Recent Advances on Graph Analytics and Its Applications in Healthcare

KDD 2020 Tutorial

August 23, morning

Fei Wang, **Peng Cui**, Jian Pei, Yangqiu Song, Chengxi Zang,

http://www.calvinzang.com/kdd2020_tutorial_medical_graph_analytics.html

KDD’20 Tutorial on Recent Advances on Graph Analytics and Its Applications in Healthcare

Network Embedding and Graph Neural Networks

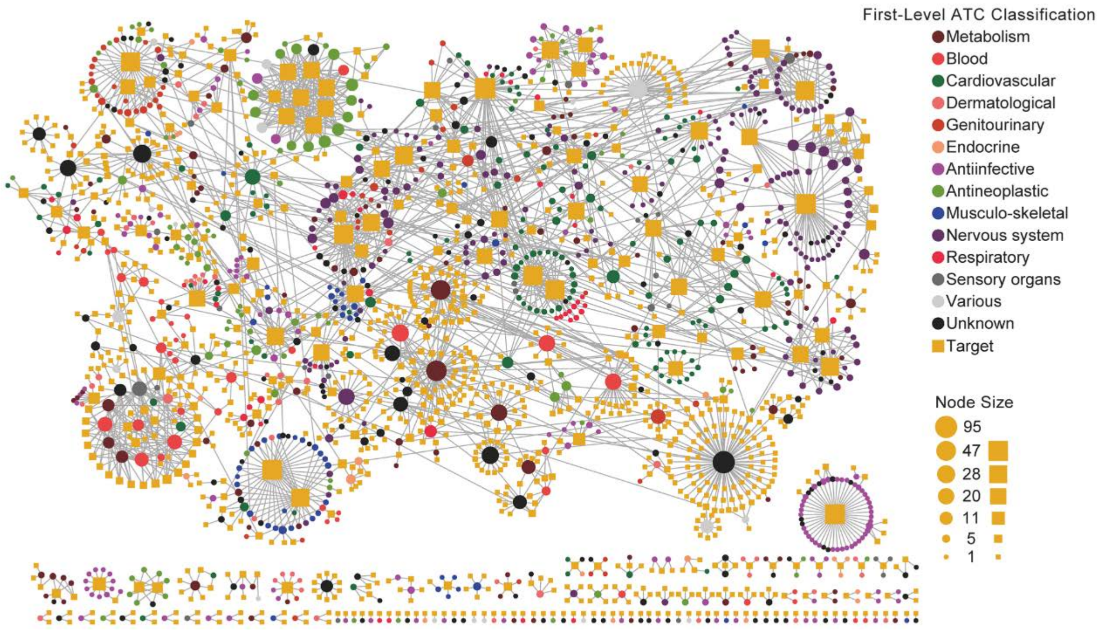
Peng Cui

Tsinghua University

Healthcare and Graph

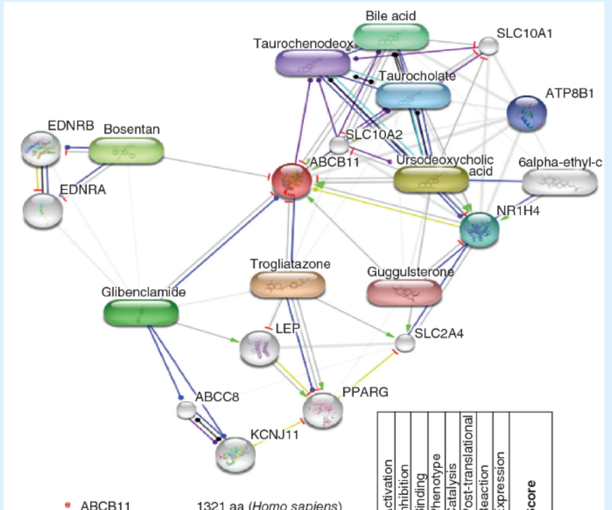
Many healthcare problems can be modeled as graph problems.

Drug retargeting



http://www.cytoscape.org/

Adverse drug reaction



ABCB11 1321 aa (*Homo sapiens*)

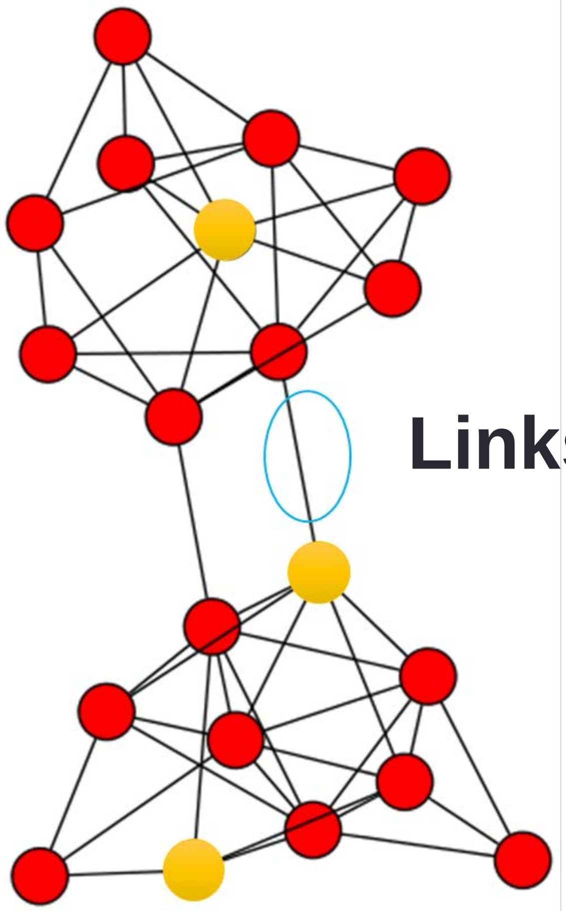
Predicted functional partners:

		Activation	Inhibition	Binding	Phenotype	Catalysis	Post-translational	Reaction	Expression	Score
Troglitazone	441.5 g/mol		•	•						0.994
Ursodeoxycholi	392.6 g/mol	•	•							0.990
Bosentan	551.6 g/mol		•							0.986
Glibenclamide	494.0 g/mol		•	•						0.982
Bile acid	408.6 g/mol	•	•							0.977
NR1H4	472 aa	•							•	0.974
Taurocholate	537.7 g/mol	•				•				0.973
ATP8B1	1251 aa									0.971
Tautochenodeox	499.7 g/mol					•				0.967
Guggulsterone	312.4 g/mol	•								0.961

https://www.future-science.com/doi/10.4155/fmc.13.202

Networks are not *learning-friendly*

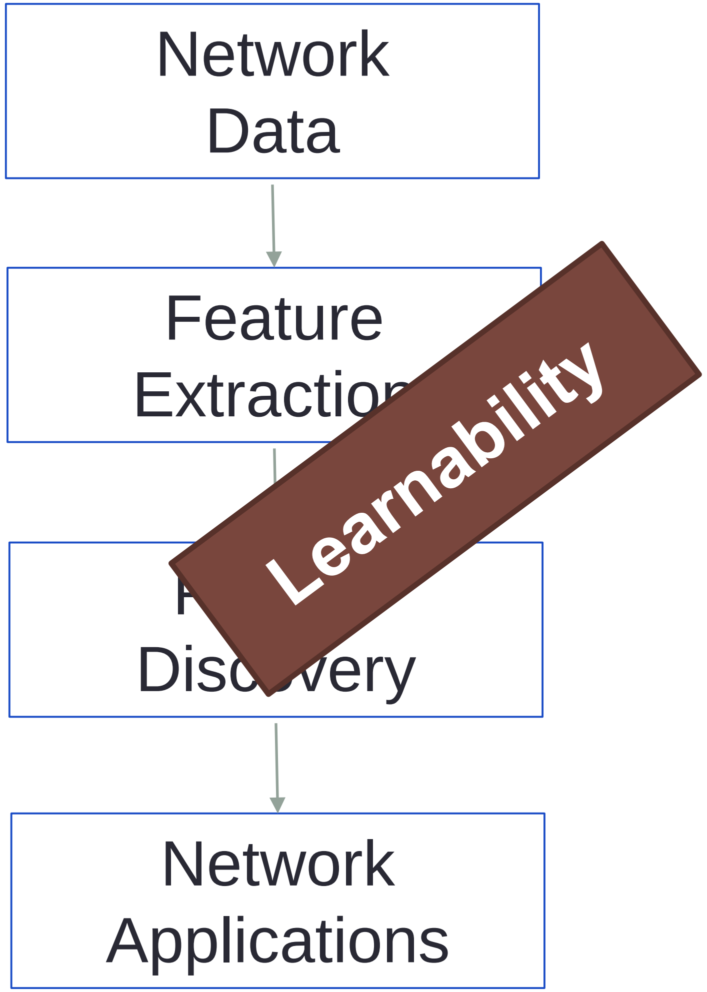
G = (V, E)



Inapplicability of ML methods

Links ⇨ Topology

Pipeline for network analysis



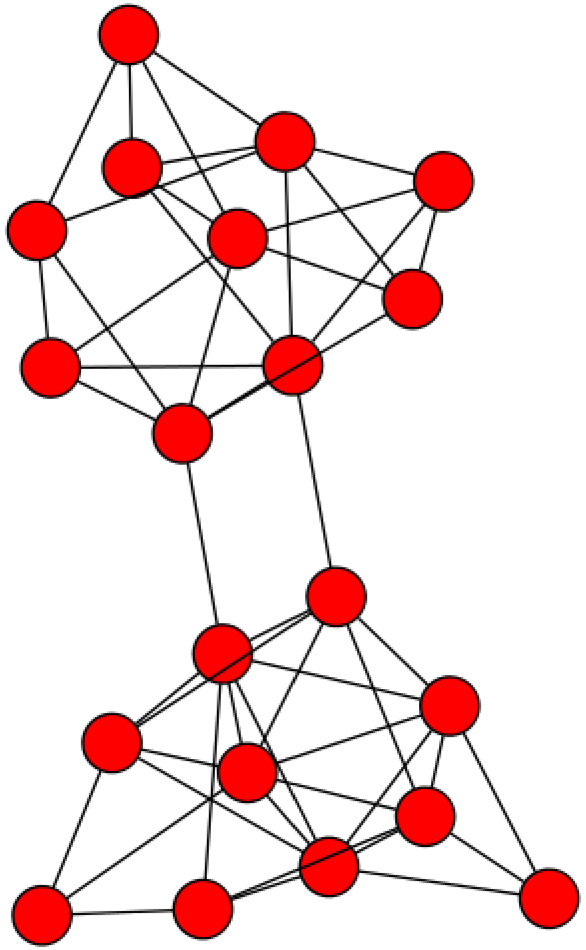
Learning from networks

Network Embedding

GCN

Network Embedding

G = (V, E)

