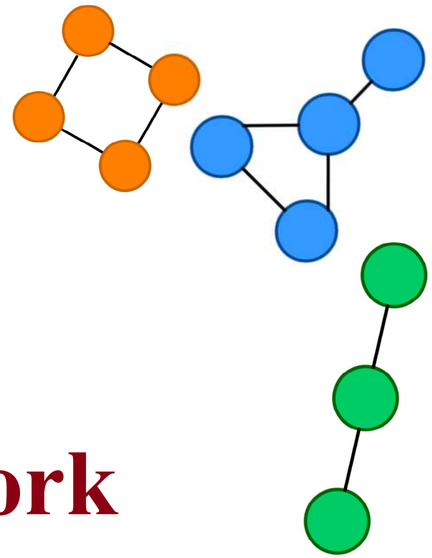
PEKING UNIVERSITY



Graph Structural-topic Neural Network

Authors: Qingqing Long*, Yilun Jin*, Guojie Song, Yi Li, and Wei Lin
Paper: http://arxiv.org/abs/2006.14278
Lab: https://www.gjsong-pku.cn/
Code: https://github.com/YimiAChack/GraphSTONE
Date: July 6, 2020

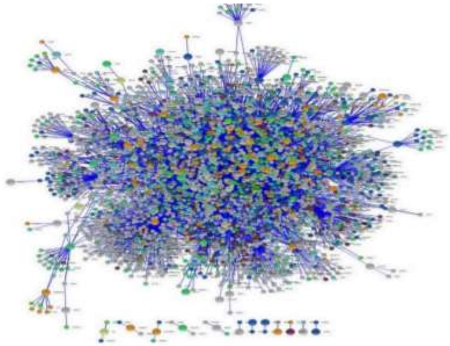
Outline

- Background and Motivation
- GraphSTONE
- Experiments
- Summary

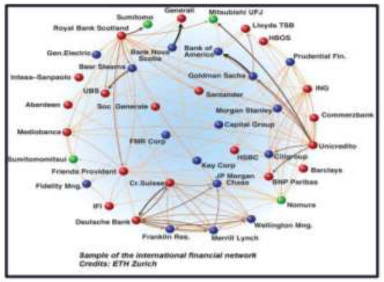
Networks

- Networks are **powerful data structures** that **encode relationships between objects.**
  - In many cases, we care not only the object itself, but also its links with other objects.

Social Networks



Biology Networks

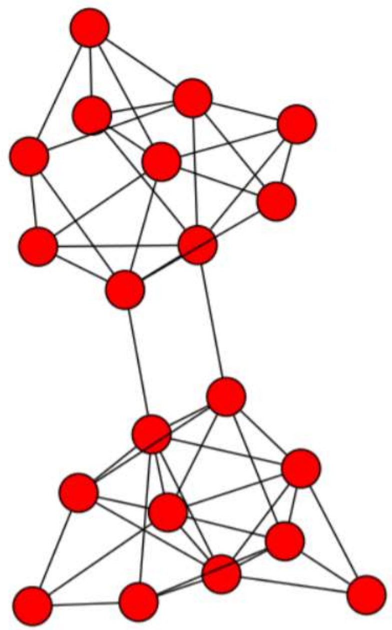


Finance Networks

Figure credit to P. Cui's tutorial at DLG workshop, KDD 2019.

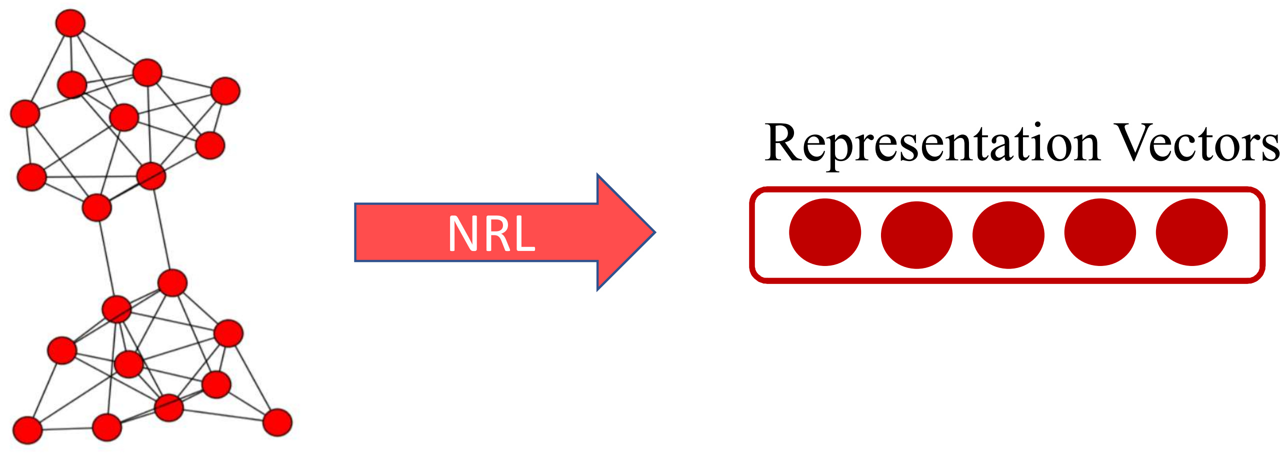
Networks are not learning friendly

- **Irregular, high-dimensional, and sparse.**
  - Degrees of nodes vary (power-law).
  - Probably millions of nodes.
  - A node only connects with very few other nodes.
- Therefore, **we need powerful learning tools!**



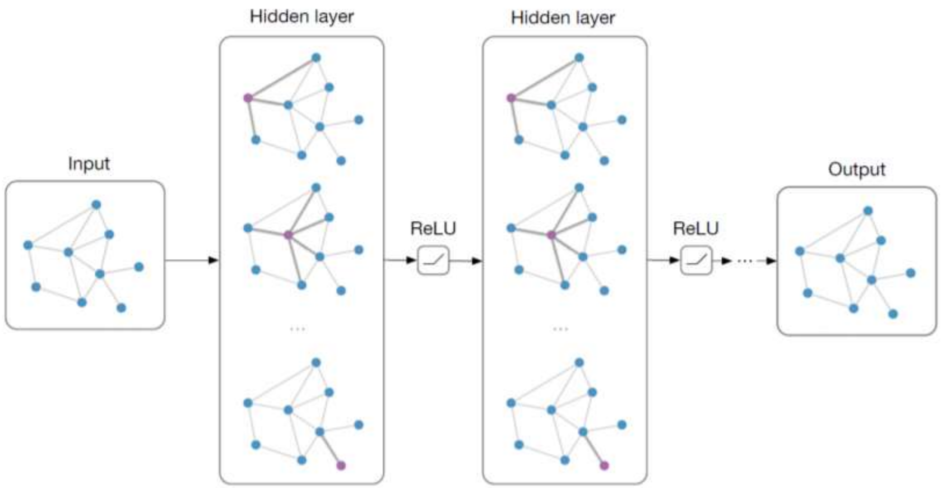
Network Representation Learning

- **Goal:** Transform irregular, high-dimensional and sparse network data (e.g. nodes, or the network itself) into *vectors*, according to network structures and node features.



