Edgeless-GNN: Unsupervised Inductive Edgeless Network Embedding

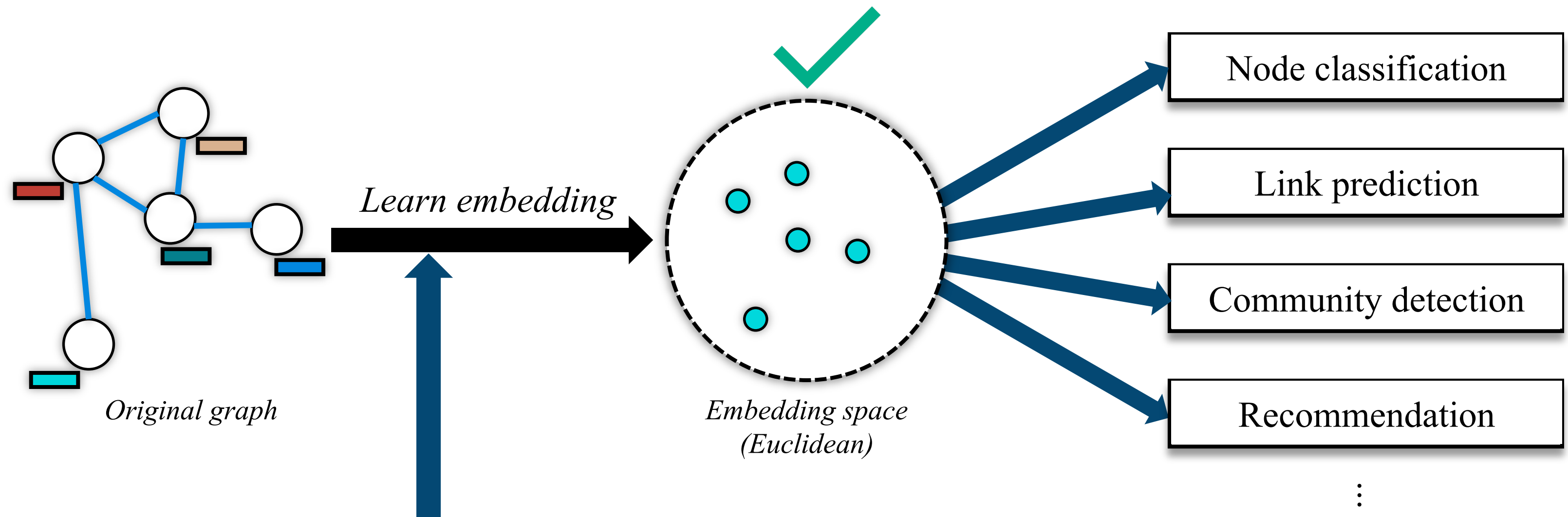
Presenter: Yong-Min Shin

28th Humantech paper award

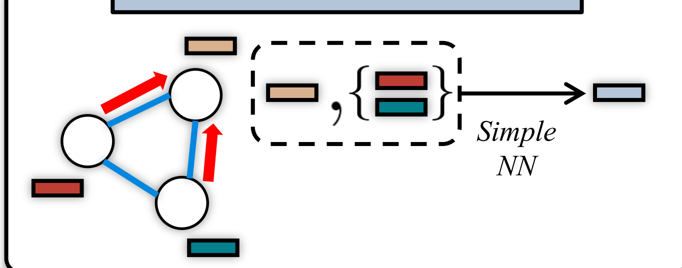
Introduction

Network embedding with graph neural networks (GNNs)

