COLLEGE OF ENGINEERING
COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF MICHIGAN

GEMS LAB

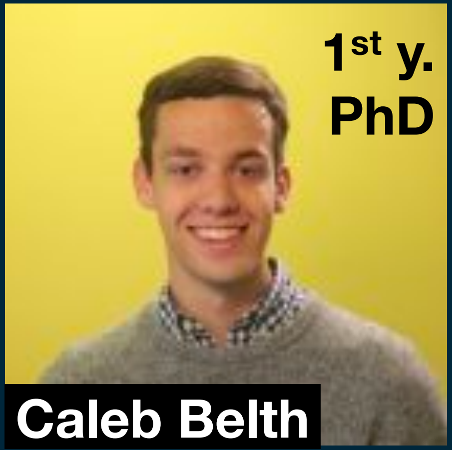
(Pocket-size) Structural Embeddings in Large-scale Networks

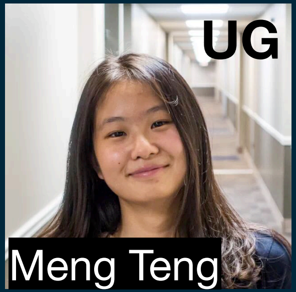
Danai Koutra

Assistant Professor, CSE

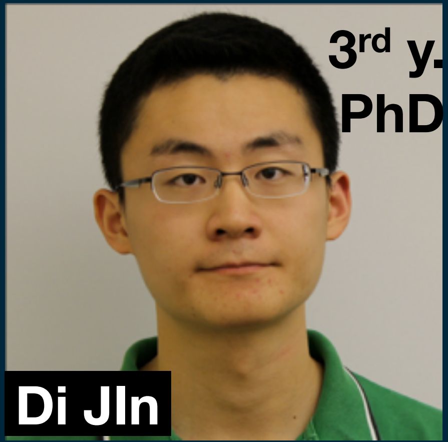
DOOCN-XII: Network Representation Learning – May 28, 2019

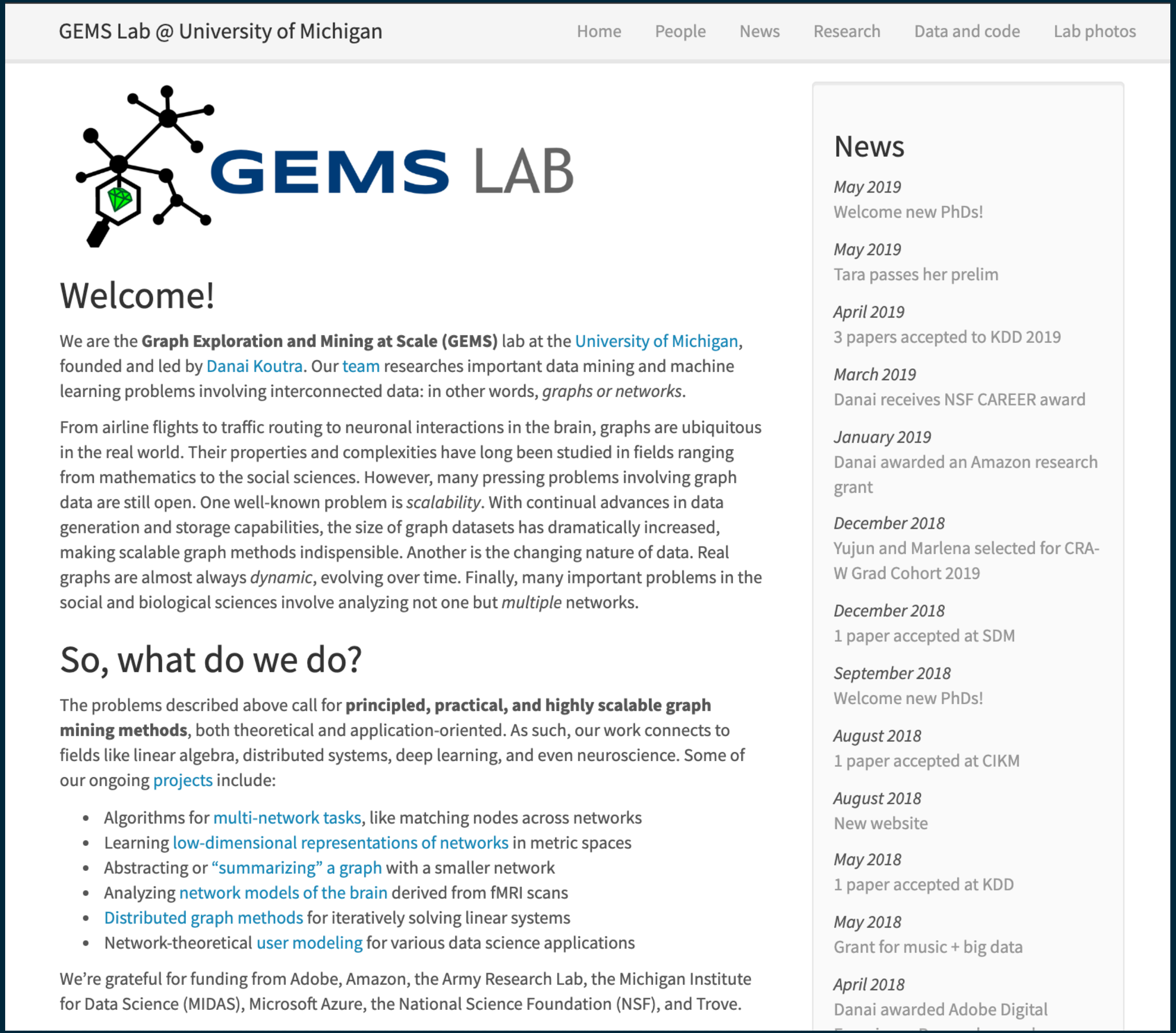
Slides at: https://bit.ly/2VVuS5B





+++




GEMS Lab @ University of Michigan

Home People News Research Data and code Lab photos

GEMS LAB

Welcome!

We are the **Graph Exploration and Mining at Scale (GEMS)** lab at the University of Michigan, founded and led by Danai Koutra. Our team researches important data mining and machine learning problems involving interconnected data: in other words, *graphs or networks*.

From airline flights to traffic routing to neuronal interactions in the brain, graphs are ubiquitous in the real world. Their properties and complexities have long been studied in fields ranging from mathematics to the social sciences. However, many pressing problems involving graph data are still open. One well-known problem is *scalability*. With continual advances in data generation and storage capabilities, the size of graph datasets has dramatically increased, making scalable graph methods indispensible. Another is the changing nature of data. Real graphs are almost always *dynamic*, evolving over time. Finally, many important problems in the social and biological sciences involve analyzing not one but *multiple* networks.

So, what do we do?

The problems described above call for **principled, practical, and highly scalable graph mining methods**, both theoretical and application-oriented. As such, our work connects to fields like linear algebra, distributed systems, deep learning, and even neuroscience. Some of our ongoing projects include:

- Algorithms for multi-network tasks, like matching nodes across networks
- Learning low-dimensional representations of networks in metric spaces
- Abstracting or "summarizing" a graph with a smaller network
- Analyzing network models of the brain derived from fMRI scans
- Distributed graph methods for iteratively solving linear systems
- Network-theoretical user modeling for various data science applications

We're grateful for funding from Adobe, Amazon, the Army Research Lab, the Michigan Institute for Data Science (MIDAS), Microsoft Azure, the National Science Foundation (NSF), and Trove.

News

May 2019
Welcome new PhDs!

May 2019
Tara passes her prelim

April 2019
3 papers accepted to KDD 2019

March 2019
Danai receives NSF CAREER award

January 2019
Danai awarded an Amazon research grant

December 2018
Yujun and Marlena selected for CRA-W Grad Cohort 2019

December 2018
1 paper accepted at SDM

September 2018
Welcome new PhDs!

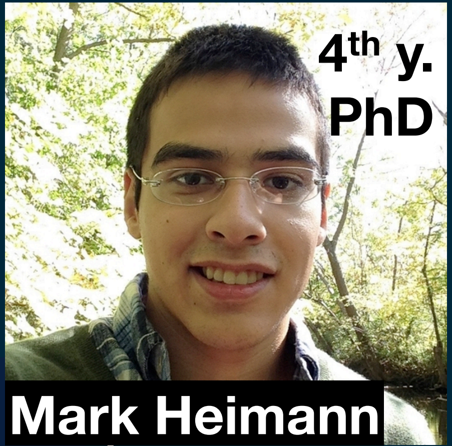
August 2018
1 paper accepted at CIKM

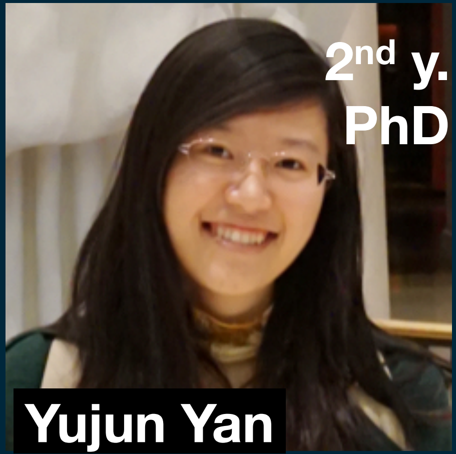
August 2018
New website

May 2018
1 paper accepted at KDD

May 2018
Grant for music + big data

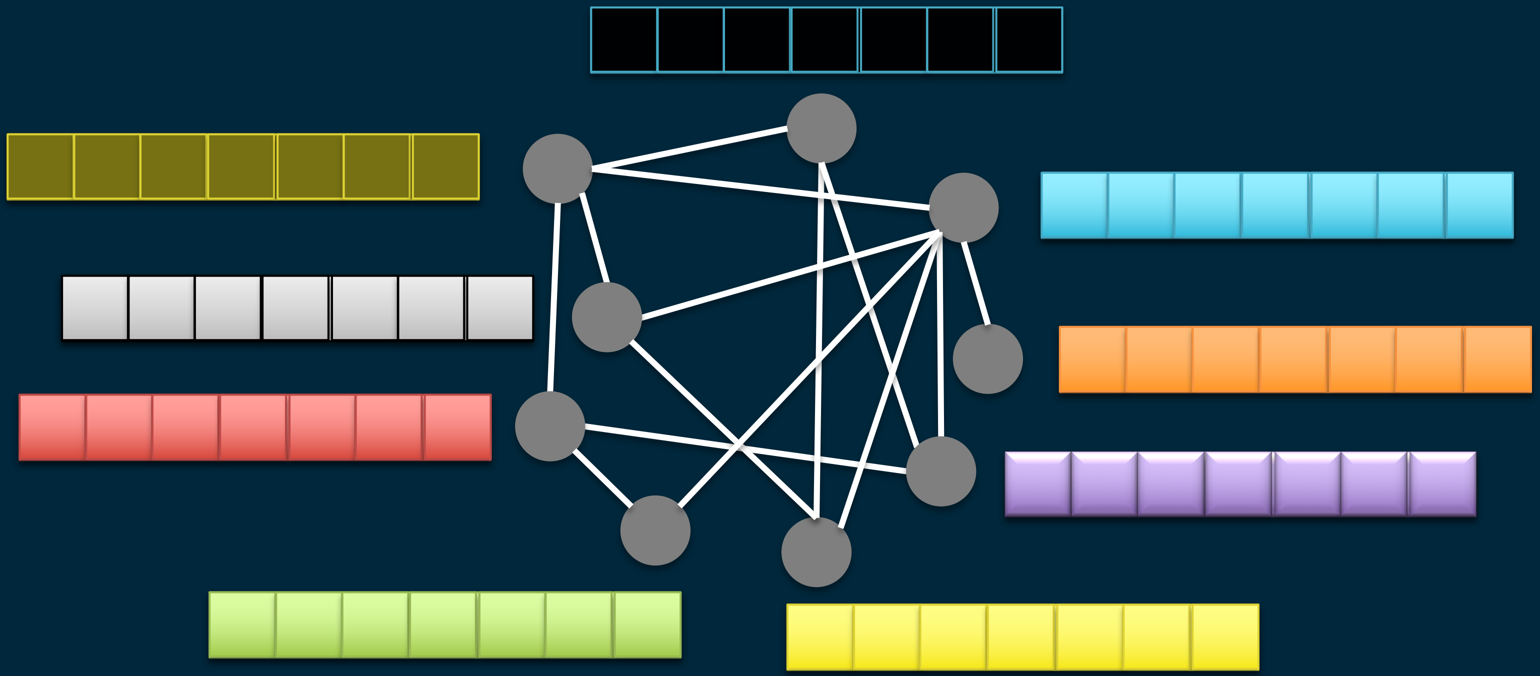
April 2018
Danai awarded Adobe Digital





Representation Learning: Goal

- Given a graph G
- Automatically learn a feature vector representation for each node



A lot of work on network representation learning!