Graph Convolutional Networks (GCNs)

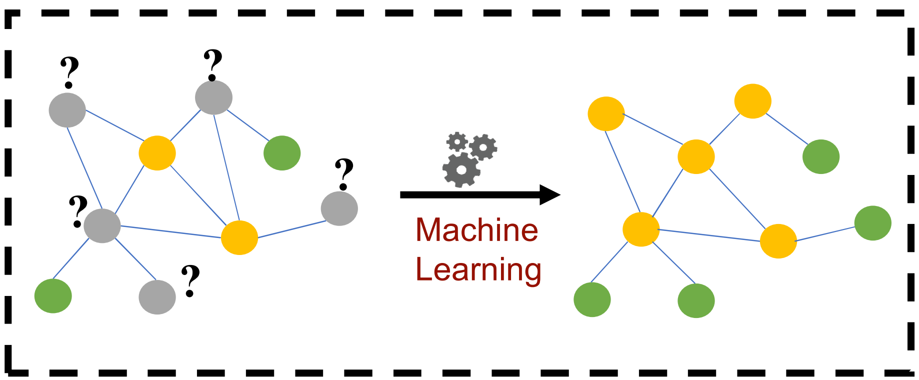
- **GCNs**
  - **Main idea:** For each layer, information is passed between each other through links, and aggregated by each node.
  - **Fuse node features with the help of network structures.**



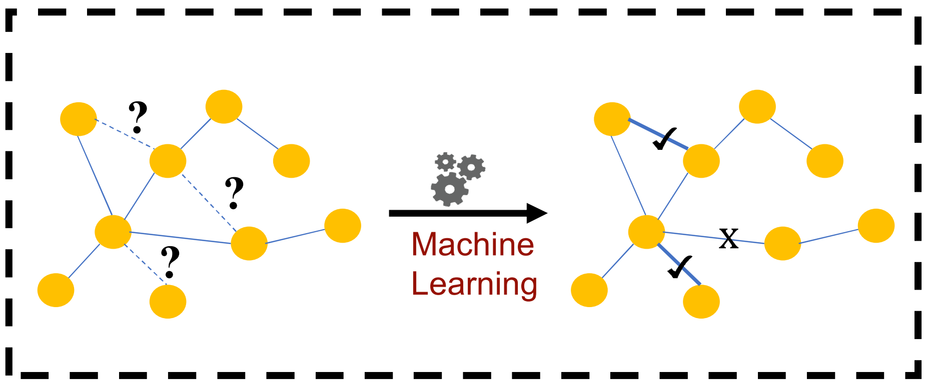
T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *ICLR, 2017.*

Graph Convolutional Networks (GCNs)

- **GCNs**
  - **Applications**: machine learning tasks in networks
  - e.g. Who is likely to know you? What items are likely to be of your interest?
  - Wide industrial applications.



Node Classification

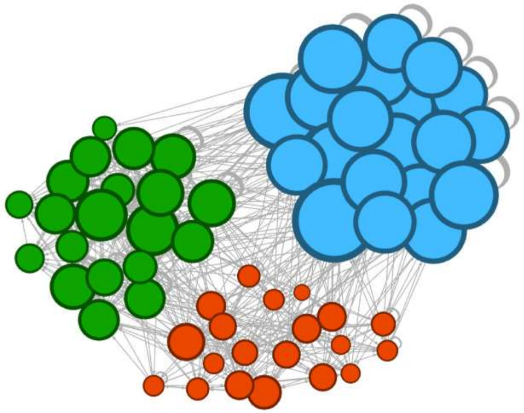


Link Prediction

Zhu et al. AliGraph: A Comprehensive Graph Neural Network Platform
Ying et al. Graph Convolutional Networks for Web-scale Recommender Systems

Graph Convolutional Networks (GCNs)

- **Rethinking: In what cases do GCNs perform badly?**
- **Synthetic data:** Stochastic block model with 10 blocks + random features.
- **GCN performs bad when network structures play the key role!**

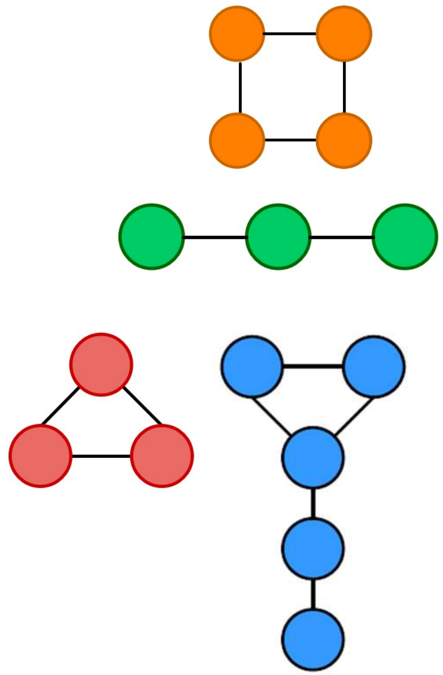


Method	Results
Random	10.0±0.1
DeepWalk	99.0±0.1
GCN	18.3±0.1

Credit to P. Cui's Tutorial at DLG Workshop, KDD 2019.
Perozzi et al. DeepWalk: Online Learning of Social Representations. In KDD, 2014

Drawbacks

- **Less capable** of expressing **structures of networks.**
  - Primarily focus on node features, as the previous example.

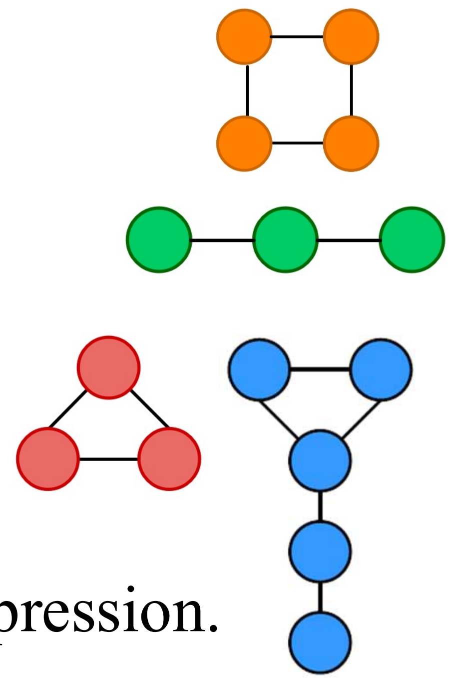


- **What network structures are important?**
  - High-order structural units (patterns) are generally indicative.
  - e.g. Motifs [1], graphlets [2].

[1] R. Milo et al. Network Motifs: Simple Building Blocks of Complex Network. Science, 2002.
[2] N. Przulj. Biological network comparison using graphlet degree distribution. Bioinformatics, 2007.