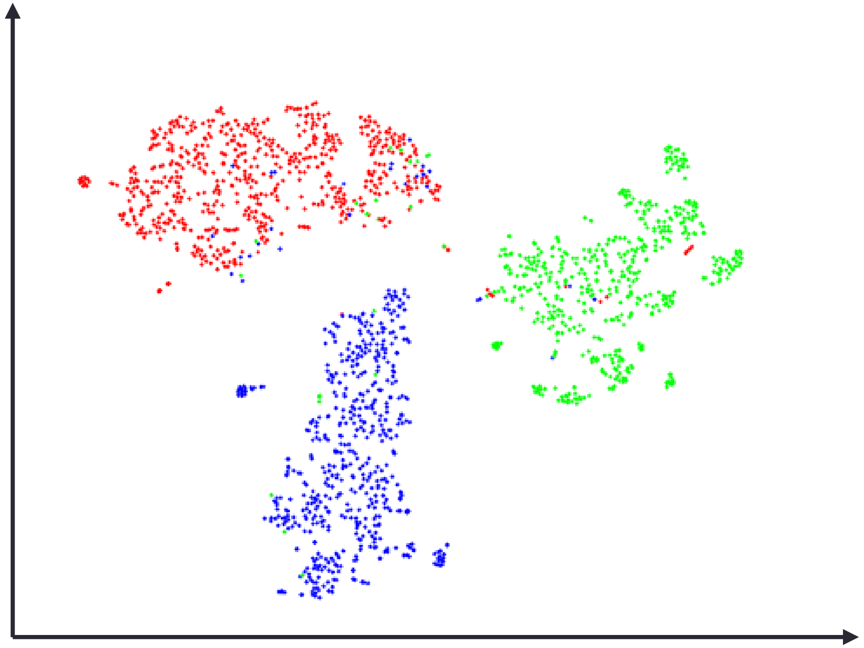


G = (V)

Vector Space

generate

embed



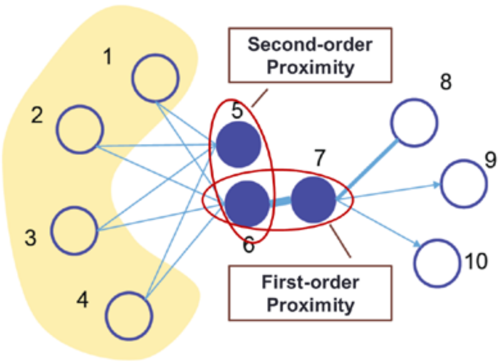
- Easy to parallel
- Can apply classical ML methods

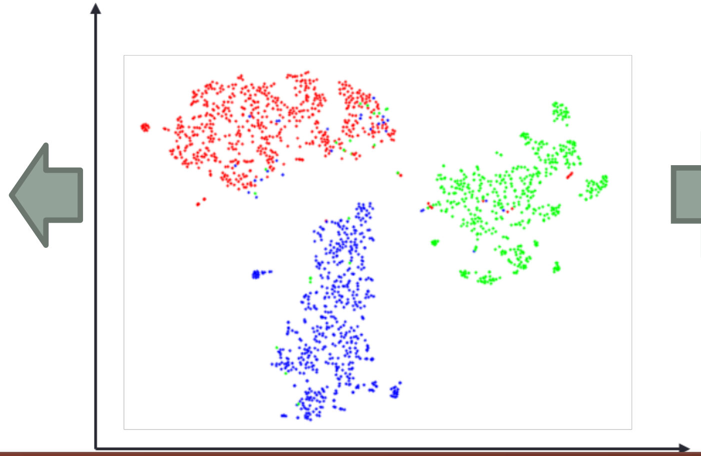
The goal of network embedding

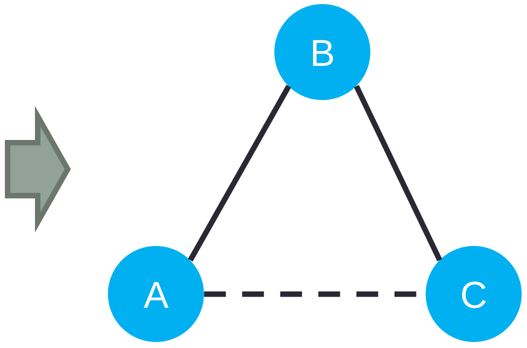
Goal Support network inference in vector space

Reflect network structure

Maintain network properties







Transform network nodes into vectors that are fit for off-the-shelf machine learning models.

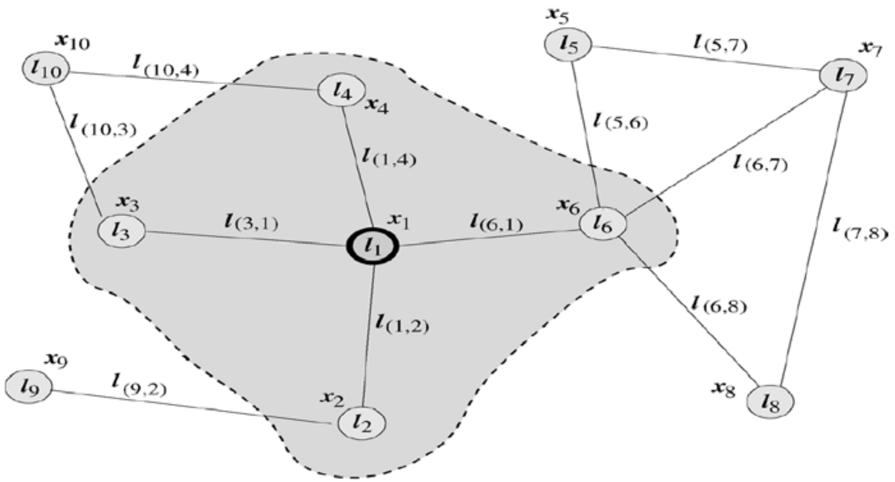
Graph Neural Networks

Design a learning mechanism on graph.

- Basic idea: recursive definition of states

$$\mathbf{s}_i = \sum_{j \in \mathcal{N}(i)} \mathcal{F}\left(\mathbf{s}_i, \mathbf{s}_j, \mathbf{F}_i^V, \mathbf{F}_j^V, \mathbf{F}_{i,j}^E\right)$$

- A simple example: PageRank



$$x_1 = f_w(l_1, \underbrace{l_{(1,2)}, l_{(3,1)}, l_{(1,4)}, l_{(6,1)}}_{l_{co[1]}}, \underbrace{x_2, x_3, x_4, x_6}_{x_{ne[1]}}, \underbrace{l_2, l_3, l_4, l_6}_{l_{ne[n]}})$$

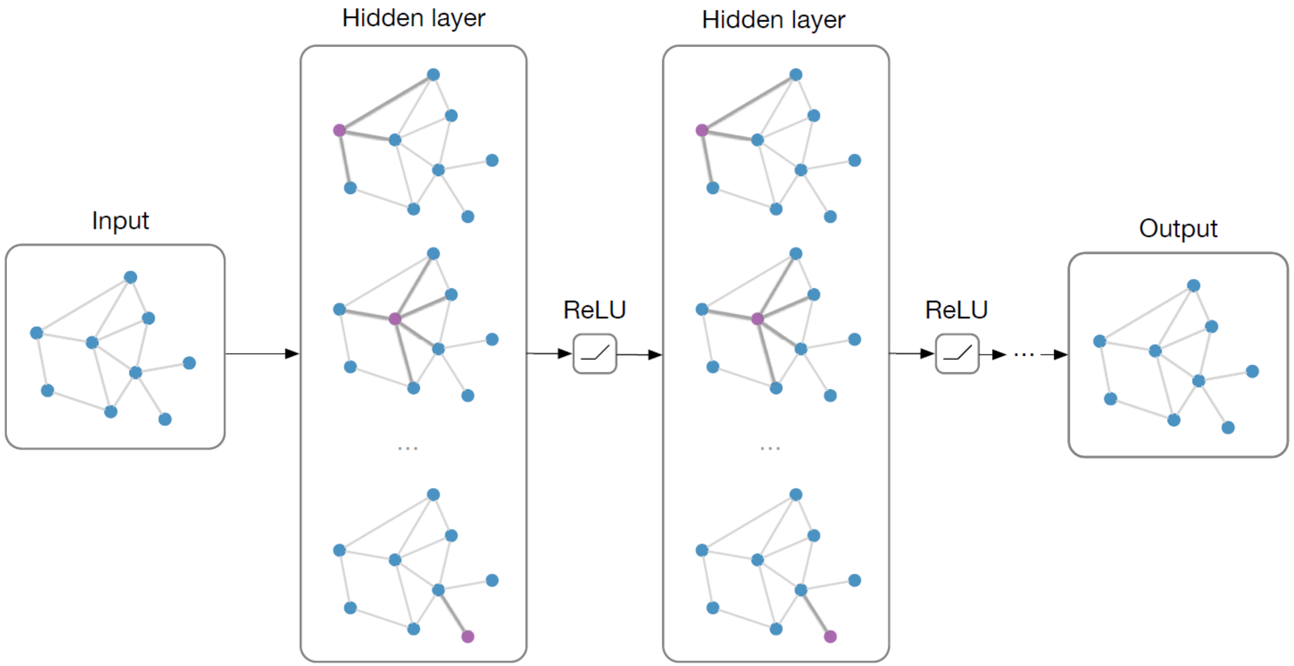
F. Scarselli, et al. The graph neural network model. *IEEE TNN, 2009.*

Graph Convolutional Networks (GCN)

- Main idea: pass messages between pairs of nodes & agglomerate

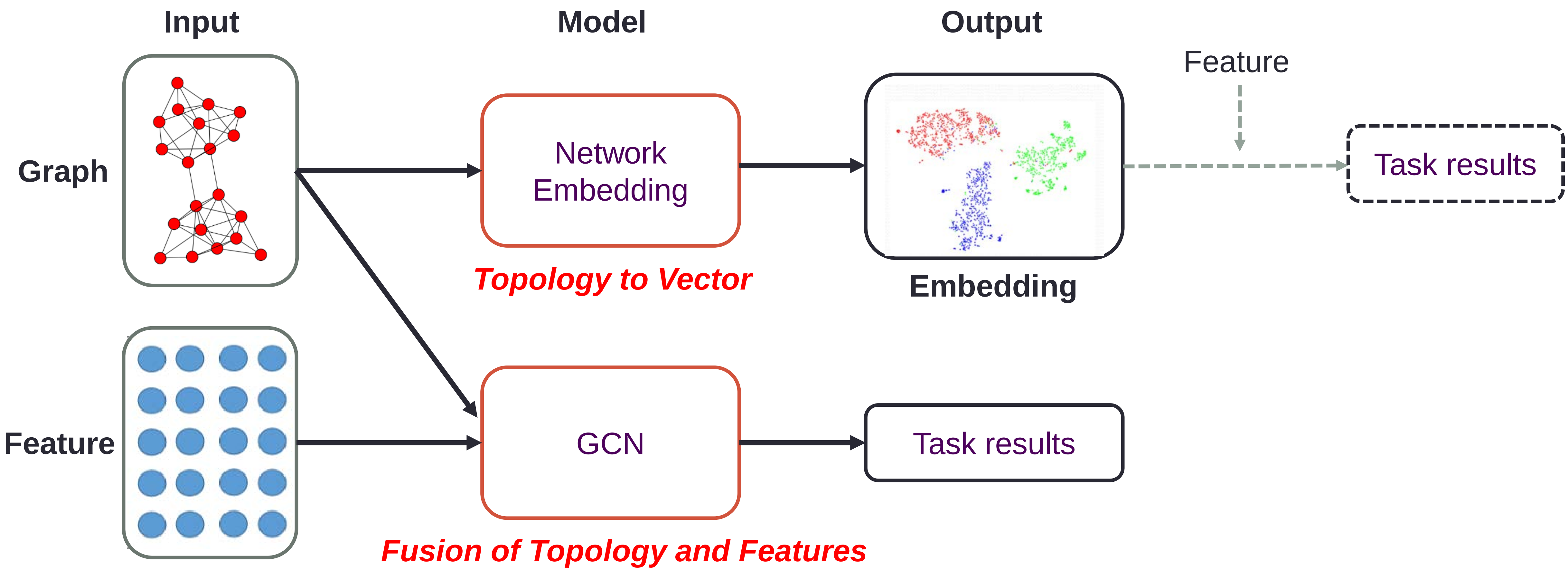
$$\mathbf{H}^{l+1} = \rho\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{l}\mathbf{\Theta}^{l}\right)$$

- Stacking multiple layers like standard CNNs:
  - State-of-the-art results on node classification



T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *ICLR, 2017.*

Network Embedding and GCN



Unsupervised v.s. (Semi-)Supervised

Learning from networks

Network Embedding

GCN

The intrinsic problems NE is solving

Reducing representation dimensionality while preserving necessary topological **structures** and **properties**.

