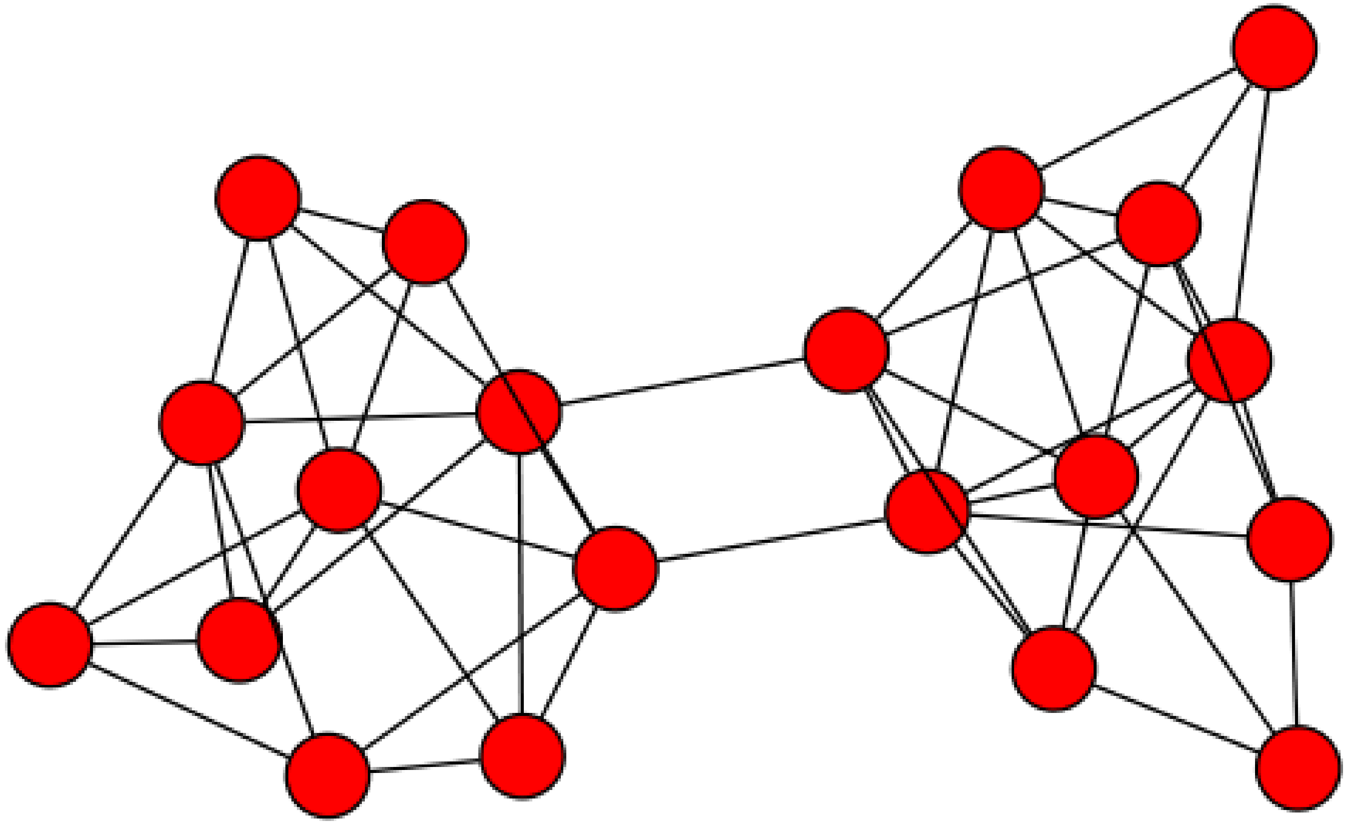
Perspectives and Outlook on
Network Embedding and GCN

Peng Cui
Tsinghua University

Network (Graph)

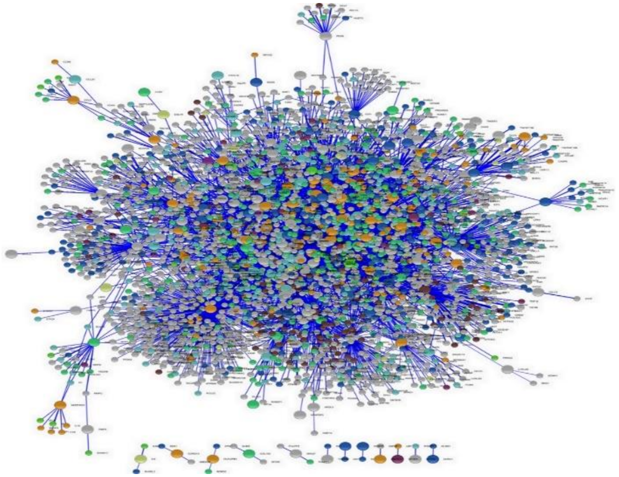
The general description of data and their relations.



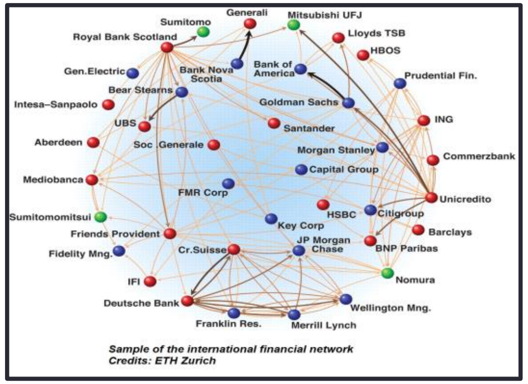
Many types of data are networks

Social Networks

Biology Networks

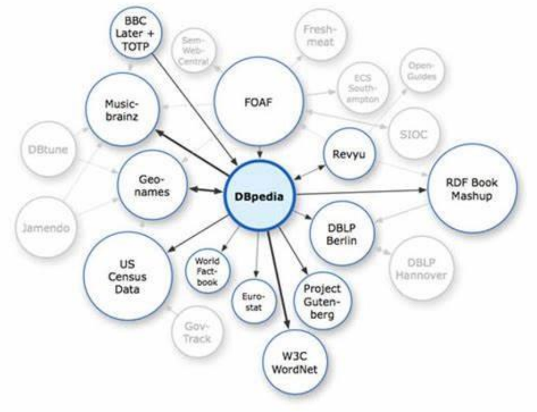


Finance Networks



Internet of Things

Information Networks

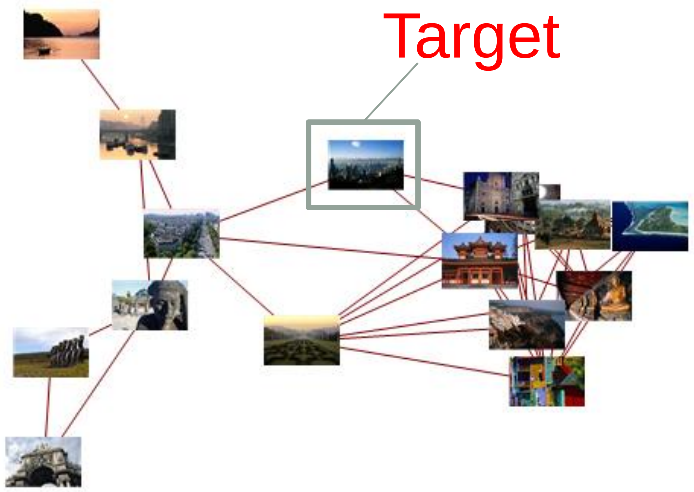


Logistic Networks

Why network is important?

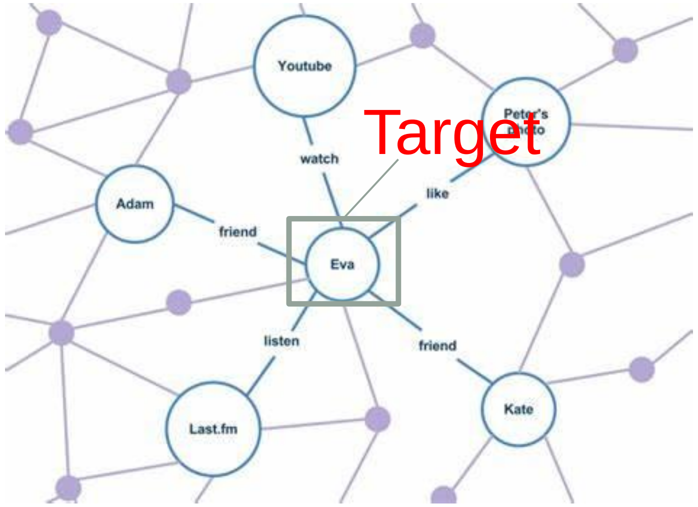
In few cases, you only care about a subject but not its relations with other subjects.

