Data Augmentation for Graph Neural Networks

Tong Zhao†, Yozen Liu‡, Leonardo Neves‡, Oliver Woodford‡, Meng Jiang†, and Neil Shah‡

† Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

‡ Snap Research, Snap Inc., Santa Monica, CA 90405, USA

UNIVERSITY OF NOTRE DAME | COLLEGE OF ENGINEERING

Backgrounds on Data Augmentation

- Why data augmentation is important for machine learning?
  - Provides more training data.
  - Reduce overfitting.
  - Improves generalization.

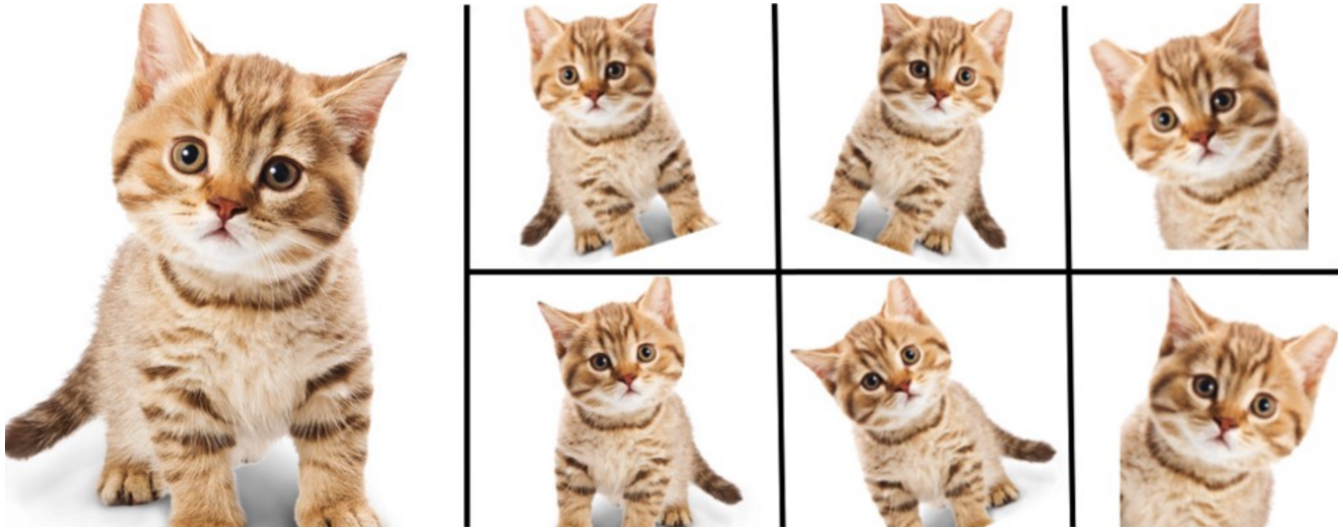
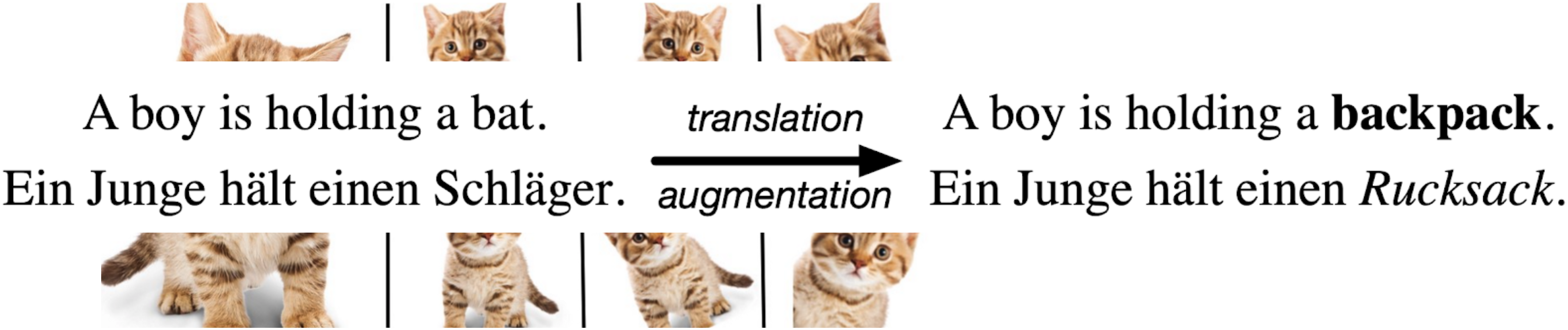


Image sources: https://www.kdnuggets.com/2018/05/data-augmentation-deep-learning-limited-data.html

Backgrounds on Data Augmentation

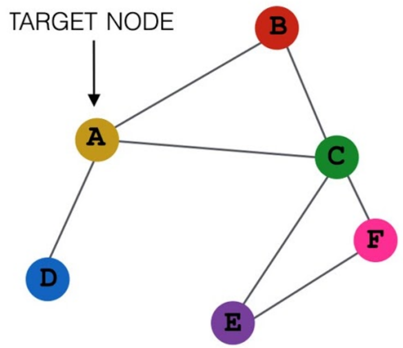
- Why data augmentation is important for machine learning?
  - Provides more training data.
  - Reduce overfitting.
  - Improves generalization.



Background on Graph Neural Networks

- Given: graph $G = (V, E)$, node features $\mathbf{x}_v \in \mathbb{R}^m, \forall v \in V$.
- Learn: low dimensional node representations $\mathbf{z}_v \in \mathbb{R}^d, \forall v \in V$.

- Neighborhood aggregation: generate node representations based on local neighborhoods.