Nodes & Links

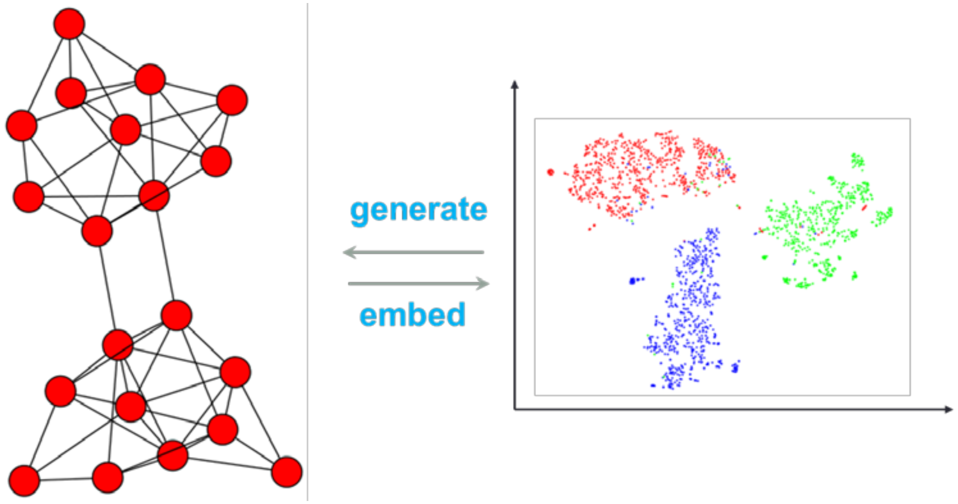
Node Neighborhood

Pair-wise Proximity

Community

Hyper Edges

Global Structure



Non-transitivity

Asymmetric Transitivity

Uncertainty

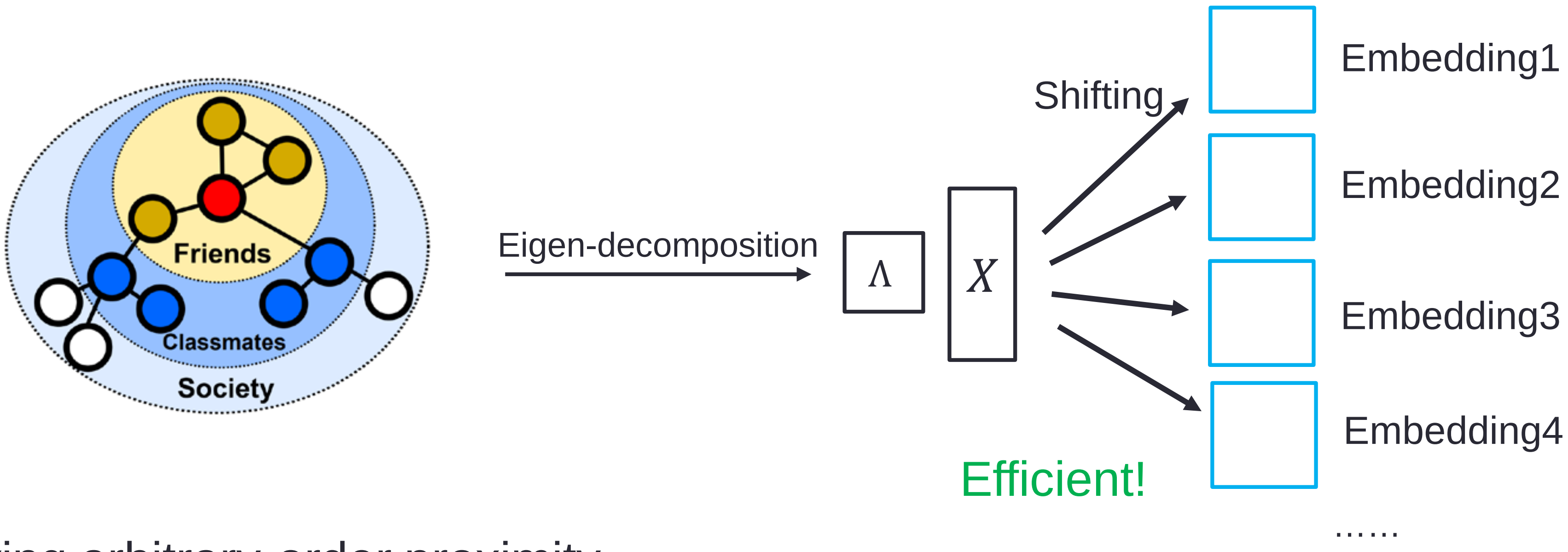
Dynamic

Heterogeneity

Interpretability

Preserving Arbitrary-Order Proximity

- Shifting across different orders/weights:



- Preserving arbitrary-order proximity
- Low marginal cost
- Accurate and efficient

Z. Zhang, et al. Arbitrary-Order Proximity Preserved Network Embedding. *KDD, 2018.*

Preserving Arbitrary-Order Proximity

- High-order proximity: a polynomial function of the adjacency matrix

$$S = f(A) = w_1 A^1 + w_2 A^2 + \cdots + w_q A^q$$

  - q: order; $w_1 \ldots w_q$: weights, assuming to be non-negative
  - A: could be replaced by other variations (such as the Laplacian matrix)
- Objective function: matrix factorization

$$\min_{U^*, V^*} \left\| S - U^* {V^*}^T \right\|_F^2$$

  - $U^*, V^* \in \mathbb{R}^{N \times d}$: left/right embedding vectors
  - d: dimensionality of the space
- Optimal solution: Singular Value Decomposition (SVD)
  - $[U, \Sigma, V]$: top-d SVD results

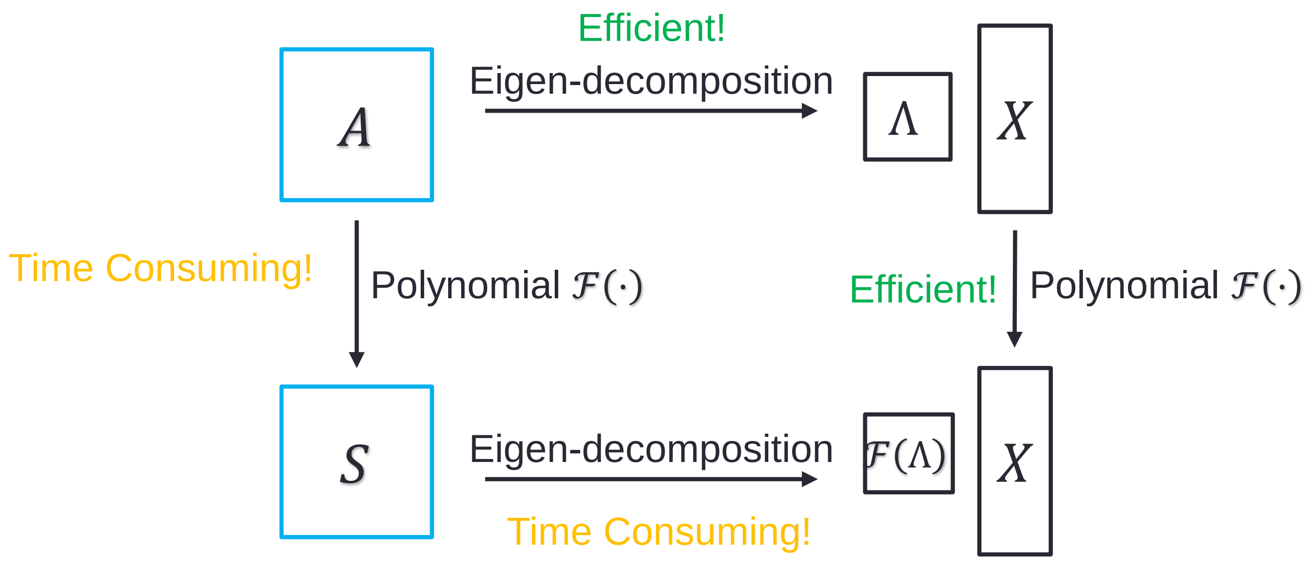
$$U^* = U\sqrt{\Sigma}, \qquad V^* = V\sqrt{\Sigma}$$