Image Characterization



Reflected by relational subjects

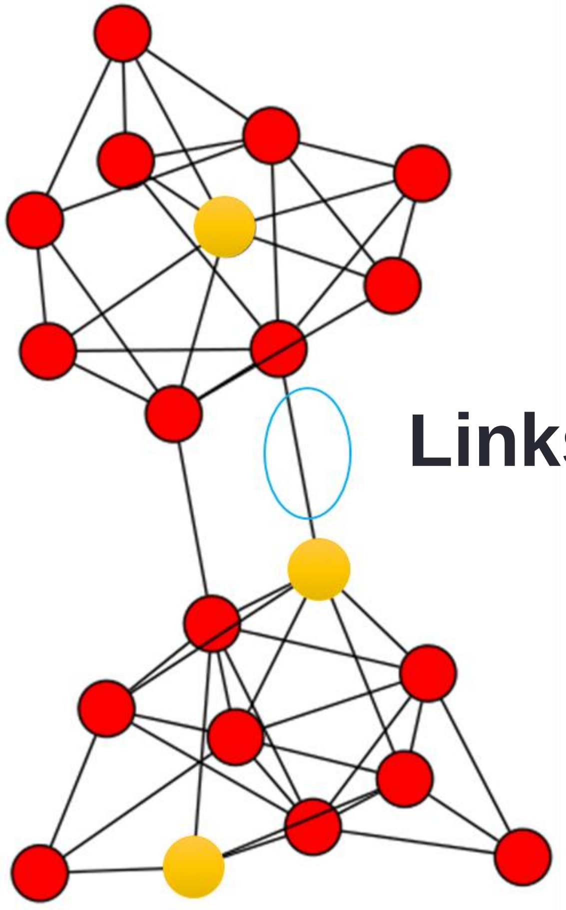
Social Capital



Decided by relational subjects

Networks are not *learning-friendly*

$G = (V, E)$



Inapplicability of ML methods

Links ⇨ **Topology**

Pipeline for network analysis

Network Data

Feature Extraction

Discovery

Network Applications

Learnability

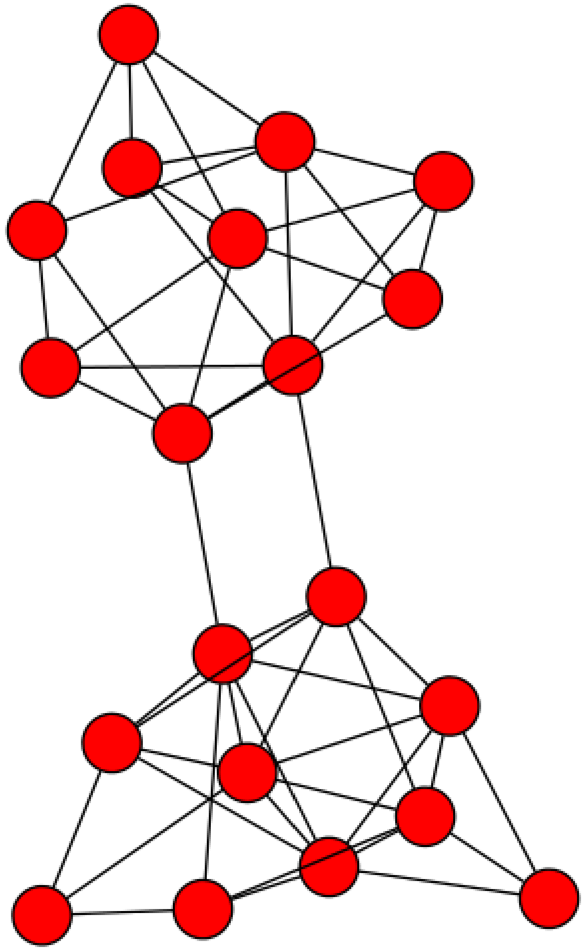
Learning from networks

Network Embedding

GCN

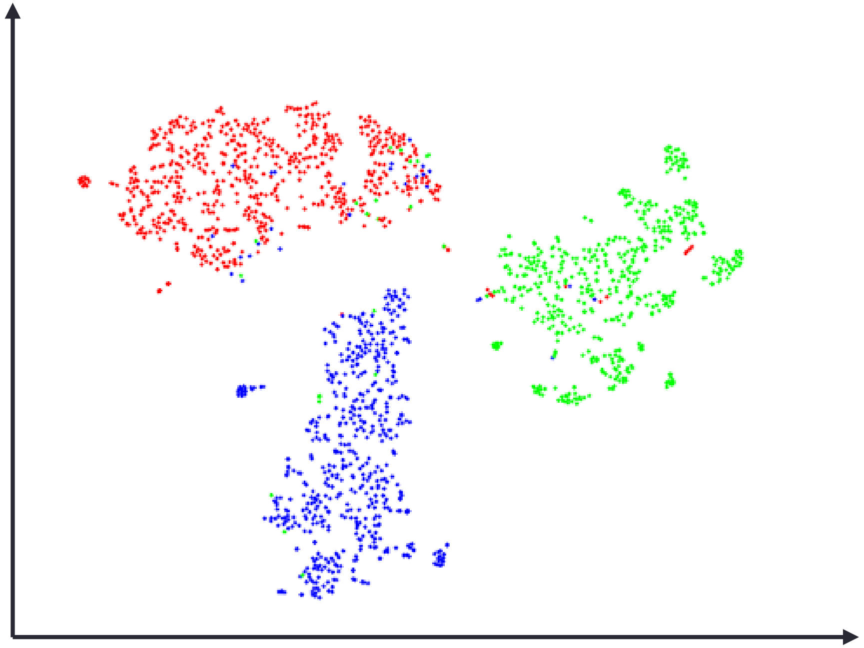
Network Embedding

G = (V, E)



G = (V)

Vector Space



generate

embed

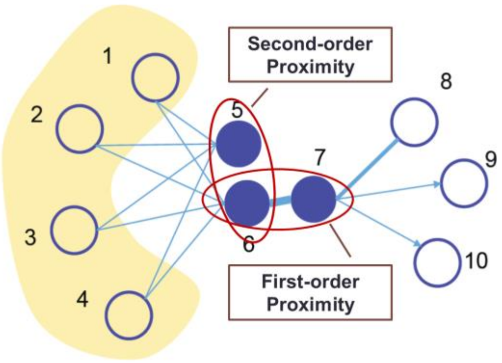
- Easy to parallel
- Can apply classical ML methods

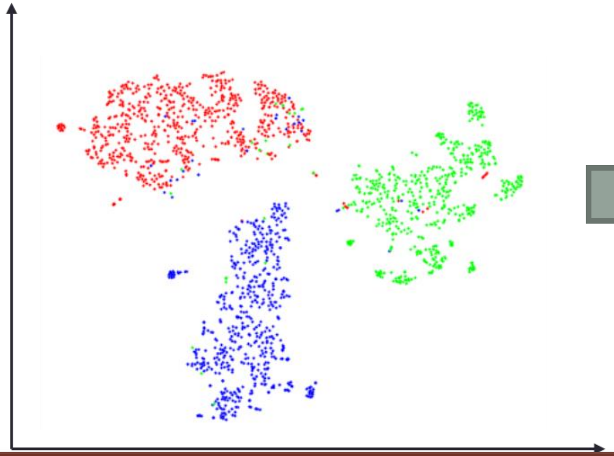
The goal of network embedding

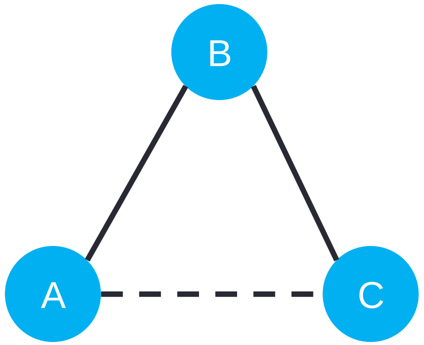
Goal **Support network inference in vector space**