Must-read papers on NRL/NE.

NRL: network representation learning. NE: network embedding.

Contributed by Cunchao Tu, Yuan Yao and Zhengyan Zhang.

We release OpenNE, an open source toolkit for NE/NRL. This repository p
Representation Learning) training and testing framework. Currently, the
DeepWalk, LINE, node2vec, GraRep, TADW and GCN.

Survey papers:

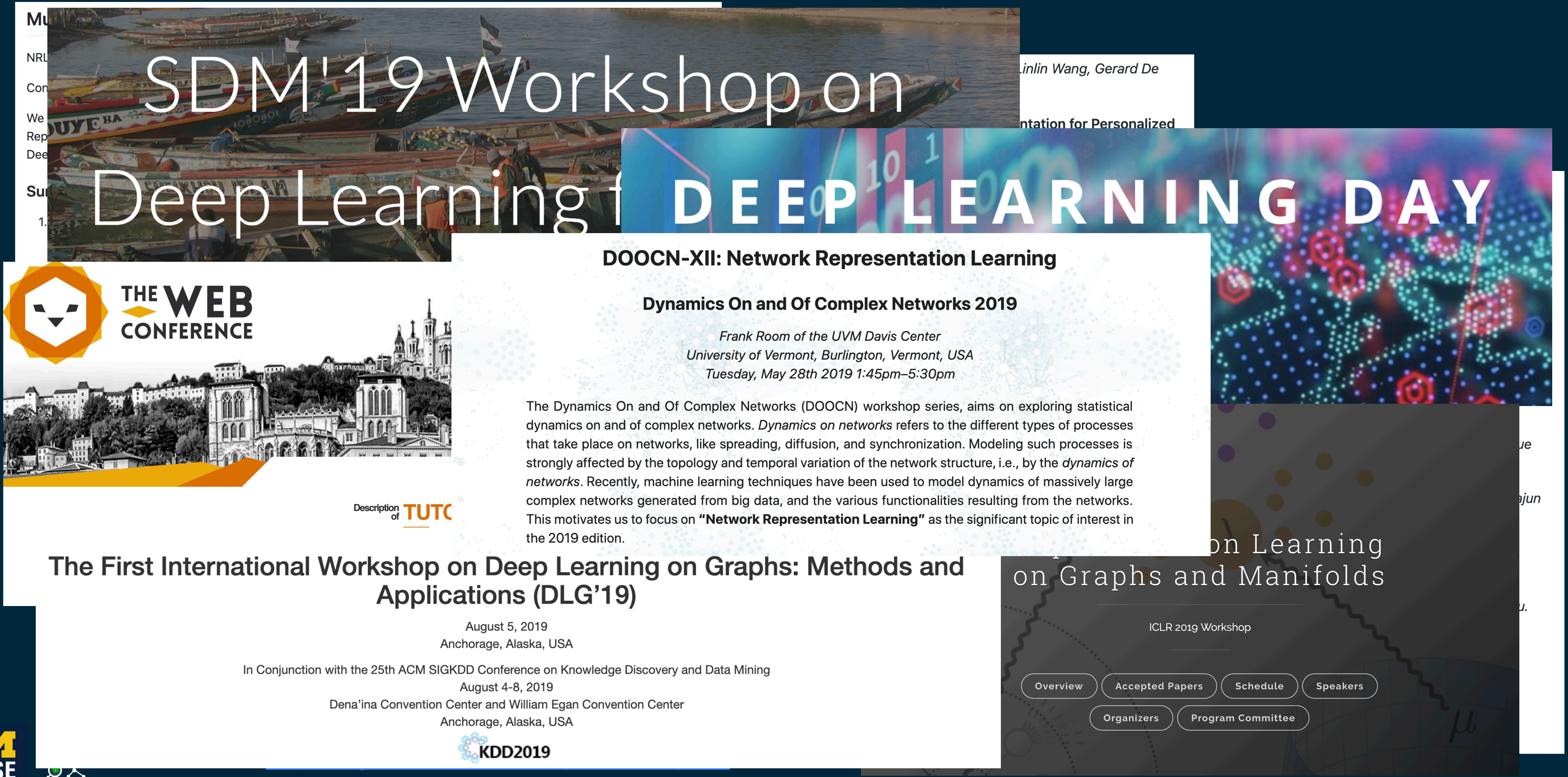
1. **Representation Learning on Graphs: Methods and Applications.** *W*
   2017. paper
2. **Graph Embedding Techniques, Applications, and Performance: A**
3. **A Comprehensive Survey of Graph Embedding: Problems, Technic**
   Zheng, Kevin Chen-Chuan Chang. 2017. paper
4. **Network Representation Learning: A Survey.** *Daokun Zhang, Jie Yi*
5. **A Tutorial on Network Embeddings.** *Haochen Chen, Bryan Perozzi,*
6. **Network Representation Learning: An Overview.(In Chinese)** *Cunc*
   2017. paper
7. ***Relational inductive biases, deep learning, and graph networks.*** *P*
   Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malino
   Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Balla
   Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas He
   Botvinick, Oriol Vinyals, Yujia Li, Razvan Pascanu. 2018. paper

54. **Link Prediction via Subgraph Embedding-Based Convex Matrix Completion.** *Zhu Cao, Linlin Wang, Gerard De melo.* AAAI 2018.
55. **Generative Adversarial Network based Heterogeneous Bibliographic Network Representation for Personalized Citation Recommendation.** *J. Han, Xiaoyan Cai, Libin Yang.* AAAI 2018.
56. **DepthLGP: Learning Embeddings of Out-of-Sample**
    Zhu. AAAI 2018. paper
57. **Structural Deep Embedding for Hyper-Networks.** *Ke*
    paper
58. **TIMERS: Error-Bounded SVD Restart on Dynamic Ne**
    Zhu. AAAI 2018. paper
59. **Community Detection in Attributed Graphs: An Emb**
    Zhang. AAAI 2018.
60. **Bernoulli Embeddings for Graphs.** *Vinith Misra, Sumi*
61. **Distance-aware DAG Embedding for Proximity Searc**
    Zhou Zhao, Fanwei Zhu, Kevin Chen-Chuan Chang, M
62. **GraphGAN: Graph Representation Learning with Ge**
    Wang, MIAO ZHAO, Weinan Zhang, Fuzheng Zhang, X
63. **HARP: Hierarchical Representation Learning for Net**
    AAAI 2018. paper code
64. **Representation Learning for Scale-free Networks.** *R*
    2018. paper
65. **Social Rank Regulated Large-scale Network Embed**
    2018. paper

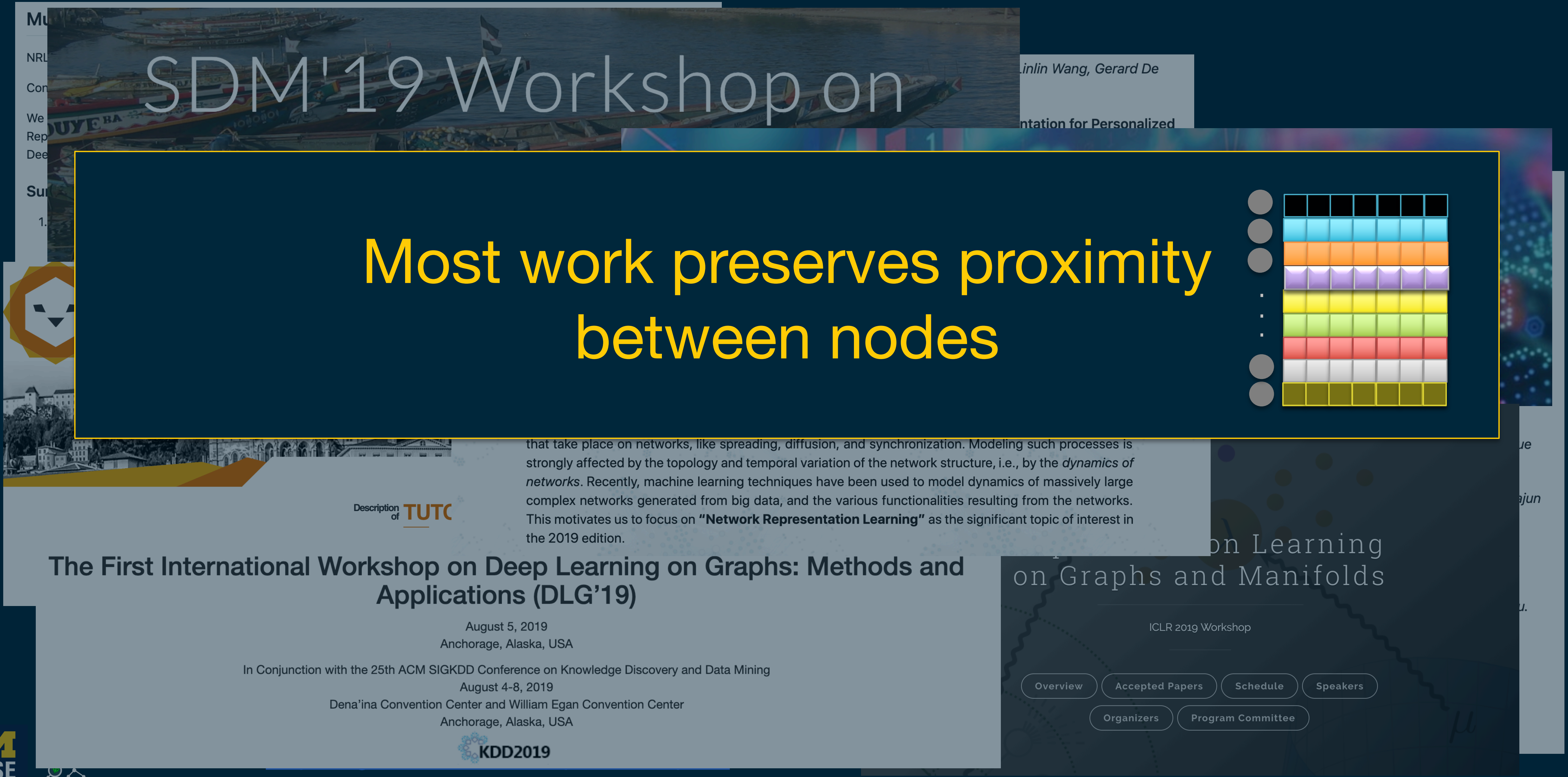
101. **Integrative Network Embedding via Deep Joint Reconstruction.** *Di Jin, Meng Ge, Liang Yang, Dongxiao He, Longbiao Wang, Weixiong Zhang.* IJCAI 2018.
102. **Scalable Multiplex Network Embedding.** *Hongming Zhang, Liwei Qiu, Lingling Yi, Yangqiu Song.* IJCAI 2018. paper
103. **Adversarially Regularized Graph Autoencoder for Graph Embedding.** *Shirui Pan, Ruiqi Hu, Guodong Long, Jing Jiang, Lina Yao, Chengqi Zhang.* IJCAI 2018.
104. **Dynamic Network Embedding : An Extended Approach for Skip-gram based Network Embedding.** *Lun Du, Yun Wang, Guojie Song, Zhicong Lu, Junshan Wang.* IJCAI 2018.
105. **Discrete Network Embedding.** *Xiaobo Shen, Shirui Pan, Weiwei Liu, Yew-Soon Ong, Quan-Sen Sun.* IJCAI 2018.
106. **Deep Attributed Network Embedding.** *Hongchang Gao, Heng Huang.* IJCAI 2018.
107. **Active Discriminative Network Representation Learning.** *Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, Yue Hu.* IJCAI 2018.
108. **ANRL: Attributed Network Representation Learning via Deep Neural Networks.** *Zhen Zhang, Hongxia Yang, Jiajun Bu, Sheng Zhou, Pinggang Yu, Jianwei Zhang, Martin Ester, Can Wang.* IJCAI 2018.
109. **Feature Hashing for Network Representation Learning.** *Qixiang Wang, Shanfeng Wang, Maoguo Gong, Yue Wu.* IJCAI 2018.
110. **Constructing Narrative Event Evolutionary Graph for Script Event Prediction.** *Zhongyang Li, Xiao Ding, Ting Liu.* IJCAI 2018. paper code
111. **Deep Inductive Network Representation Learning.** *Ryan A. Rossi, Rong Zhou, Nesreen K. Ahmed.* WWW 2018. paper
112. **A Unified Framework for Community Detection and Network Representation Learning.** *Cunchao Tu, Xiangkai Zeng, Hao Wang, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun, Bo Zhang, Leyu Lin.* TKDE 2018. paper