Embedding based methods



GNN architecture



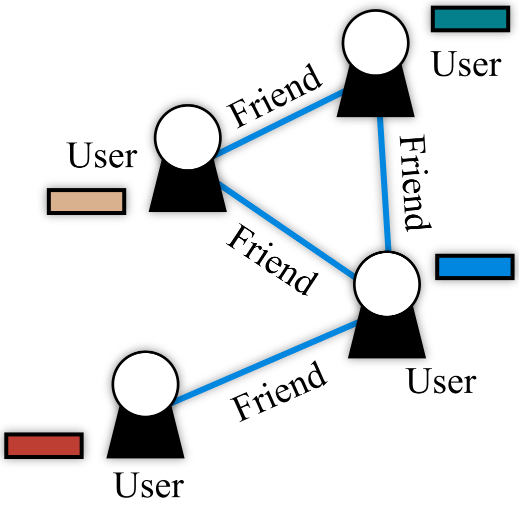
- Network embedding: Map each node to a vector in embedding space
- Embeddings can be used to perform various downstream tasks
- **GNNs**: Neural network architecture, need edges to feed-forward

Introduction

Edgeless nodes: Real-world example

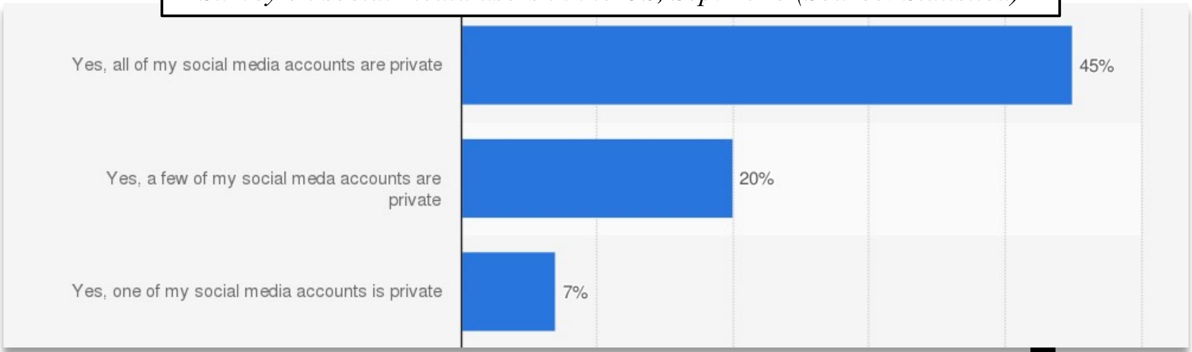
Examples

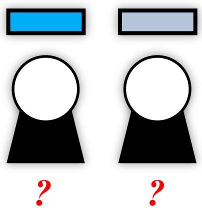
- Social network: Completely new user or hide relations due to privacy in social platforms (e.g., Facebook)
- Citation network: Single / isolated author



1. Some users hide their friendship information online due to privacy concerns.

Survey on social media users in the US, Sep. 2018 (Source: Statistica)