Reflect network structure

Maintain network properties







Transitivity

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

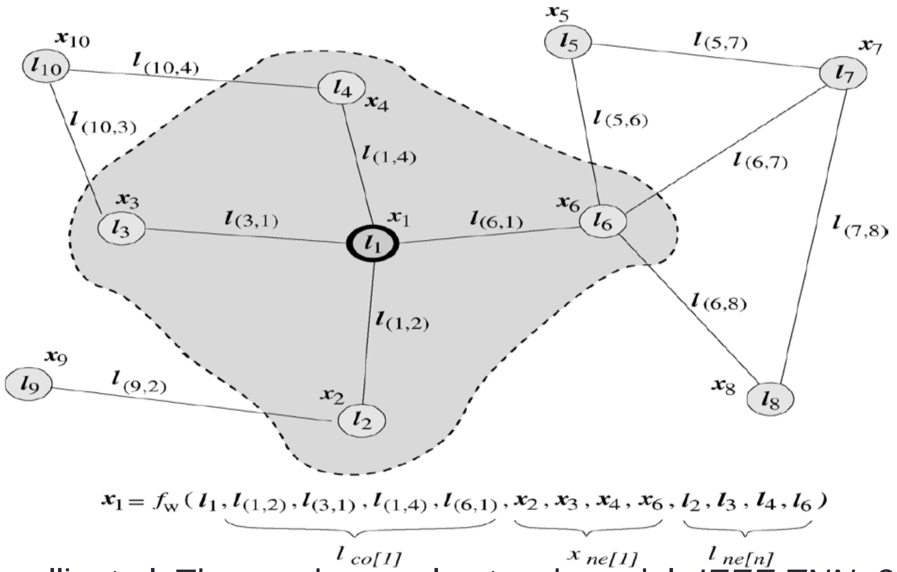
Graph Neural Networks

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



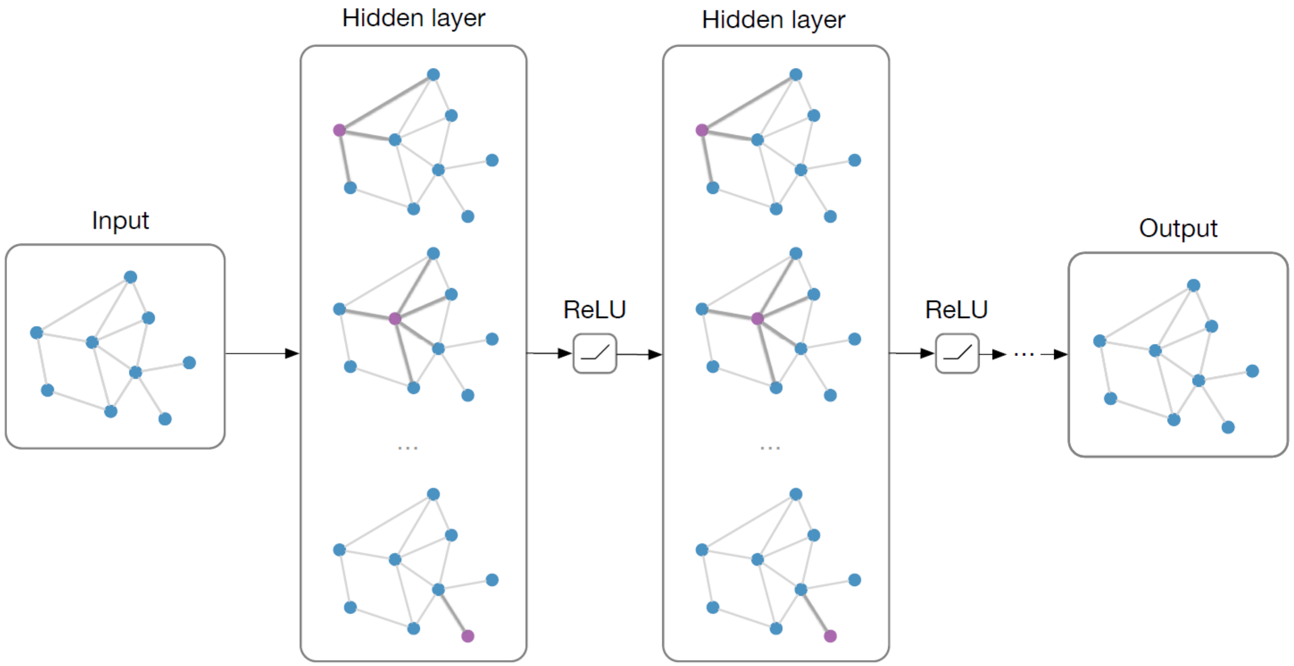
F. Scarselli, et al. The graph neural network model. *IEEE TNN, 2009.*

Graph Convolutional Networks (GCN)

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

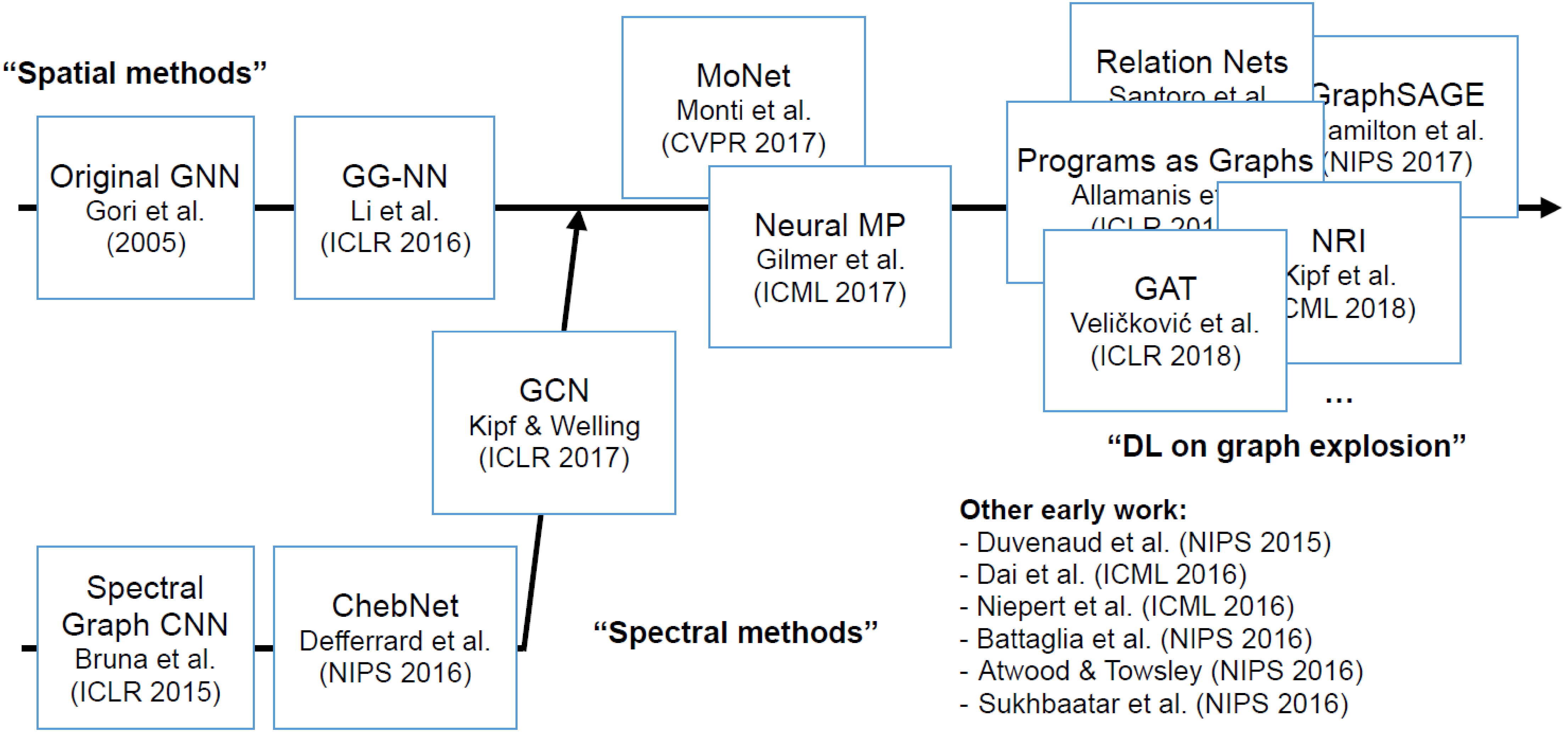
$$\mathbf{H}^{l+1} = \rho\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{l}\mathbf{\Theta}^{l}\right)$$

- 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.*

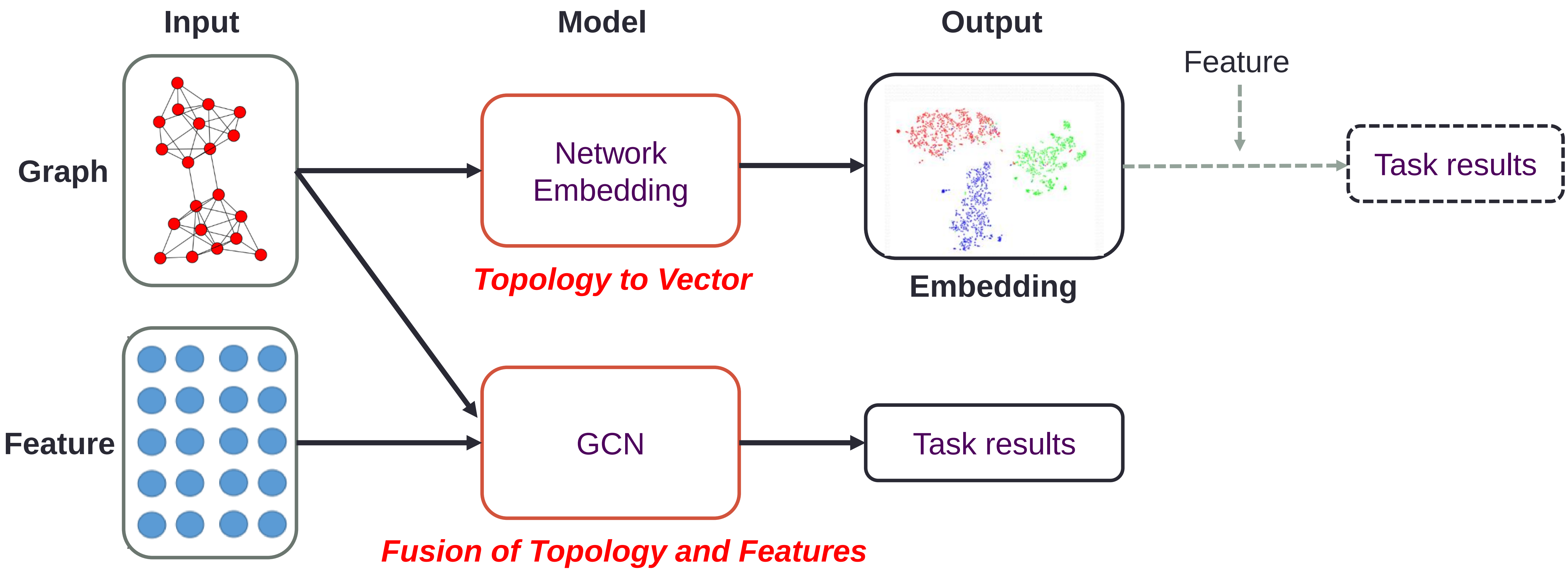
A brief history of GNNs



Other early work:
- Duvenaud et al. (NIPS 2015)
- Dai et al. (ICML 2016)
- Niepert et al. (ICML 2016)
- Battaglia et al. (NIPS 2016)
- Atwood & Towsley (NIPS 2016)
- Sukhbaatar et al. (NIPS 2016)

(slide inspired by Alexander Gaunt's talk on GNNs)