Z. Zhang, et al. Arbitrary-Order Proximity Preserved Network Embedding. *KDD, 2018.*

Preserving Arbitrary-Order Proximity

- Eigen-decomposition reweighting

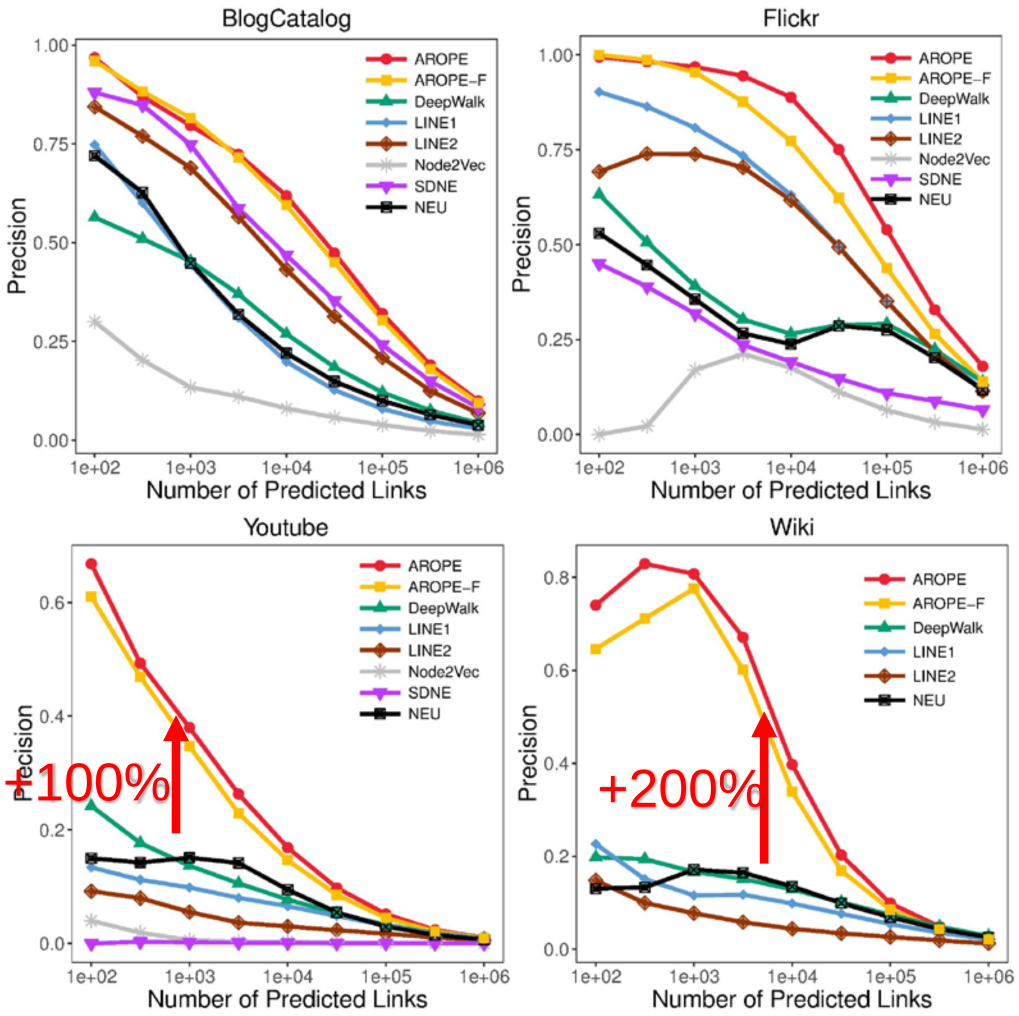
THEOREM 4.2 (EIGEN-DECOMPOSITION REWEIGHTING). *If* $[\lambda, \mathbf{x}]$ *is an eigen-pair of* $\mathbf{A}$, *then* $[\mathcal{F}(\lambda), \mathbf{x}]$ *is an eigen-pair of* $\mathbf{S} = \mathcal{F}(\mathbf{A})$.