Drawbacks

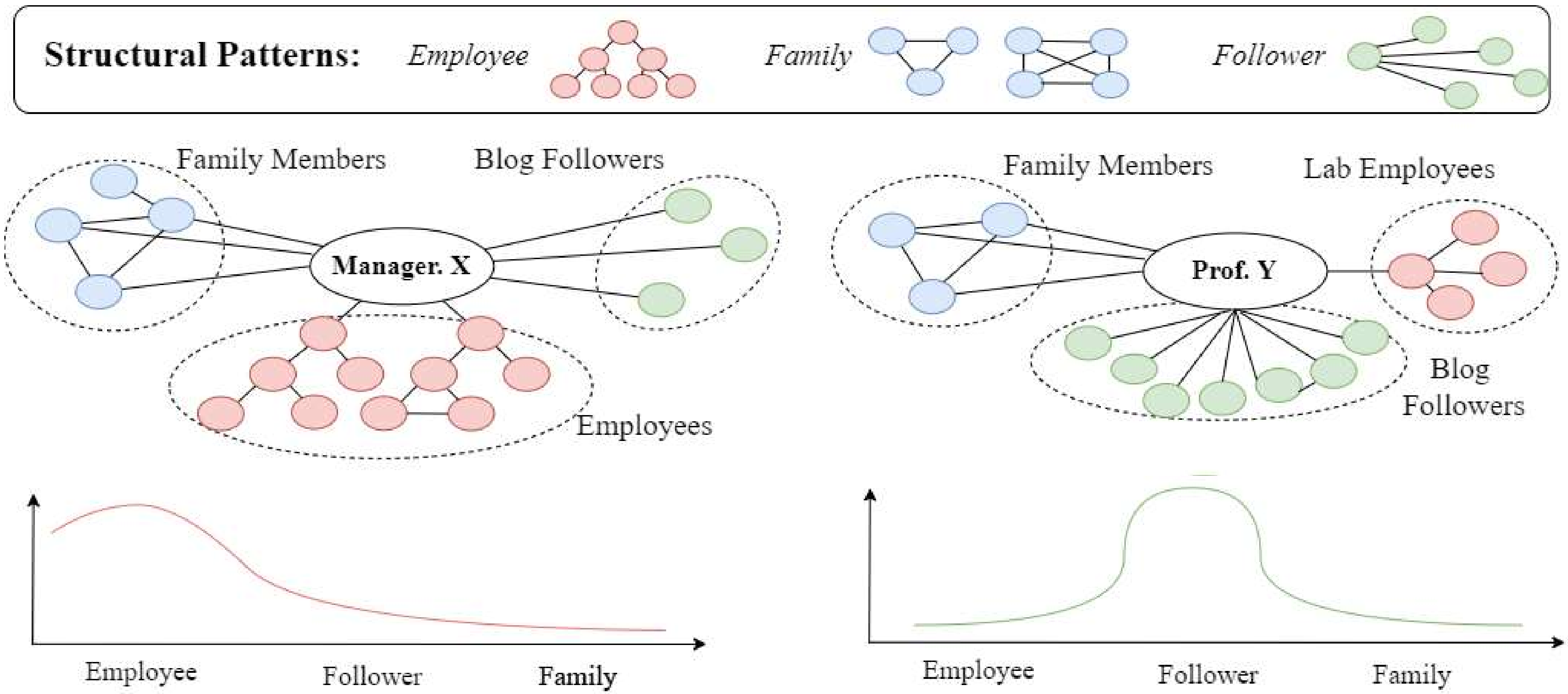
- **Can we use very deep GCNs, just as ResNet?**
  - Yes. However, even very deep GCNs are unable to learn complex structures in networks [1].
- **Alternative: Can we design new GCNs that incorporate such information?**
  - Yes. However…
  - Only **few motifs** [2] are selected — insufficient expression.
  - All **possible** structures are selected [3] — poor efficiency.



[1] Oono et al. Graph neural networks exponentially lose expressive power for node classification. In ICLR, 2020
[2] Lee, Rossi et al. Graph Convolutional Networks with Motif-based Attention. In CIKM, 2019
[3] Jin, Song et al. GraLSP: Graph Neural Networks with Local Structural Patterns. In AAAI, 2020.

Why selecting a few motifs is insufficient?

- **An Example :**

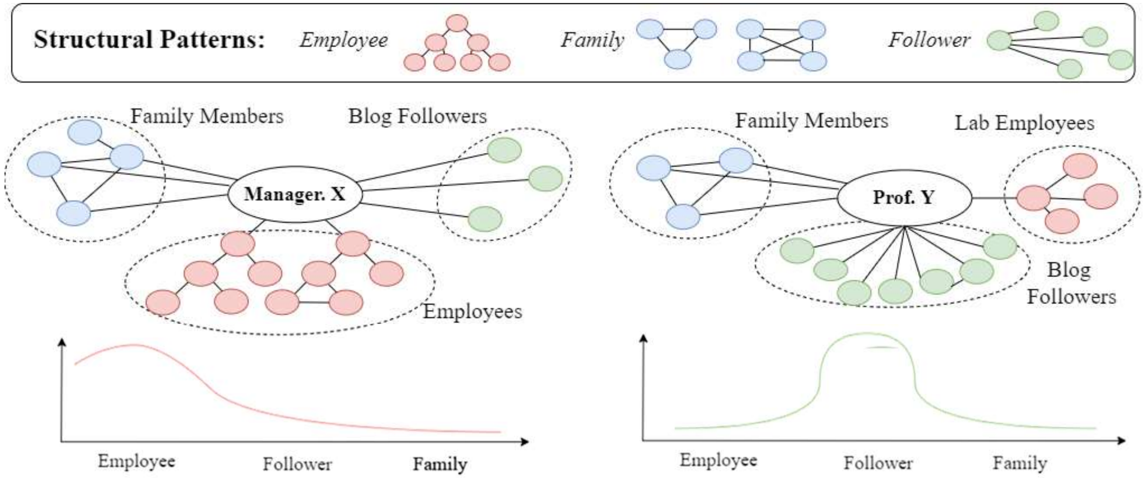


Research Goal and Challenges

- **Goal:** Design a novel GCN framework that adequately describes and models network local structures in an efficient manner, which means:
  - To consider local structures of nodes **as a whole**.
  - To be efficient, which means selecting concise and accurate representations of structures.

Why topics?

