# Disentangled Graph Convolutional Networks

Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, Wenwu Zhu
Team : Tsinghua University
Conference : ICML '19

CAU
Junseo, Yu

DMAIS Lab Meeting
10.04.2024

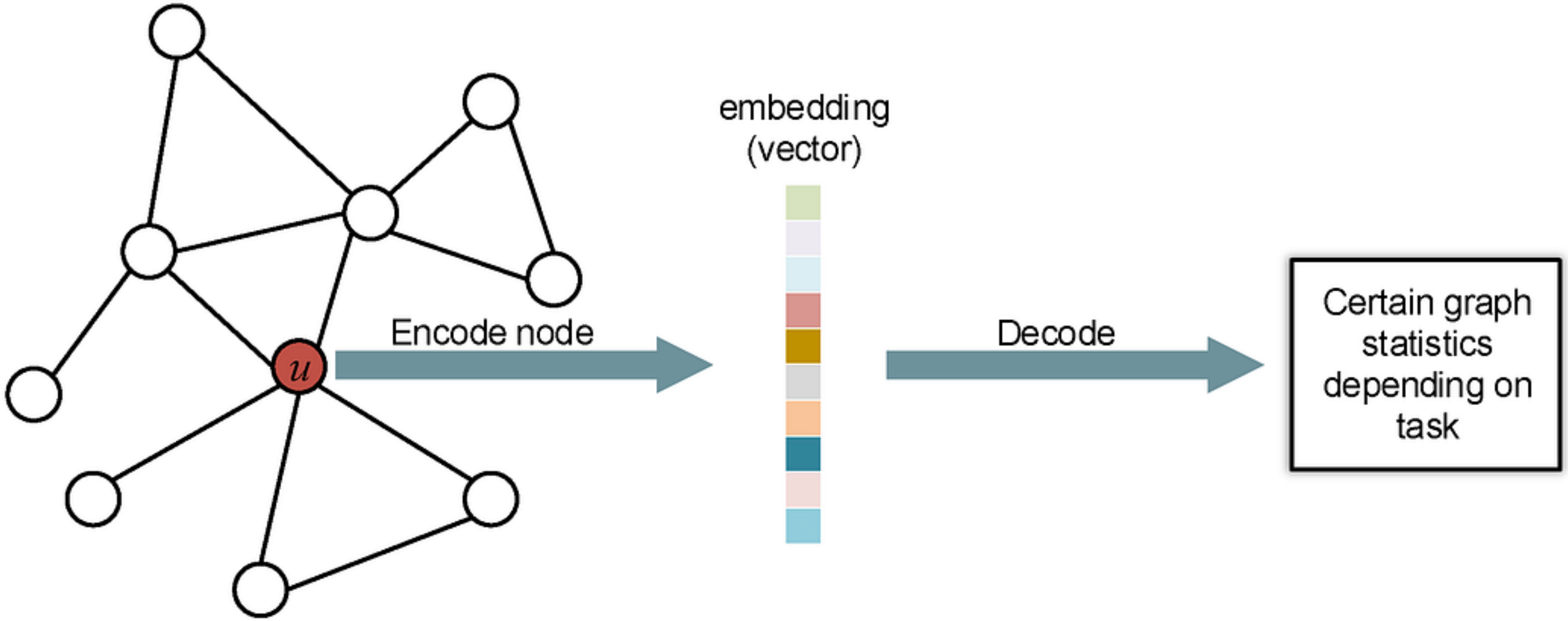
# Contents

# 1. Introduction

- Problem & Motivation
- Background
- Research Objective

# Problem & Motivation






- ❑ Representation Learning for a node
  - ❑ Describe node's neighborhood information
  - ❑ Matrix Factorization, Random walk, **GNNs**