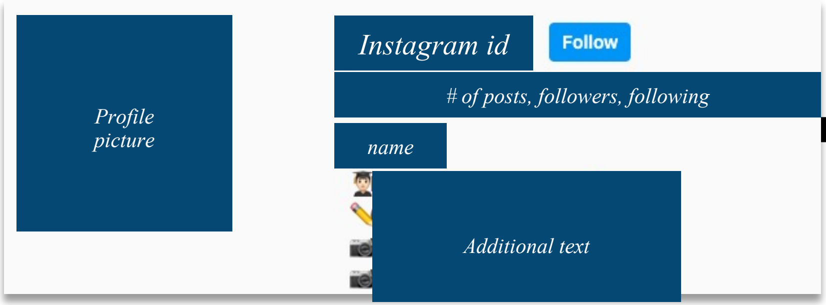




Edgeless node example

- Social networks
- Edgeless due to privacy concerns

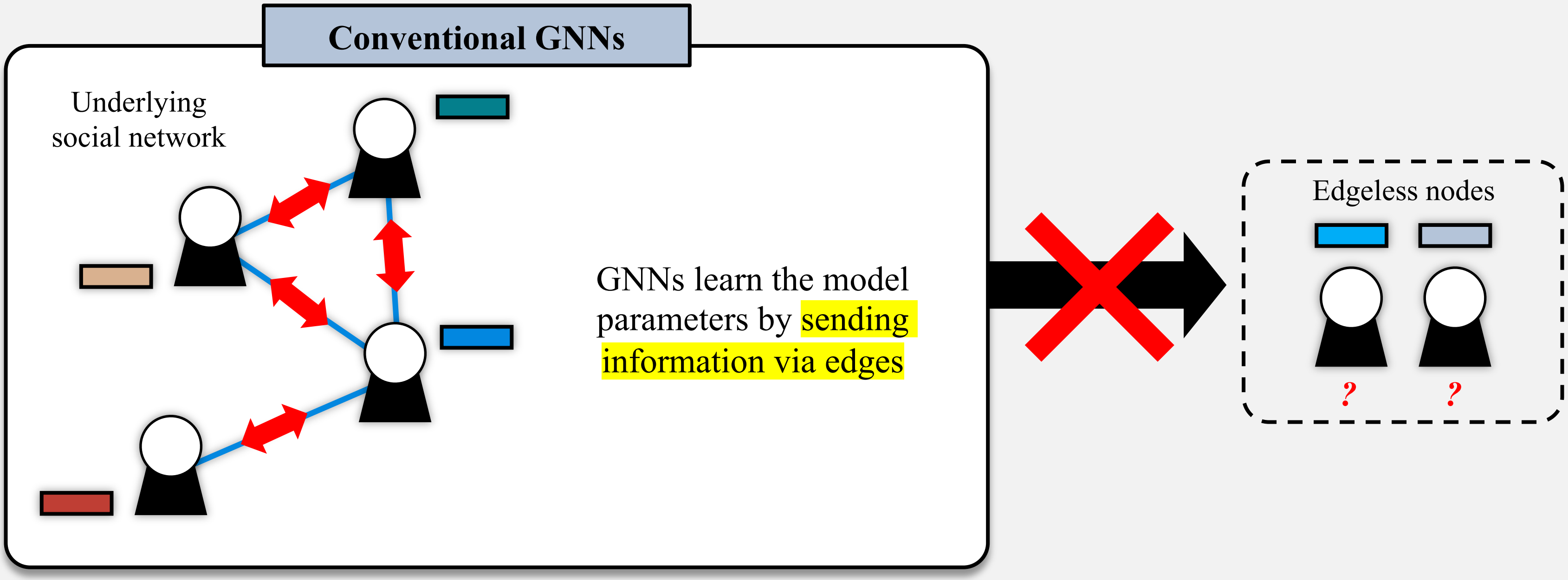
2. Attribute information is relatively easy to find



Target: Edgeless node with attribute information

Challenge

How to apply GNNs to nodes without edges?