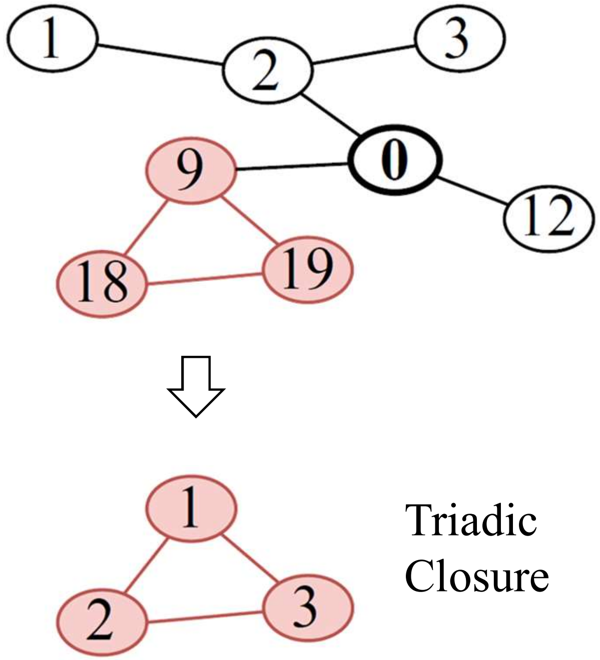
- **What are topics?**
  - In NLP (*Latent Dirichlet Allocation*), topics are defined by a collection of words, and texts are described by a collection of topics.
  - Similar?



D. Blei, A. Ng, M. I. Jordan. Latent Dirichlet Allocation, In JMLR, 2013.

Preliminaries

- **Anonymous Walks**
  - Node is represented by the first position where it appears.
  - **Example**
    - Random walk sequence: (9, 18, 19, 9)
    - Anonymous walk sequence: (1, 2, 3, 1)
    - Highly likely generated through a *triadic closure*.
  - More theoretical analysis see [1].



[1] Micali and Zhu. Reconstructing markov processes from independent and anonymous experiments. In Applied Discrete Maths, 2016.

Preliminaries

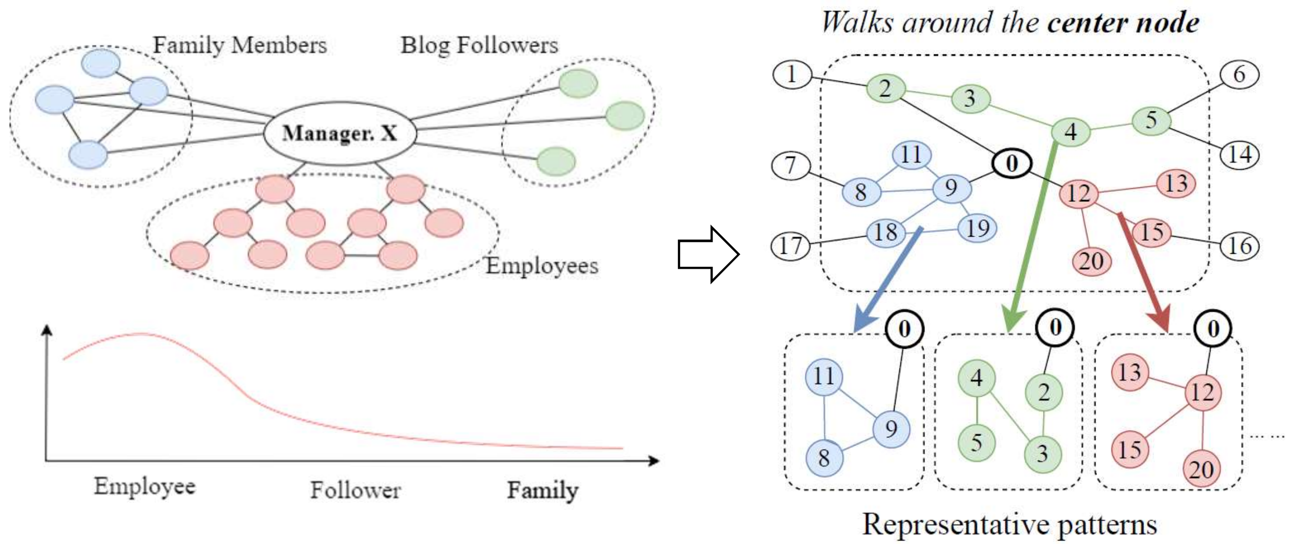
❑ Anonymous Walks

Theorem 1.

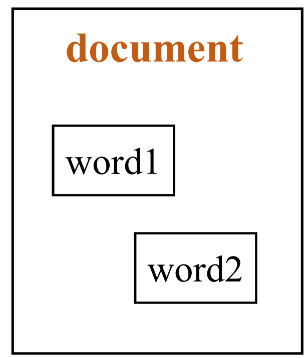
Let $B(v, r)$ be the subgraph induced by all nodes u such that $dist(v, u) \leq r$, and P_L be the distribution of anonymous walks of length L, one can reconstruct $B(v, r)$ using $(P_1, \ldots, P_L)$.

Topic Modeling for Graphs

- An analogy to topic modeling in NLP
  - **Structural patterns** (anonymous walks) ⇔ ***Words***
  - **Sets of walks** starting from each node ⇔ ***Documents***