INPUT GRAPH

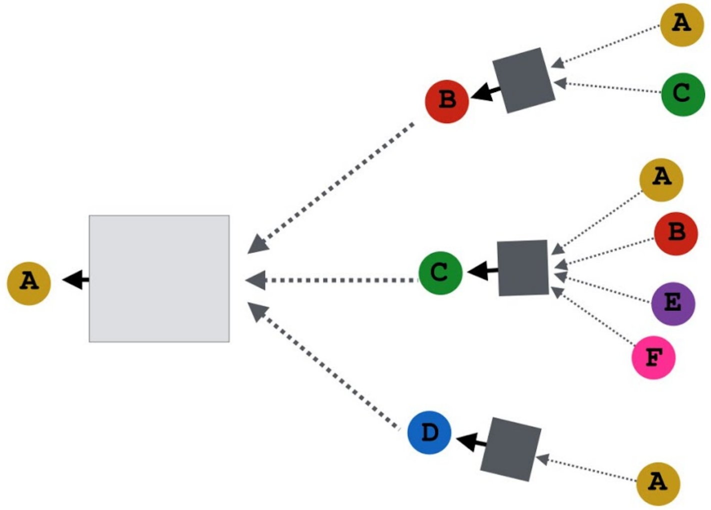


Image sources: Ying, Rex, et al. "Graph convolutional neural networks for web-scale recommender systems." Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2018.

Data Augmentation for GNNs

- **Goal:** use graph data augmentation to improve the performance of GNNs on the task of node classification.

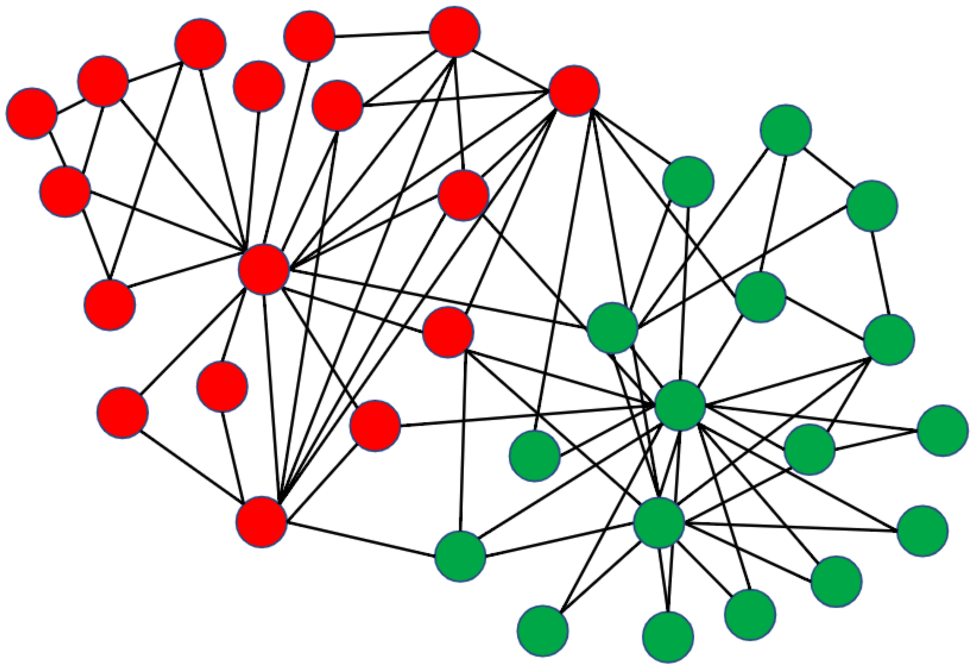
- **Challenges:**
  - There's no direct analogs of traditional data augmentation operations (flipping, rotating, blurring, etc.) on graphs.
  - Very limited operations exist for perturbing graphs.
    - Any manipulation would affect the whole graph (dataset).
    - Adding/removing edges are the best strategy available.
    - But *which* edges to add/remove?

Manipulating Edges to Augment Graph Data

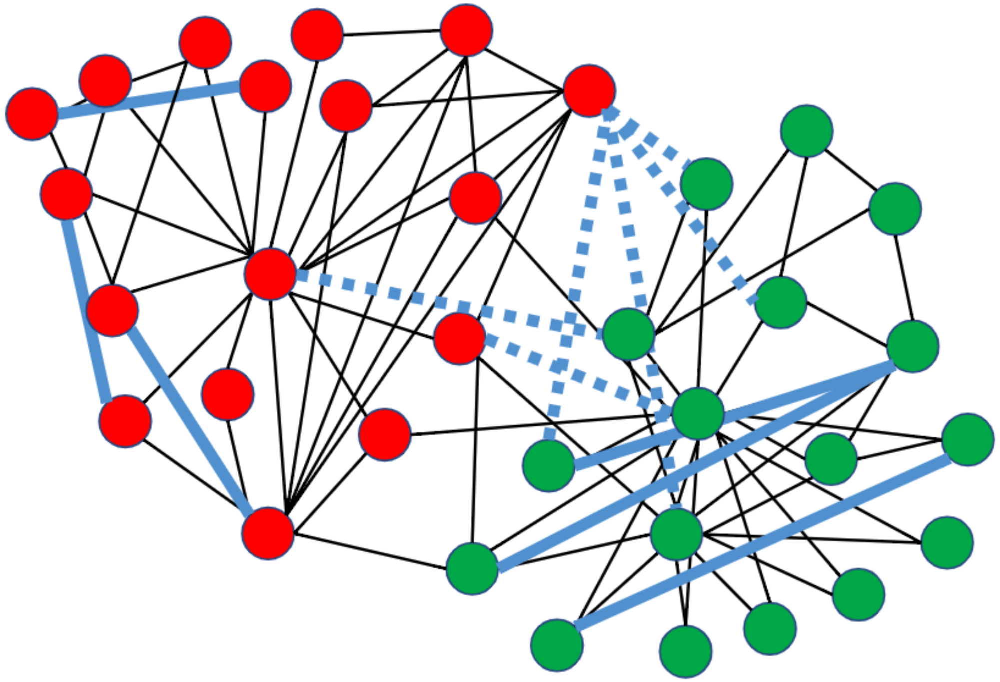
- For node classification task,
  - There could be noise edges generated by spammers, anomalies, adversarial attacks, etc.