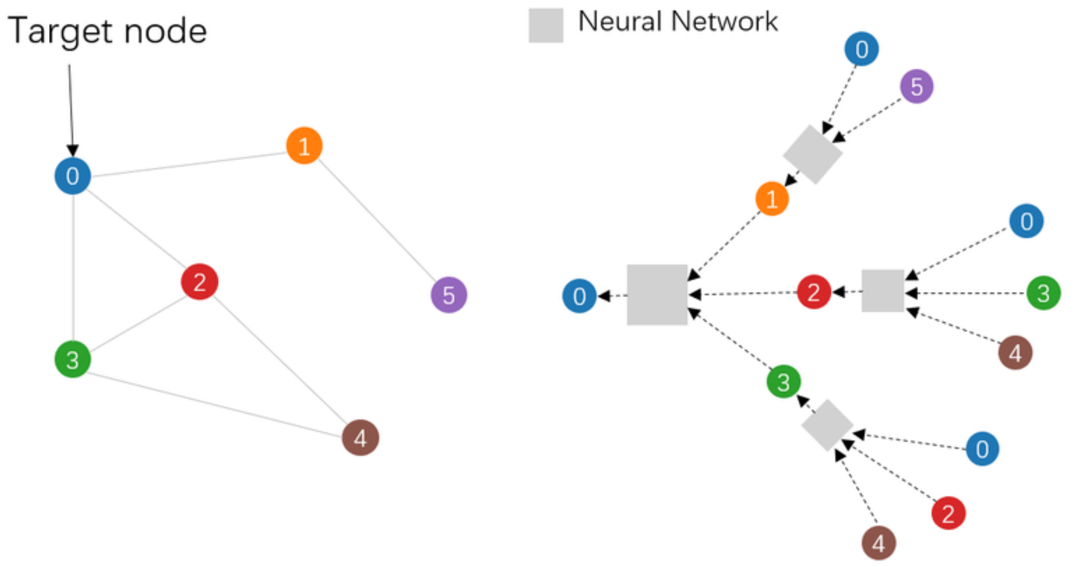
# Problem & Motivation



❑ Problem of GNNs

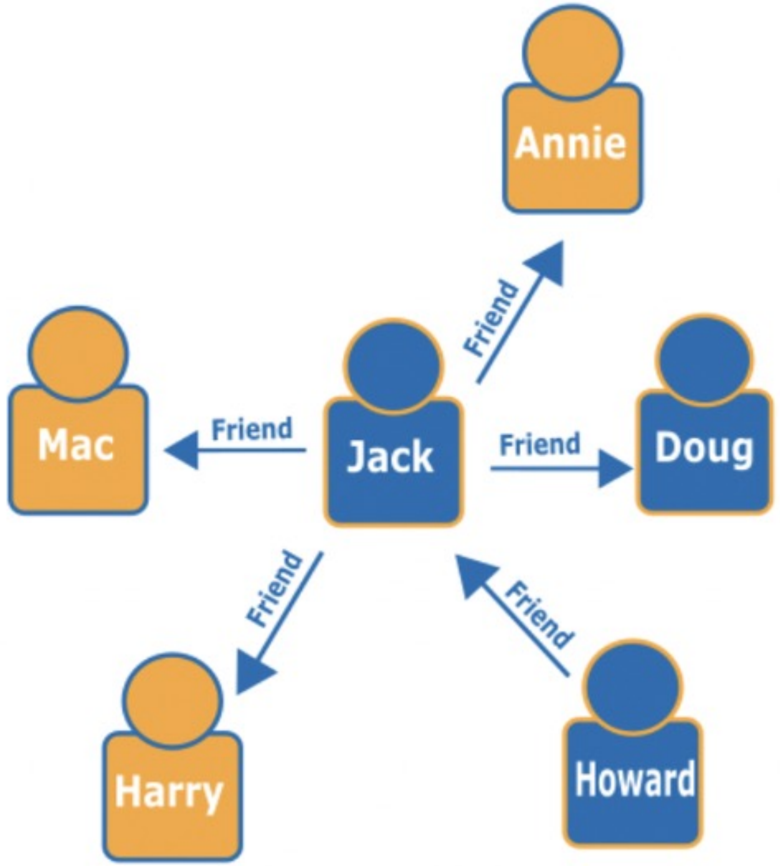
- Generally, take a holistic(Integrated) approach

  ➔ The nuances between the different parts of the neighborhood are ignored
- However, the real-world graph are complex and heterogeneous




(a) Input graph (b) Neighborhood aggregation

# Problem & Motivation



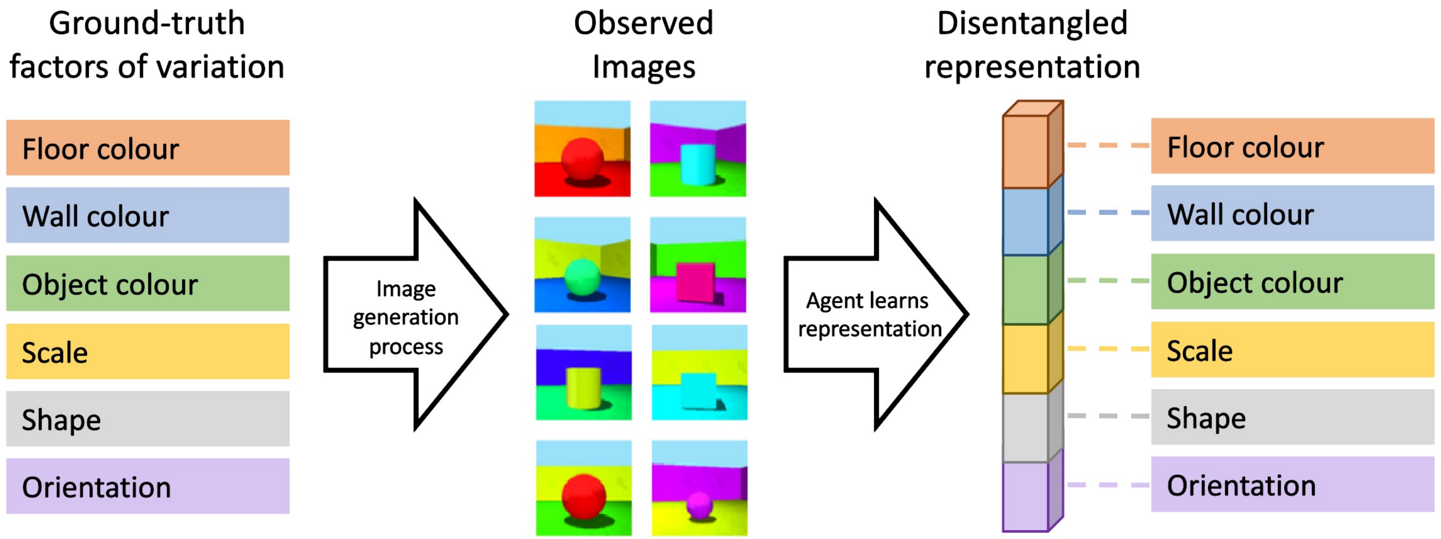

- ❖ Scenario
  - ❑ Jack in a social network
  - ❑ Mac, Harry, and Annie are high school friends
  - ❑ Doug and Howard are co-workers
  - ❑ Connects with others for various reasons

**Needs to be performed separately for each component**

# Background



- Disentangled representation learning
  - In Computer Vision field, this method has gained attentions
  - Able to bring enhanced generalization ability, robustness, and interpretability




- Scenario
  - Try to detect object shape
  - Orange floor color and blue wall color and **round shape**
  - **Rainbow floor color** and blue wall color and **round shape**
  - Does not need all combination patterns in train data