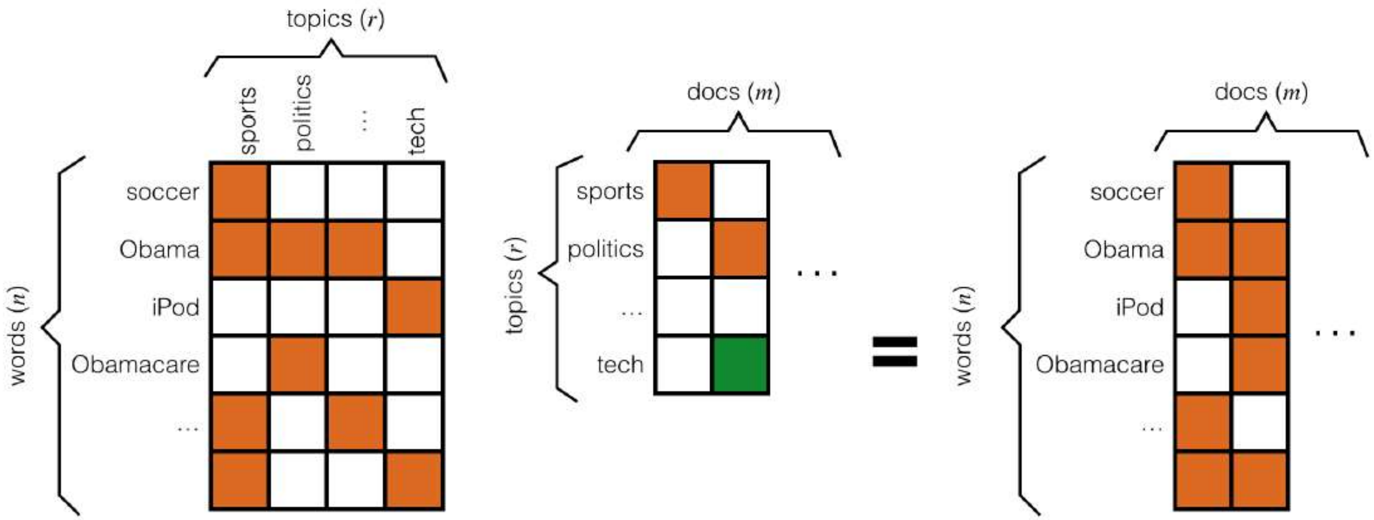
Concepts for graphs



Concepts in NLP

Topic Modeling for Graphs

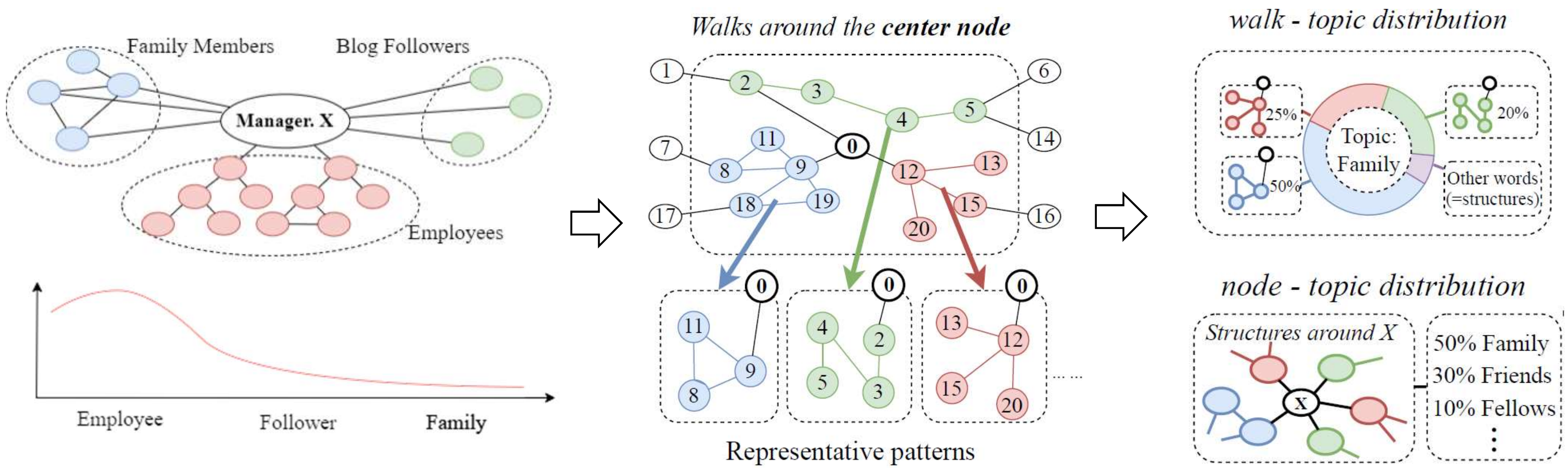
- An analogy to topic modeling in NLP
  - Parameters to learn in NLP [1]:
    - A **word-topic** distribution matrix
    - A **document-topic** distribution matrix



[1] Arora et al. Learning Topic Models — Going beyond SVD. In NIPS, 2012

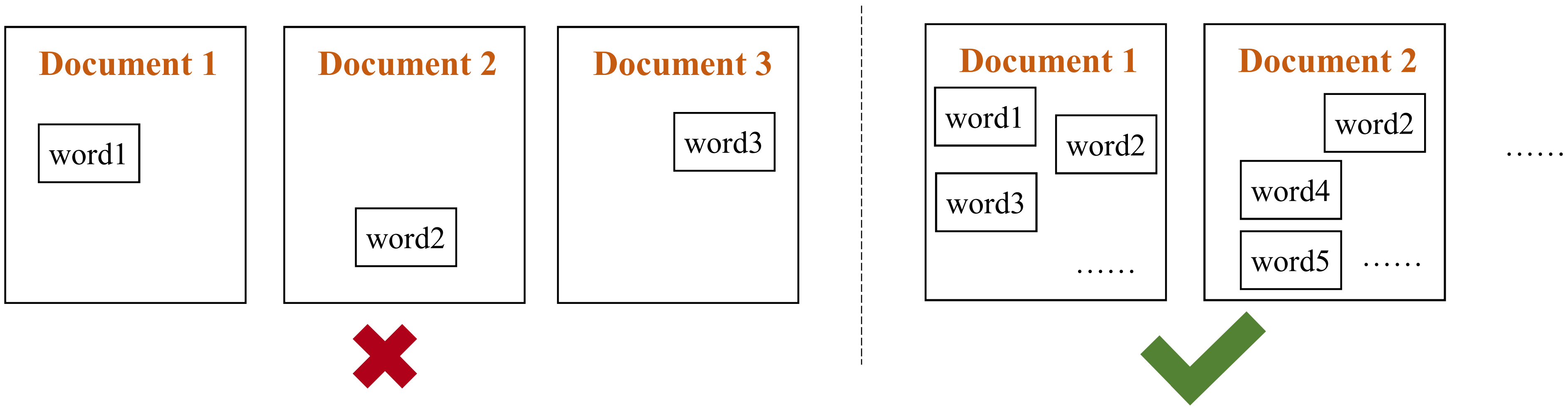
Topic Modeling for Graphs

- **Parameters to learn**
  - A **walk-topic** matrix $U \in \mathbb{R}^{K \times |\mathcal{W}_l|}$
  - A **node-topic** matrix $R \in \mathbb{R}^{|V| \times K}$



Topic Modeling for Graphs

- **Not in all cases** can we learn topic distributions in NLP
- Example:
  - Only one word in each document
  - No word co-occurrences ⇨ No topics !
- Input cases need satisfying some constraints …



Topic Modeling for Graphs

- An analogy to topic modeling in NLP

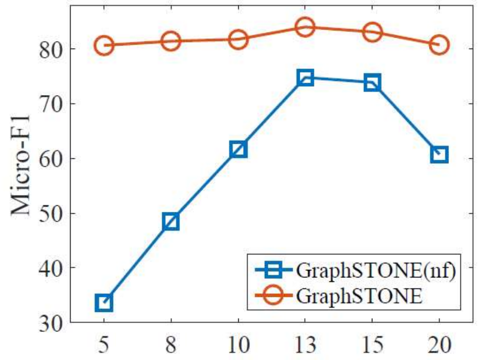
Lemma 1.

There is a polynomial-time algorithm that fits a topic model on a graph with error ϵ, if N and the length of walks l satisfy $\frac{N}{l} \geq O(\frac{b^4K^6}{\epsilon^2 p^6 \gamma^2 |V|})$.

  - For more details, see Section 3.1.2 in our paper.