Network Embedding and GCN



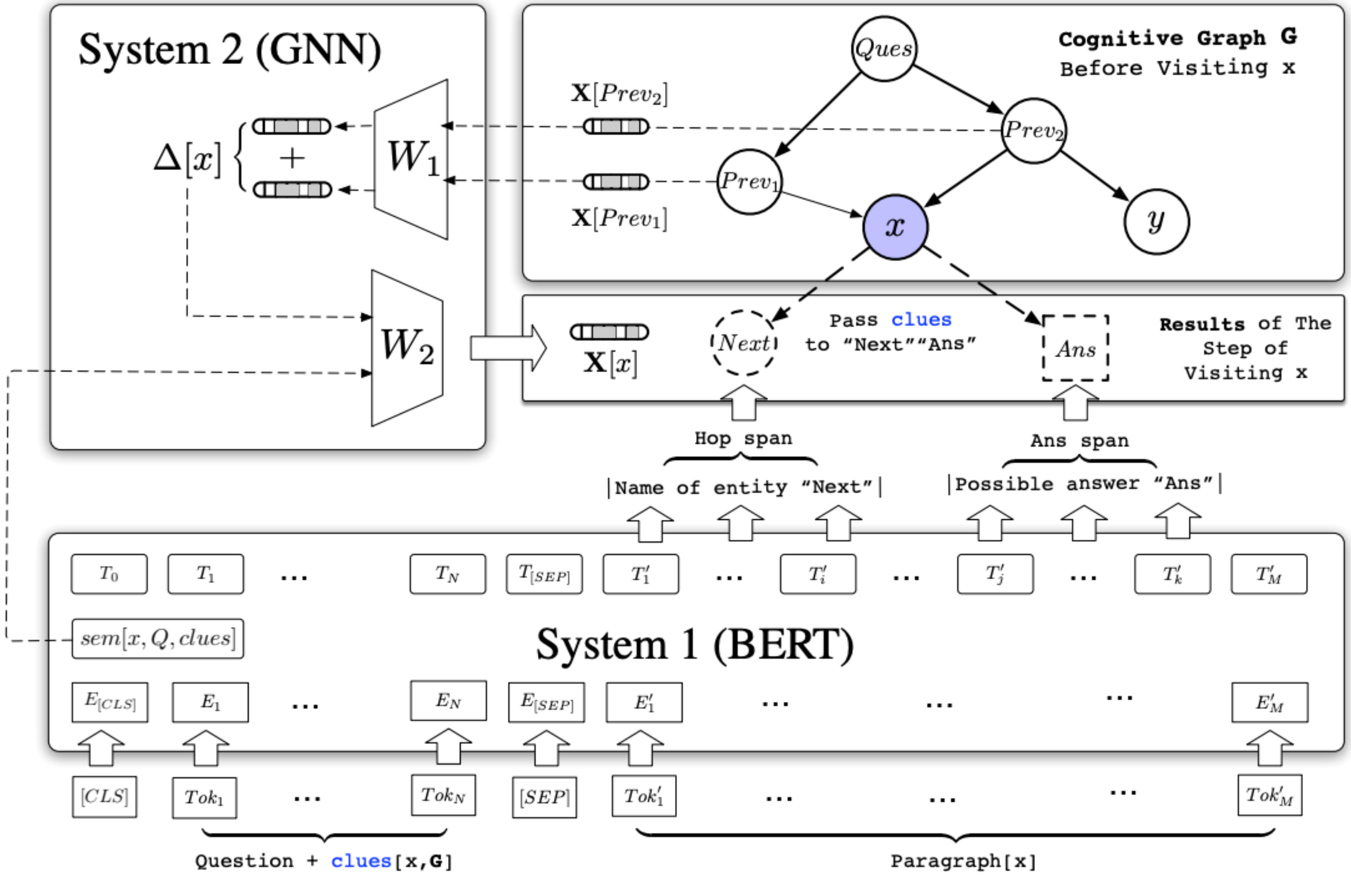
Unsupervised v.s. (Semi-)Supervised

Graph convolutional network v.s. Network embedding

- In some sense, they are different.
- **Graphs** exist in ***mathematics***. (Data Structure)
  - Mathematical structures used to model pairwise relations between objects
- **Networks** exist in the ***real world***. (Data)
  - Social networks, logistic networks, biology networks, transaction networks, etc.
- A network can be represented by a graph.
- A dataset that is not a network can also be represented by a graph.

GCN for Natural Language Processing

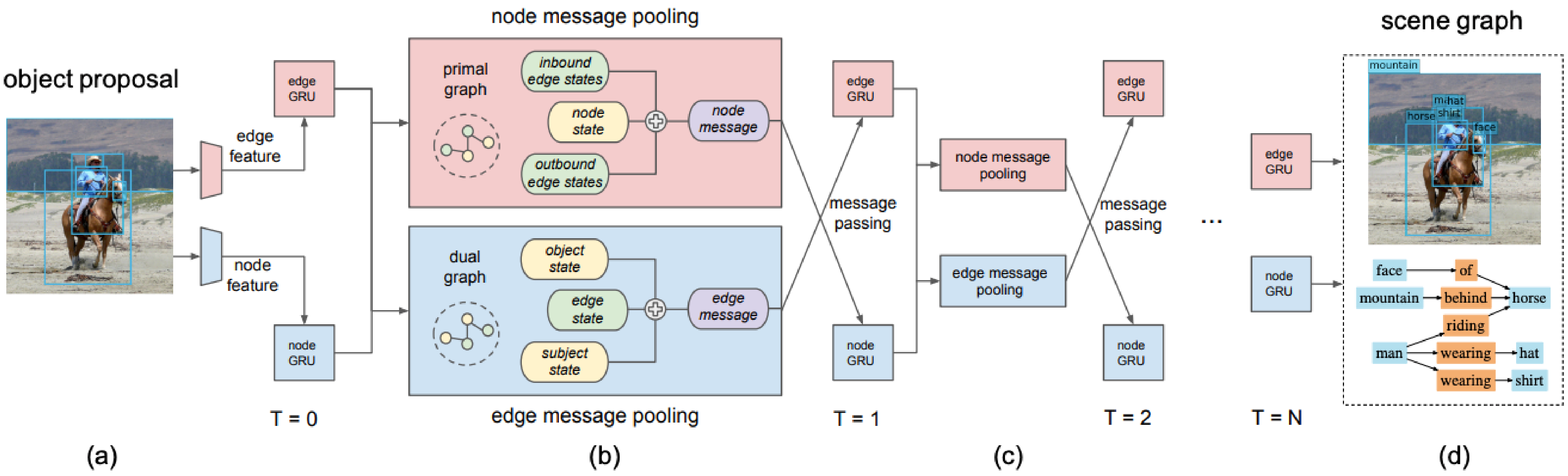
- Many papers on BERT + GNN.
- BERT is for retrieval.
  - It creates an initial graph of relevant entities and the initial evidence.
- GNN is for reasoning.
  - It collects evidence (i.e., old messages on the entities) and arrive at new conclusions (i.e., new messages on the entities), by passing the messages around and aggregating them.



Cognitive Graph for Multi-Hop Reading Comprehension at Scale. Ding et al., ACL 2019.
Dynamically Fused Graph Network for Multi-hop Reasoning. Xiao et al., ACL 2019.

GCN for Computer Vision

- A popular trend in CV is to construct a graph during the learning process.
  - To process multiple objects or parts in a scene, and to infer their relationships.
- Example: Scene graphs.



Scene Graph Generation by Iterative Message Passing. Xu et al., CVPR 2017.
Image Generation from Scene Graphs. Johnson et al., CVPR 2018.

GCN for Symbolic Reasoning

- We can view the process of symbolic reasoning as a directed acyclic graph.
- Many recent efforts use GNNs to perform symbolic reasoning.