https://github.com/thunlp/NRLPapers

A lot of work on network representation learning!

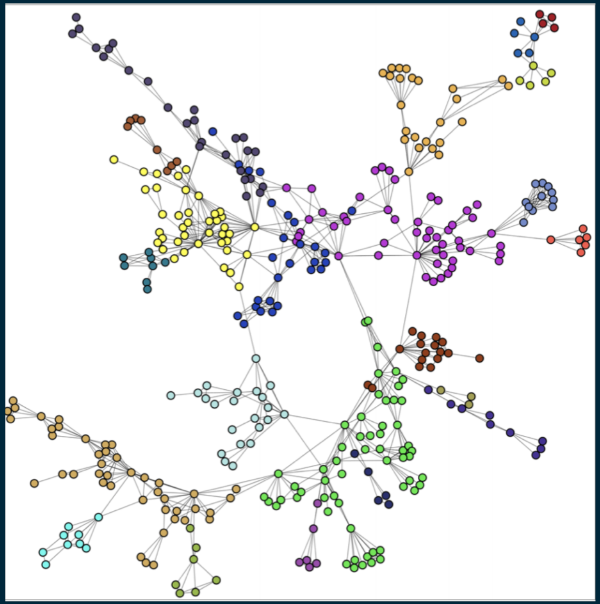


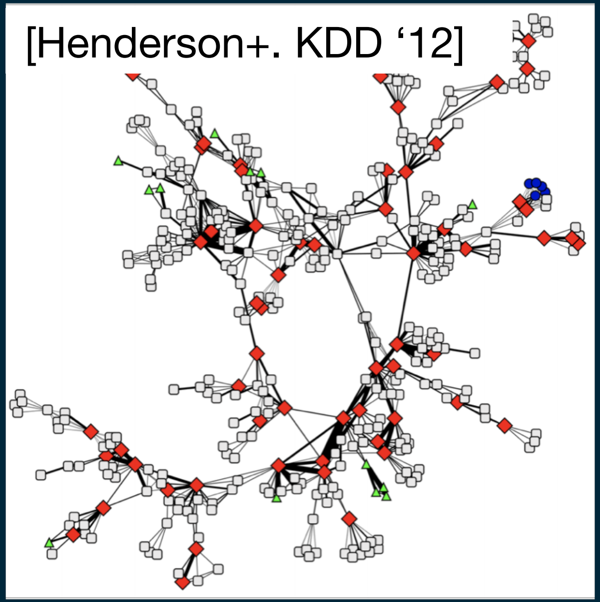
A lot of work on network representation learning!



Most work preserves proximity between nodes

Proximity vs. Structural Similarity





Find similar nodes in the same part of the network

Useful for link prediction, clustering, classification assuming homophily

[Grover+ '16; Perozzi+ '14, …]

Find nodes with similar roles all over the network

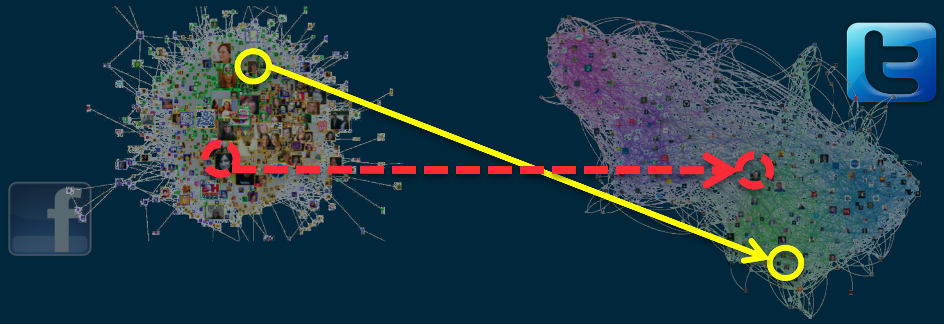
Useful for role-based classification, transfer learning, …

[Ribeiro+ '17; Donnat+ '18]

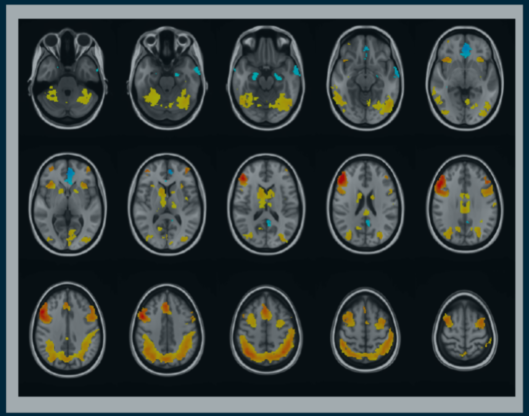
Sometimes structural similarity is more appropriate than proximity

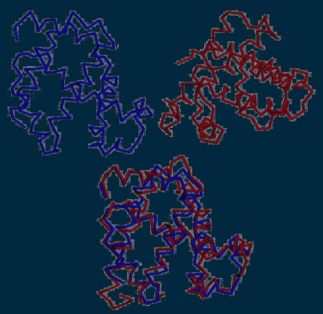
Multiple networks

Alignment or matching



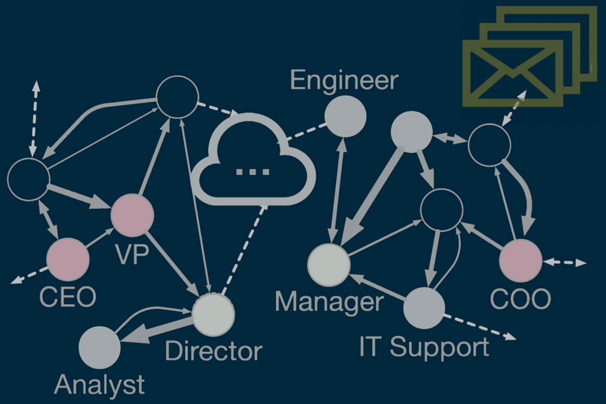
Graph similarity / classification



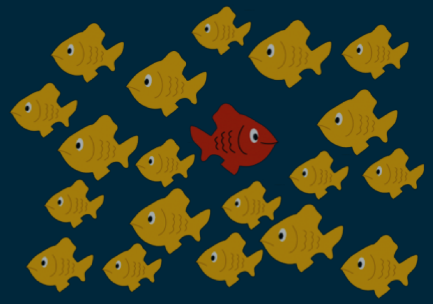


Single network

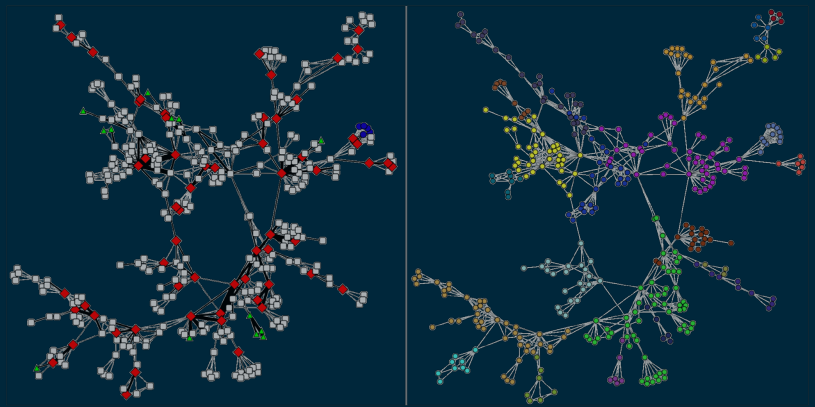
Node classification



Anomaly detection



Roles vs. communities



GEMS LAB

[Henderson+, RolX; KDD'12]

What we've found to be powerful…

- Histogram representations as a way to encode neighborhood information (instead of RWR)
  - ✧ Capture structural properties or features/attributes that generalize

- (Implicit) Matrix factorization instead of SGNS
  - ✧ Removes randomness
  - ✧ Speed / scalability

[Qiu, Jiezhong, et al. "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec." WSDM '18]

Talk Outline: Structural Embeddings for…

- Cross-network tasks [ACM CIKM’18]
  - ✧ Node (role) classification [ACM KDD’19]

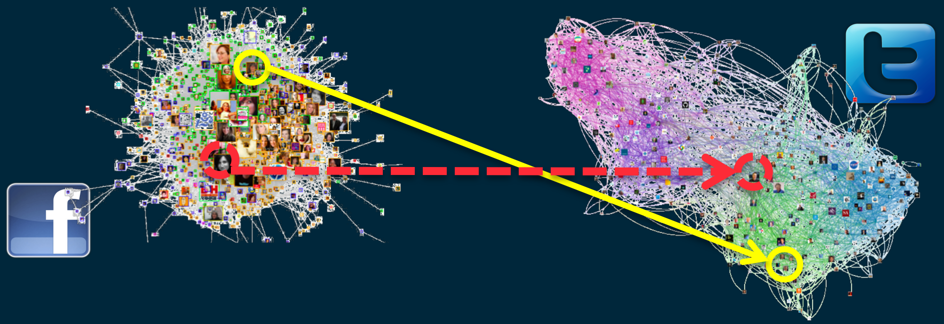
- Latent summarization [ACM KDD’19]

Based on:

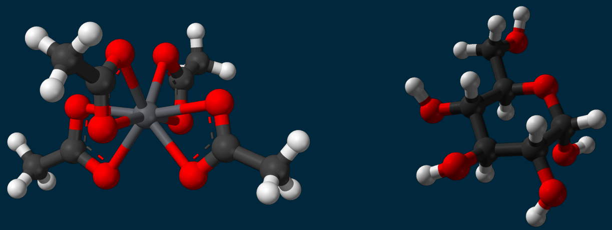
- M. Heimann, H. Shen, T. Safavi, D. Koutra. REGAL: Representation Learning-based Graph Alignment. ACM CIKM’18.
- D. Jin*, M. Heimann*, T. Safavi, M. Wang, W. Lee, L. Snider, D. Koutra. Smart Roles: Inferring Professional Roles in Email Networks. ACM KDD’19.
- D. Jin, R. Rossi, E. Koh, S. Kim, A. Rao, D. Koutra. Latent Network Summarization. ACM KDD’19.
- Y. Liu, T. Safavi, A. Dighe, D. Koutra. Graph Summarization Methods and Applications: A Survey. ACM Computing Surveys 2018.
- D. Jin, M. Heimann, R. Rossi, D. Koutra. node2bits: Compact Time- and Attribute-aware Node Representations for User Stitching. Arxiv 1904.08572
- Y. Yan, J. Zhu, Marlena Duda, Eric Solarz, Chandra Sripada, Danai Koutra. GroupINN: Grouping-based Interpretable Neural Network-based Classification of Limited, Noisy Brain Data. ACM KDD’19.