- Adding(removing) intra(inter)-class edges improves node classification performance.

Zackary's Karate Club (ZKC) Graph

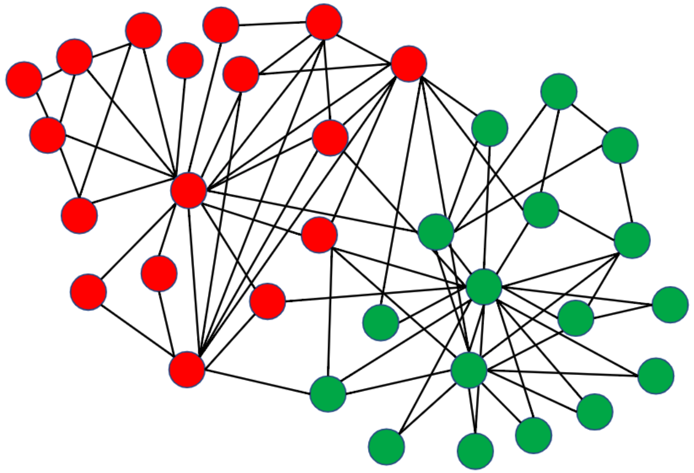


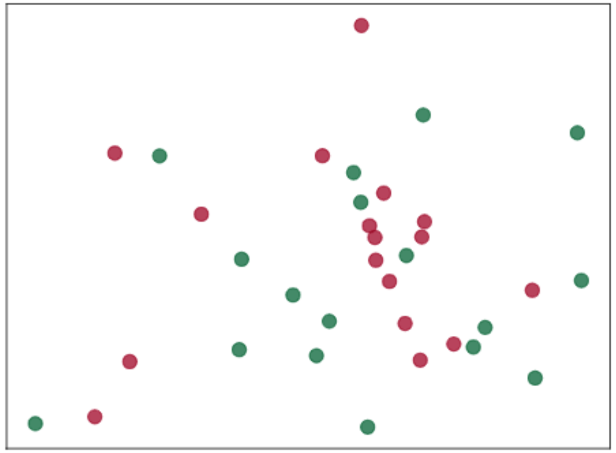
Original graph.
GCN Performance: 92.4



Omniscient modified graph.
GCN Performance: 98.6

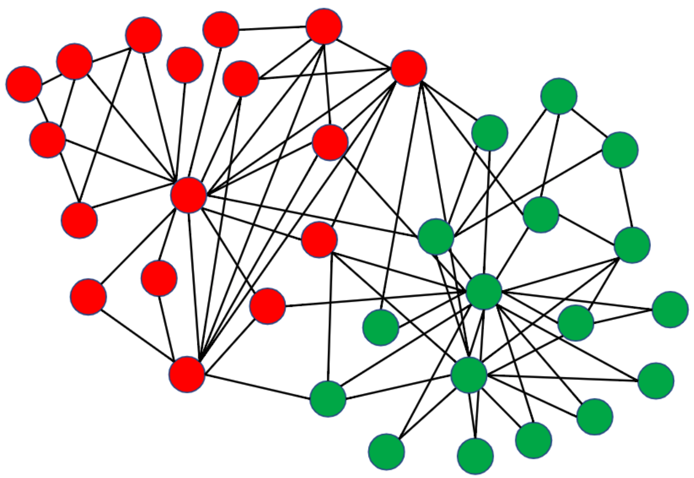
Random Initialized Features of ZKC Graph