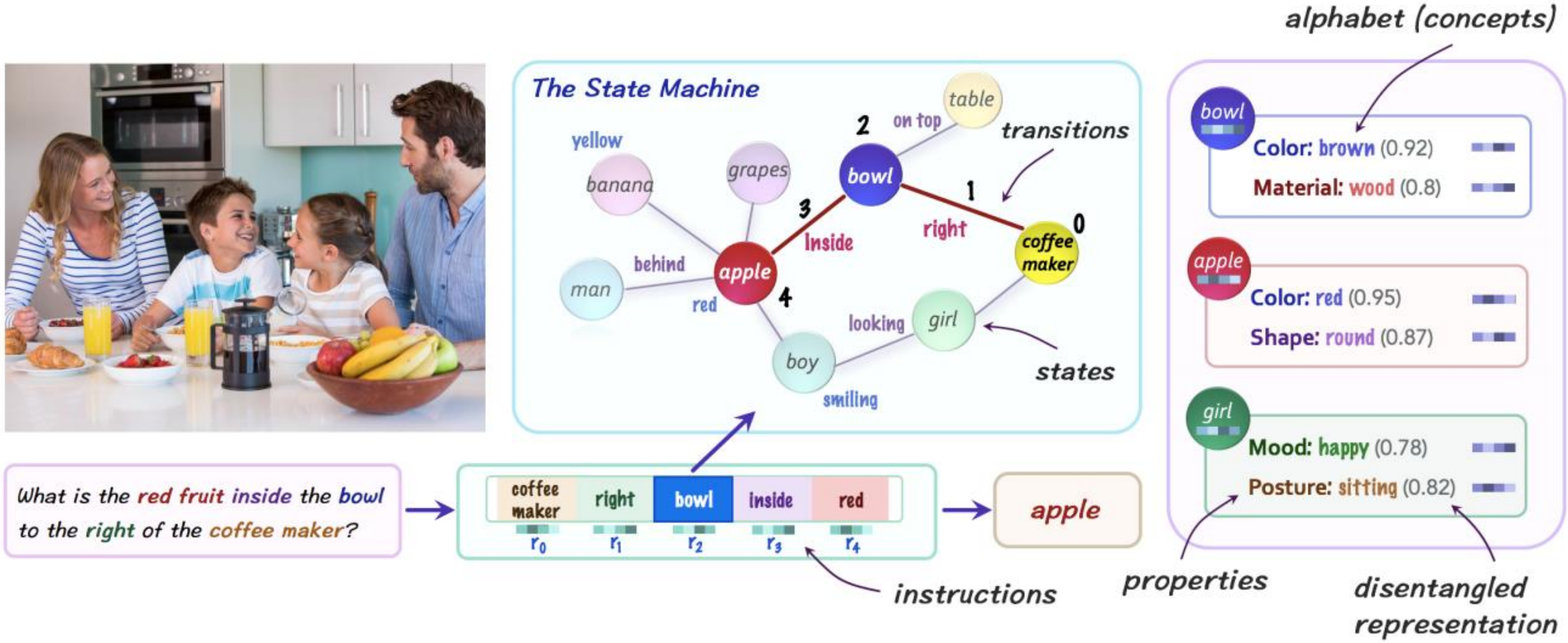
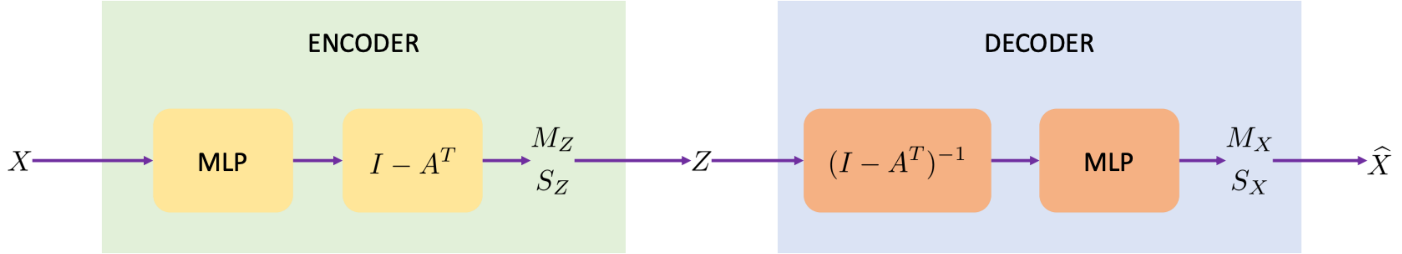


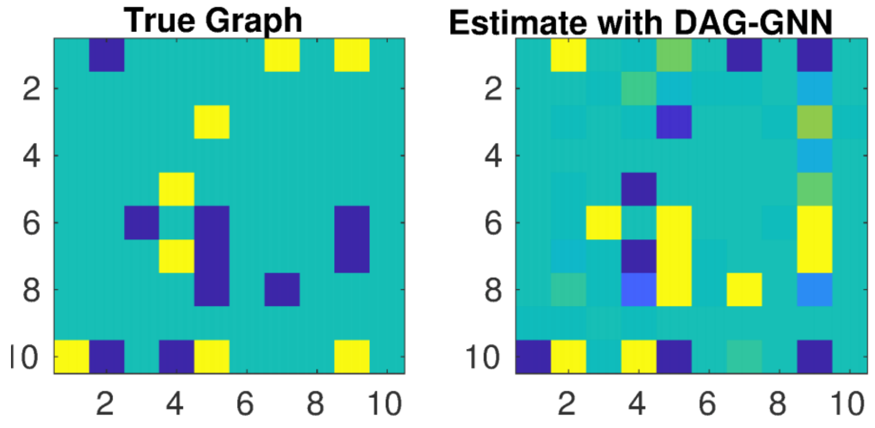
Figure 1: The Neural State Machine is a graph network that simulates the computation of an automaton.

Learning by Abstraction: The Neural State Machine. Hudson & Manning, 2019.
Can Graph Neural Networks Help Logic Reasoning? Zhang et al., 2019.
Symbolic Graph Reasoning Meets Convolutions. Liang et al., NeurIPS 2018.

GCN for Structural Equation Modeling

- Structural equation modeling, a form of causal modeling, tries to describe the relationships between the variables as a directed acyclic graph (DAG).
- GNN can be used to represent a nonlinear structural equation and help find the DAG, after treating the adjacency matrix as parameters.





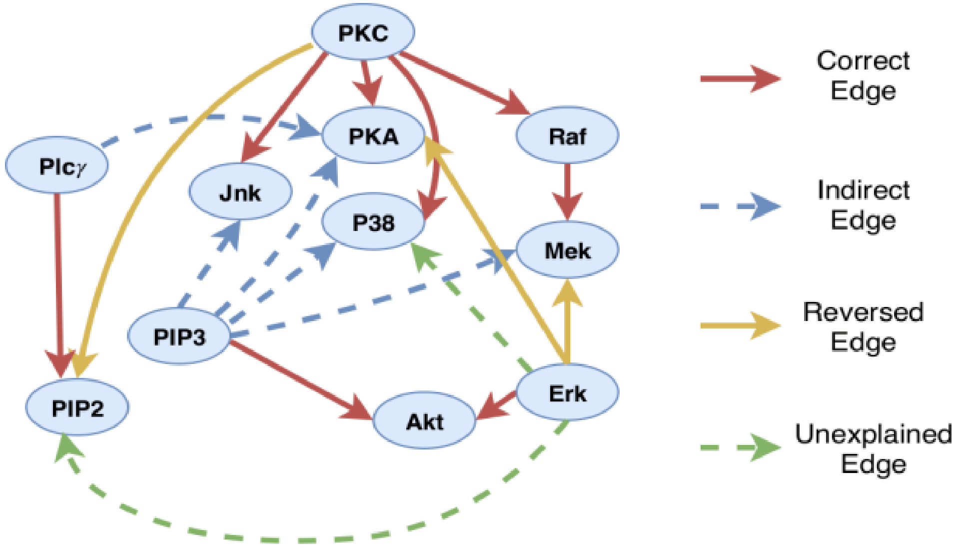
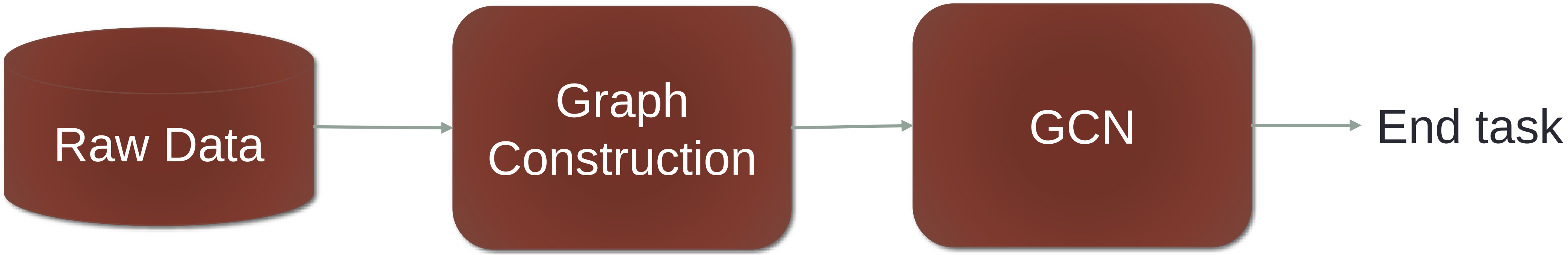


Figure 8. Estimate protein signaling network.

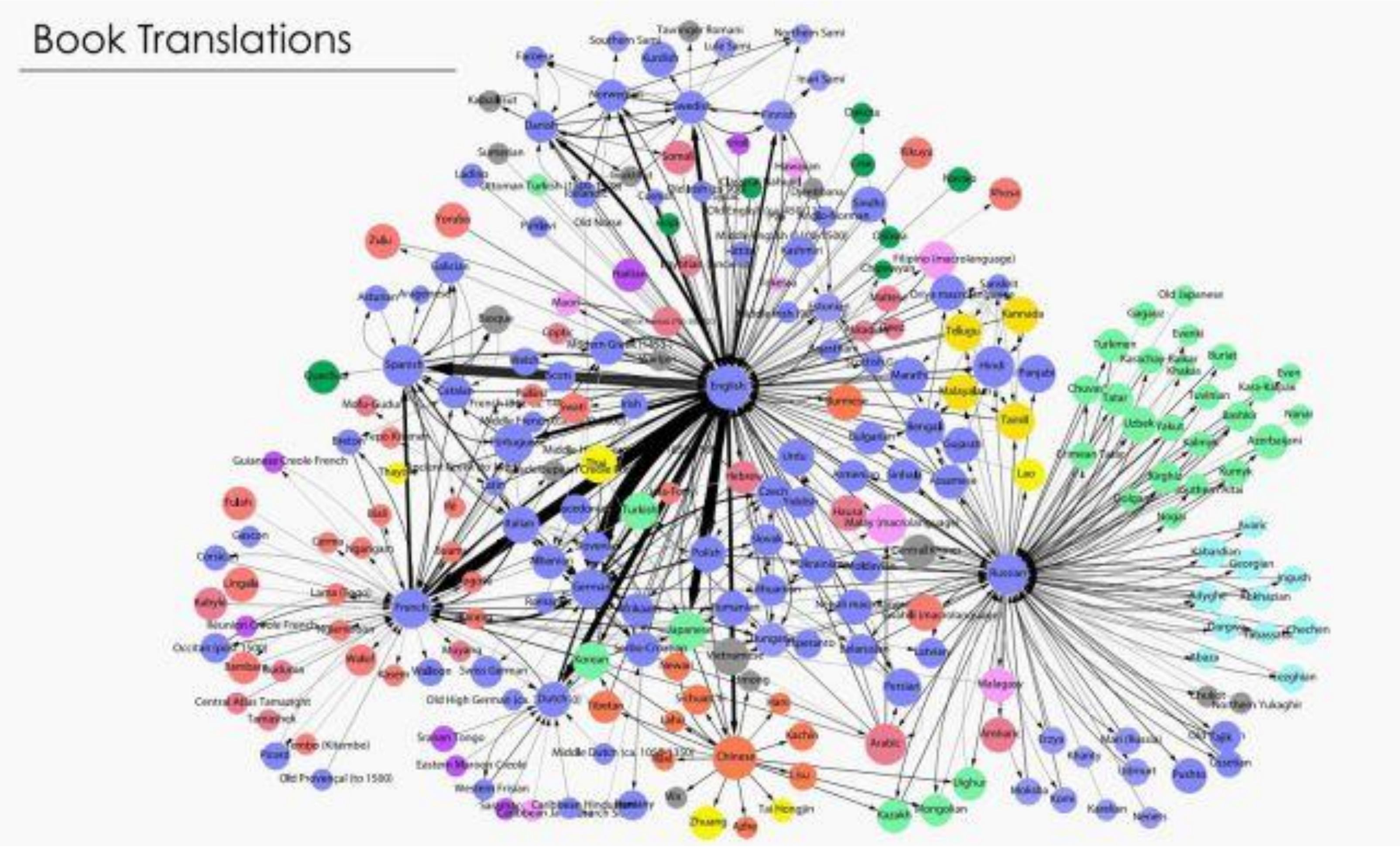
DAG-GNN: DAG Structure Learning with Graph Neural Networks. Yu et al., ICML 2019.