- Example:
  - Performance is sensitive to **length of walks.**

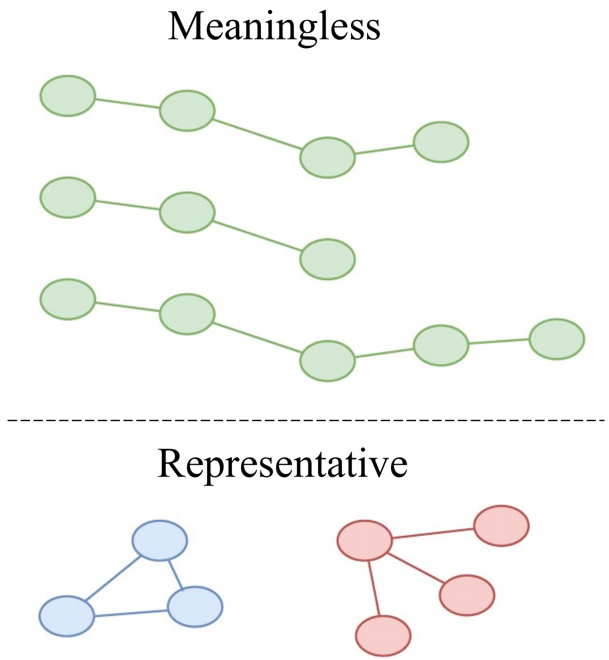
    (number of “words” in a “document”)



Graph Anchor LDA

- Selection of indicative structural patterns
  - Due to the irregularity of graphs, **large number of walk sequences** will be generated.
  - Topic model may focus on **meaningless sequences** and ignore more important structural patterns.
  - These meaningless sequences are like ***stopwords*** in NLP.

For Example:



Graph Anchor LDA

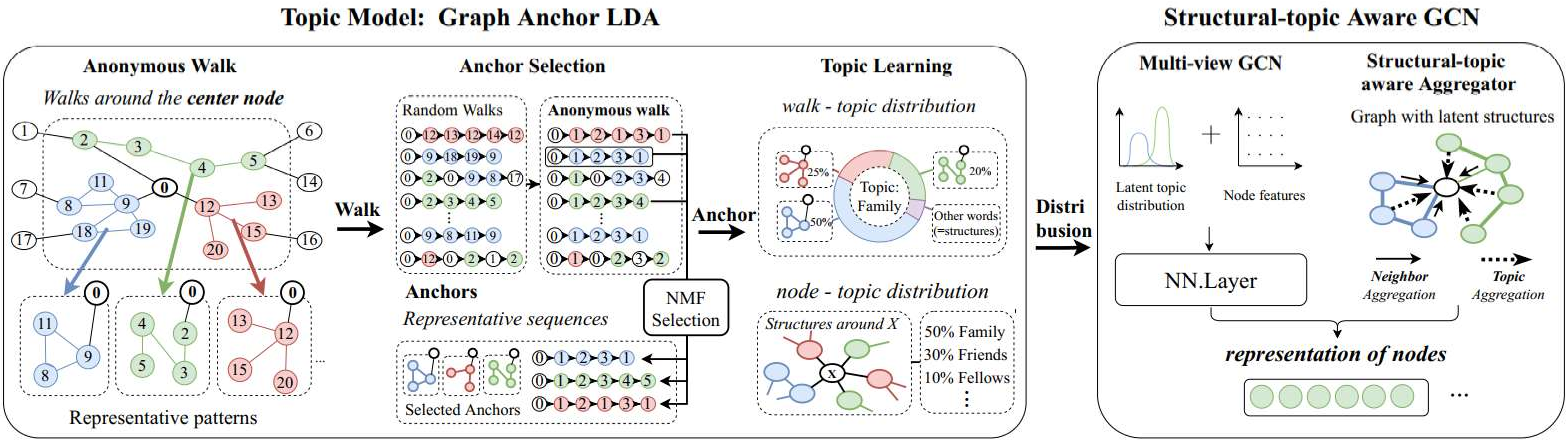
- **Anchor Selection**
  - Select indicative structures patterns based on non-negative matrix factorization (NMF) [1].
  - NMF is able to find principal components (anchors in our model).
- **Topic Learning**
  - Based on selected anchors [2]

$$\arg\min D_{KL}\left(Q_i \| \sum_{k \in \mathbf{A}} U_{ik} \operatorname{diag}^{-1}(Q\vec{1}) Q_{A_k}\right)$$

- **More theoretical analysis** and details see Section 3.1.4 in our paper.

[1] Lee et al. Learning the parts of objects by non-negative matrix factorization. In Nature, 1999
[2] Arora et al. A practical algorithm for topic modeling with provable guarantees. In ICLR, 2013

Overview of GraphSTONE



Structural-topic Aware GCN

- **Multi-view GCN**

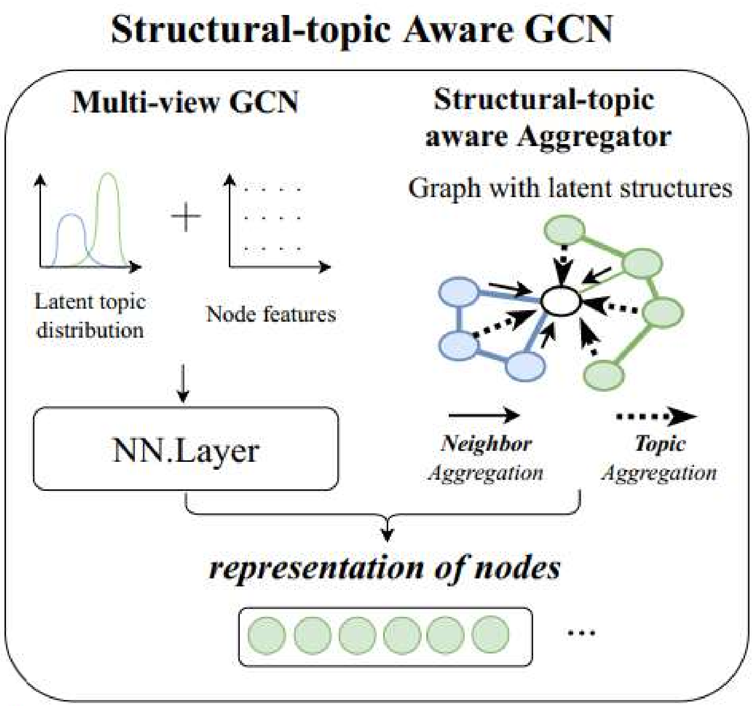
$$h_i^{(L)} = (\mathbf{W} \cdot \mathrm{ReLU}([h_{i,n}^{(L)} \otimes h_{i,s}^{(L)}]) + \mathbf{b})$$

- **Structural-topic Aware Aggregator**

$$h_i^{(k)} = \mathrm{AGGREGATE}\left(\left\{\frac{R_i^T R_j}{\sum_j R_i^T R_j} h_j^{(k-1)}, v_j \in N(v_i)\right\}\right)$$

- **Unsupervised objective function**
  - Like GraphSAGE [1]

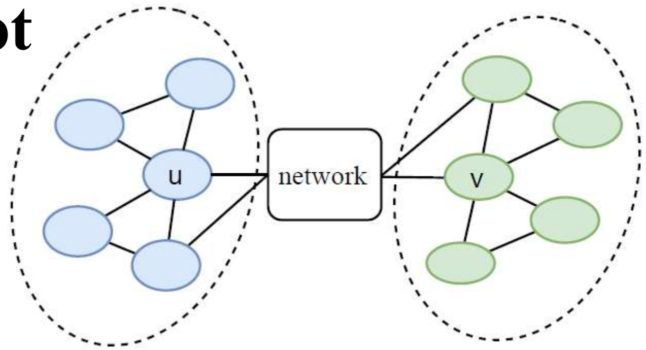
$$\mathcal{L} = -\log[\sigma(h_i^{(L)T} h_j^{(L)}] - q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log[\sigma(h_i^{(L)T} h_n^{(L)})]$$