Examples

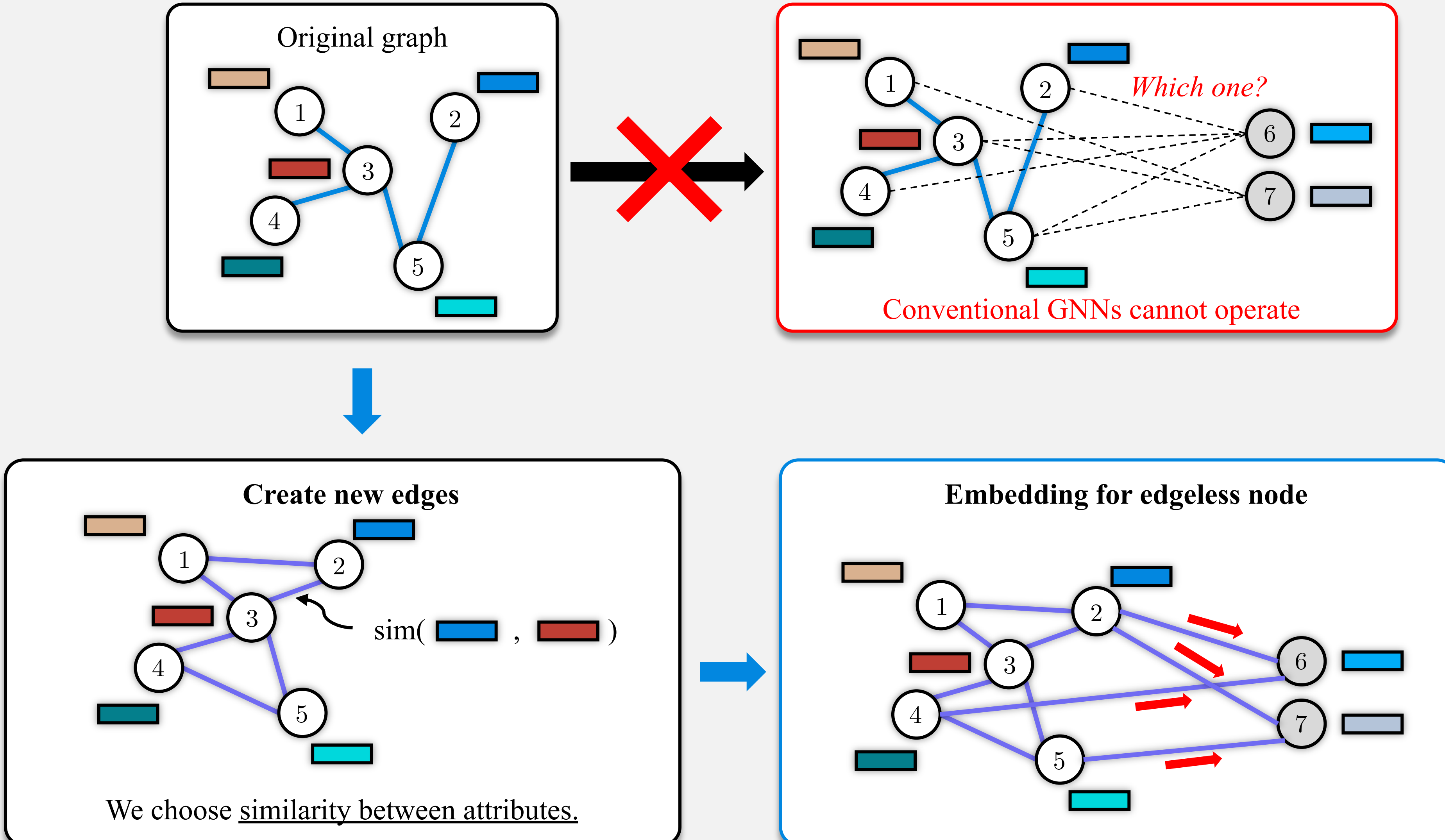
- Social network: Completely new user or hide relations due to privacy in social platforms (e.g., Facebook)
- Citation network: Single / isolated author

- Conventional GNNs cannot operate for nodes without any edges
- **Can we design a new GNN-model-agnostic framework that can learn representations for such edgeless nodes?**

Methodological contributions

Key idea 1: Computation graph construction

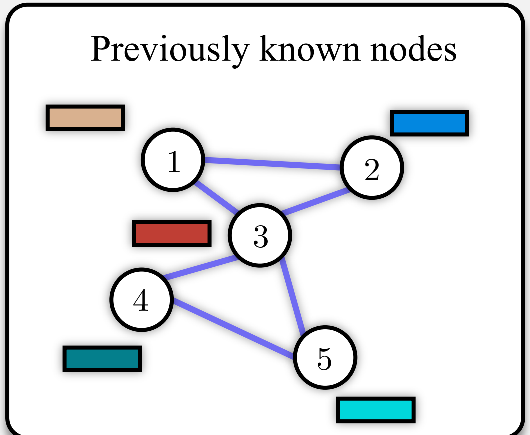
Biggest challenge: GNNs cannot perform computation to nodes without edges