Pipeline for (most) GCN works



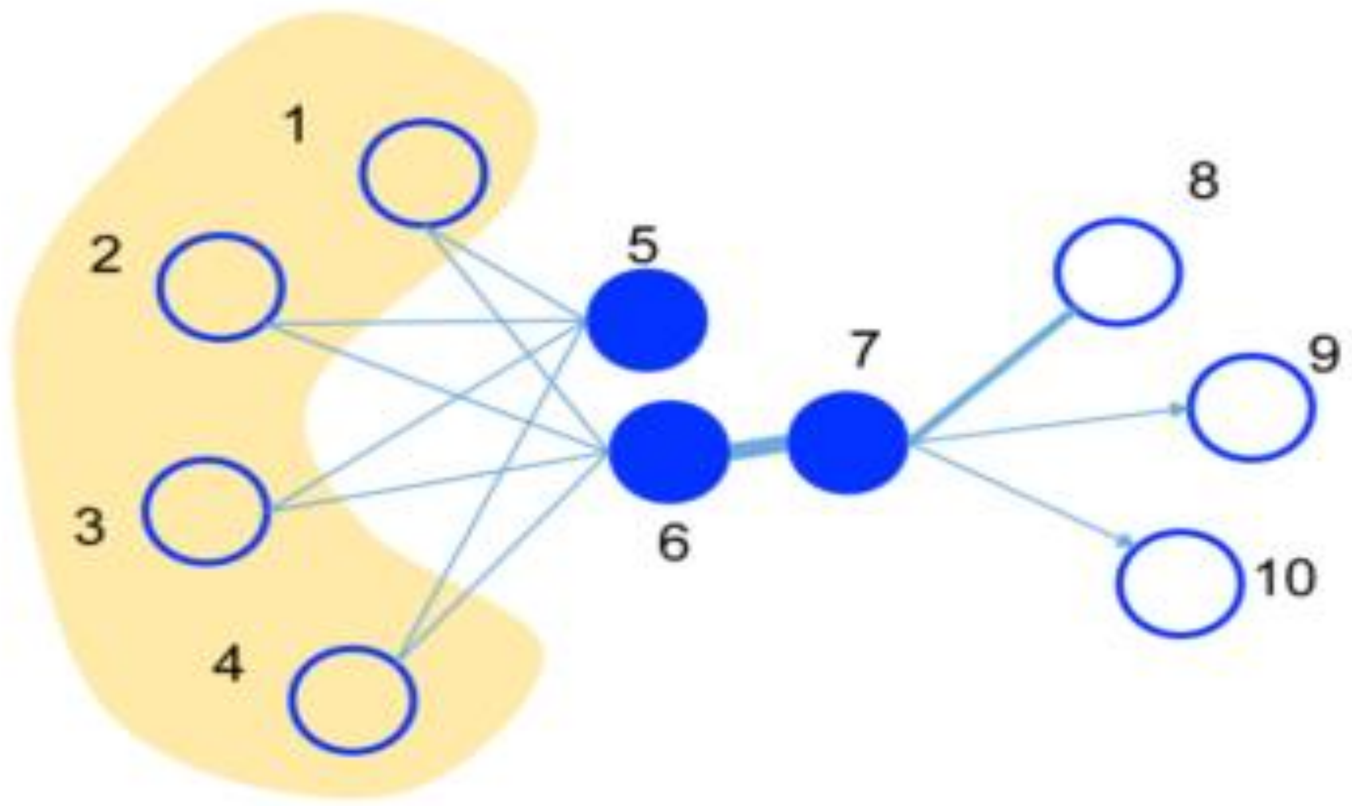
Network embedding: topology to vector

- **Co-occurrence (neighborhood)**



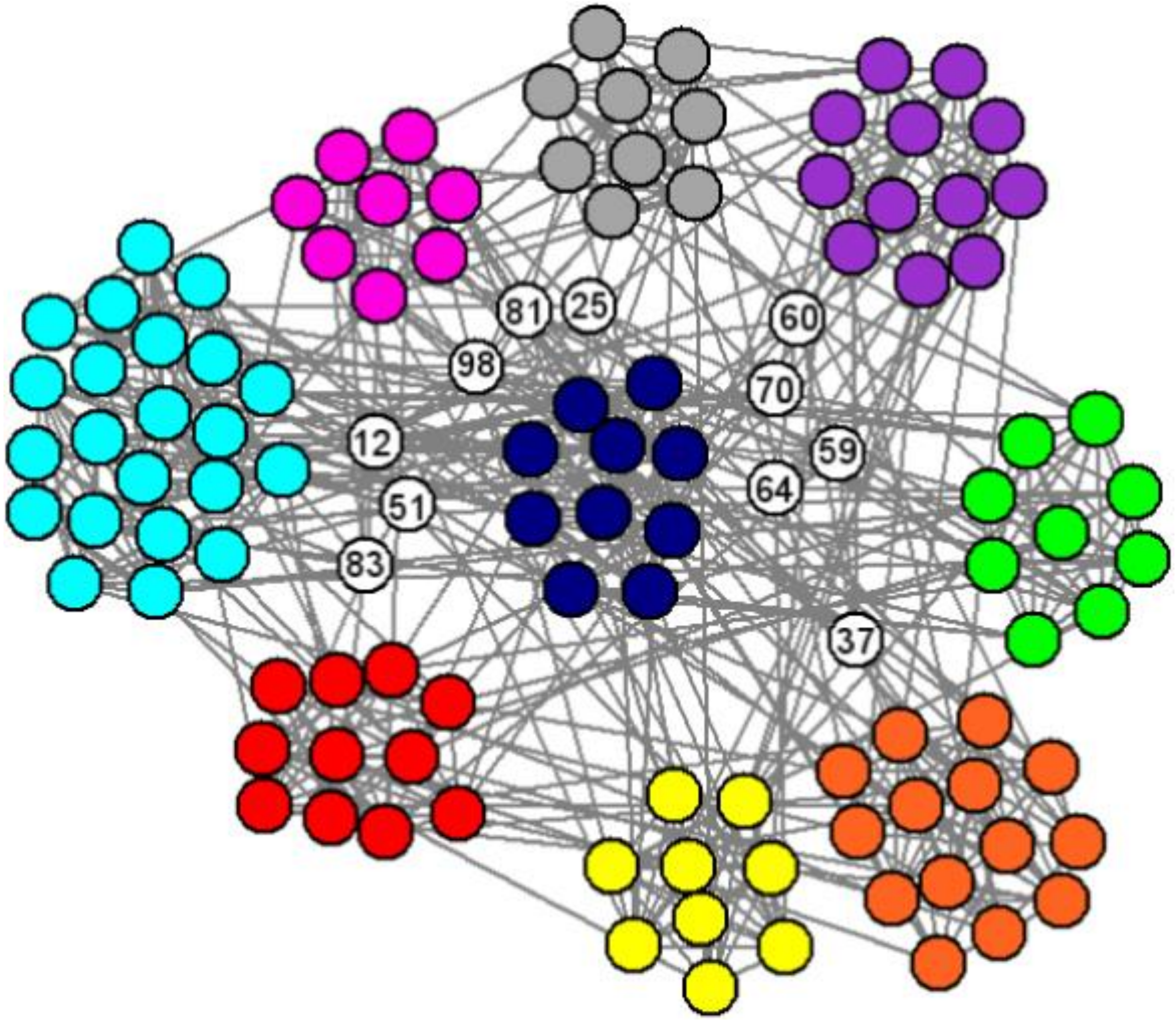
Network embedding: topology to vector

- **High-order proximities**



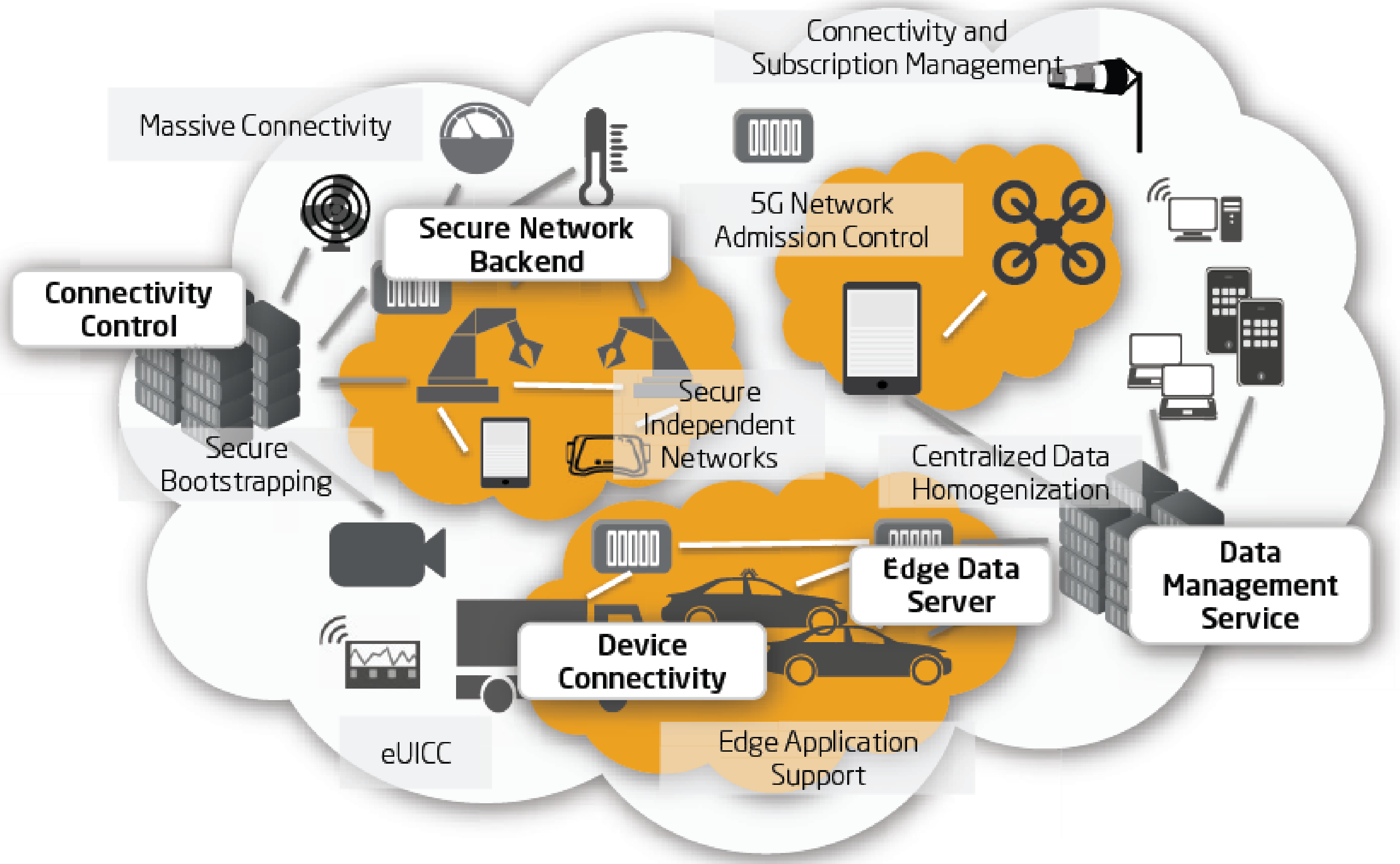
Network embedding: topology to vector