Task: Network Alignment

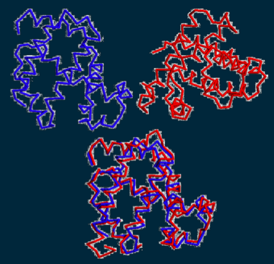
entity resolution (link user accounts)



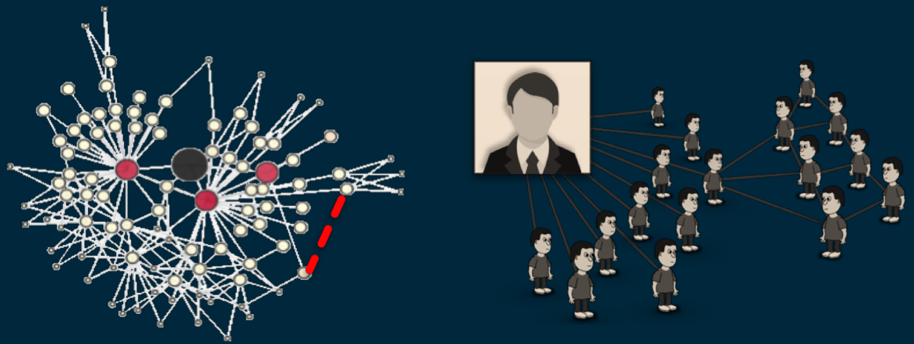
chemical compound comparison



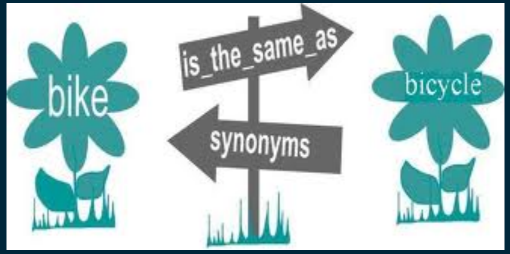
protein-protein alignment

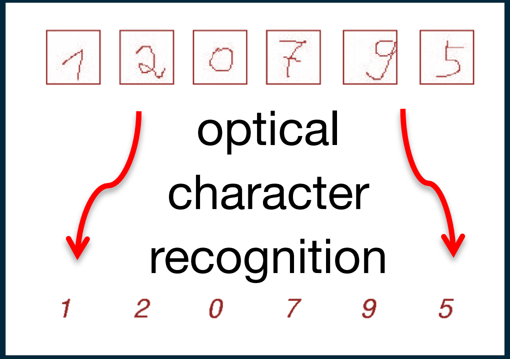


link prediction &viral marketing

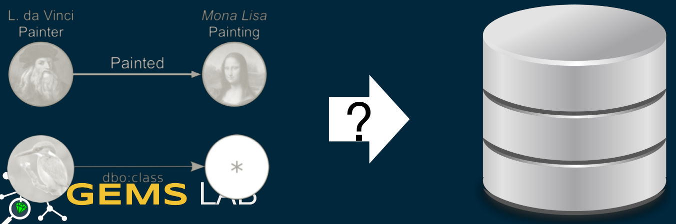


IR: synonym extraction





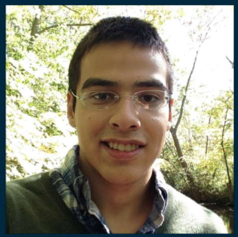
structure matching in DB



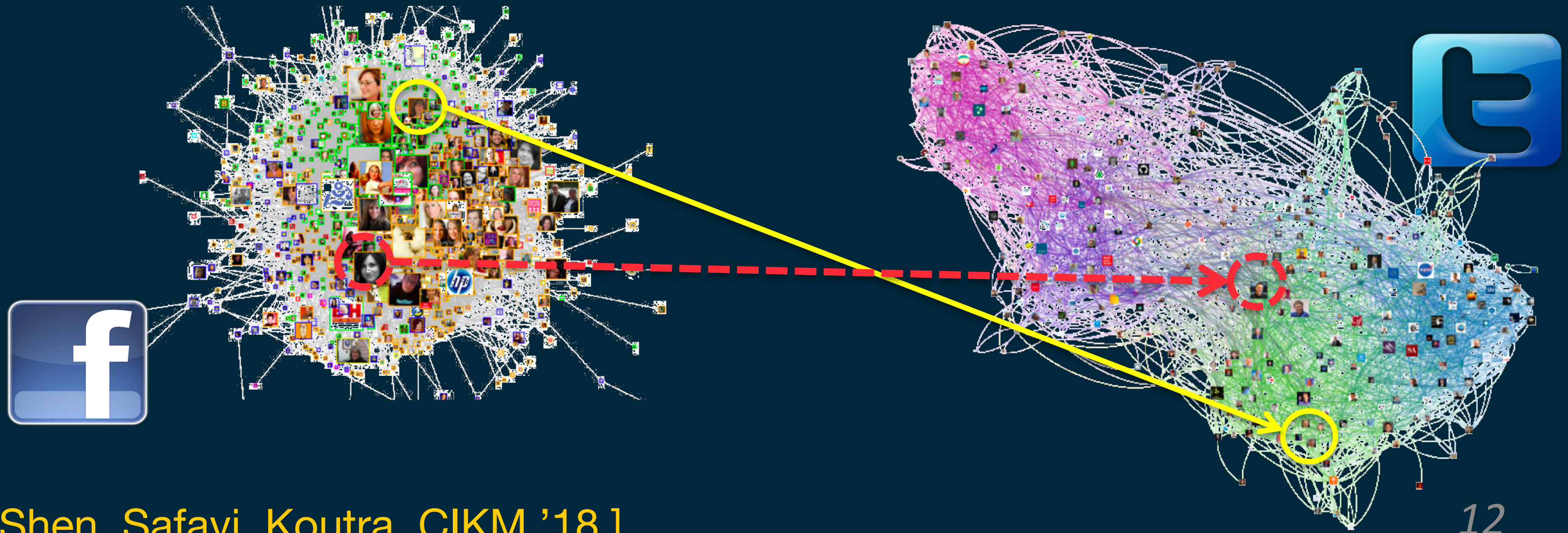
wiki translation

… and many
more applications

Network Alignment



- Given: >=2 unweighted, undirected, potentially labeled graphs
- Find: the correspondence between their nodes
    - Efficiently
    - Using node embeddings



Traditional vs. Proposed Approach

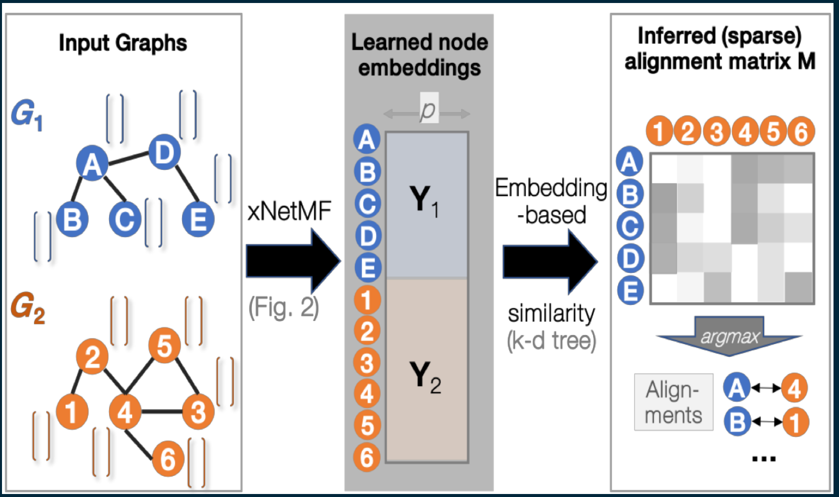
- Classic optimization (+ variants)

$$\min_P \|\mathbf{PAP}^T - \mathbf{B}\|_F$$

- Potential drawbacks
  - (-) Computationally expensive
    - ✧ e.g. $O(n^3)$ Hungarian algorithm
  - (-) 1-to-1 or hard mappings
  - (-) Require different formulation for attributed graphs, different sizes

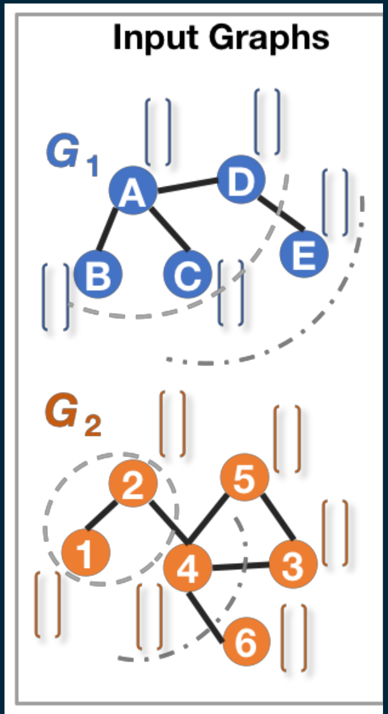
[Umeyama '88]; [Bayati+ '09]; [**Koutra**+ ICDM '13]; [Zhang+ '16] [Singh+ '08]; [Klau+ '09]; [Zhang+ '15]; [Heimann, Lee+ '18] …

- Our idea: match nodes with similar (learned) embeddings
- Challenges:
  - ✧ Comparability of embeddings across networks
  - ✧ Scalability



[Heimann, Shen, Safavi, Koutra. CIKM '18]

REGAL: Graph Alignment Framework



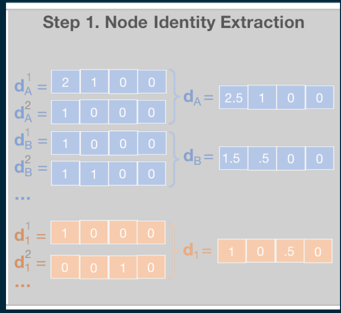
Idea 1: capture structure + labels for comparability

Most embedding methods effectively factorize a similarity matrix [Qiu+'18]

xNetMF node embeddings

Idea 2: implicit matrix factorization (Nystrom low-rank) for scalability

Step 1: Comparable Node Identity

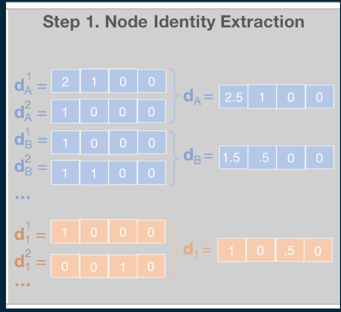


- Proximity to other nodes
  - ✧ Common for single-network tasks
  - ✧ Not *comparable* across networks
- Structural Identity
  - ✧ Used for transfer learning in graphs [Henderson+ ’12]
- Attribute Information
  - ✧ Used for graph alignment [Zhang+ ’16]