Methodological contributions

Key idea 2: GNN-agnostic inductive framework

Training phase

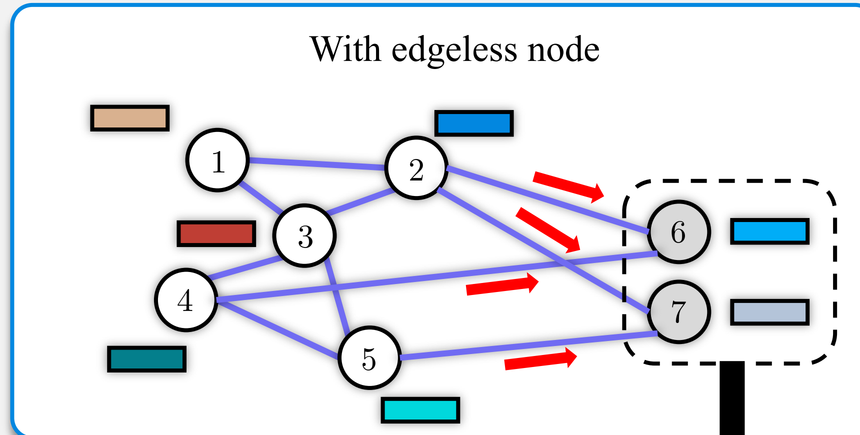


Train parameters

GNN model

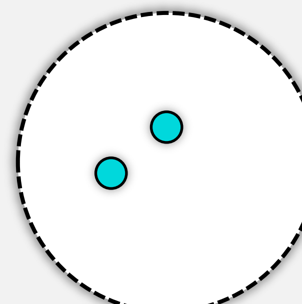
- GCN (Kipf & Welling, ICLR)
- GraphSAGE (Hamilton et al., NeurIPS)
- GAT (Veličković et al., ICLR)
- …

Test phase



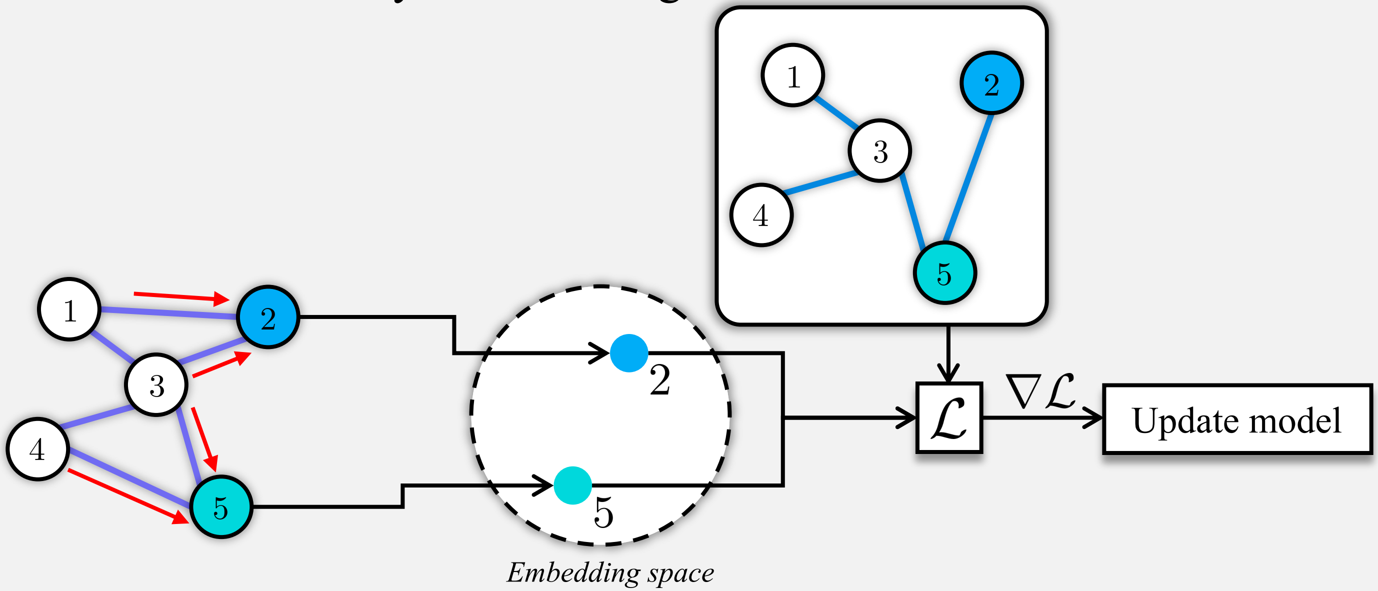
GNN model with trained parameters

- Acquire embedding for edgeless node by one feed-forward pass
- Plug & play framework