Z. Zhang, et al. Arbitrary-Order Proximity Preserved Network Embedding. *KDD, 2018.*

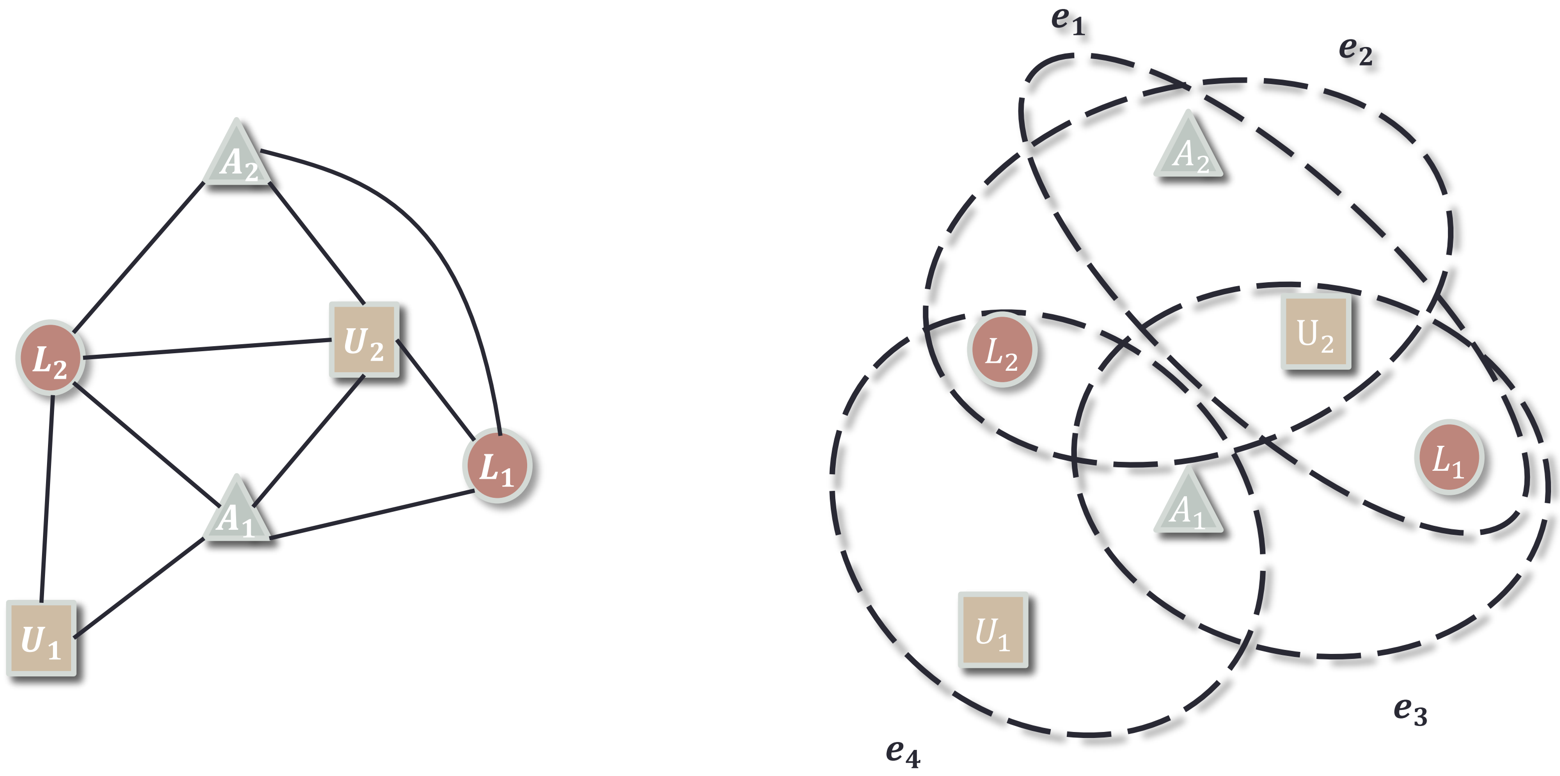
Experimental Results

- Link Prediction



Z. Zhang, et al. Arbitrary-Order Proximity Preserved Network Embedding. *KDD, 2018.*

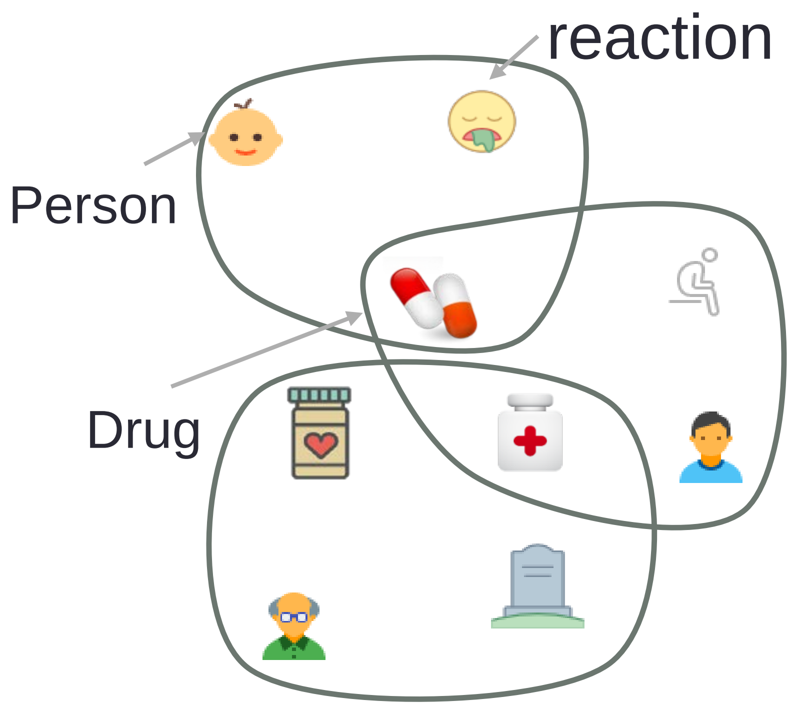
Hyper-network embedding



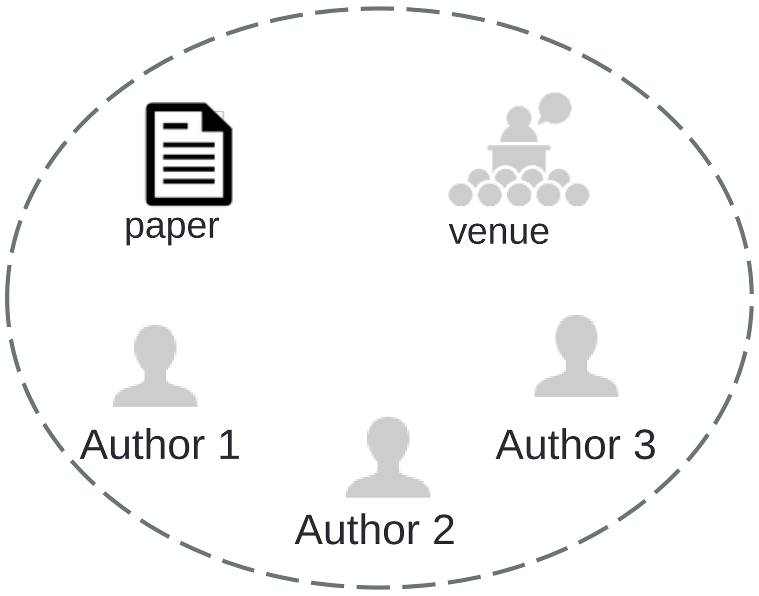
Networks Hyper-Networks

- A hyper-network is a network in which an edge can include any number of nodes

Hyper-edges are often indecomposable



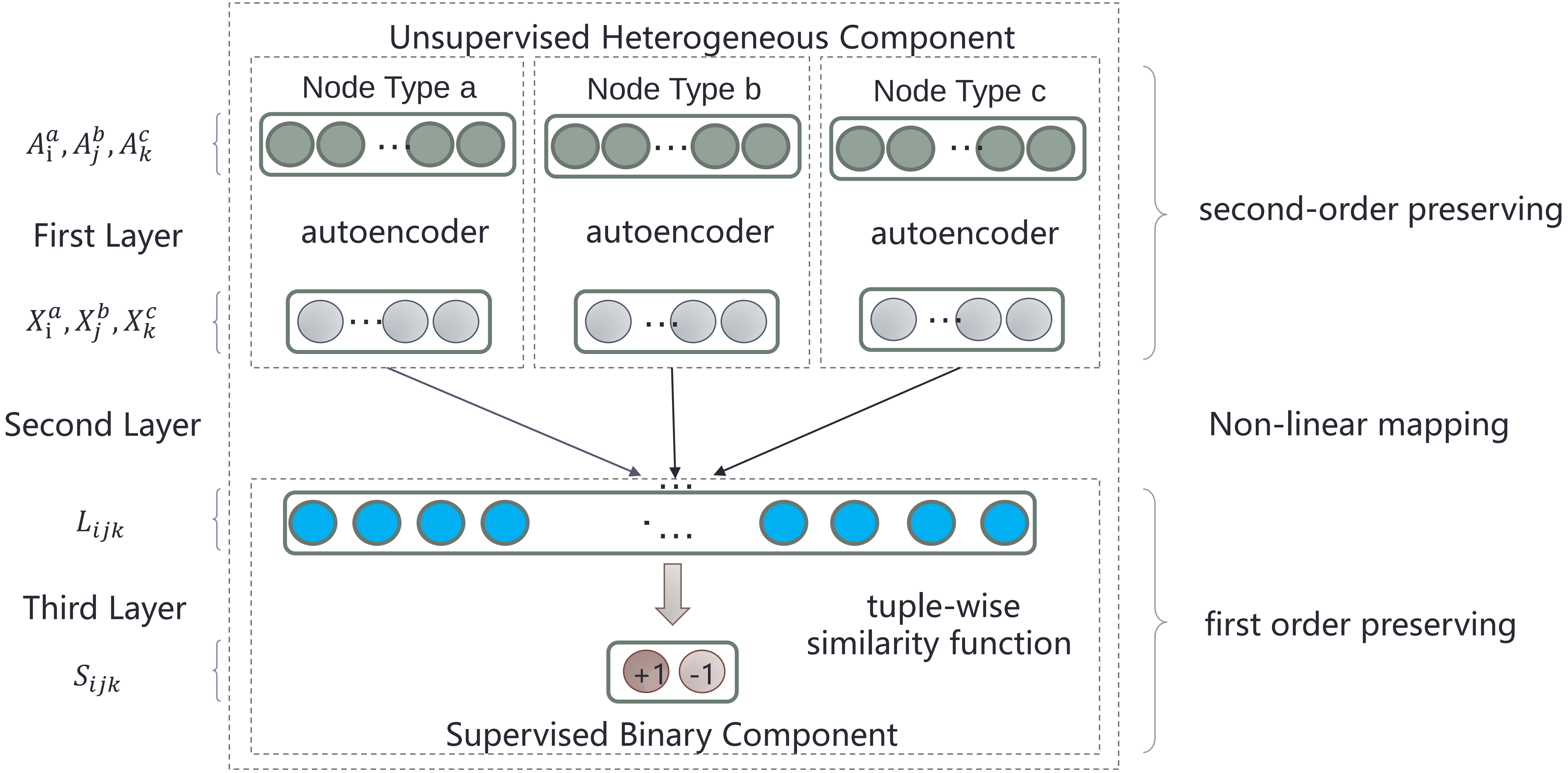
Adverse Drug Network



Bibliographic Network

Ke Tu, et al. Structural Deep Embedding for Hyper-Networks. ***AAAI***, 2018.

Structural Deep Network for Hyper-network

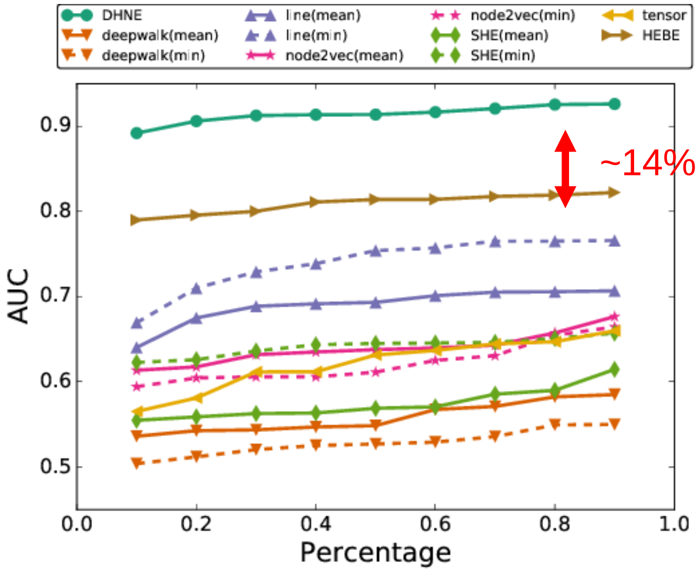


Ke Tu, et al. Structural Deep Embedding for Hyper-Networks. ***AAAI***, 2018.

Experiment: link prediction

Table 4: AUC value for link prediction

methods		GPS	MovieLens	drug	wordnet
DHNE		**0.9166**	**0.8676**	**0.9254**	**0.8268**
mean	deepwalk	0.6593	0.7151	0.5822	0.5952
	line	0.7795	0.7170	0.7057	0.6819
	node2vec	0.5835	0.8211	0.6573	0.8003
	SHE	0.8687	0.7459	0.5899	0.5426
min	deepwalk	0.5715	0.6307	0.5493	0.5542
	line	0.7219	0.6265	0.7651	0.6225
	node2vec	0.5869	0.7675	0.6546	0.7985
	SHE	0.8078	0.8012	0.6508	0.5507
tensor		0.8646	0.7201	0.6470	0.6516
HEBE		0.8355	0.7740	0.8191	0.6364



The overall performance

Performance on networks of different sparsity

Ke Tu, et al. Structural Deep Embedding for Hyper-Networks. ***AAAI***, 2018.

Learning from networks

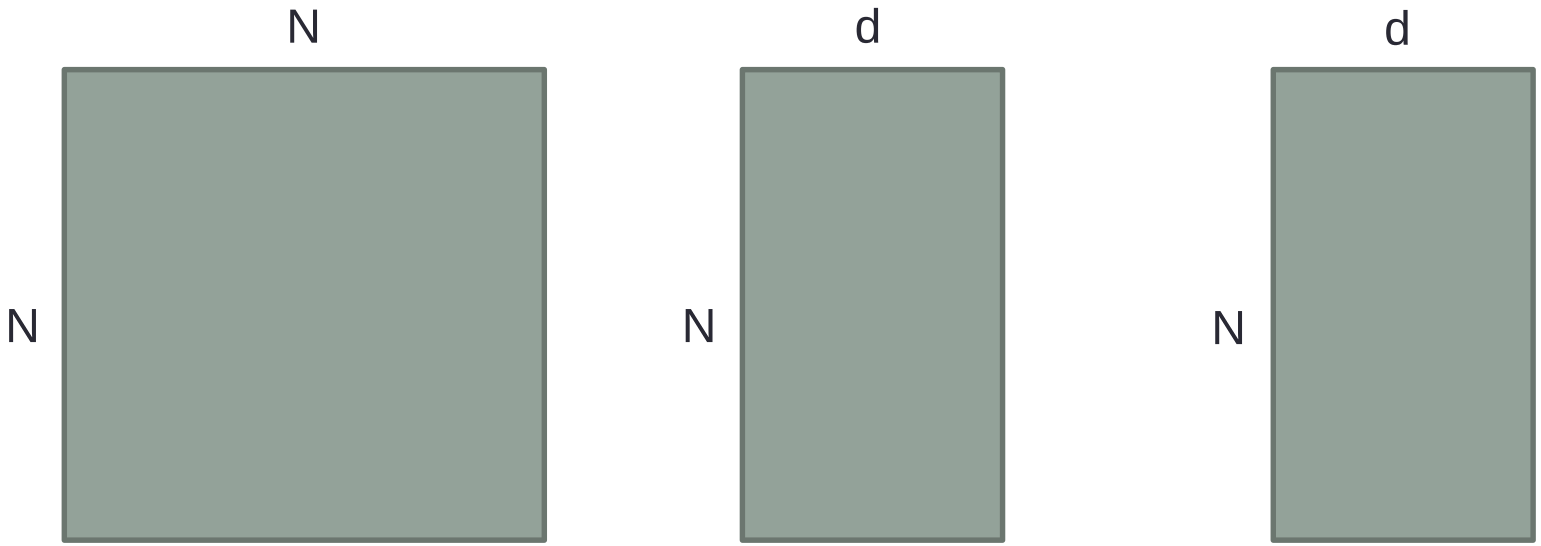
Network Embedding

GCN

The intrinsic problem GCN is solving

Fusing topology and features in the way of smoothing features with the assistance of topology.

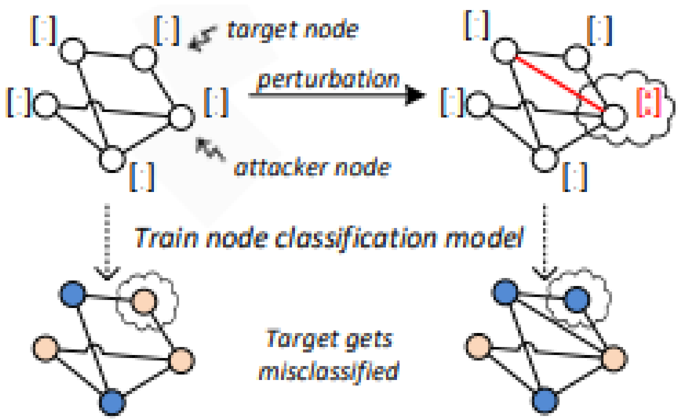
$$\mathbf{H}^{l+1} = \rho\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{l}\mathbf{\Theta}^{l}\right)$$

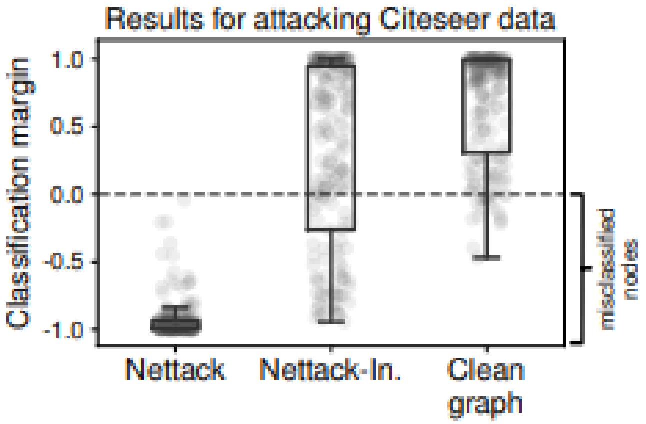


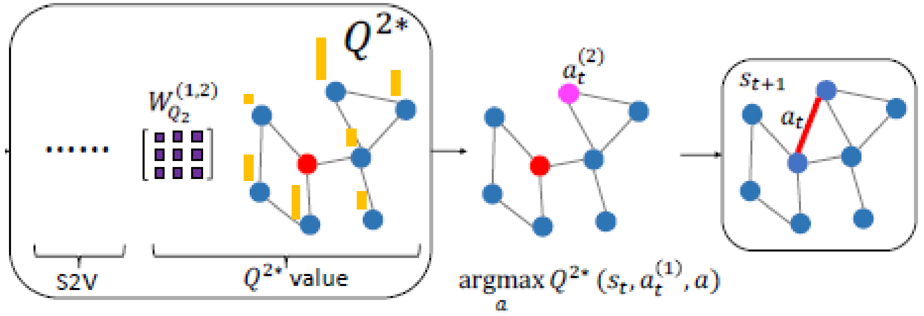
$$\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}} \quad \text{X} \quad \mathbf{H}^{l} \quad = \quad \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{l}$$