Use *node-ID invariant* quantities for cross-network comparison

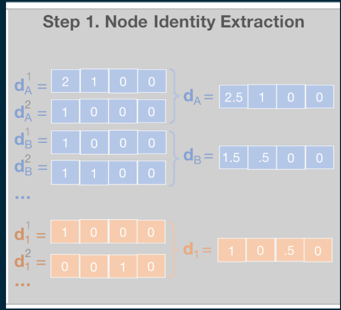
[Henderson, Keith, et al. "Rolx: structural role extraction & mining in large graphs." KDD 2012]
[Zhang, Si, and Hanghang Tong. "Final: Fast attributed network alignment." KDD 2016]

Step 1: Structural Identity Intuition



- Requirement: comparability
- Solution: Capture degrees of neighbors
  - ✧ Typical Assumption: aligning nodes have similar degrees
  - ✧ Used in structural node representation learning (struc2vec)

[**Koutra**, Tong, Lubensky. "Big-align: Fast bipartite graph alignment." ICDM '13]
[Ribeiro, Saverese, Figueiredo. "struc2vec: Learning node representations from structural identity." KDD '17]
[**Koutra**, Vogelstein, Faloutsos. "Deltacon: A principled massive-graph similarity function." SDM '13]

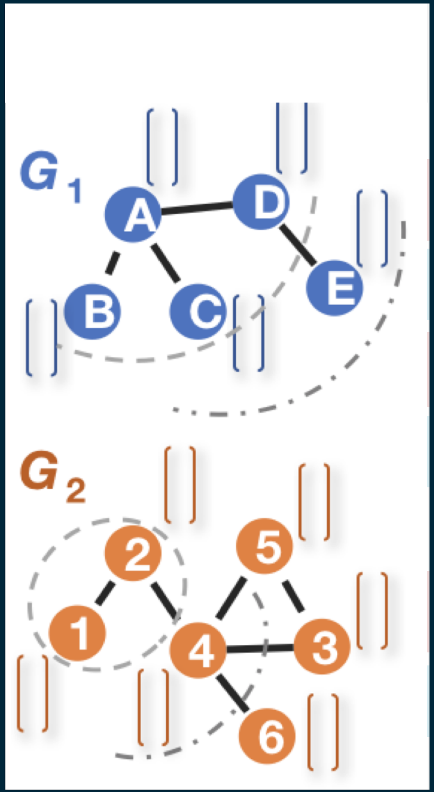


Step 1: Node Identity Extraction

- Requirement: comparability
- Solution: Degree histograms of the k-hop neighbors
  - ✧ Naive approach: j^{th} entry is # neighbors with degree j
  - ✧ Robust & compact approach: logarithmic binning

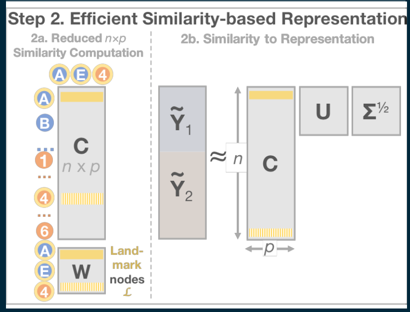
$$\mathbf{d}_u = \sum_{k=1}^{K} \delta^{k-1} \mathbf{d}_u^k$$

combine across K hops — discount distant hops



K = 2 hops, discount δ = 0.5, no logarithmic binning

Step 2: Node Similarity Representation



- Requirement: scalability
  - ✧ Avoid expensive RW
- Solution: matrix factorization
  - ✧ Most embedding methods effectively factorize a similarity matrix [Qiu+'18]
  - ✧ Cross-network similarity matrix **S** from node identities (+ attributes)

$$\mathbf{S}_{uv} = \mathrm{sim}(u, v) = \exp\left[-\gamma_s \cdot \|\mathbf{d}_u - \mathbf{d}_v\|_2^2 - \gamma_a \cdot \mathrm{dist}(\mathbf{f}_u, \mathbf{f}_v)\right]$$

structural distance

attribute distance
f: attribute vectors

M CSE GEMS LAB [Heimann, Shen, Safavi, Koutra. CIKM '18.]

[Qiu, et al. "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec." WSDM '18]

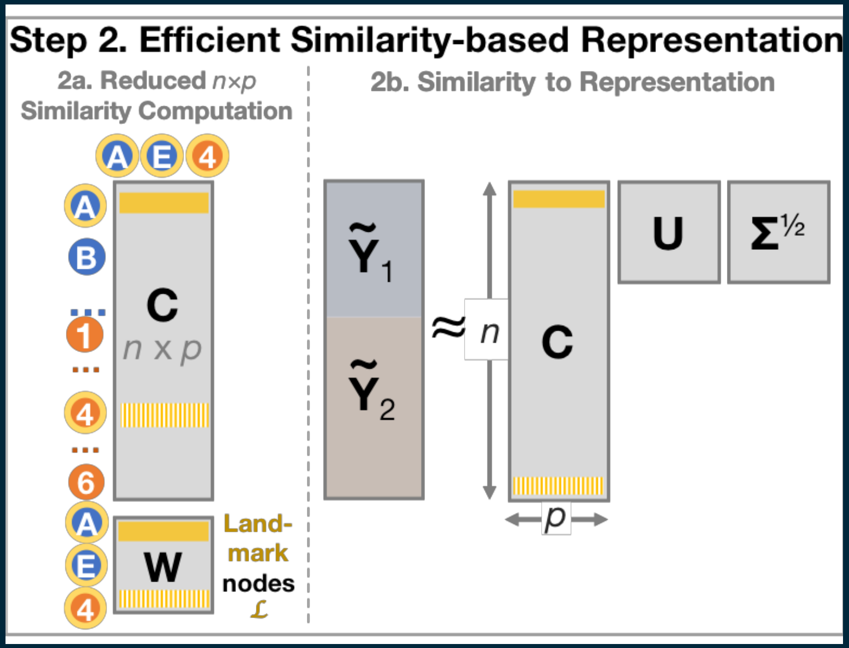
Step 2: Node Similarity Representation

- Requirement: scalability
  - ✧ Avoid expensive RW
- Solution: implicit matrix factorization
  - ✧ Based on the Nystrom low-rank approximation

THEOREM 3.1. *Given graphs $G_1(\mathcal{V}_1, \mathcal{E}_1)$ and $G_2(\mathcal{V}_2, \mathcal{E}_2)$ with $n \times n$ joint combined structural and attribute-based similarity matrix $\mathbf{S} \approx \mathbf{Y}\mathbf{Z}^T$, its node embedding matrix $\mathbf{Y}$ can be approximated as*

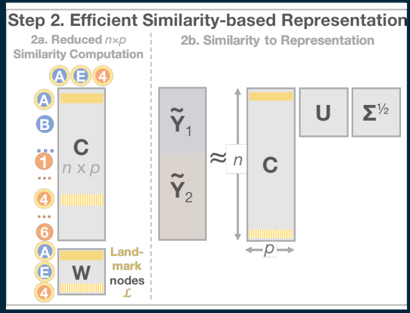
$$\tilde{\mathbf{Y}} = \mathbf{C}\mathbf{U}\mathbf{\Sigma}^{1/2},$$

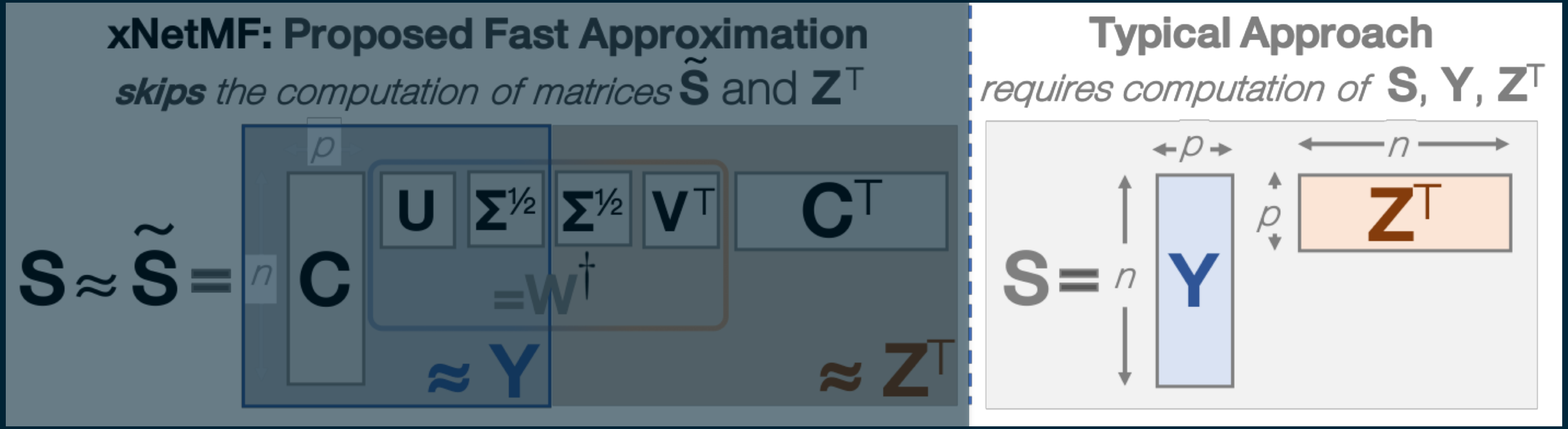
where $\mathbf{C}$ is the $n \times p$ matrix of similarities between the n nodes and p randomly chosen landmark nodes, and $\mathbf{W}^{\dagger} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}$ is the full rank singular value decomposition of the pseudoinverse of the small $p \times p$ landmark-to-landmark similarity matrix $\mathbf{W}$.



[Qiu, et al. "Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec." WSDM '18]
[Drineas and Mahoney. "On the Nyström method for approximating a Gram matrix for improved kernel-based learning." JMLR '05]

Step 2: Contrast to Typical Approach

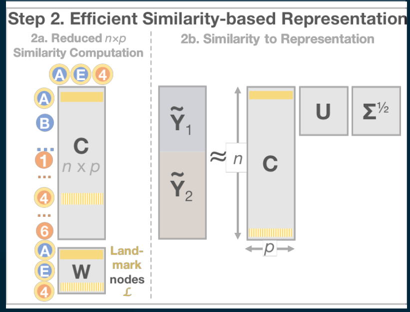


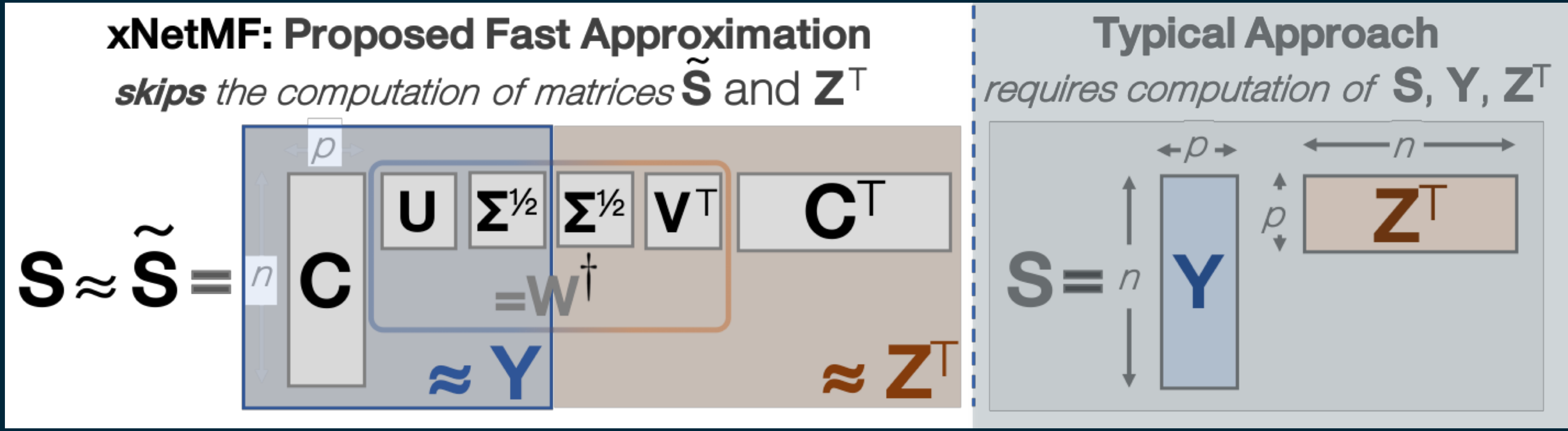


+ Exact factorization of Nystrom low-rank approximation
+ Decomposition known
+ O(np) similarities needed, for p landmarks

- Approximate factorization of exact similarity matrix
- Decomposition learned
- O(n^2) similarities + time for full factorization

Step 2: Contrast to Typical Approach



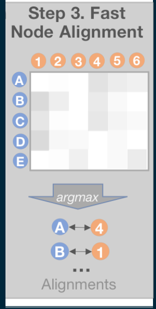


+ Exact factorization of Nystrom low-rank approximation
+ Decomposition known
+ O(np) similarities needed, for p landmarks

- Approximate factorization of exact similarity matrix
- Decomposition learned
- O(n^2) similarities + time for full factorization

GEMS LAB

[Heimann, Shen, Safavi, Koutra. CIKM '18.]