[1] Hamilton et al. Inductive representation learning on large graphs. In NIPS, 2017

Comparison with community detection

- **Our model**
  - Focuses on distribution of local structures, i.e. …
  - **will discover structurally similar**, but **not necessarily connected** nodes
- Community detection
  - Focuses on **dense connections** [1]
- An example
  - Nodes u and v are structurally similar…
  - but belong to distinct communities

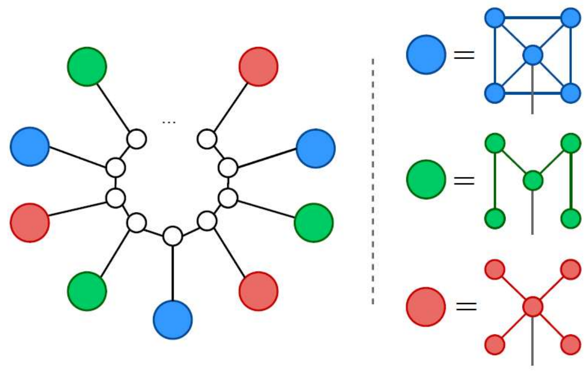


Our model

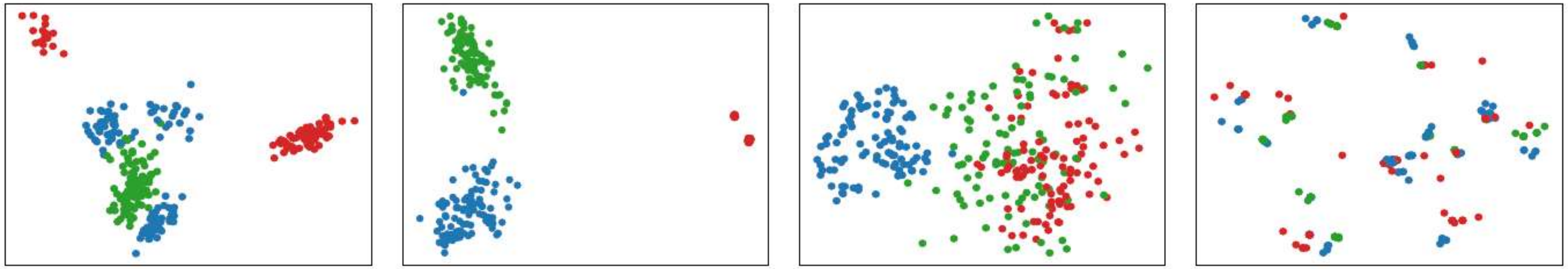
[1] Jin et al. Incorporating network embedding into markov random field for better community detection. In AAAI, 2019

Proof-of-concept Visualization

- Synthetic dataset
  - $G(n)$ with 3 structures (constituents).
- Results
  - Our model can **mark** different **structural patterns** more clearly



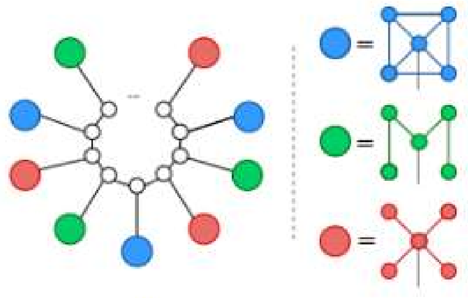
(a) Illustration of $G(n)$



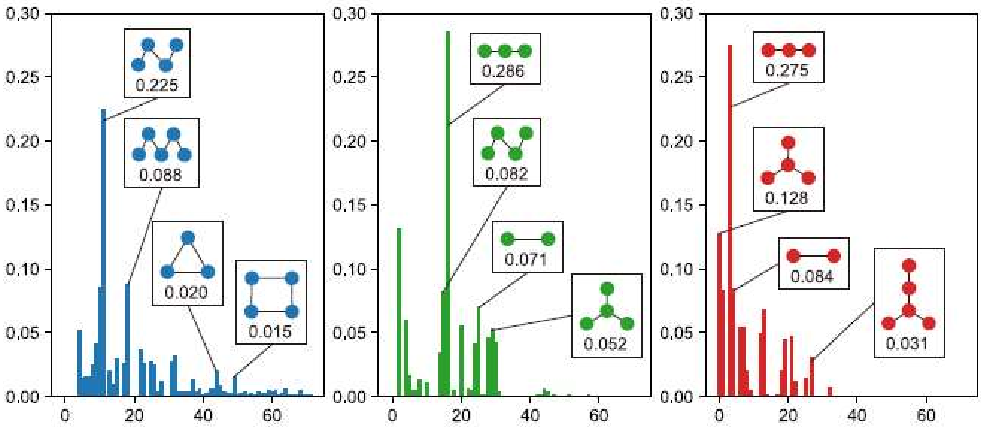
(b) Graph Anchor LDA (c) GraphSTONE (d) GraLSP (e) MNMF

Proof-of-concept Visualization

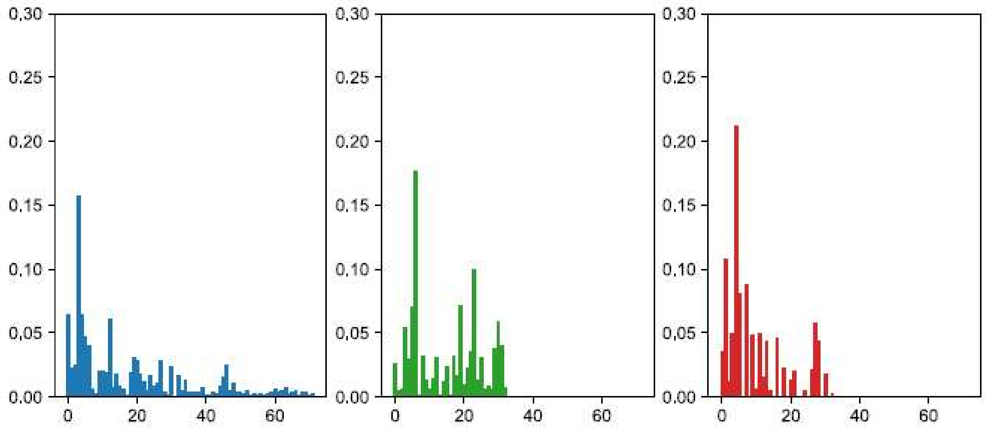
- Learned distributions
  - Distributions of local structures are **different** among 3 structural topics
  - Our model **amplifies** indicative structures within each topic