Robust GCN

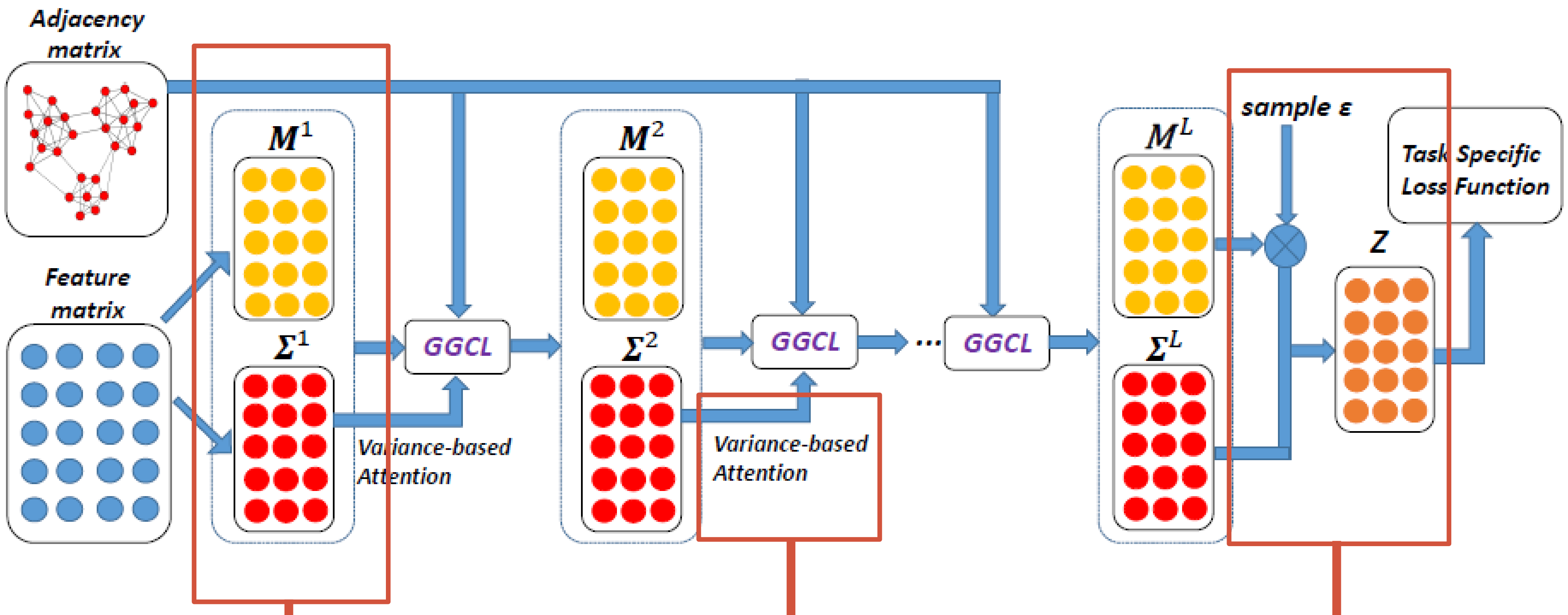
- **Adversarial attacks**
  - small perturbations in graph structures and node attributes
  - great challenges for applying GCNs to node classification







Robust GCN



Gaussian Based hidden representations:
Variance terms absorb the effects of adversarial attacks

Attention mechanism:
Remedy the propagation of adversarial attacks

Sampling process:
Explicitly considers mathematical relevance between means and variances

Dingyuan Zhu, Ziwei Zhang, Peng Cui, Wenwu Zhu. Robust Graph Convolutional Networks Against Adversarial Attacks. ***KDD***, 2019.

Robust GCN

Node Classification on Clean Datasets

	Cora	Citeseer	Pubmed
GCN	81.5	70.9	79.0
GAT	83.0	**72.5**	79.0
RGCN	**83.1**	71.3	**79.2**

Against Non-targeted Adversarial Attacks

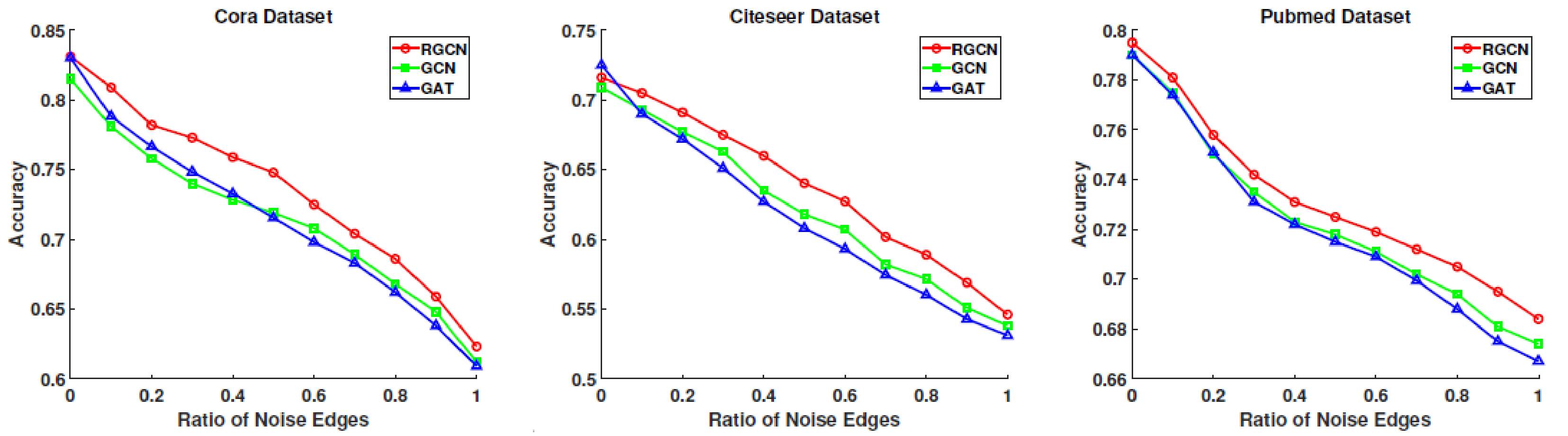
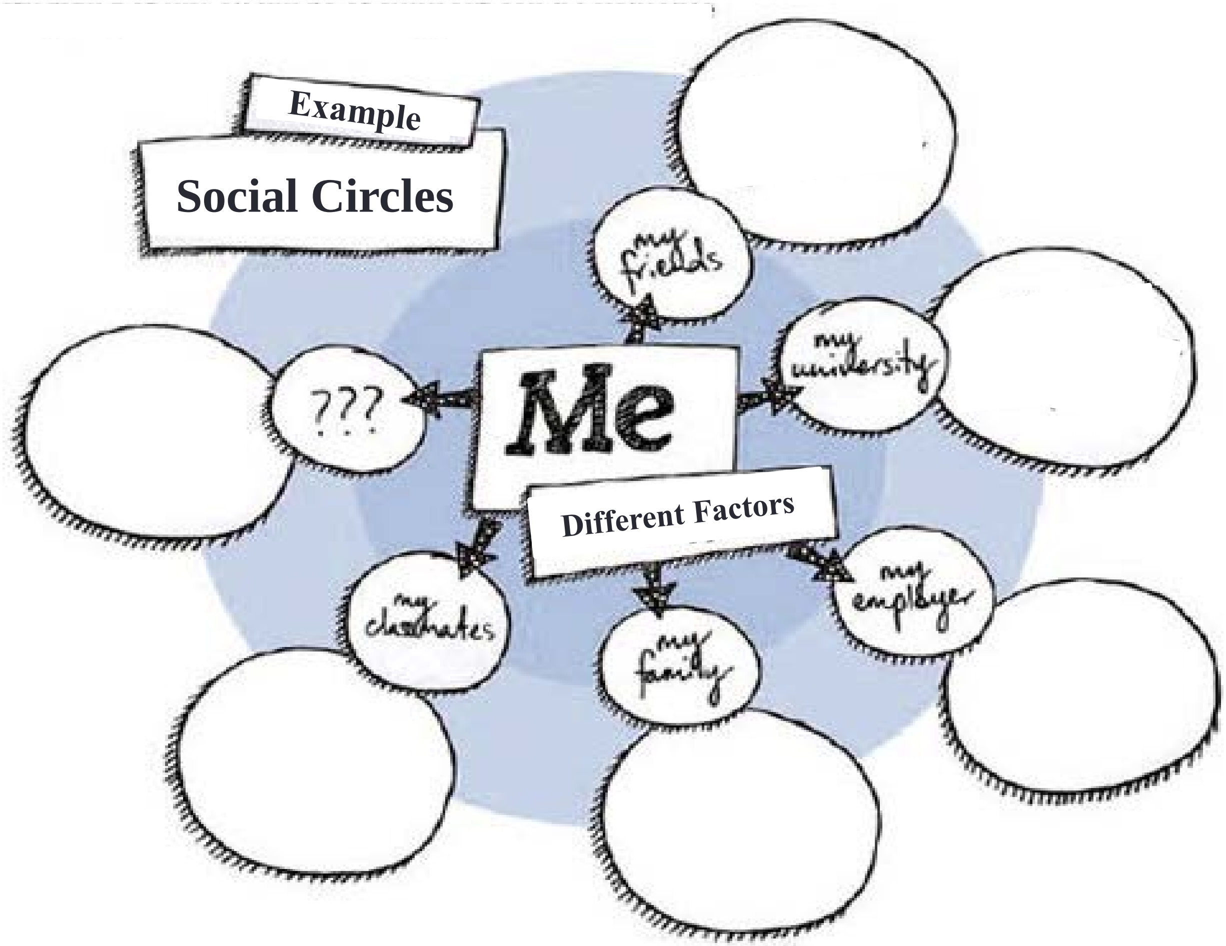


Figure 2: Results of different methods when adopting Random Attack as the attack method.

Dingyuan Zhu, Ziwei Zhang, Peng Cui, Wenwu Zhu. Robust Graph Convolutional Networks Against Adversarial Attacks. ***KDD***, 2019.