Step 3: Fast Embedding Matching

- Given: structural embeddings of nodes in G_1 and G_2
- Find: the node correspondence

- Requirement: scalability
  - ✧ avoid computing all pairwise node embedding comparisons
- Solution: use a *k-d* tree to find top-α most similar embeddings
  - ✧ Can find “soft” or “hard” alignments

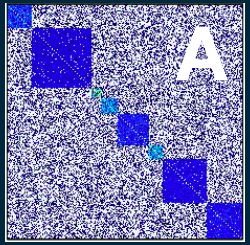
Simple greedy approach, but works well with *comparable* features

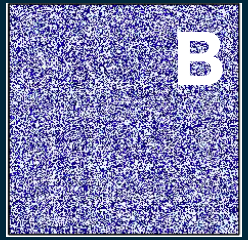
Experiments: Baselines & Setup

- Baselines: Classic, spectral and optimization-based alignment methods
  - ✧ NetAlign, FINAL, IsoRank, Klau
- Our embedding-based methods
  - ✧ REGAL
  - ✧ REGAL-node2vec (node2vec + *k-d* tree)
  - ✧ REGAL-struc2vec (struc2vec + *k-d* tree)

https://github.com/GemsLab/REGAL

- Setup: Align graphs with adj matrices **A** and **B** = **PAP**^{T} + noise

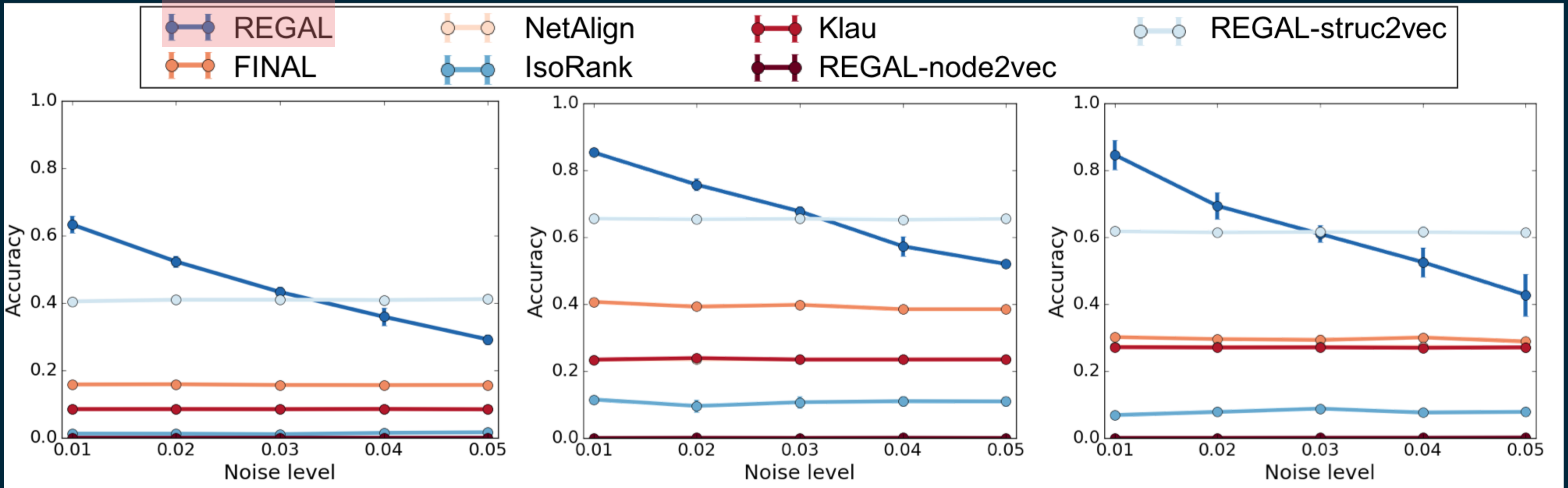




+ structural and attribute noise with probability p_s and p_a

[Bayati+"Algorithms for large, sparse network alignment problems." ICDM '09]
[Zhang+ "Final: Fast attributed network alignment." KDD '16]
[Singh, Rohit, et al. "Global alignment of multiple protein interaction networks with application to functional orthology detection." PNAS '08]
[Klau, Gunnar W. "A new graph-based method for pairwise global network alignment." BMC bioinformatics 10.1 2009.

Non-attributed Graphs



(a) **Arxiv** (198 110 **edges**) (b) **PPI** (76 584 **edges**) (c) **Arenas Email** (5 451 **edges**)

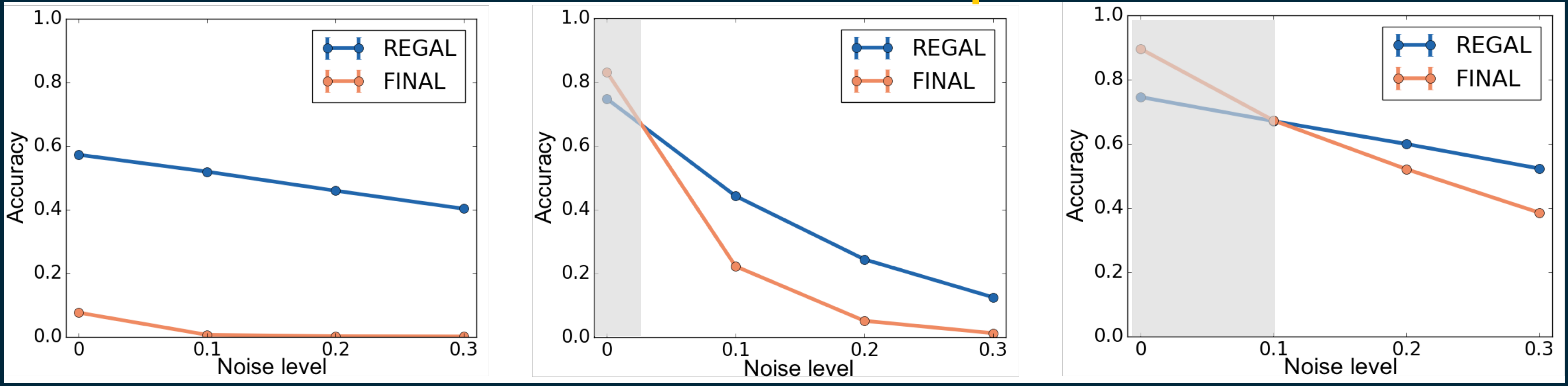
- REGAL variants are more accurate than traditional alignment methods.
- Structural embeddings outperform the proximity-based ones.

- REGAL is up to 22-31× faster than other representation-learning-based alignment methods.
  - Avoids the expense of RW

Dataset	Arxiv	PPI	Arenas
FINAL	4182 (180)	62.88 (32.20)	3.82 (1.41)
NetAlign	149.62 (282.03)	22.44 (0.61)	**1.89 (0.07)**
IsoRank	**17.04 (6.22)**	**6.14 (1.33)**	**0.73 (0.05)**
Klau	1291.00 (373)	476.54 (8.98)	43.04 (0.80)
REGAL-node2vec	709.04 (20.98)	139.56 (1.54)	15.05 (0.23)
REGAL-struc2vec	1975.37 (223.22)	441.35 (13.21)	74.07 (0.95)
REGAL	**86.80 (11.23)**	**18.27 (2.12)**	2.32 (0.31)

Attributed Graphs

16K edges

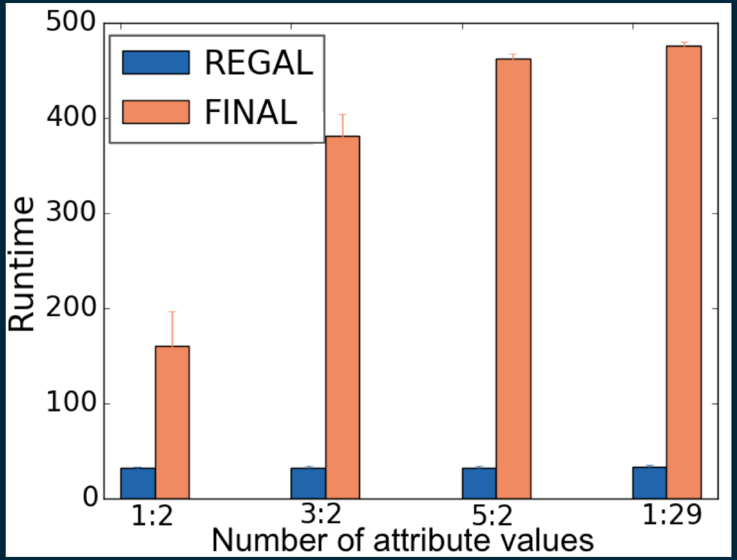


1 synthetic binary attribute

5 synthetic binary attributes

1 real attribute (29 values)

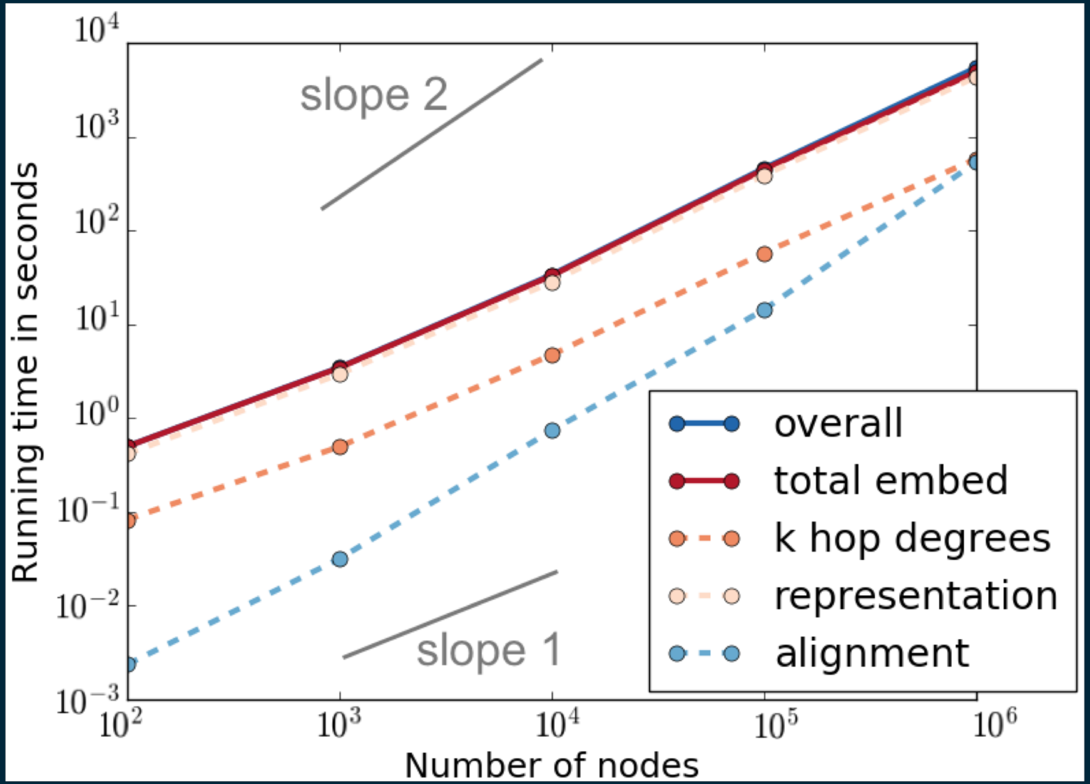
- REGAL outperforms FINAL without extensive, reliable attribute information.
- REGAL is significantly faster than FINAL, especially with more attribute information.