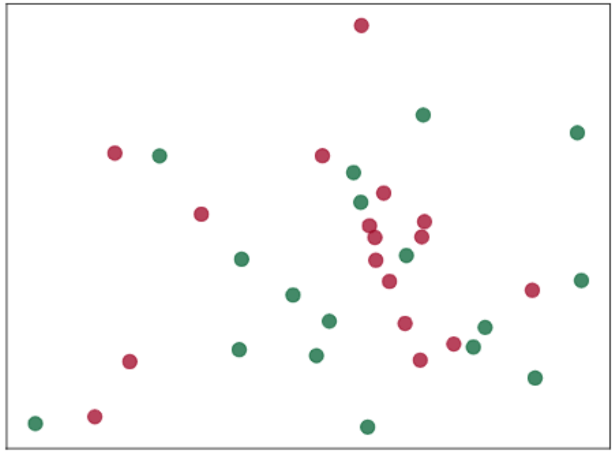




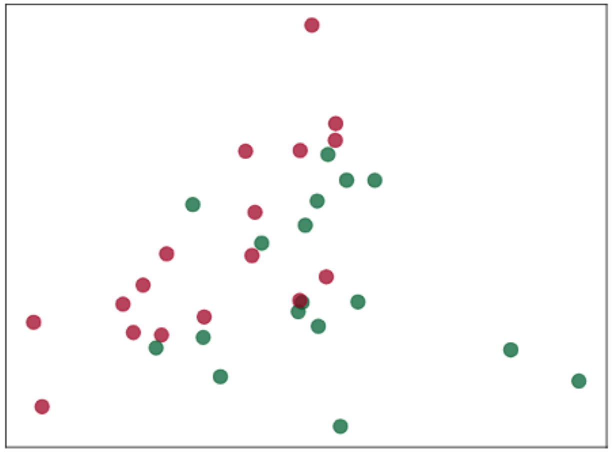
Random initialized features

Embeddings of ZKC Graph after GCN Layer



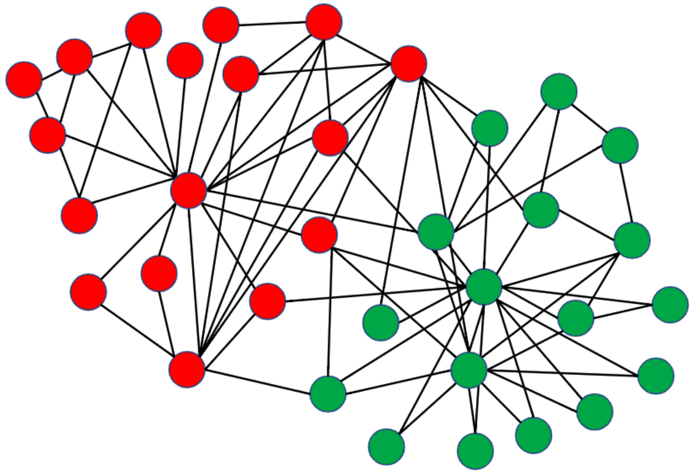


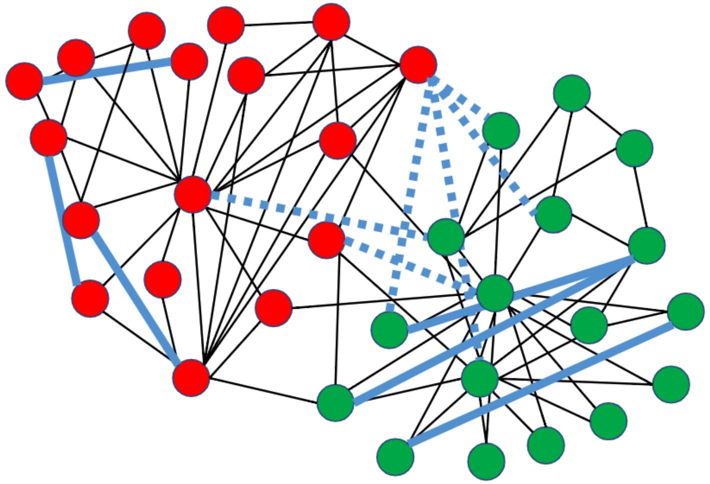
Random initialized features

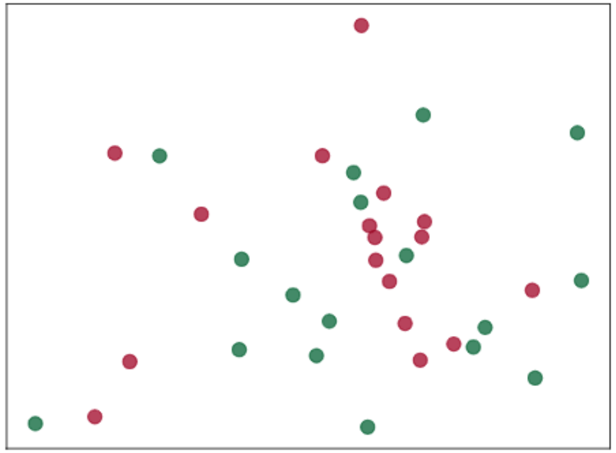


Original Graph

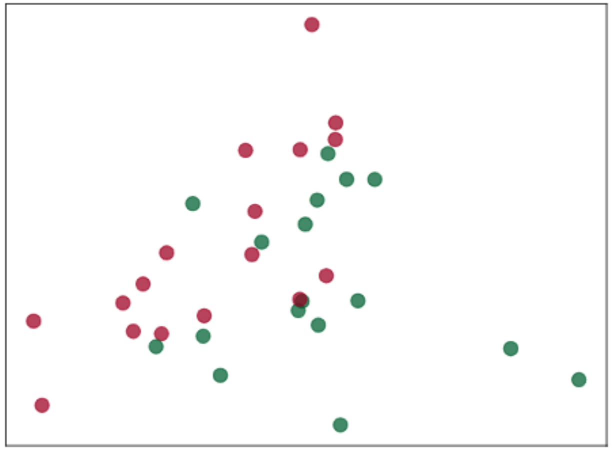
Embeddings of ZKC Graph after GCN Layer



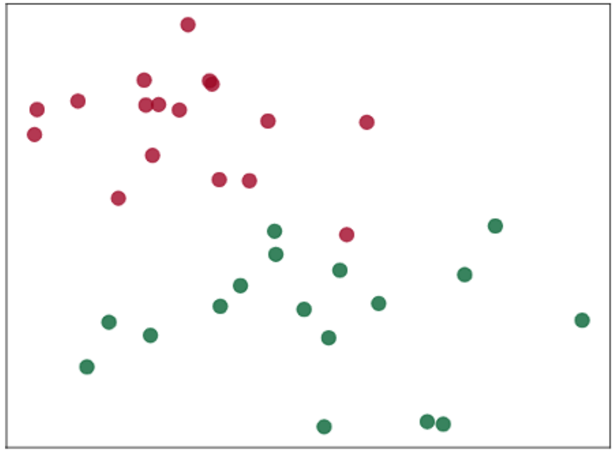




Random initialized features

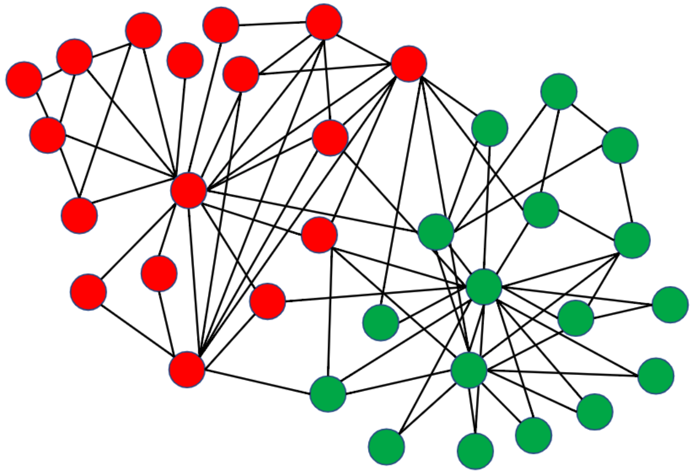