Methodological contributions

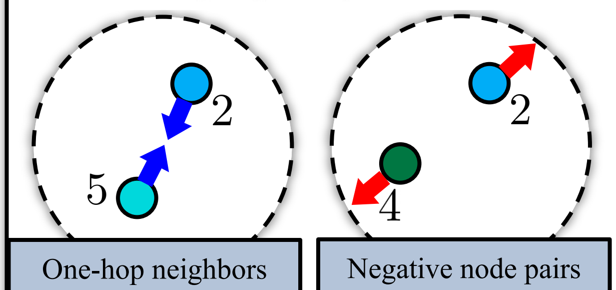
Key idea 3: Design of loss function



Loss function (Energy-based learning)

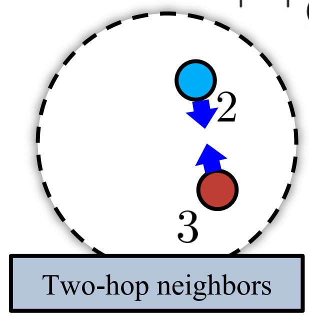
$$\mathcal{L} = \mathcal{L}_0 + \alpha\mathcal{L}_{\text{2nd}}$$

$$\mathcal{L}_0 = \frac{1}{|\mathcal{B}'|}\sum_{(v_i,v_j,v_n,v_t)\in\mathcal{B}'}(E_{ij}^{+} + D_{in}E_{in}^{-})$$



Pulls one-hop neighbors & pushes negative node pairs

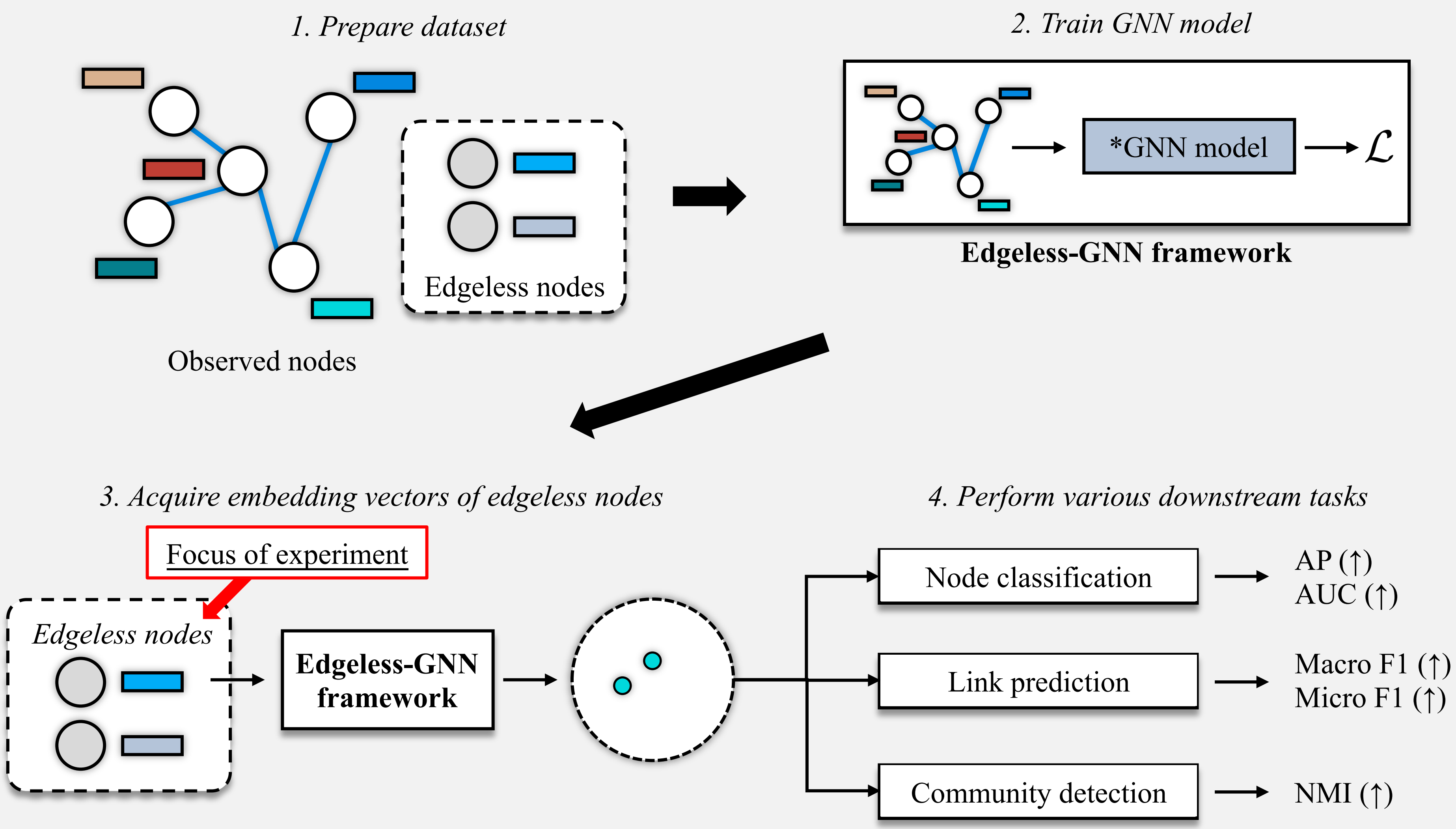
$$\mathcal{L}_{\text{2nd}} = \frac{1}{|\mathcal{B}'|}\sum_{(v_i,v_j,v_n,v_t)\in\mathcal{B}'}J_{it}E_{it}^{+}$$