- Communities

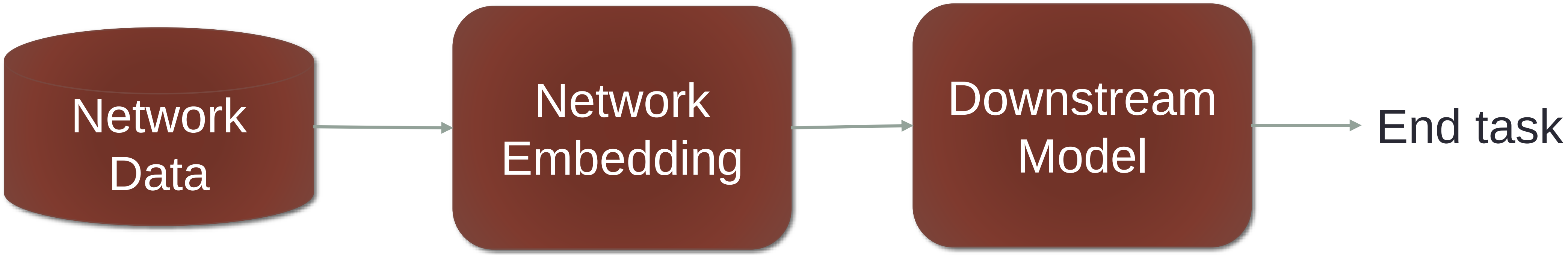


Network embedding: topology to vector

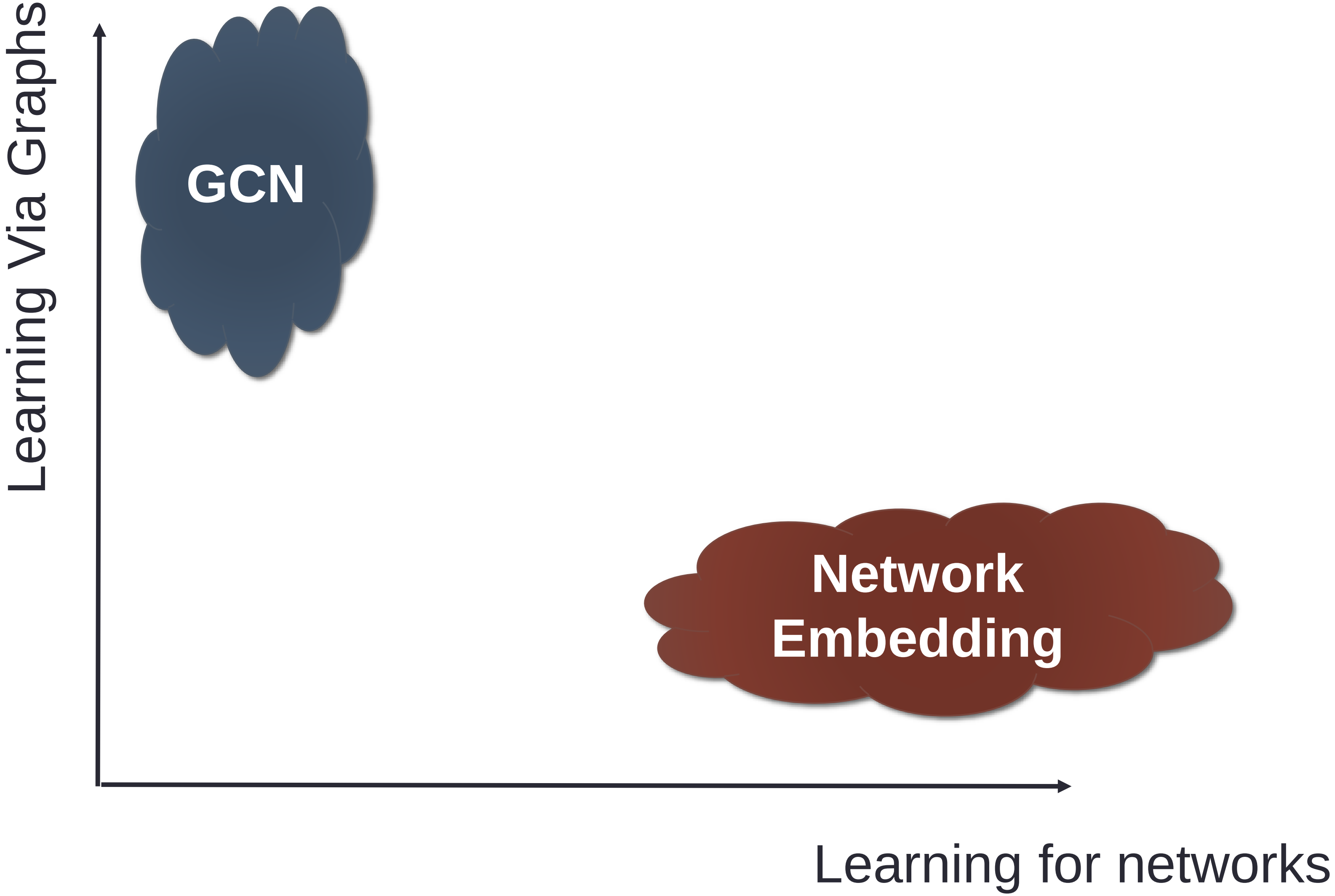
- **Heterogeneous networks**



Pipeline for (most) Network Embedding works



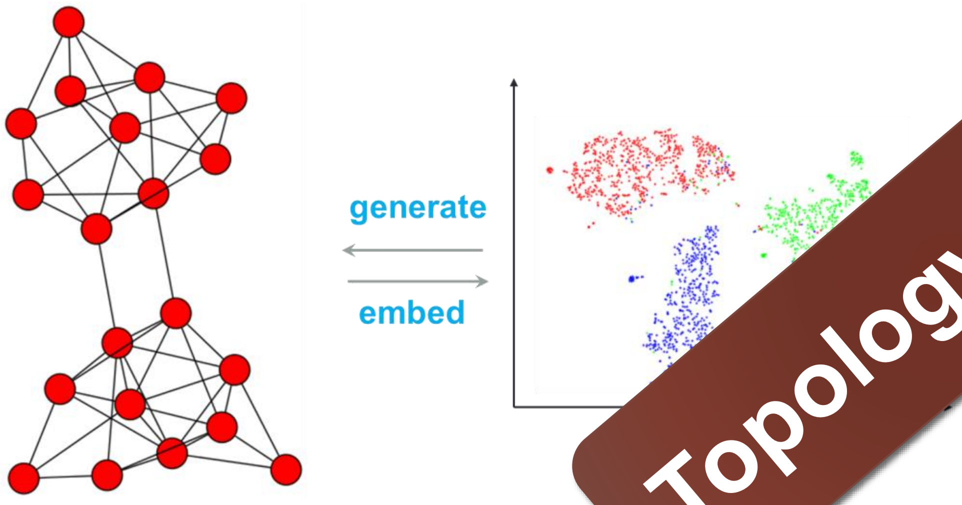
Learning for Networks vs. Learning via Graphs