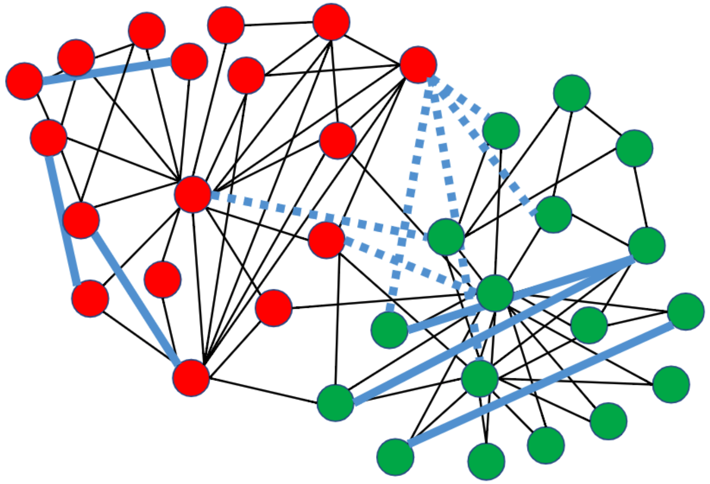
Original Graph

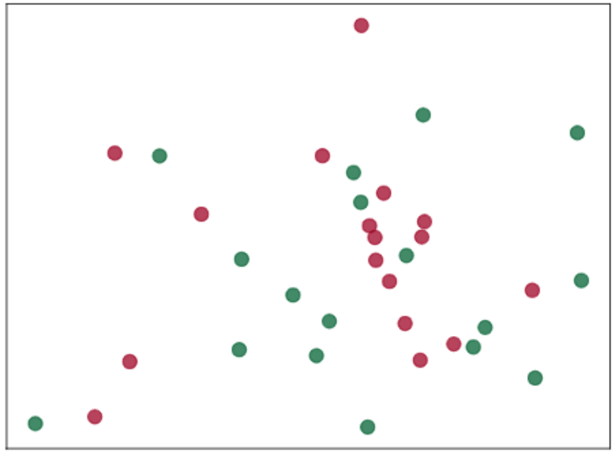


Modified Graph

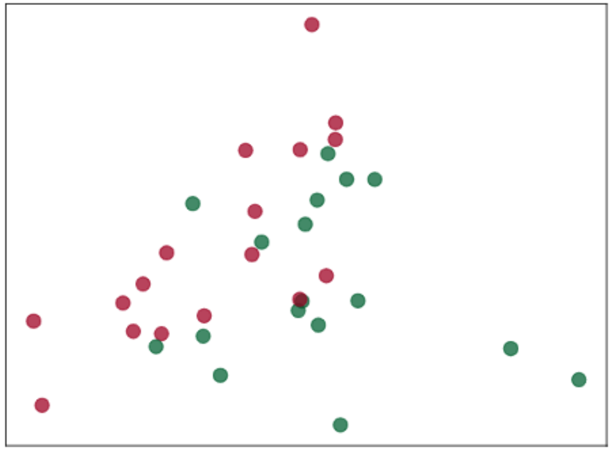
Embeddings of ZKC Graph after GCN Layer



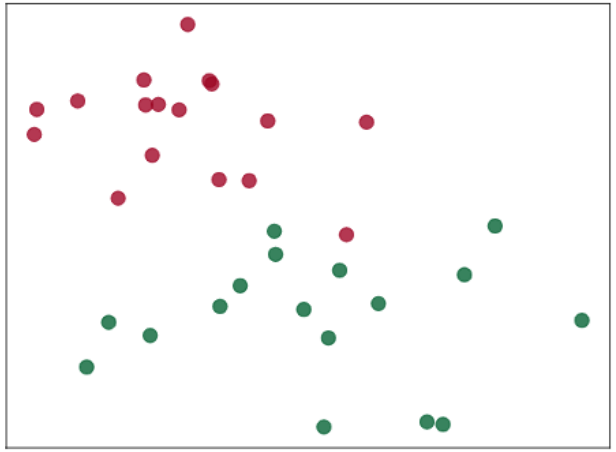




Random initialized features



Original Graph



Modified Graph

Ideal Graph

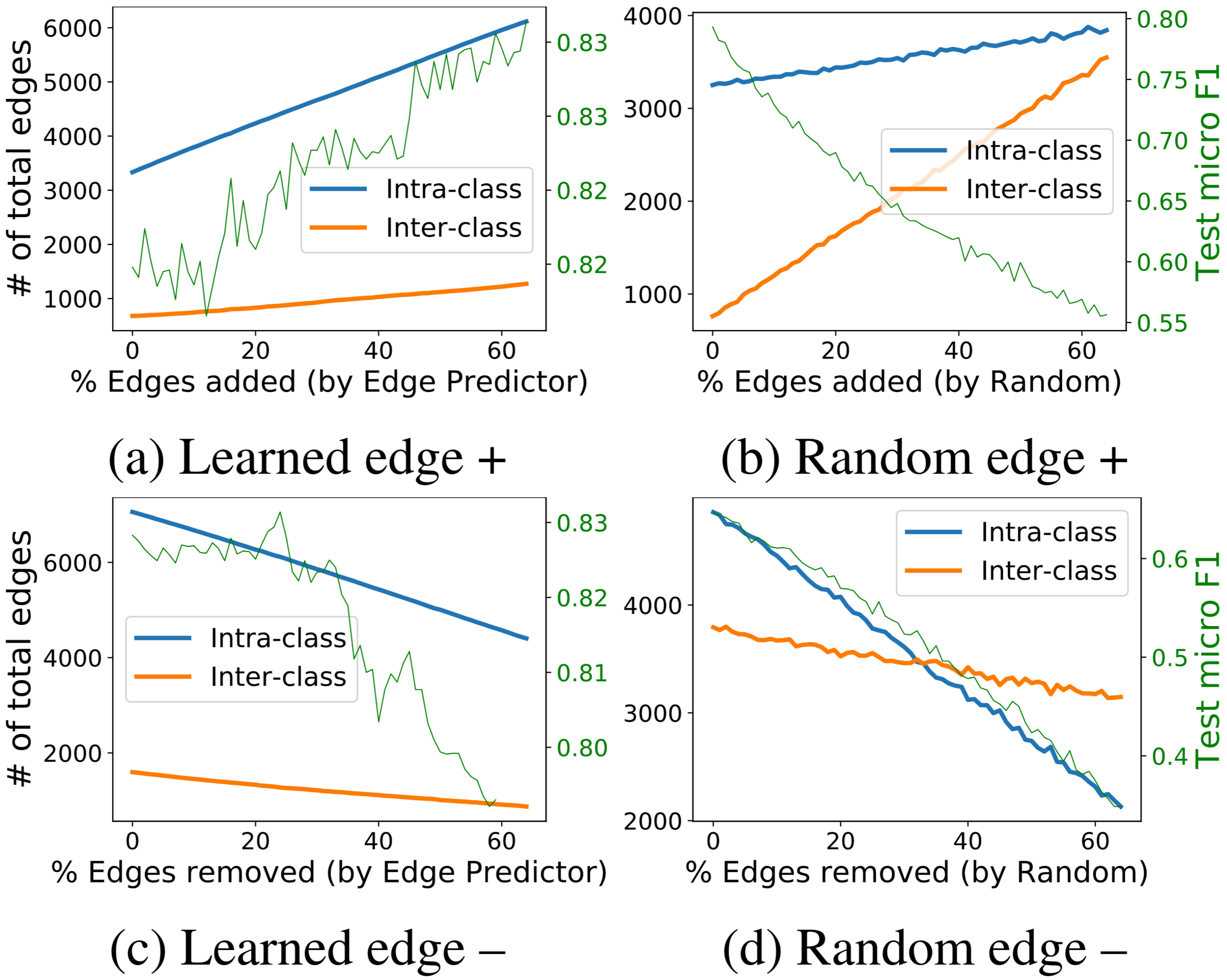
Evaluating on Modified or Original Graph

- How traditional data augmentation methods in CV works:
  - Generate augmented data variants for each data object.
  - Train model with both original and augmented data.

- On graphs: augmentation results with a new graph.
  - Training & inference on different graphs → train-test gap.
  - For real-life social networks that are consistently growing/changing, ability of inferencing with original graph is preferred.

GAug-M

1. Use an edge predictor to predict edge probabilities for all node pairs.
2. Based on the edge probabilities, deterministically add (remove) new (existing) edges to create a modified graph.

