.52±1.68%	72.66±1.73%	9.07±0.63%	91.50±0.34%	93.05±0.42%	4.02±0.33%	93.30±0.37%	95.13±0.26%	3.29±0.15%
RawlsGCN	77.12±2.88%	77.29±2.96%	9.05±0.67%	91.69±0.58%	92.89±0.49%	3.70±0.26%	93.08±0.71%	95.61±0.46%	3.10±0.31%
GRACE	**77.32±2.40%**	**77.35±2.41%**	7.73±0.71%	**92.92±0.61%**	**93.97±0.57%**	**3.42±0.33%**	**93.67±0.60%**	**95.89±0.43%**	**2.81±0.21%**

Experiments

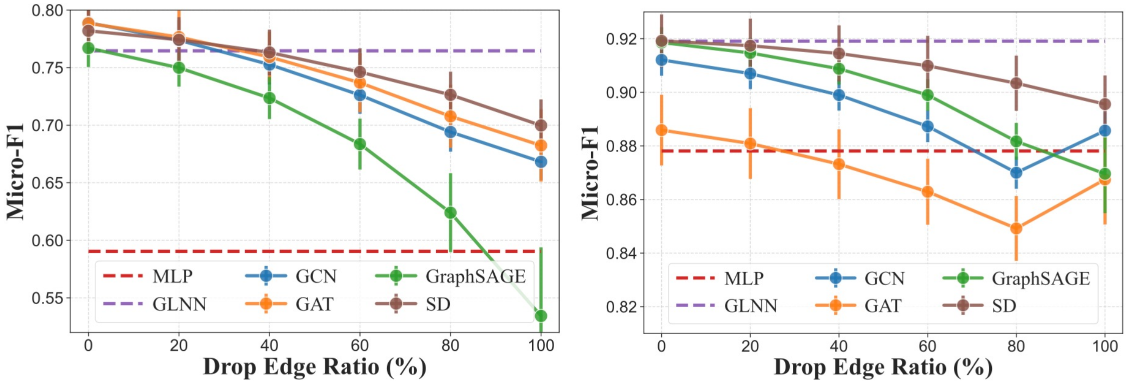
- Performance of all benchmarks on different degree thresholds

Dataset	Cora			Citeseer			Amazon-Photo		
K	**1**	**5**	**10**	**1**	**5**	**10**	**1**	**5**	**10**
GCN	71.03 ± 2.51%	78.69 ± 1.69%	78.67 ± 1.66%	61.74 ± 1.61%	67.60 ± 1.46%	68.18 ± 1.48%	68.32 ± 2.81%	70.77 ± 5.03%	74.41 ± 4.88%
GraphSAGE	67.89 ± 3.07%	76.20 ± 1.84%	76.36 ± 1.79%	59.83 ± 1.62%	66.51 ± 1.57%	67.20 ± 1.58%	61.14 ± 6.06%	76.63 ± 2.28%	81.70 ± 1.77%
GAT	71.21 ± 3.38%	78.23 ± 2.22%	78.22 ± 2.17%	59.63 ± 2.11%	65.51 ± 1.88%	66.19 ± 1.88%	59.52 ± 8.38%	69.28 ± 5.77%	74.20 ± 4.84%
AKGNN	71.98 ± 2.19%	79.19 ± 1.50%	79.32 ± 1.48%	60.83 ± 1.77%	67.07 ± 1.62%	67.74 ± 1.63%	63.14 ± 5.78%	74.21 ± 4.97%	79.18 ± 4.17%
Ordered GNN	70.45 ± 2.54%	77.67 ± 1.83%	77.74 ± 1.83%	59.28 ± 1.82%	64.72 ± 1.70%	65.36 ± 1.69%	70.05 ± 3.17%	79.03 ± 2.09%	82.82 ± 1.89%
Demo-Net	68.41 ± 4.04%	76.35 ± 2.13%	76.28 ± 2.10%	58.01 ± 2.48%	64.03 ± 2.28%	64.64 ± 2.28%	57.18 ± 8.42%	63.18 ± 6.06%	64.92 ± 5.52%
Tail-GCN	67.42 ± 3.32%	76.89 ± 2.00%	77.09 ± 1.94%	57.27 ± 3.00%	64.69 ± 2.56%	65.51 ± 2.49%	59.35 ± 6.52%	72.26 ± 3.47%	76.75 ± 3.21%
ColdBrew-S	51.31 ± 2.98%	55.07 ± 2.28%	55.43 ± 2.19%	49.65 ± 2.15%	53.20 ± 2.06%	53.72 ± 2.09%	61.35 ± 2.97%	67.89 ± 2.52%	70.81 ± 2.12%
ColdBrew-T	71.46 ± 2.41%	79.16 ± 1.73%	79.20 ± 1.69%	60.95 ± 2.04%	66.85 ± 1.81%	67.49 ± 1.78%	70.60 ± 2.15%	78.78 ± 1.92%	82.81 ± 1.58%
RawlsGCN	66.85 ± 2.73%	75.24 ± 2.14%	75.52 ± 2.08%	59.97 ± 2.11%	65.92 ± 2.00%	66.58 ± 2.01%	69.89 ± 2.36%	78.32 ± 1.89%	82.45 ± 1.71%
GRACE	**72.41 ± 3.10%**	**79.70 ± 2.08%**	**79.72 ± 2.06%**	**62.42 ± 2.26%**	**68.26 ± 2.11%**	**68.84 ± 2.10%**	**72.21 ± 2.74%**	**80.49 ± 1.78%**	**84.56 ± 1.58%**