Pulls two-hop neighbors considering common neighbors

Empirical contributions

Experiment procedure

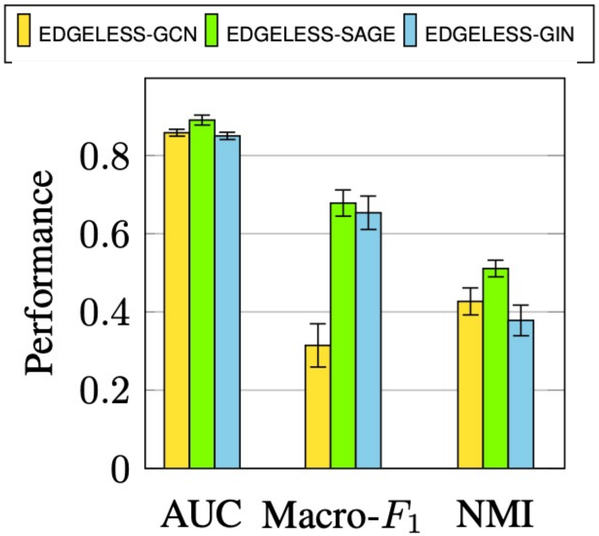


* As Edgeless-GNN can operate on most GNNs, we investigate on well-known models from the literature.

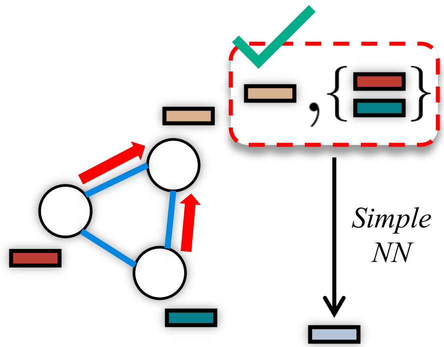
Empirical contributions

Empirical analysis on Edgeless-GNN

Empirical finding 1



GraphSAGE consistently outperforms: **Concatenation** allows the model to assign different weights for self and neighbors.