# Background

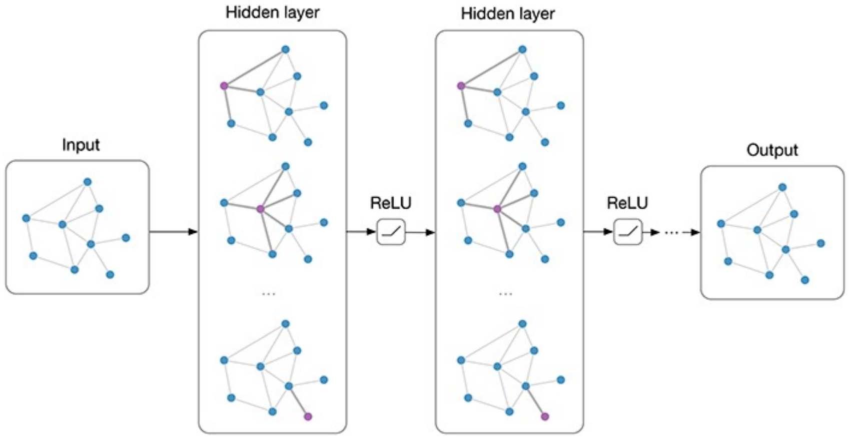


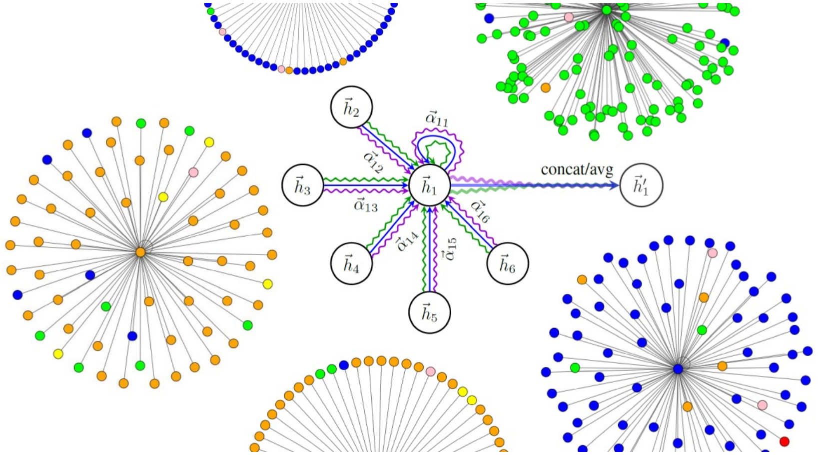
- Disentangled representation learning **in Graph**
  - Remained largely **unexplored** due to the characteristics of graphs
  - The complex formation and the limited information available make hard to infer the latent factor

# Background



- Existing methods
  - Graph Convolutional Networks
    - Cannot learn disentangled node representations
    - May produce overly smoothed representations (over-smoothing)
  - Attention Mechanism
    - Try to prune the irrelevant elements.
    - However, remains a holistic approach

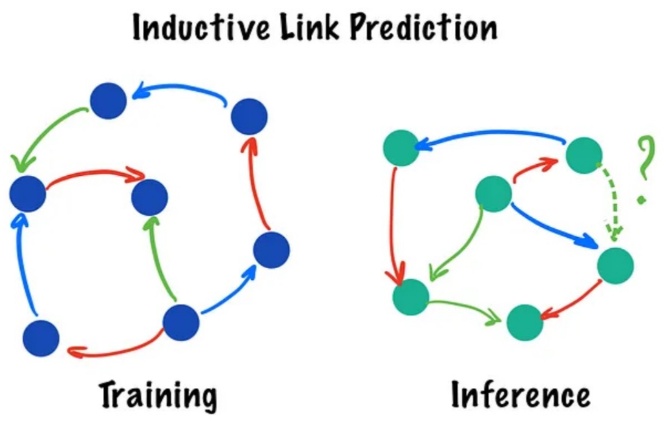


# Research Objective



- ❖ Goal
  - ❑ Identify the subset of neighbors that are connected due to factor k
- ❖ Conditions
  - ❑ Differentiable to support end-to-end training
  - ❑ Conduct inductive learning for OOD generalization
  - ❑ Propose neighborhood routing (Not only one-hop distance)
  - ❑ Theoretically analyze the convergence properties

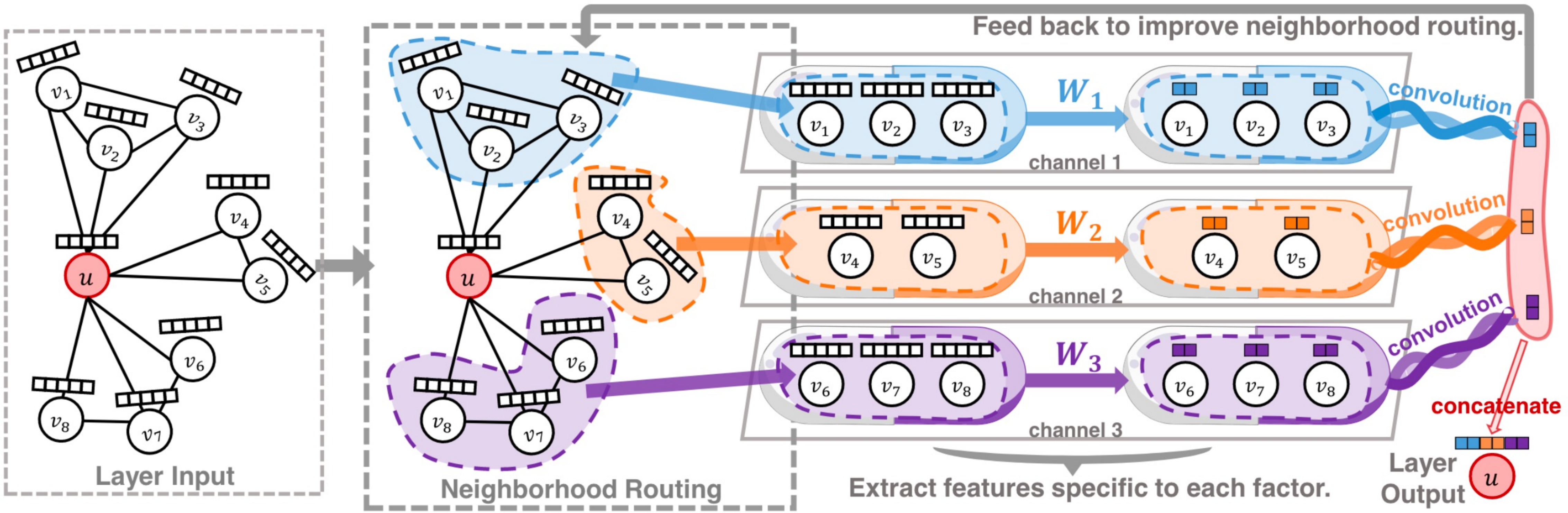


# 2. Proposed Method

- Overview
- The DisenConv Layer
- Network Architecture
- Theoretical Analysis

# Overview






- ❑ Layer input
- ❑ Neighborhood Routing
- ❑ Extract features specific
- ❑ Layer output