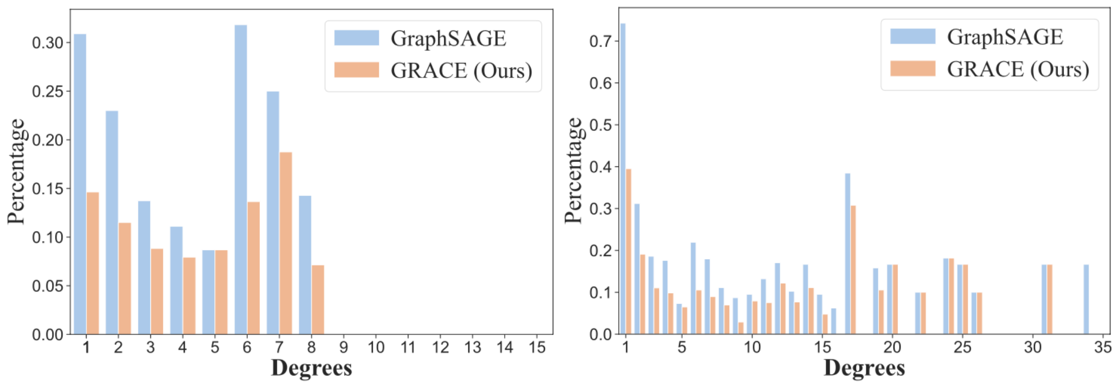
Dataset	Amazon-Computers			Coauthor-CS			Coauthor-Physics		
K	**1**	**5**	**10**	**1**	**5**	**10**	**1**	**5**	**10**
GCN	38.07 ± 3.61%	50.49 ± 4.85%	55.95 ± 6.09%	87.52 ± 1.02%	90.62 ± 0.73%	90.11 ± 0.63%	87.51 ± 1.24%	90.26 ± 0.97%	91.19 ± 0.95%
GraphSAGE	38.30 ± 7.53%	59.19 ± 3.76%	67.07 ± 3.15%	84.84 ± 0.87%	90.97 ± 0.64%	91.43 ± 0.63%	85.65 ± 1.21%	89.80 ± 1.12%	90.69 ± 1.10%
GAT	38.29 ± 7.85%	54.95 ± 6.36%	62.29 ± 5.92%	84.34 ± 1.53%	88.18 ± 1.11%	88.38 ± 1.22%	83.24 ± 1.79%	87.11 ± 1.59%	88.50 ± 1.50%
AKGNN	39.10 ± 4.24%	57.24 ± 4.40%	65.04 ± 4.53%	84.70 ± 1.18%	88.44 ± 0.79%	88.69 ± 0.76%	65.74 ± 12.0%	71.11 ± 11.2%	73.64 ± 10.9%
Ordered GNN	43.20 ± 2.72%	60.95 ± 2.88%	68.01 ± 2.81%	86.42 ± 1.17%	92.00 ± 0.66%	92.24 ± 0.61%	87.15 ± 1.30%	90.48 ± 1.01%	91.24 ± 1.01%
Demo-Net	32.85 ± 5.42%	42.10 ± 4.86%	45.96 ± 4.49%	80.50 ± 1.88%	88.77 ± 1.06%	89.50 ± 1.49%	81.80 ± 2.70%	88.00 ± 1.29%	89.50 ± 1.49%
Tail-GCN	36.75 ± 5.30%	55.79 ± 4.17%	63.36 ± 4.11%	80.50 ± 1.88%	88.77 ± 1.06%	89.50 ± 1.49%	81.80 ± 2.70%	88.00 ± 1.29%	89.50 ± 1.49%
ColdBrew-S	37.59 ± 3.25%	50.99 ± 3.07%	56.75 ± 3.09%	88.29 ± 1.02%	88.47 ± 1.18%	88.23 ± 1.20%	86.46 ± 2.55%	86.75 ± 2.44%	87.43 ± 2.47%
ColdBrew-T	41.65 ± 2.96%	62.05 ± 3.42%	70.05 ± 3.08%	88.18 ± 4.11%	91.28 ± 0.42%	91.67 ± 0.38%	88.52 ± 0.41%	91.28 ± 0.52%	92.08 ± 0.46%
RawlsGCN	42.36 ± 3.22%	59.56 ± 3.62%	67.07 ± 3.60%	89.17 ± 1.09%	91.20 ± 0.81%	91.51 ± 0.67%	87.84 ± 1.44%	89.86 ± 1.05%	90.89 ± 0.92%
GRACE	**48.22 ± 3.99%**	**63.68 ± 3.72%**	**70.88 ± 3.52%**	**89.21 ± 1.03%**	**92.87 ± 0.77%**	**93.08 ± 0.70%**	**89.13 ± 0.60%**	**91.51 ± 0.77%**	**92.20 ± 0.69%**