CSE

GEMS LAB

Experiments: Scalability



Erdős-Renyi random graphs

- Dominant factors: O(*n p*) node similarities, forming embeddings.
- REGAL is subquadratic in practice.

Code: https://github.com/GemsLab/REGAL

GEMS LAB

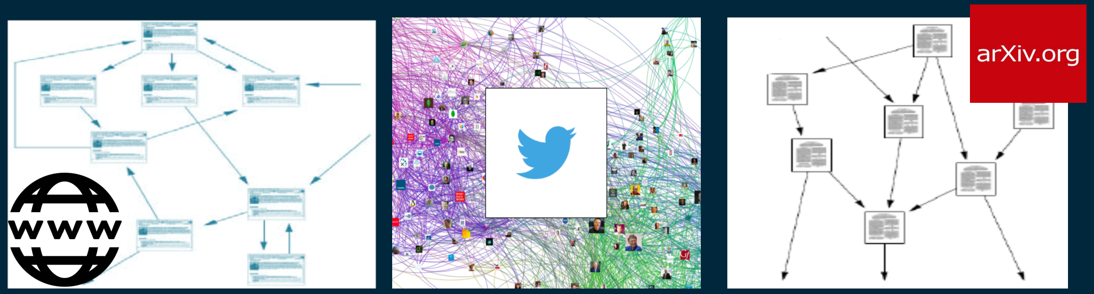
[Heimann, Shen, Safavi, Koutra. CIKM '18.]

Extension to weighted, directed graphs

- Analyze incoming & outgoing neighborhoods separately

$$\mathbf{b}_u^+ = \sum_{k=0}^{K} \delta^k \mathbf{b}_u^{k+}$$

combine across K hops — discount distant hops



✧ Concatenate incoming/outgoing histograms $\mathbf{b}_u = [\mathbf{b}_u^+, \mathbf{b}_u^-]$

✧ "Weighted" histograms: capture a node's contribution to another node's structural identity

[Di Jin, Mark Heimann, Tara Safavi et al. ACM KDD'19]

Predicting professional roles in email networks

	SNA	RolX	LinBP	LINE	DeepWalk	node2vec	struc2vec	DNGR	Graphwave	EMBER-U	EMBER-D	EMBER-W	EMBER
Trove-318	.7605	.5670	.6908	.6618	.7602	.7648	.7799	.7131	.7685	.7749	.7563	.7625	**.8045***
Trove-183	.7648	.5787	.7718	.5657	.8071	.8223	**.8264**	4925	.6391	.7986	.7838	.8186	.8241
Trove-141	.6738	.5591	.7409	.7102	.7191	.7474	.7391	.6235	.7112	.7291	.7309	.6971	**.7568***
Trove-98	.6676	.5177	.6323	.6872	.5587	.6198	.6498	.5329	**.7177***	.6040	.5857	.6333	.6911
Trove-19	.5429	.6981	.6248	.7184	.5531	.5959	.6102	.6089	.7157	.6837	.7204	.6939	**.7337***
Trove-2K	.6305	.5212	.6622	.6771	.6769	.6780	**.6802**	.6527	.6594	.6689	.6345	.6677	.6745
Trove	.6633	.5280	5454	—	.6866	.6951	—	—	—	.6905	.7141	.7122	**.7162***
Enron	.6205	.5197	.5000	.6931	.7201	.7389	—	.5709	—	**.7393**	.7347	.7305	.7305

EMBER outperforms its unweighted/undirected variants → importance of accounting for the volume + reciprocity in email exchanges.

Professional Roles:

- Officers (“C-Suite” employees)
- Middle-level managers
- Workers

[Di Jin, Mark Heimann, Tara Safavi et al. ACM KDD’19]

Fun fact

Comparing academic & industrial roles

- Academic email network with 3,078 users and 231,470 email exchanges
- Employee *u* at a university "maps" to employee *v* at organization X
  - ✧ if $\underset{v \in X}{\operatorname{argmin}} || b_u - b_v ||_2$

bigger size

	Officer	Mgmt.	Worker
Trove-318	0.13	0.62	0.25
Trove-183	0.24	0.51	0.26
Trove-141	0.17	0.69	0.15
Trove-98	0.57	0.31	0.11
Trove-19	0.75	0.11	0.13

Professors are similar to:
- CEOs of smaller companies (Trove-98 and Trove-19), and
- more like managers in bigger companies (Trove-318 through Trove-141).

GEMS LAB

[Di Jin, Mark Heimann, Tara Safavi et al. ACM KDD'19]

Talk Outline: Structural Embeddings for…

- Cross-network tasks [ACM CIKM’18]
  - ✧ Node (role) classification [ACM KDD’19]

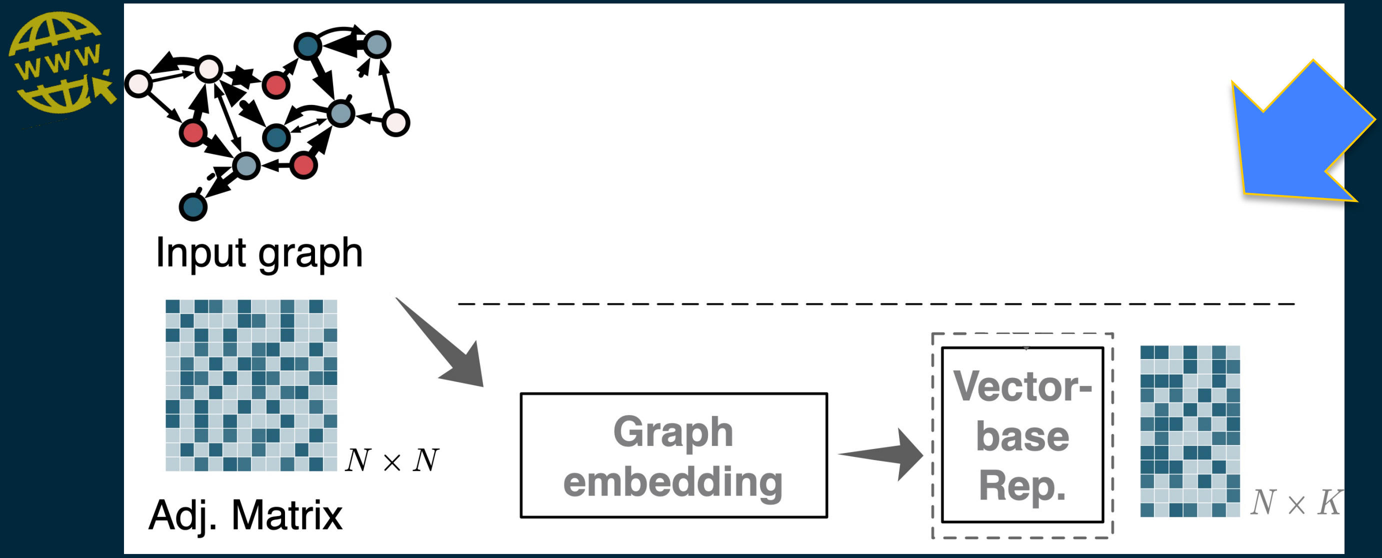
- Latent summarization [ACM KDD’19]

Based on:

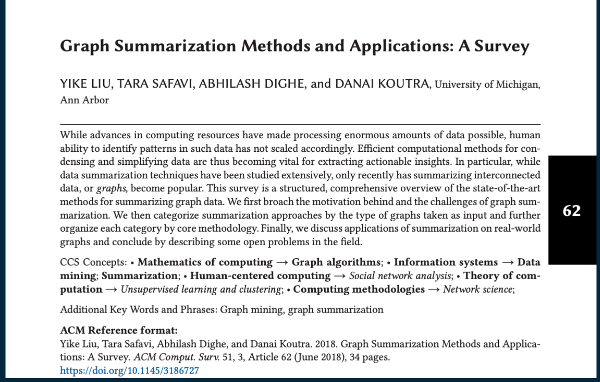
- M. Heimann, H. Shen, T. Safavi, D. Koutra. REGAL: Representation Learning-based Graph Alignment. ACM CIKM’18.
- D. Jin*, M. Heimann*, T. Safavi, M. Wang, W. Lee, L. Snider, D. Koutra. Smart Roles: Inferring Professional Roles in Email Networks. ACM KDD’19.
- D. Jin, R. Rossi, E. Koh, S. Kim, A. Rao, D. Koutra. Latent Network Summarization. ACM KDD’19.
- Y. Liu, T. Safavi, A. Dighe, D. Koutra. Graph Summarization Methods and Applications: A Survey. ACM Computing Surveys 2018.
- D. Jin, M. Heimann, R. Rossi, D. Koutra. node2bits: Compact Time- and Attribute-aware Node Representations for User Stitching. Arxiv 1904.08572
- Y. Yan, J. Zhu, Marlena Duda, Eric Solarz, Chandra Sripada, Danai Koutra. GroupINN: Grouping-based Interpretable Neural Network-based Classification of Limited, Noisy Brain Data. ACM KDD’19.

Embeddings are powerful, but can take up a lot of space!

- For 1B nodes and K=128 → 1TB to store the embeddings!
- Can we summarize them?



Graph Summarization Survey



Graph Summarization Methods and Applications: A Survey

YIKE LIU, TARA SAFAVI, ABHILASH DIGHE, and DANAI KOUTRA, University of Michigan, Ann Arbor

62

While advances in computing resources have made processing enormous amounts of data possible, human ability to identify patterns in such data has not scaled accordingly. Efficient computational methods for condensing and simplifying data are thus becoming vital for extracting actionable insights. In particular, while data summarization techniques have been studied extensively, only recently has summarizing interconnected data, or *graphs*, become popular. This survey is a structured, comprehensive overview of the state-of-the-art methods for summarizing graph data. We first broach the motivation behind and the challenges of graph summarization. We then categorize summarization approaches by the type of graphs taken as input and further organize each category by core methodology. Finally, we discuss applications of summarization on real-world graphs and conclude by describing some open problems in the field.