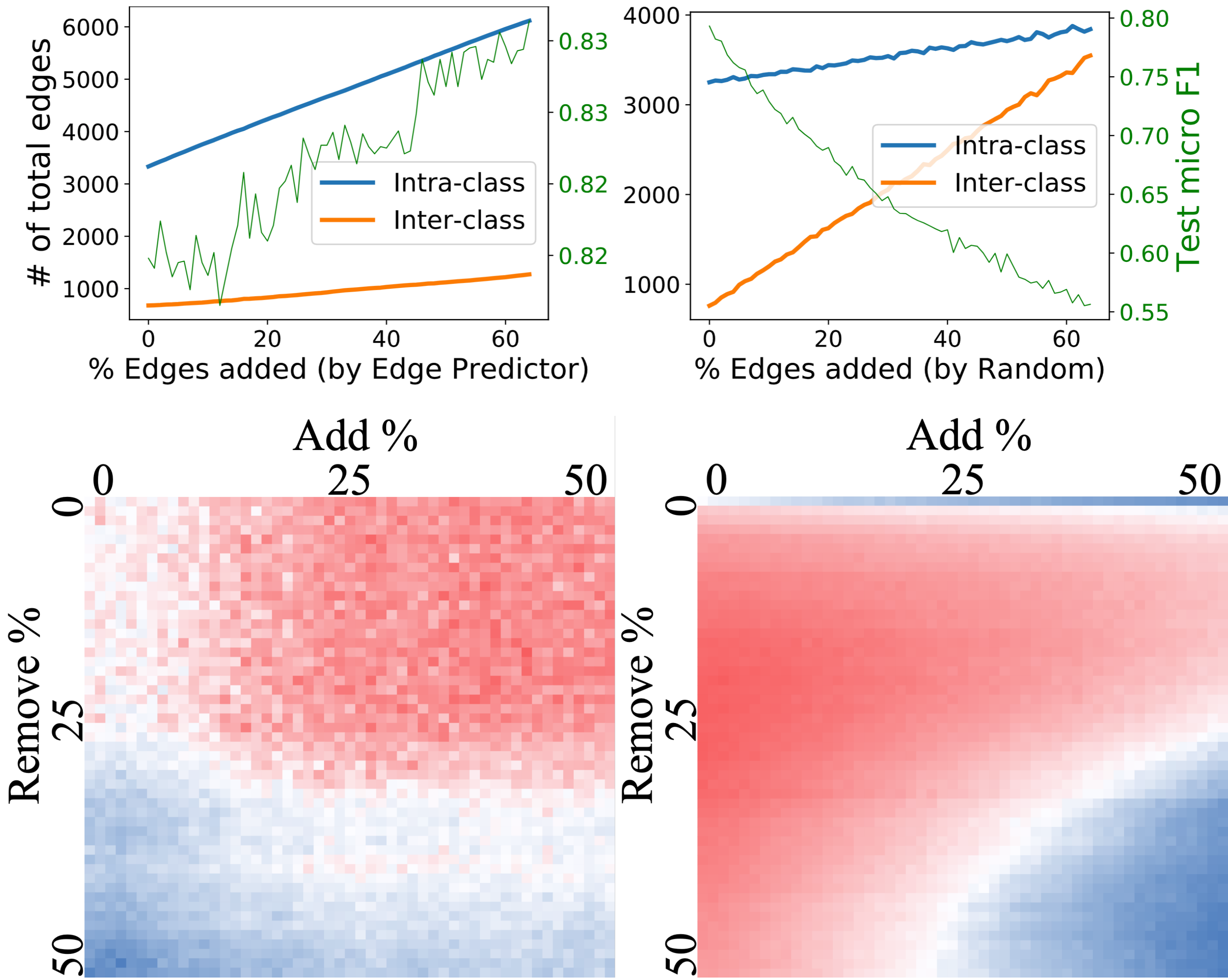


(a) Learned edge +
(b) Random edge +
(c) Learned edge –
(d) Random edge –

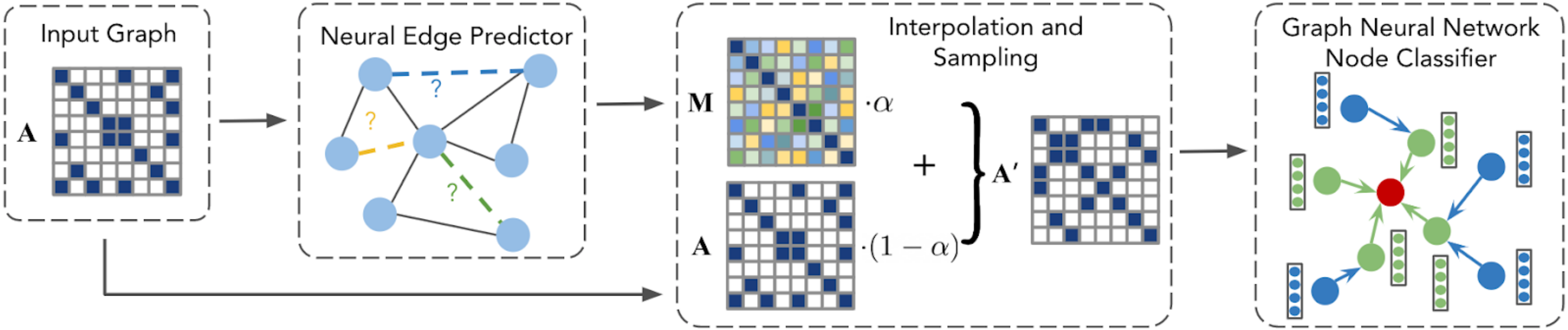
GAug-M

1. Use an edge predictor to predict edge probabilities for all node pairs.

2. Based on the edge probabilities, deterministically add (remove) new (existing) edges to create a modified graph.



GAug-O for Evaluating on Original Graph



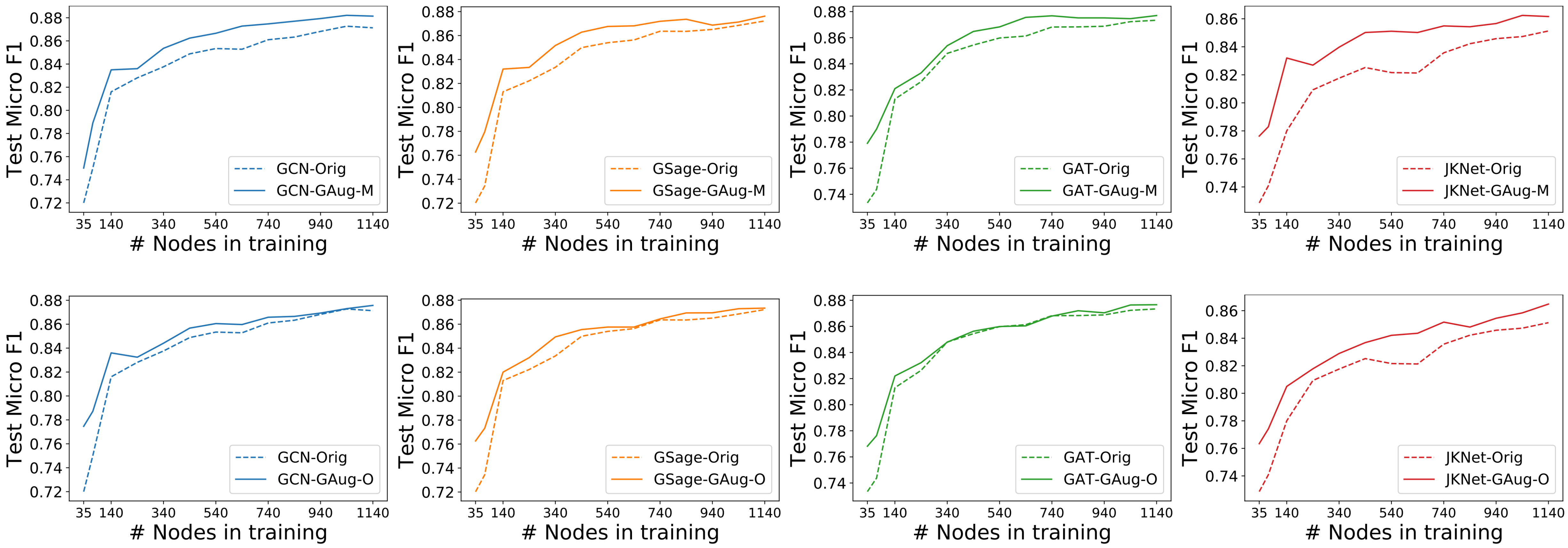
$$\mathbf{A}'_{ij} = \left\lfloor \frac{1}{1+e^{-(\log \mathbf{P}_{ij}+G)/\tau}} + \frac{1}{2} \right\rfloor, \quad \text{where} \quad \mathbf{P}_{ij} = \alpha \mathbf{M}_{ij} + (1-\alpha)\mathbf{A}_{ij}$$

$$\mathcal{L} = \mathcal{L}_{nc} + \beta \mathcal{L}_{ep}, \quad \text{where} \quad \mathcal{L}_{nc} = CE(\hat{\mathbf{y}}, \mathbf{y}) \;\; \text{and} \;\; \mathcal{L}_{ep} = BCE(\sigma(f_{ep}(\mathbf{A}, \mathbf{X})), \mathbf{A})$$

Results

GNN Arch.	Method	Cora	Citeseer	PPI	BlogC	Flickr	Air-USA
GCN	Original	81.6±0.7	71.6±0.4	43.4±0.2	75.0±0.4	61.2±0.4	56.0±0.8
	+AdaEdge	81.9±0.7	72.8±0.7	43.6±0.2	75.3±0.3	61.2±0.5	57.2±0.8
	+GAug-M	83.5±0.4	72.3±0.4	43.5±0.2	**77.6±0.4**	**68.2±0.7**	61.2±0.5
	+DropEdge	82.0±0.8	71.8±0.2	43.5±0.2	75.4±0.3	61.4±0.7	56.9±0.6
	+GAug-O	**83.6±0.5**	**73.3±1.1**	**46.6±0.3**	75.9±0.2	62.2±0.3	**61.4±0.9**