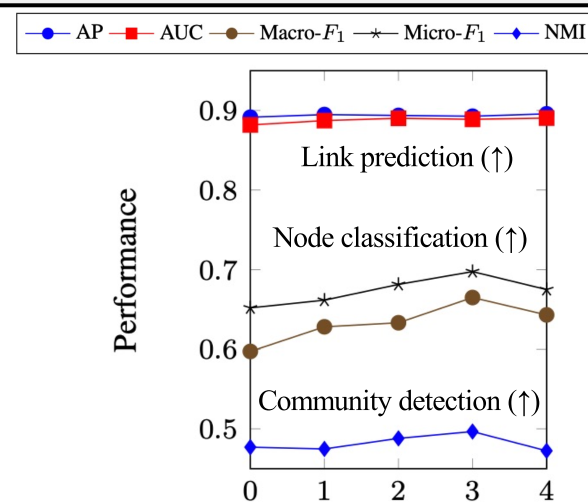
- GraphSAGE (Hamilton et al., NeurIPS) *Concatenation – Does not mix*
- GCN (Kipf & Welling, ICLR) & GIN (Xu et al., ICLR) *Weighted average*

Empirical finding 2

Loss function (Energy-based learning)

$$\mathcal{L} = \mathcal{L}_0 + \alpha \mathcal{L}_1$$

- Second-hop proximity information affects **node classification & community detection** more
- The value of α is optimal at 3 for Cora dataset
- **Relative strength** between two loss terms matter



Empirical & Analytical contributions

Performance comparison & Complexity

Experimental results on Cora dataset (partial)

Task	Link prediction	Node classification	Community detection
Metric	AUC (↑)	Macro F1 (↑)	NMI (↑)
Edgeless-GNN	0.8905 ± 0.012	0.6783 ± 0.014	0.5109 ± 0.021
DEAL (Hao et al., 2020)	0.8585 ± 0.011	0.6410 ± 0.033	0.4321 ± 0.019
G2G (Bojchevski et al., 2018)	0.7966 ± 0.047	0.5983 ± 0.017	0.4089 ± 0.035
***Att-Only**	0.7546 ± 0.013	0.4923 ± 0.035	0.2213 ± 0.060

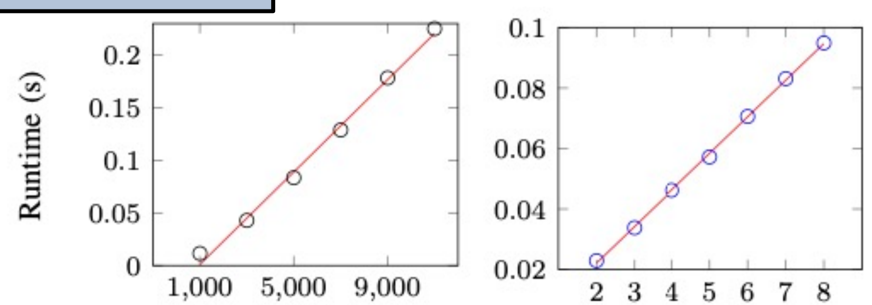
***Att-Only**: Use given node attributes as representations

Performance comparison

- The superiority is achieved from the loss function design.
  1. Pushing negative node pairs w.r.t. shortest-path distance
  2. Additionally considering second-order positive relations
- DEAL: Originally designed for link prediction
- G2G: KL divergence in loss does not preserve transitivity

Analytical analysis

- Computational complexity: $\mathcal{O}(k|\mathcal{V}|)$
- This linearity with respect to k and the number of nodes are also shown empirically.



(a) $|\mathcal{V}|$ (b) k

Conclusion

1. World's first attempt to apply GNNs to edgeless nodes
2. New inductive and unsupervised framework design of GNNs for edgeless nodes
3. Our Edgeless-GNN framework outperforms previous state-of-the-art methods by virtue of our judiciously designed loss

- Thank you for your listening -