**Make convolution process more special!**

# The DisenConv Layer



$$\mathbf{y}_u = f\left(\mathbf{x}_u, \{\mathbf{x}_v : (u, v) \in G\}\right).$$

- GCN(Graph Convolutional Network) Layer
  - $y_u$ is the representation of node u, learned by the layer
  - The neighborhood of a node provides a rich information for the $y_u$
- Goal of the DisenGCN layer
  - Aim to derive a layer f(•) such that the output $y_u$ is a disentangled representation into K independent components

$$\mathbf{y}_u = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K], \text{ where } \mathbf{c}_k \in \mathbb{R}^{\frac{d_{out}}{K}} \ (1 \leq k \leq K),$$

# The DisenConv Layer



❖ How to identify disentangled subsets?

❑ Assume that the K channels can extract different features

$$\mathbf{z}_{i,k} = \frac{\sigma(\mathbf{W}_k^\top \mathbf{x}_i + \mathbf{b}_k)}{\left\|\sigma(\mathbf{W}_k^\top \mathbf{x}_i + \mathbf{b}_k)\right\|_2},$$

➔ Separately learn parameters in terms of k

❑ Is it sufficient to represent the latent factors?

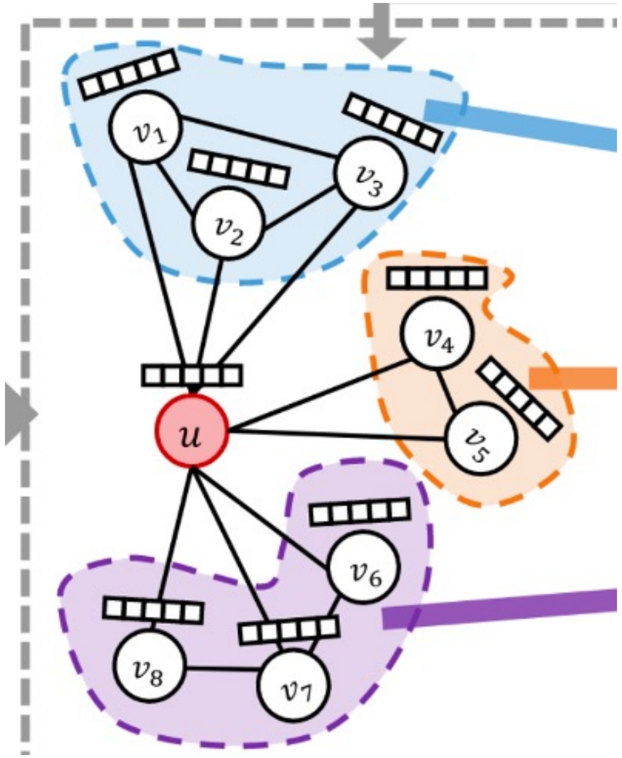
➔ No! We need to require neighborhood information

➔ Therefore, we are going to construct $\mathbf{c}_K$

# The DisenConv Layer



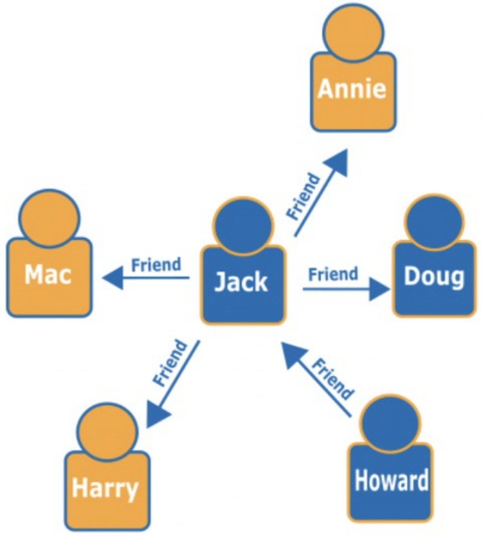
- ❖ Hypothesis1 : Second-order proximity
  - ❑ If the subset of its neighbors is large and they are similar w.r.t. aspect k
    → Factor k is likely to the reason why node u connects with them.
  - ❑ E.g., If most people in **u**'s neighborhood cluster like gaming, they're likely connected to **u** through gaming.
  - ❑ Need to search for the largest cluster in each of the K subspaces projected from the original space



# The DisenConv Layer

- Hypothesis2 : First-order proximity
  - If u and v are similar in terms of factor k, then the factor k is likely to be the reason why they are connected
  - E.g., If u and v like gaming, they're likely connected through gaming
  - However, How can we know u and v have similar in terms of aspect k?

    ➔ Apply hypothesis 1, then use hypothesis 2 like an assistance



# The DisenConv Layer



- Method
  - In the neighborhood routing mechanism, Combine $z_{u,k}$ and $z_{v,k}$ by using $p_{v,k}$ that is the prob that factor k is the reason why u and v are connected.

$$p_{v,k}^{(t)} = \frac{\exp(\mathbf{z}_{v,k}{}^{\top}\mathbf{c}_k^{(t)}/\tau)}{\sum_{k'=1}^{K}\exp(\mathbf{z}_{v,k'}{}^{\top}\mathbf{c}_{k'}^{(t)}/\tau)},$$

$$\mathbf{c}_k^{(t)} = \frac{\mathbf{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k}^{(t-1)}\mathbf{z}_{v,k}}{\|\mathbf{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k}^{(t-1)}\mathbf{z}_{v,k}\|_2},$$

- $p_{v,k}$ reflect hypo2
- τ means controls the hardness of the assignments (due to index of softmax)
- $c_k$ reflect hypo1

# The DisenConv Layer



- Method
  - Searches for the largest cluster in each subspace
  - Can view $c_k$ as the center of each subspace cluster k
  - All process are differentiable operations

$$\mathbf{c}_k^{(t)} = \frac{\mathbf{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k}^{(t-1)} \mathbf{z}_{v,k}}{\|\mathbf{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k}^{(t-1)} \mathbf{z}_{v,k}\|_2},$$

# The DisenConv Layer



**Algorithm 1** The proposed DisenConv layer, with $K$ channels. It performs $T$ iterations of routing. Typically $T \approx 5$.