Disentangled GCN

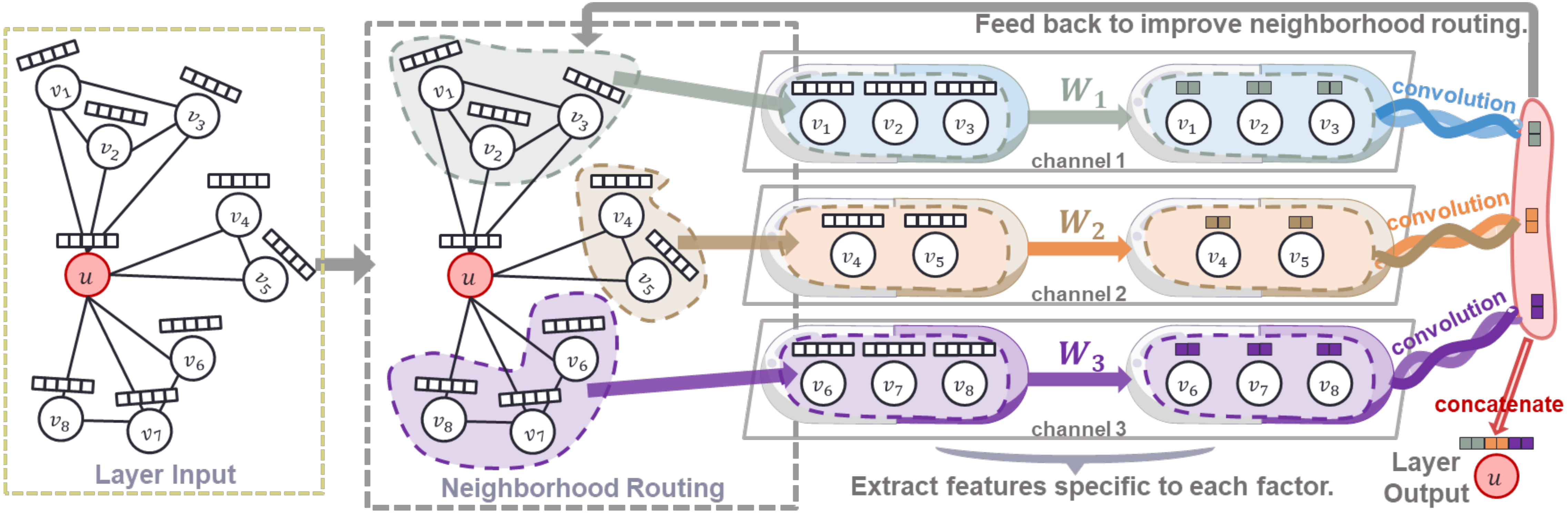
- **A real-world graph is typically formed due to *many* latent factors.**



- Existing GNNs/GCNs:
  - A holistic approach, that takes in the *whole* neighborhood to produce a *single* node representation.
- We suggest:
  - To disentangle the latent factors. (By segmenting the heterogeneous parts, and learning multiple factor-specific representations for a node.)
  - Robustness (e.g., not overreact to an irrelevant factor) & Interpretability.

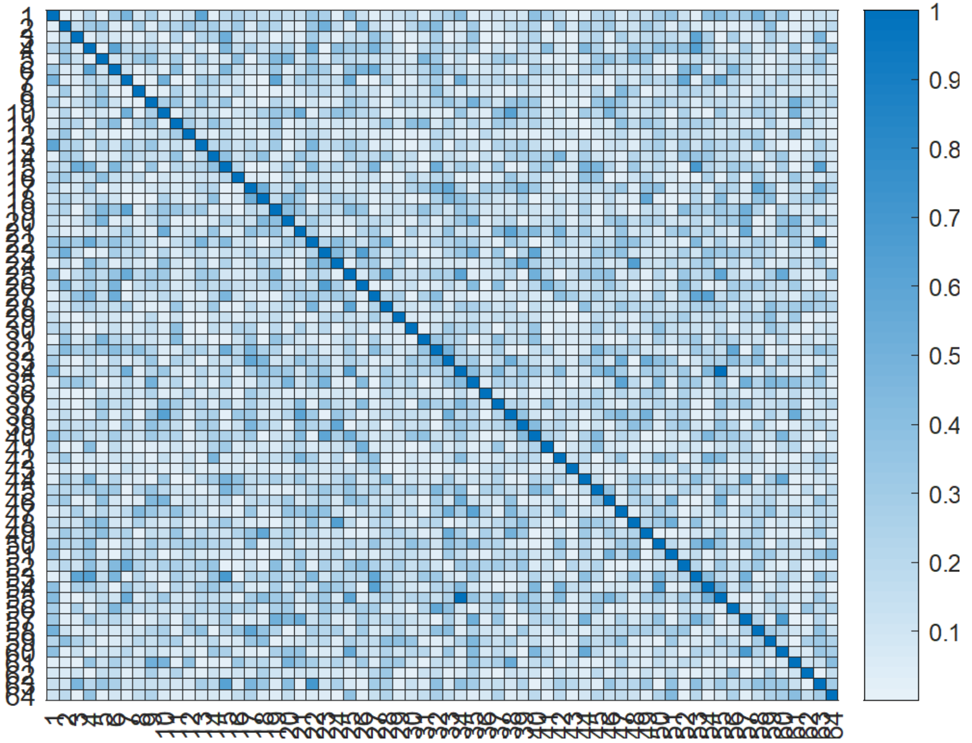
Disentangled GCN

- We present DisenGCN, the *disentangled* graph convolutional network.
  - DisenConv, a disentangled multichannel convolutional layer (figure below).
  - Each channel convolutes features related with a single latent factor.

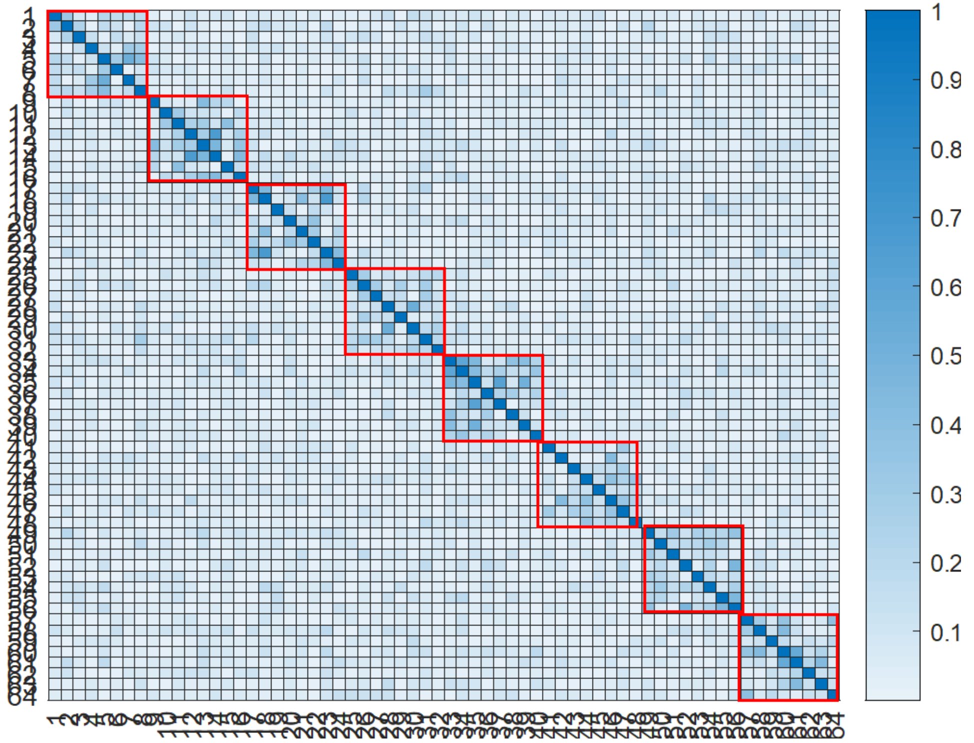


Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, Wenwu Zhu. Disentangled Graph Convolutional Networks. ***ICML***, 2019.

Disentangled GCN



(a) GCN.



(b) DisenGCN (this work).

Jianxin Ma, Peng Cui, Kun Kuang, Xin Wang, Wenwu Zhu. Disentangled Graph Convolutional Networks. ***ICML***, 2019.

Some interesting questions for GCN…

What if the problem is topology-driven?

- Since GCN is filtering features, it is inevitably feature-driven
  - Structure only provides auxiliary information (e.g. for filtering/smoothing)
- When feature plays the key role, GNN performs good …
- How about the contrary?
- Synthesis data: stochastic block model + random features

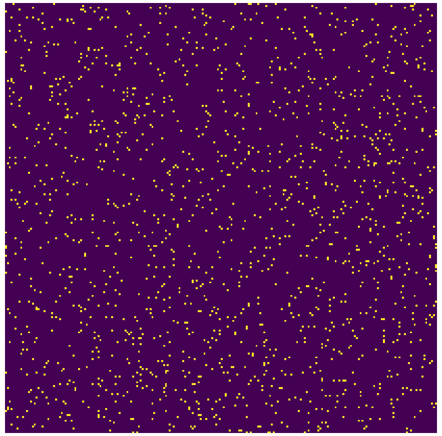
Method	Results
Random	10.0
GCN	18.3±1.1
DeepWalk	99.0±0.1

Does GCN fuse feature and topology optimally?

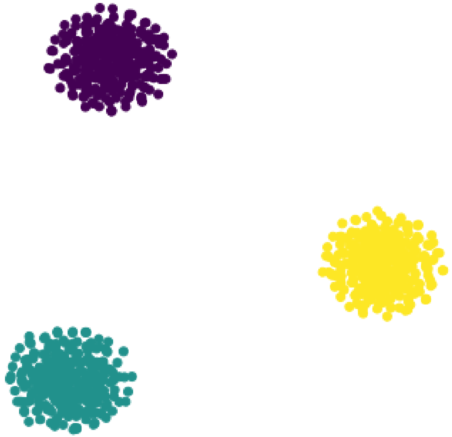
Fusion Capability of GCNs

- Ideal Solution: extract the most correlated information for task

Case 1



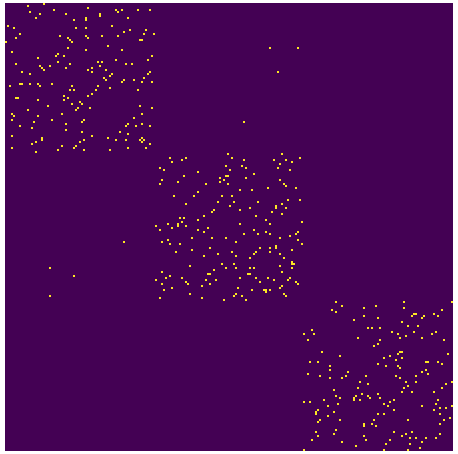
Random topology



Correlated Features

MLP(100%) > GCN(75.2%)

Case 2



Correlated Topology

Random Features

DeepWalk(100%) > GCN(87%)

Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, Jian Pei. AM-GCN: Adaptive Multi-channel Graph Convolutional Networks. ACM SIGKDD, 2020.

Rethinking: Is GCN truly a Deep Learning method?

- Recall GNN formulation:

$$H^{(k+1)} = \sigma\left(SH^{(k)}W^{(k)}\right), S = \widetilde{D}^{-1/2}\tilde{A}\widetilde{D}^{-1/2}$$

- How about removing the non-linear component:

$$H^{(k+1)} = SH^{(k)}W^{(k)}$$

  - Stacking multiple layers and add softmax classification:

$$\begin{aligned}\hat{Y} &= softmax\left(H^{(K)}\right)\\ &= softmax\left(SS\ldots SH^{(0)}W^{(0)}W^{(1)}\ldots W^{(K-1)}\right)\\ &= softmax\left(\boxed{S^K}H^{(0)}W\right)\end{aligned}$$

High-order proximity

Wu, Felix, et al. Simplifying graph convolutional networks. *ICML, 2019.*

Rethinking: Is GCN truly a Deep Learning method?