Experiments

- Performance of various GNNs with different drop- ping ratios on Cora (Left) and Coauthor-CS (Right) datasets.

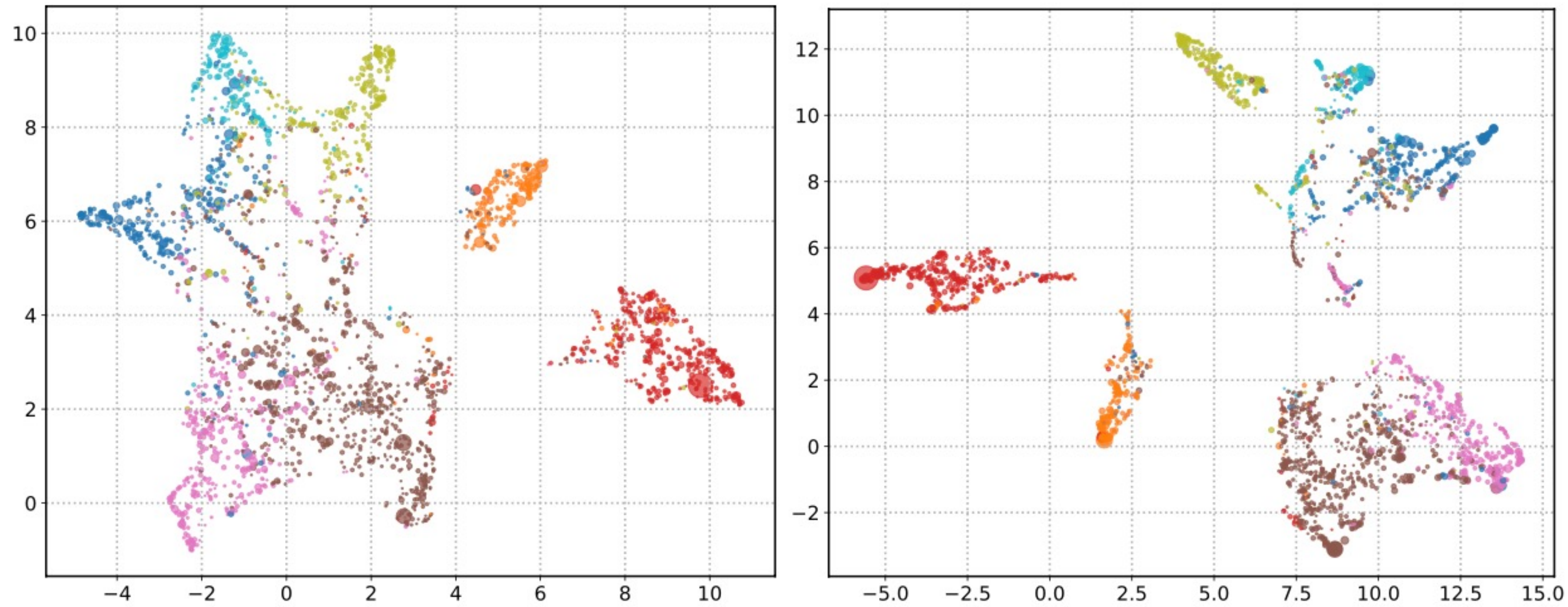


- Percentage of low-NHR (<0.2) nodes of GraphSAGE's misclassified nodes on Cora (Left) and Coauthor-CS (Right) dataset.



Experiments

- Visualization of node representations learned by GraphSAGE (Left) and Grace (Right) on Cora dataset. Different colors denote different classes of nodes.



Conclusion

- We study the problem of degree-related bias on GNNs for long tailed degree distribution, and propose a new framework Grace to solve it
  - The graph self-distillation module is proposed to enhance the self-transformation part in GNNs
  - The graph completion module is proposed to improve the NHR of low-degree nodes
  - Directed completed edges and one-hop label propagation can avoid the error propagation and amplification
- Experiment results demonstrate the effectiveness of our model

Thank You!

Q&A