(a) Illustration of $G(n)$



(a) Walk-topic distribution by Graph Anchor LDA



(b) Walk-topic distribution by ordinary LDA

Experiments

Datasets

Datasets	Type	$\lvert V\rvert$	$\lvert E\rvert$	# Classes
Cora	Citation	2,708	5,429	7
AMiner	Social	3,121	7,219	4
Pubmed	Citation	19,717	44,338	3
PPI	Protein	14,755	228,431	121

Baselines

- Struc2Vec [Ribeiro *et al.*, 2017]
- GCN [Kipf *et al.*, 2017]
- GAT [Veličković *et al,* 2017]
- GraphSAGE [Hamilton *et al.*, 2017]
- GraLSP [Jin *et al.*, 2019]

Link Reconstruction

Input	Model	Cora		AMiner		Pubmed	
		AUC	Recall@0.5	AUC	Recall@0.5	AUC	Recall@0.5
No features	Struc2Vec	54.29	54.38	47.55	47.63	53.14	53.14
	GraLSP	66.28	66.38	65.40	65.50	57.62	57.63
	GCN	74.60	74.71	71.98	72.07	59.20	59.22
	GraphSTONE (nf)	**92.44**	**92.56**	**89.87**	**89.91**	**87.47**	**87.48**
Features	GCN	94.14	94.26	94.47	94.55	92.23	92.25
	GAT	94.66	94.78	95.24	95.34	92.36	92.38
	GraLSP	94.39	94.51	94.85	94.89	90.83	90.84
	GraphSAGE	95.30	95.42	94.92	95.02	91.52	91.54
	GraphSTONE	**96.37**	**96.70**	**95.94**	**96.06**	**94.25**	**94.27**

Table 2: Results of link reconstruction on different datasets.

- GraphSTONE is competitive against all the baselines
- Especially **in the absence of node features**

Vertex Classification

Input	Model	Cora				AMiner				Pubmed				PPI			
		Macro-f1		Micro-f1		Macro-f1		Micro-f1		Macro-f1		Micro-f1		Macro-f1		Micro-f1	
		30%	70%	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%	30%	70%
No features	Struc2Vec	17.55	18.92	29.07	31.34	23.17	21.80	36.11	38.44	31.29	31.31	41.50	41.49	12.89	13.53	40.49	40.74
	GraLSP	58.86	61.62	60.88	62.45	43.19	43.03	45.85	45.92	38.89	38.84	45.88	46.01	10.19	10.72	37.65	37.88
	GCN	11.65	11.94	32.30	32.83	14.86	16.81	41.24	42.51	35.07	36.51	46.56	47.83	8.75	9.08	36.70	37.46
	GraphSTONE (nf)	**70.25**	**71.33**	**71.73**	**72.42**	**57.11**	**56.70**	**58.21**	**58.91**	**56.87**	**58.88**	**60.47**	**60.69**	10.28	11.20	38.93	38.96
Features	GCN	79.84	81.09	80.97	81.94	65.02	67.33	64.89	66.72	76.93	77.21	76.42	77.49	12.57	12.62	40.40	40.44
	GAT	79.33	82.08	80.41	83.43	68.76	69.10	67.92	68.16	76.94	76.92	77.64	77.82	11.91	11.97	39.92	40.10
	GraLSP	82.43	83.27	83.67	84.31	68.82	70.15	69.12	69.73	**81.21**	**81.38**	**81.43**	**81.52**	11.34	11.89	39.55	39.80
	GraphSAGE	80.52	81.90	82.13	83.17	67.40	68.32	66.59	67.54	76.61	77.24	77.36	77.84	11.81	12.41	39.80	40.08
	GraphSTONE	**82.78**	**83.54**	**83.88**	**84.73**	**69.37**	**71.16**	**69.51**	**69.93**	78.61	78.87	79.53	81.03	**15.55**	**15.91**	**43.60**	**43.64**

Table 3: Macro-f1 and Micro-f1 scores of transductive node classification.

- GraphSTONE is competitive against all the baselines
- Especially **in the absence of node features**

Vertex Classification (Inductive)

- **Settings**
  - PPI dataset, including 22 separate protein graphs
  - Train all GNNs on 20 graphs, and **directly** predict on 2 test graphs
  - Test nodes are unobserved during training
- Structural topic features generalize **well across graphs**

Model	Macro-f1	Micro-f1
Struc2Vec	-	-
GCN	12.15	40.85
GAT	12.31	39.76
GraLSP	12.59	40.81
GraphSAGE	11.92	40.05
GraphSTONE	**18.14**	**46.02**

Table 4: Inductive node classification results on PPI.

Efficiency

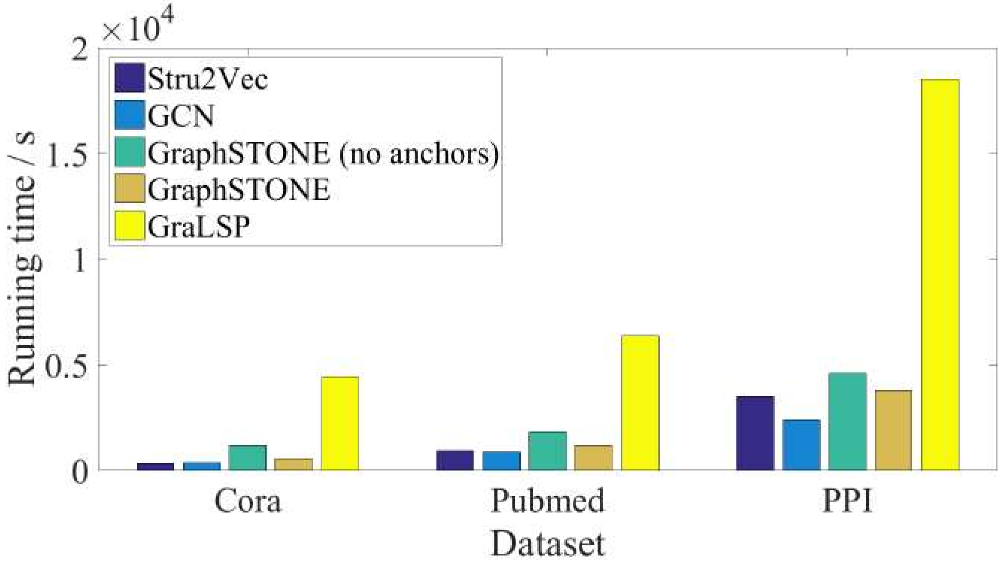


Figure 7: Running time on different datasets.

- Anchors improve efficiency
- With anchors, GraphSTONE barely takes more time than GCN

Summary

- We present **GraphSTONE**, a GCN framework that captures local structural patterns. To the best of our knowledge, it is the **first attempt** on topic models on graphs and GCNs.
- We design the Graph Anchor LDA algorithm and a multi-view GCN unifying node features with structural-topic features.
- Extensive experiments demonstrate that GraphSTONE is **competitive** against its various counterparts.

See More Details …

Paper： http://arxiv.org/abs/2006.14278

Code: https://github.com/YimiAChack/GraphSTONE

Lab: https://www.gjsong-pku.cn/