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



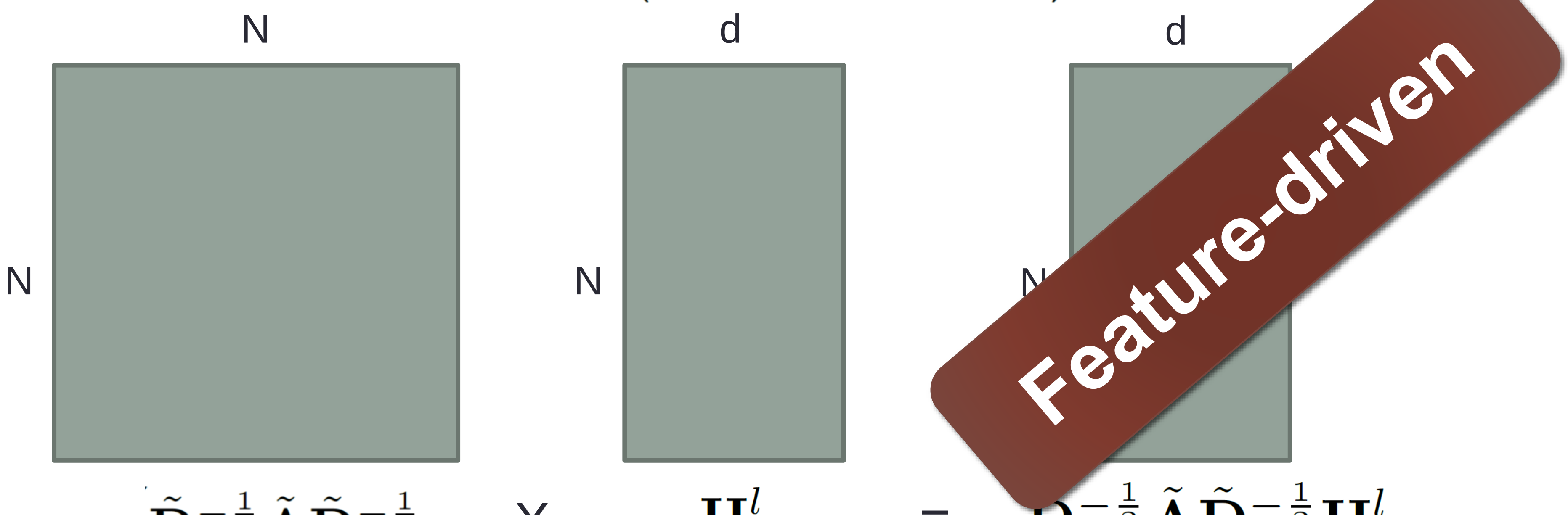
- Non-transitivity
- ansitivity
- Uncertainty
- Dynamic
- Heterogeneity
- Interpretability

Topology-driven

The intrinsic problem GCN is solving

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

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



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

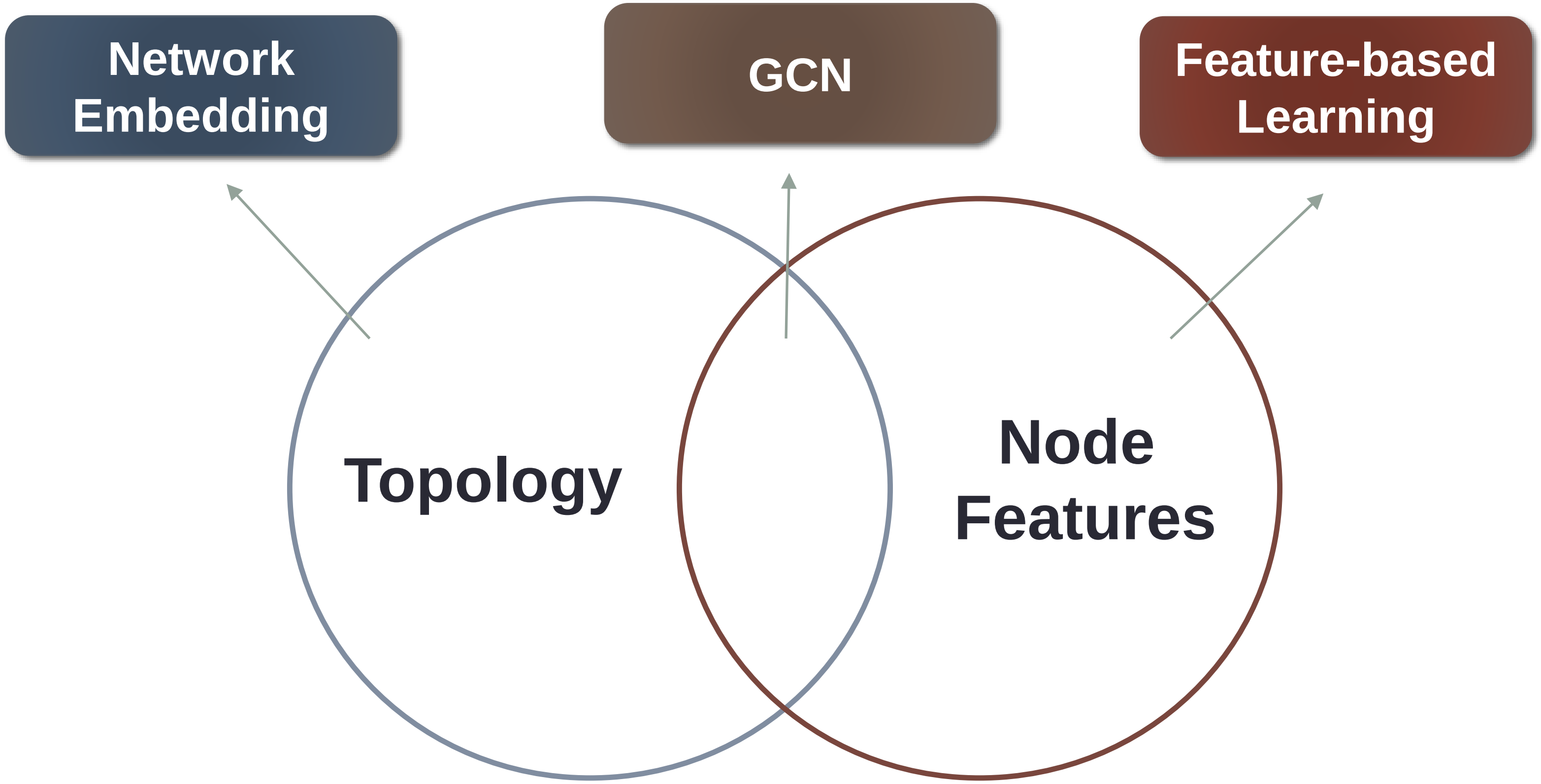
What if the problem is topology-driven?

- 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

Network Embedding v.s. GCN

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



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

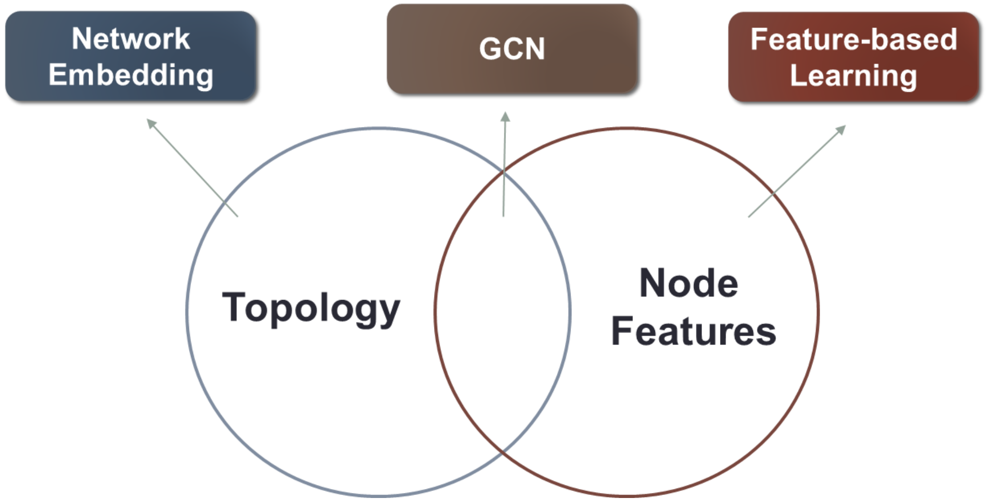
Text Classification

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.*

Summaries and Conclusions

- Unsupervised v.s. (Semi-)Supervised
- Learning for Networks v.s. Learning via Graphs
- Topology-driven v.s. Feature-driven
- Both GCN and NE need to treat the counterpart as the baselines



Thanks!

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