**Input:** $\mathbf{x}_u \in \mathbb{R}^{d_{in}}$ (the feature vector of node $u$), and $\{\mathbf{x}_v \in \mathbb{R}^{d_{in}} : (u,v) \in G\}$ (its neighbors' features).

**Output:** $\mathbf{y}_u \in \mathbb{R}^{d_{out}}$ (the representation of node $u$).

**Param:** $\mathbf{W}_k \in \mathbb{R}^{d_{in} \times \frac{d_{out}}{K}}$, $\mathbf{b}_k \in \mathbb{R}^{\frac{d_{out}}{K}}$, $k = 1, \ldots, K$.

**for** $i \in \{u\} \cup \{v : (u,v) \in G\}$ **do**

  **for** $k = 1, 2, \ldots, K$ **do**

    $\mathbf{z}_{i,k} \leftarrow \sigma(\mathbf{W}_k^\top \mathbf{x}_i + \mathbf{b}_k)$.

    $\mathbf{z}_{i,k} \leftarrow \mathbf{z}_{i,k} / \|\mathbf{z}_{i,k}\|_2$. // The $k^{\text{th}}$ aspect of node $i$.

  **end for**

**end for**

$\mathbf{c}_k \leftarrow \mathbf{z}_{u,k}, \forall k = 1, 2, \ldots, K$. // Initialize $K$ channels.

**for** routing iteration $t = 1, 2, \ldots, T$ **do**

  **for** $v$ that satisfies $(u,v) \in G$ **do**

    $p_{v,k} \leftarrow \mathbf{z}_{v,k}^\top \mathbf{c}_k / \tau, \forall k = 1, 2, \ldots, K$.

    $[p_{v,1} \ldots p_{v,K}] \leftarrow \mathrm{softmax}([p_{v,1} \ldots p_{v,K}])$.

  **end for**

  **for** channel $k = 1, 2, \ldots, K$ **do**

    $\mathbf{c}_k \leftarrow \mathbf{z}_{u,k} + \sum_{v:(u,v)\in G} p_{v,k}\, \mathbf{z}_{v,k}$. // Update.

    $\mathbf{c}_k \leftarrow \mathbf{c}_k / \|\mathbf{c}_k\|_2$.

  **end for**

**end for**

$\mathbf{y}_u \leftarrow$ the concatenation of $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_K$.

# Network Architecture



❖ The number of layers

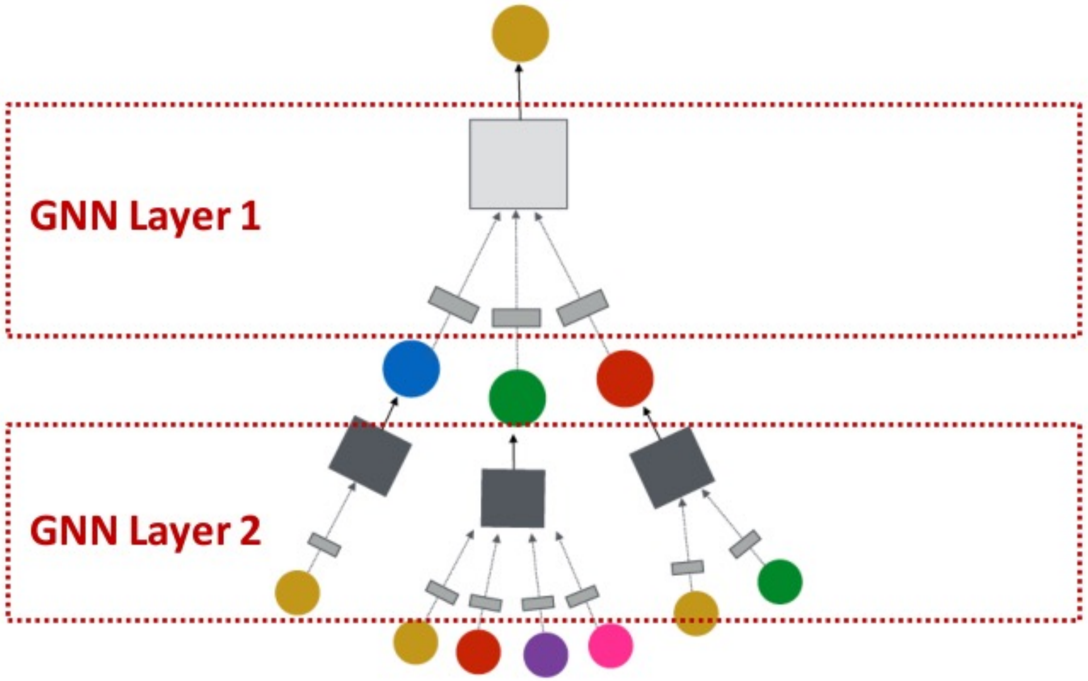
❑ May be desirable to stack multiple DisenConv layers

1. Allows us to access data beyond one-hop nodes
2. Can potentially learn hierarchical representations, by gradually decreasing the number of channels

➔ $K^{(1)} \geq K^{(2)} \geq \ldots \geq K^{(L)}$

→ Is it real general benefit?