GSAGE	Original	81.3±0.5	70.6±0.5	40.4±0.9	73.4±0.4	57.4±0.5	57.0±0.7
	+AdaEdge	81.5±0.6	71.3±0.8	41.6±0.8	73.6±0.4	57.7±0.7	57.1±0.5
	+GAug-M	**83.2±0.4**	71.2±0.4	41.1±1.0	**77.0±0.4**	**65.2±0.4**	**60.1±0.5**
	+DropEdge	81.6±0.5	70.8±0.5	41.1±1.0	73.8±0.4	58.4±0.7	57.1±0.5
	+GAug-O	82.0±0.5	**72.7±0.7**	**44.4±0.5**	73.9±0.4	56.3±0.6	57.1±0.7
GAT	Original	81.3±1.1	70.5±0.7	41.5±0.7	63.8±5.2	46.9±1.6	52.0±1.3
	+AdaEdge	82.0±0.6	71.1±0.8	42.6±0.9	68.2±2.4	48.2±1.0	54.5±1.9
	+GAug-M	82.1±1.0	71.5±0.5	42.8±0.9	70.8±1.0	**63.7±0.9**	**59.0±0.6**
	+DropEdge	81.9±0.6	71.0±0.5	**45.9±0.3**	70.4±2.4	50.0±1.6	52.8±1.7
	+GAug-O	**82.2±0.8**	**71.6±1.1**	44.9±0.9	**71.0±1.1**	51.9±0.5	54.6±1.1
JK-Net	Original	78.8±1.5	67.6±1.8	44.1±0.7	70.0±0.4	56.7±0.4	58.2±1.5
	+AdaEdge	80.4±1.4	68.9±1.2	44.8±0.9	70.7±0.4	57.0±0.3	59.4±1.0
	+GAug-M	**81.8±0.9**	68.2±1.4	47.4±0.6	**71.9±0.5**	**65.7±0.8**	60.2±0.6
	+DropEdge	80.4±0.7	69.4±1.1	46.3±0.2	70.9±0.4	58.5±0.7	59.1±1.1
	+GAug-O	80.5±0.9	**69.7±1.4**	**53.1±0.3**	71.0±0.6	55.7±0.5	**60.4±1.0**

Results



Test Micro F1
Nodes in training
35 140 340 540 740 940 1140
GCN-Orig
GCN-GAug-M
GSage-Orig
GSage-GAug-M
GAT-Orig
GAT-GAug-M
JKNet-Orig
JKNet-GAug-M
GCN-Orig
GCN-GAug-O
GSage-Orig
GSage-GAug-O
GAT-Orig
GAT-GAug-O
JKNet-Orig
JKNet-GAug-O

Thank you!

Any Questions?

COLLEGE OF ENGINEERING

UNIVERSITY OF NOTRE DAME