- This simplified GNN (SGC) shows remarkable results:

Node classification

	Cora	Citeseer	Pubmed
GCN	81.4 ± 0.4	70.9 ± 0.5	79.0 ± 0.4
GAT	83.3 ± 0.7	72.6 ± 0.6	78.5 ± 0.3
FastGCN	79.8 ± 0.3	68.8 ± 0.6	77.4 ± 0.3
GIN	77.6 ± 1.1	66.1 ± 0.9	77.0 ± 1.2
LNet	$80.2 \pm 3.0^{\dagger}$	67.3 ± 0.5	$78.3 \pm 0.6^{\dagger}$
AdaLNet	$81.9 \pm 1.9^{\dagger}$	$70.6 \pm 0.8^{\dagger}$	$77.8 \pm 0.7^{\dagger}$
DGI	82.5 ± 0.7	71.6 ± 0.7	78.4 ± 0.7
SGC	81.0 ± 0.0	71.9 ± 0.1	78.9 ± 0.0

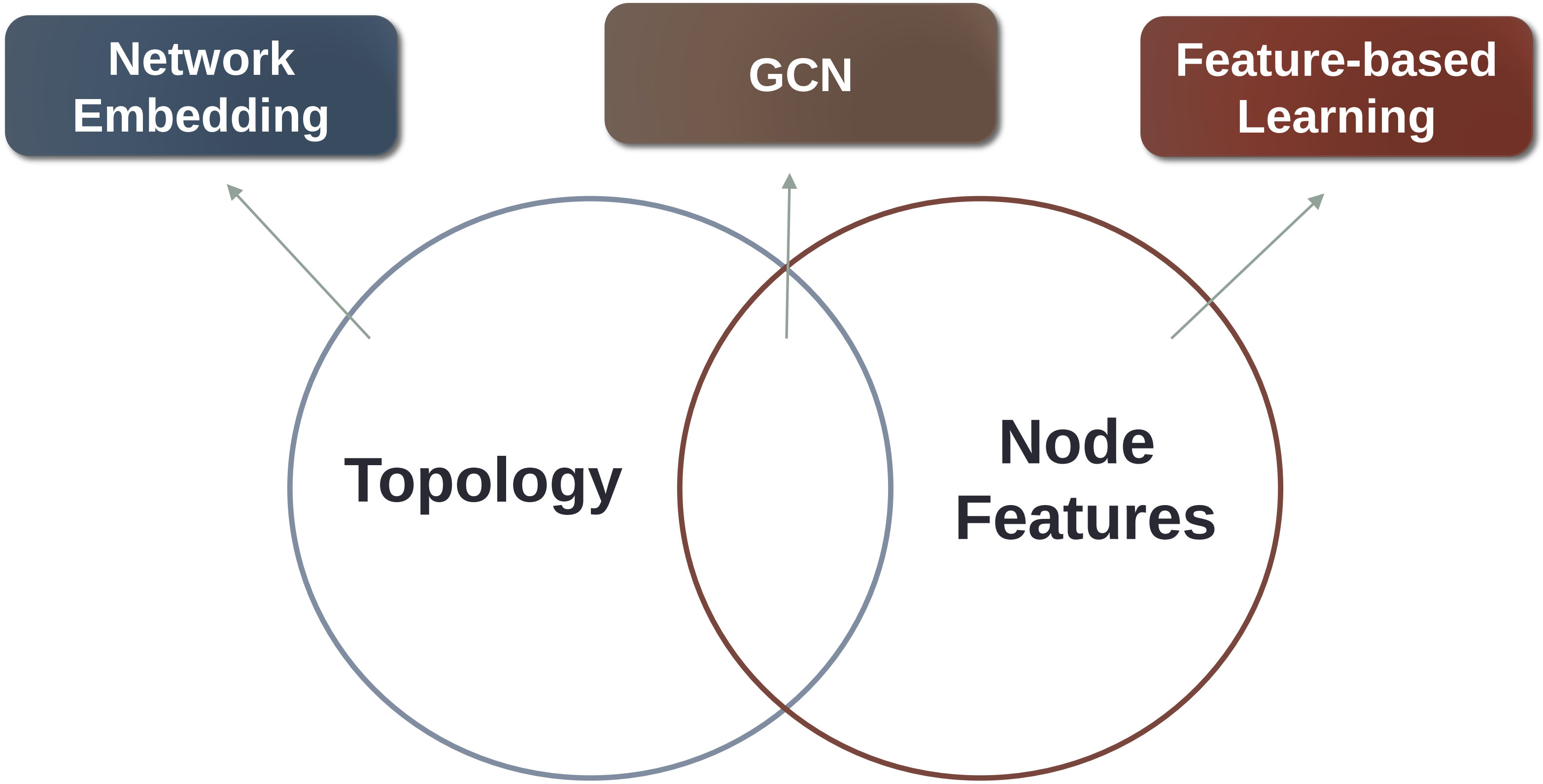
Text Classification

Dataset	Model	Test Acc. ↑	Time (seconds) ↓
20NG	GCN	87.9 ± 0.2	1205.1 ± 144.5
	SGC	88.5 ± 0.1	19.06 ± 0.15
R8	GCN	97.0 ± 0.2	129.6 ± 9.9
	SGC	97.2 ± 0.1	1.90 ± 0.03
R52	GCN	93.8 ± 0.2	245.0 ± 13.0
	SGC	94.0 ± 0.2	3.01 ± 0.01
Ohsumed	GCN	68.2 ± 0.4	252.4 ± 14.7
	SGC	68.5 ± 0.3	3.02 ± 0.02
MR	GCN	76.3 ± 0.3	16.1 ± 0.4
	SGC	75.9 ± 0.3	4.00 ± 0.04

Wu, Felix, et al. Simplifying graph convolutional networks. *ICML, 2019.*

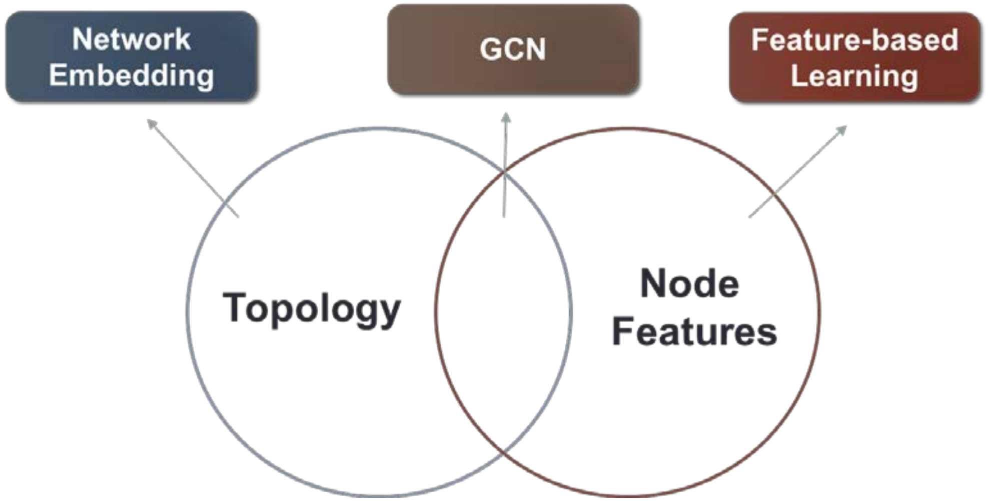
Network Embedding v.s. GCN

There is no better one, but there is more proper one.



Summaries and Conclusions

- Unsupervised v.s. (Semi-)Supervised
- Topology-driven v.s. Feature-driven
- For different healthcare tasks, there is no best one, but there is more proper one.



A Survey on Network Embedding

IEEE TRANSACTIONS ON
KNOWLEDGE AND
DATA ENGINEERING

A Survey on Network Embedding

Issue No. 01 - (preprint vol.)
ISSN: 1041-4347
pp: 1
DOI Bookmark: http://doi.ieeecomputersociety.org/10.1109/TKDE.2018.2849727

Peng Cui , Computer Science Department, Tsinghua University, Beijing, Beijing China (e-mail: cuip@tsinghua.edu.cn)
Xiao Wang , Computer Science, Tsinghua University, Beijing, Beijing China (e-mail: wangxiao007@mail.tsinghua.edu.cn)
Jian Pei , School of Computing Science, Simon Fraser Univeristy, Burnaby, British Columbia Canada (e-mail: jpei@cs.sfu.ca)
Wenwu Zhu , Department of Computer Science, Tsinghua Univerisity, Beijing, Beijing China (e-mail: wwzhu@tsinghua.edu.cn)

ABSTRACT
Network embedding assigns nodes in a network to low-dimensional representations and effectively preserves the network structure. Recently, a significant amount of progresses have been made toward this emerging network analysis paradigm. In this survey, we focus on categorizing and then reviewing the current development on network embedding methods, and point out its future research directions. We first summarize the motivation of network embedding. We discuss the classical graph embedding algorithms and their relationship with network embedding. Afterwards and primarily, we provide a comprehensive overview of a large number of network embedding methods in a systematic manner, covering the structure- and property-preserving network embedding methods, the network embedding methods with side information and the advanced information preserving network embedding methods. Moreover, several evaluation approaches for network embedding and some useful online resources, including the network data sets and softwares, are reviewed, too. Finally, we discuss the framework of exploiting these network embedding methods to build an effective system and point out some potential future directions.

Peng Cui, Xiao Wang, Jian Pei, Wenwu Zhu. **A Survey on Network Embedding**. *IEEE TKDE, 2019*.

Deep Learning on Graphs: A Survey

Deep Learning on Graphs: A Survey

Ziwei Zhang, Peng Cui and Wenwu Zhu

Abstract—Deep learning has been shown successful in a number of domains, ranging from acoustics, images to natural language processing. However, applying deep learning to the ubiquitous graph data is non-trivial because of the unique characteristics of graphs. Recently, a significant amount of research efforts have been devoted to this area, greatly advancing graph analyzing techniques. In this survey, we comprehensively review different kinds of deep learning methods applied to graphs. We divide existing methods into three main categories: semi-supervised methods including Graph Neural Networks and Graph Convolutional Networks, unsupervised methods including Graph Autoencoders, and recent advancements including Graph Recurrent Neural Networks and Graph Reinforcement Learning. We then provide a comprehensive overview of these methods in a systematic manner following their history of developments. We also analyze the differences of these methods and how to composite different architectures. Finally, we briefly outline their applications and discuss potential future directions.

Index Terms—Graph Data, Deep Learning, Graph Neural Network, Graph Convolutional Network, Graph Autoencoder.

1 Introduction

In the last decade, deep learning has been a "crown jewel" in artificial intelligence and machine learning [1], showing superior performance in acoustics [2], images [3] and natural language processing [4]. The expressive power of deep learning to extract complex patterns underlying data has been well recognized. On the other hand, graphs[1] are ubiquitous in the real world, repre-

- **Scalability and parallelization**. In the big-data era, real graphs can easily have millions of nodes and edges, such as social networks or e-commerce networks [8]. As a result, how to design scalable models, preferably with a linear time complexity, becomes a key problem. In addition, since nodes and edges in the graph are interconnected and often need to be modeled as a whole, how to conduct parallel computing is another critical issue.

Ziwei Zhang, Peng Cui, Wenwu Zhu. Deep Learning on Graphs: A Survey. Arxiv, *2019*.

Thanks!

Peng Cui
cuip@tsinghua.edu.cn
http://pengcui.thumedialab.com