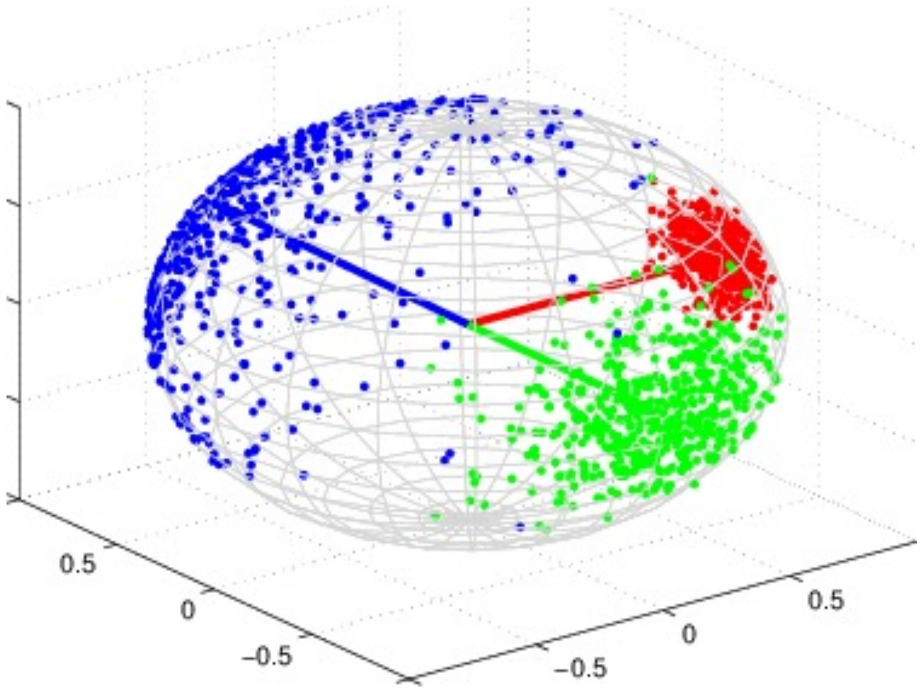
❖ The final layer is FC



# Theoretical Analysis



- ❖ Need to answer about neighborhood routing mechanism
  - ❑ Whether it converges after enough iterations
  - ❑ To what solution it converges if it does




A von Mises-Fisher(vMF) mixture model

- ❖ vMF distribution
  - ❑ Probability distribution about direction
  - ❑ Used for analyzing embedding vectors
- ❖ vMF mixture model
  - ❑ Mixture of several vMF distribution
  - ❑ Each vMF distribution represents a latent feactures

# Theoretical Analysis



- How to prove?
  - Whether it converges after enough iterations
    - Convert the mechanism into an expectation-maximization (EM) algorithm for the mixture model that is known as convergence in specific condition
    - Show the mechanism satisfy that condition

      → Compact parameter set & Continuous likelihood function
  - To what solution it converges if it does
    - Each distribution represents K latent factors under three conditions

      →

$$
\begin{aligned}
&\mathbf{z}_{u,k} \sim \mathrm{vMF}(\mathbf{c}_k, 1),\\
&r_v \sim \mathrm{Categorical}\left([1/K, 1/K, \ldots, 1/K]\right),\\
&\mathbf{z}_{v,r_v} \mid r_v \sim \mathrm{vMF}(\mathbf{c}_{r_v}, 1/\tau),\\
&\mathbf{z}_{v,k'} \mid r_v \sim \mathrm{vMF}(\boldsymbol{\mu}, 0), \quad k' \neq r_v \ \wedge \ 1 \leq k' \leq K,
\end{aligned}
$$

# 3. Experimental Results

- Setup
- Experiment

# Setup



❖ Baselines

- GCN and GAT(state-of-the-art)
- Additionally add DeepWalk, LINE, and node2vec for multi-label tasks

❖ Datasets

| Dataset | Type | Nodes | Edges | Classes | Features | Multi-label |
|---|---|---|---|---|---|---|
| Citeseer | Citation network | 3,327 | 4,732 | 6 | 3,703 | No |
| Cora | Citation network | 2,708 | 5,429 | 7 | 1,433 | No |
| Pubmed | Citation network | 19,717 | 44,338 | 3 | 500 | No |
| Blogcatalog | Social network | 10,312 | 333,983 | 39 | - | Yes |
| PPI | Biological network | 3,890 | 76,584 | 50 | - | Yes |
| POS | Word co-occurrence | 4,777 | 184,812 | 40 | - | Yes |

- The latter three do not have node-features

  → use only their adjacency matrices

❖ Hyper parameters

- Set T = 7 and τ = 1. The others use 'hyperopt'

# Experiment



- Semi-Supervised Node Classification
  - Each dataset contains only 20 labeled instance for each class
  - The optimal number of layers for DisenGCN is 5, while GCN and GAT is 2
    ➔ Robust for over-smoothing by taking a disentangled approach

*Table 2.* Semi-supervised classification accuracies (%).

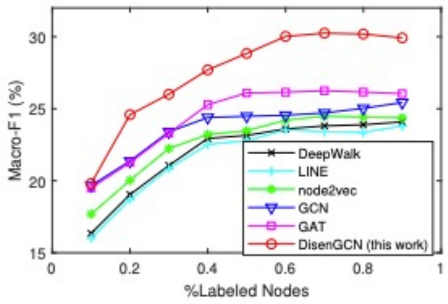
| Method | Datasets | | |
|---|---|---|---|
| | **Cora** | **Citeseer** | **Pubmed** |
| MLP | 55.1 | 46.5 | 71.4 |
| ManiReg (Belkin et al., 2006) | 59.5 | 60.1 | 70.7 |
| SemiEmb (Weston et al., 2012) | 59.0 | 59.6 | 71.1 |
| LP (Zhu et al., 2003) | 68.0 | 45.3 | 63.0 |
| DeepWalk (Perozzi et al., 2014) | 67.2 | 43.2 | 65.3 |
| ICA (Lu & Getoor, 2003) | 75.1 | 69.1 | 73.9 |
| Planetoid (Yang et al., 2016) | 75.7 | 64.7 | 77.2 |
| ChebNet (Defferrard et al., 2016) | 81.2 | 69.8 | 74.4 |
| GCN (Kipf & Welling, 2017) | 81.5 | 70.3 | 79.0 |
| MoNet (Monti et al., 2017) | 81.7 | - | 78.8 |
| GAT (Veličković et al., 2018) | 83.0 | 72.5 | 79.0 |
| DisenGCN (this work) | **83.7** | **73.4** | **80.5** |

# Experiment

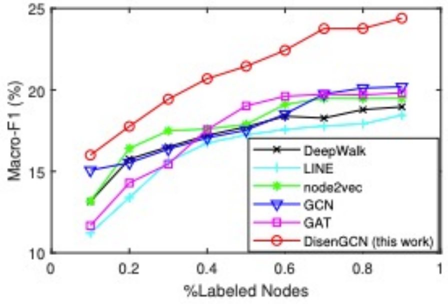


❖ Multilabel Node Classification

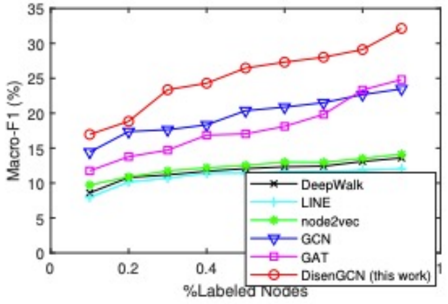
- ❑ Vary the number of nodes labeled for training from 10%|V| to 90%|V|
- ❑ GCN : Relatively high Macro-F1 but low Micro-F1 → Class imbalance
- ❑ GAT : Performance is much lower in train-set → Over-fitting



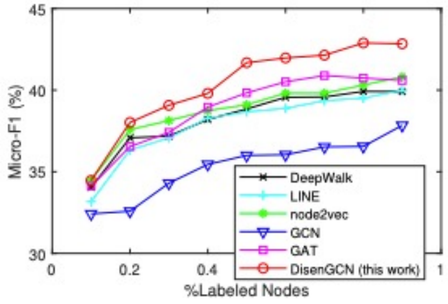

(a) Macro-F1(%), BlogCatalog.



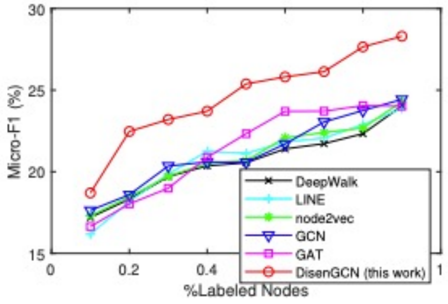

(c) Macro-F1(%), PPI.



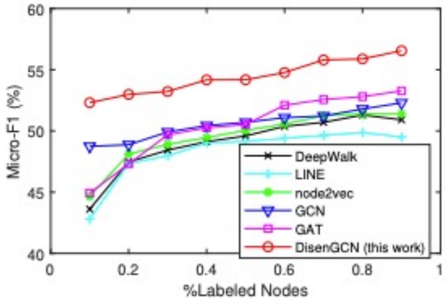

(e) Macro-F1(%), POS.




(b) Micro-F1(%), BlogCatalog.




(d) Micro-F1(%), PPI.




(f) Micro-F1(%), POS.

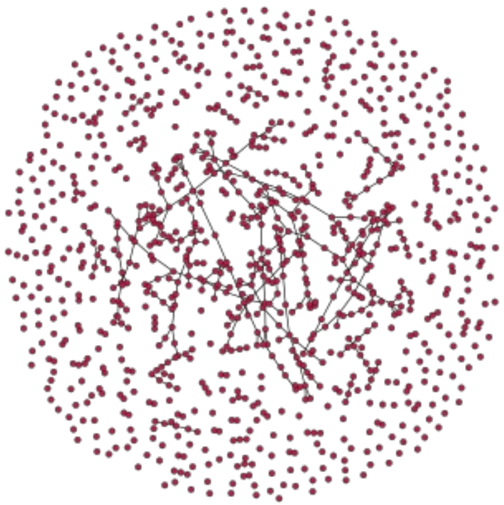
# Experiment



- Disentangling Synthetic Graphs
  - Generate K Erdos-Renyi random graphs
  - Each has 1,000 nodes and 16 communities (Randomly allocate)
  - Sum the adjacency matrices (I guess max value is 1)

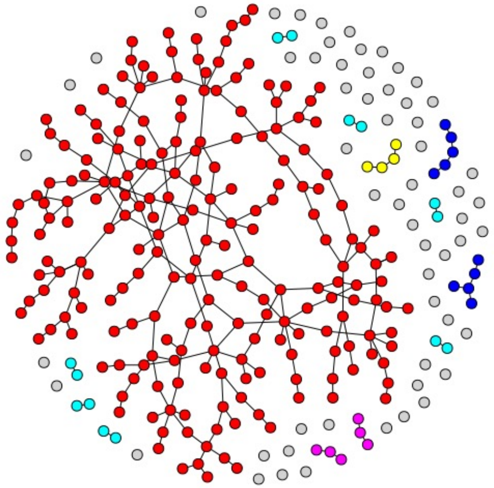
    ➔ Generate the graph with K latent factors

    E.g, $K_1$ graph means hobby and $K_2$ graphs means local information



← In this $K_1$ graph, the edges are connected due to $K_1$

In this $K_2$ graph, the edges → are connected due to $K_2$



# Experiment

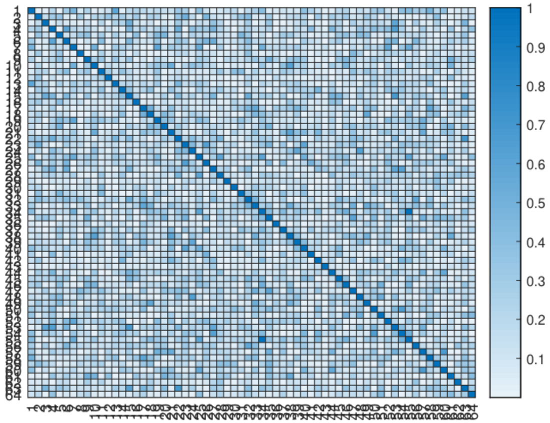


❖ Disentangling Synthetic Graphs

*Table 3.* Micro-F1 scores on synthetic graphs generated with different numbers of latent factors.

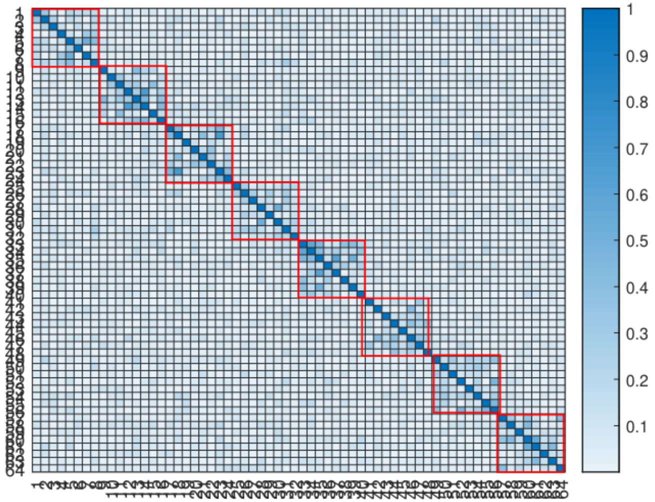
| Method | Number of latent factors 4 | 6 | 8 | 10 | 12 | 14 | 16 |
|---|---|---|---|---|---|---|---|
| GCN | 78.78 ± 1.52 | 65.73 ± 1.94 | 46.55 ± 1.55 | 37.37 ± 1.52 | 24.49 ± 1.03 | 18.14 ± 1.50 | 16.43 ± 0.92 |
| GAT | 83.77 ± 2.32 | 60.89 ± 3.75 | 45.88 ± 3.79 | 36.72 ± 3.58 | 24.77 ± 3.47 | 20.89 ± 3.57 | 19.53 ± 3.97 |
| DisenGCN (this work) | **93.84** ± 1.12 | **74.68** ± 1.92 | **54.57** ± 1.79 | **43.96** ± 1.45 | **28.17** ± 1.22 | **23.57** ± 1.28 | **21.99** ± 1.34 |
| Relative improvement | +12.02% | +13.62% | +17.23% | +17.63% | +13.73% | +12.83% | +12.6% |

❑ When K is very large i.e., K > 12, the performance starts to fall



(a) GCN.

❑ Visualize the absolute values of the correlations between the 64-dimensional node representation



(b) DisenGCN (this work).

❑ 8 Channels leads eight clear diagonal blocks

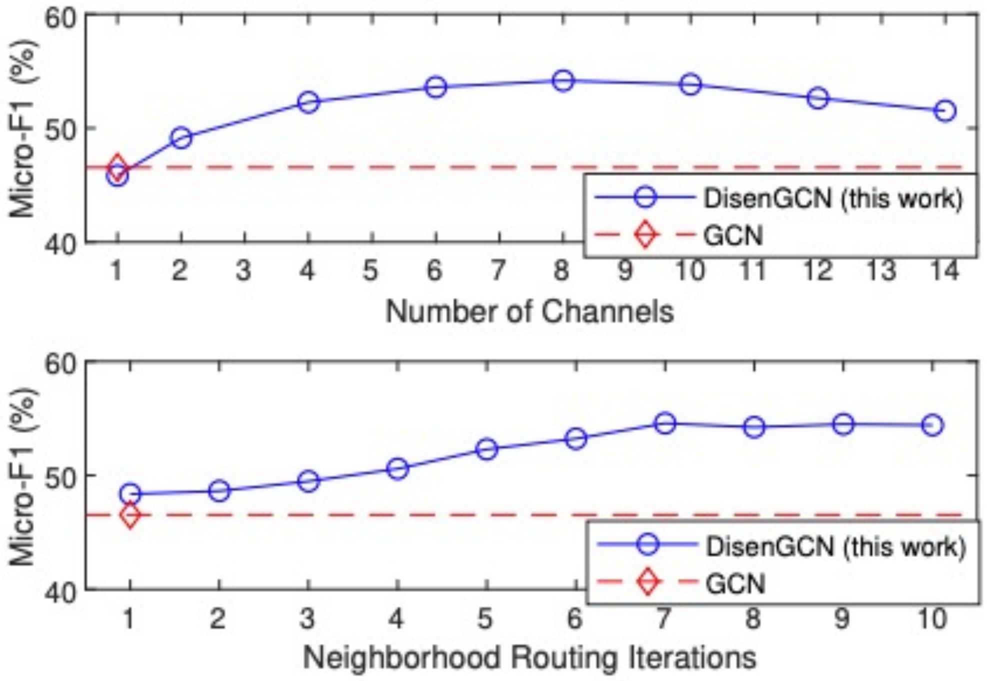
➔ DisenGCN are likely Capture mutually exclusive information

# Experiment



❖ Hyperparameter Sensitivity

- ❑ Crucial hyperparameters
  - The number of channels, C
  - The number of routing iterations, T
- ❑ The best when…
  - The number of channels is around the actual number of latent factors
  - More iterations generally leads to better performance before saturation T ≈ 5



# 4. Conclusion

- Contribution & Limitation
- Further Directions

# Contribution & Limitation



- Contributions
  - First try to disentangle the latent factors in the graph structure
- Limitations
  - ER random graph method are quite not good
  - One edge is only can represented as one relationship (But in the real world?)
    → I though this idea at first time, but I think it is a subtle. However…
  - How can we know k latent factors are really fundamental ones?
    (Are they just ad-hoc ones?)

# Further Directions

- What if we had humans manually label the latent factors identified by the model, and then used that information for further analysis?
- Are there really no more sophisticated methods?