CCS Concepts: • **Mathematics of computing** → **Graph algorithms**; • **Information systems** → **Data mining**; **Summarization**; • **Human-centered computing** → *Social network analysis*; • **Theory of computation** → *Unsupervised learning and clustering*; • **Computing methodologies** → *Network science*;

Additional Key Words and Phrases: Graph mining, graph summarization

ACM Reference format:
Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. 2018. Graph Summarization Methods and Applications: A Survey. *ACM Comput. Surv.* 51, 3, Article 62 (June 2018), 34 pages.
https://doi.org/10.1145/3186727

Graph Summarization

Network type

- Static
  - Plain
    - Structure only
      - Grouping
      - Compression
      - Simplification
      - Influence
  - Labeled
    - Structure + labels
      - Grouping
      - Compression
      - Influence
- Dynamic
  - Plain
    - Temporal structure
      - Grouping
      - Compression
      - Influence

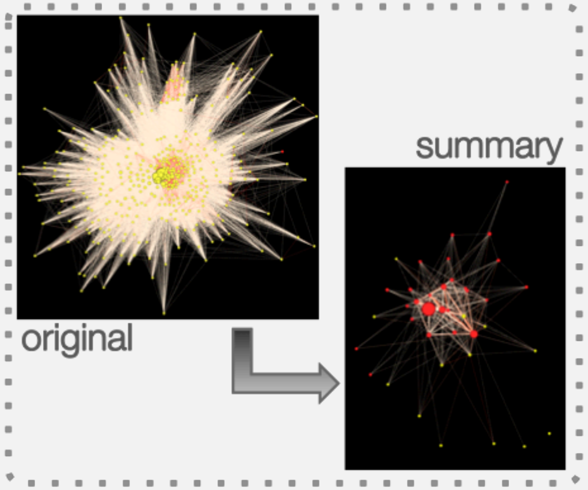
Core techniques employed

Applications

- Query Efficiency
- Compression
- Visualization
- Pattern Discovery
- Influence Analysis

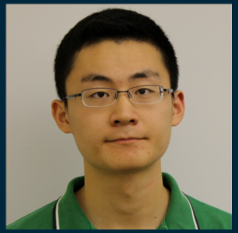
Open Problems

- Richer Data
- Streaming / Incremental Summarization
- Automated Insight Extraction
- Evaluation

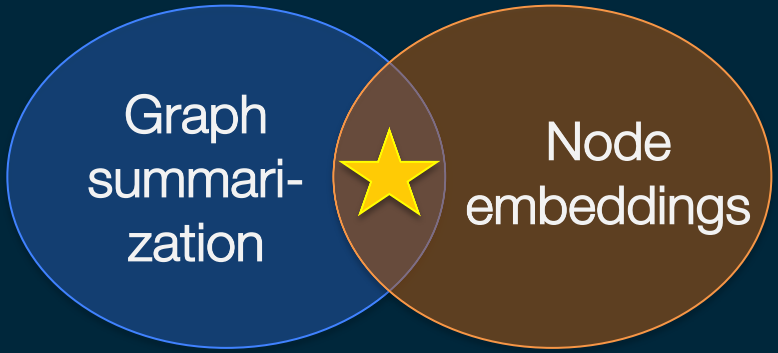


Related Research Areas:
graph clustering, partitioning, community detection, sampling, sparsification, sketches, compression

Latent Network Summarization

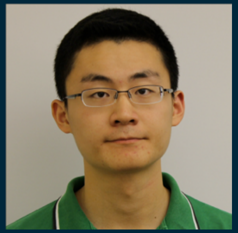


- Given: a graph G(V, E)
- Find: a compressed representation that captures the key structural information such that it is
  - ✧ independent of graph size (|V|, |E|), and
  - ✧ capable of deriving node representations



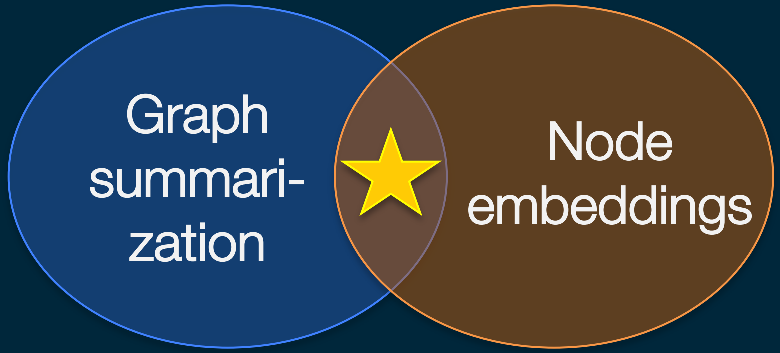
[Di Jin, Rossi et al. ACM KDD'19]

Latent Network Summarization



- Given: a graph G(V, E)
- Find: a compressed representation that captures the key structural information such that it is
  - ✧ independent of graph size (|V|, |E|), and
  - ✧ capable of deriving node representations

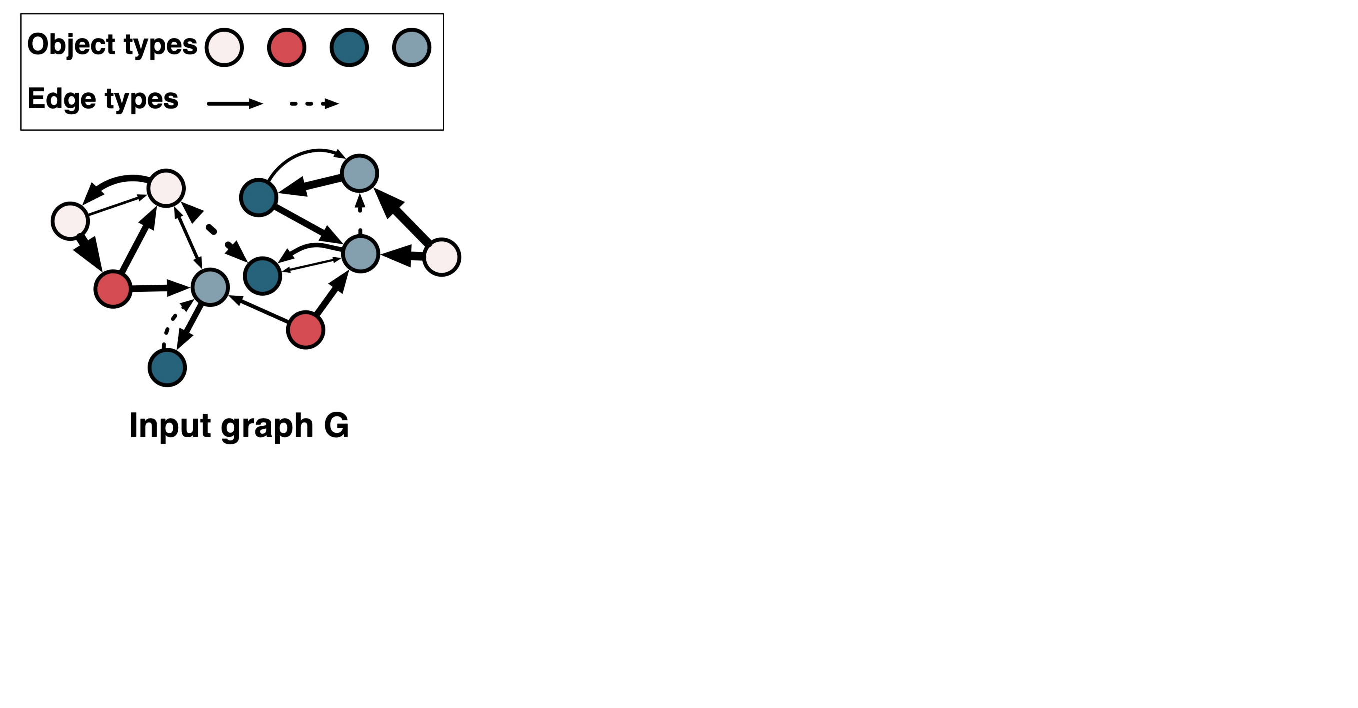
- Desired Properties
  - ✧ (P1) generality to handle arbitrary network
  - ✧ (P2) high compression rate
  - ✧ (P3) natural support of inductive learning
  - ✧ (P4) ability to on-the-fly derive node embeddings



Comparison to Related Work

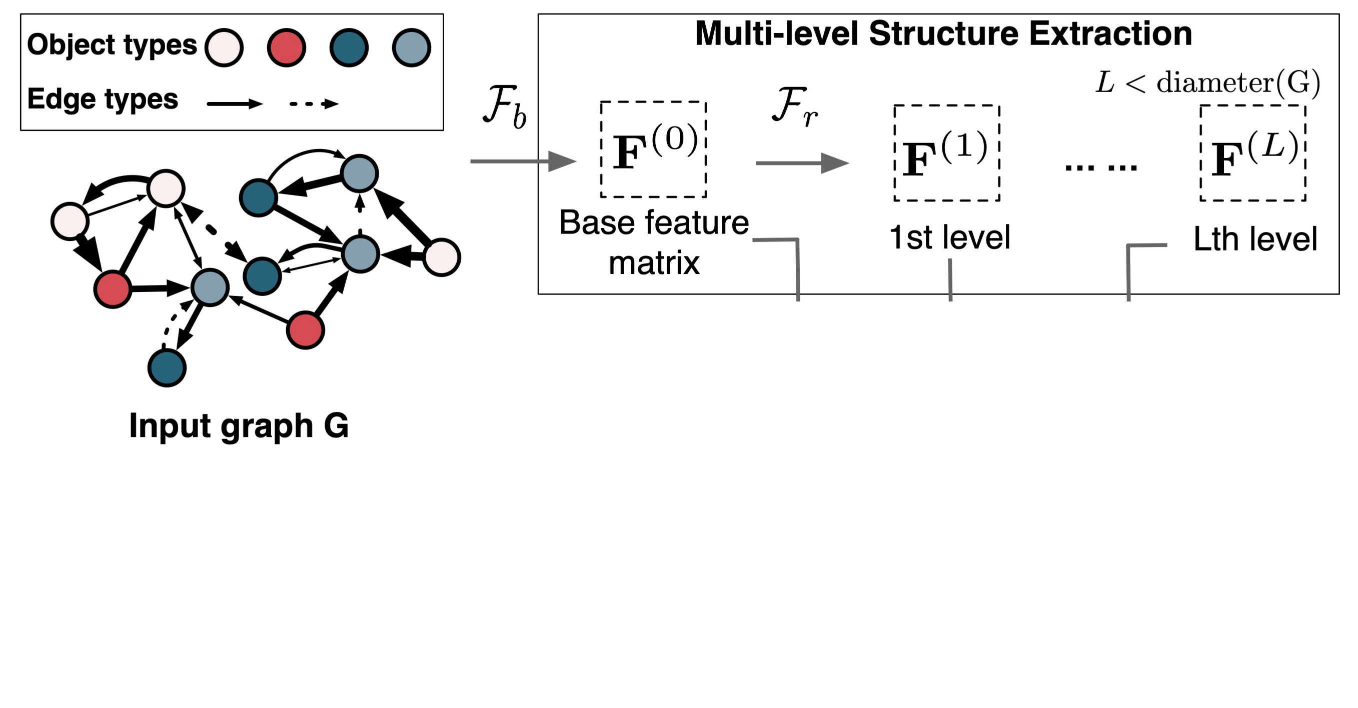
	Input	Representations / Output			Method	
	Hetero-geneity	Size indep.	Node specific	Proxim. indep.	Scalable	Induc.
Aggregation [2]	✓	✗	✗	✗	✓	✗
Cosum [34]	✗	✗	✗	✓	✗	✗
AspEm [31]	✓	✗	✓	✗	✓	✗
metapath2vec [8]	✓	✗	✓	✗	✓	✗
n2vec [11], LINE [32]	✗	✗	✓	✗	✓	✗
struc2vec [26]	✗	✗	✓	✓	✗	✗
DNGR [6]	✗	✗	✓	✗	✗	✗
GraphSAGE [12]	✓	✗	✓	✓	✓	✓
Multi-LENS	✓	✓	✓	✓	✓	✓

Latent Network Summarization: Overview



Latent Network Summarization: Overview

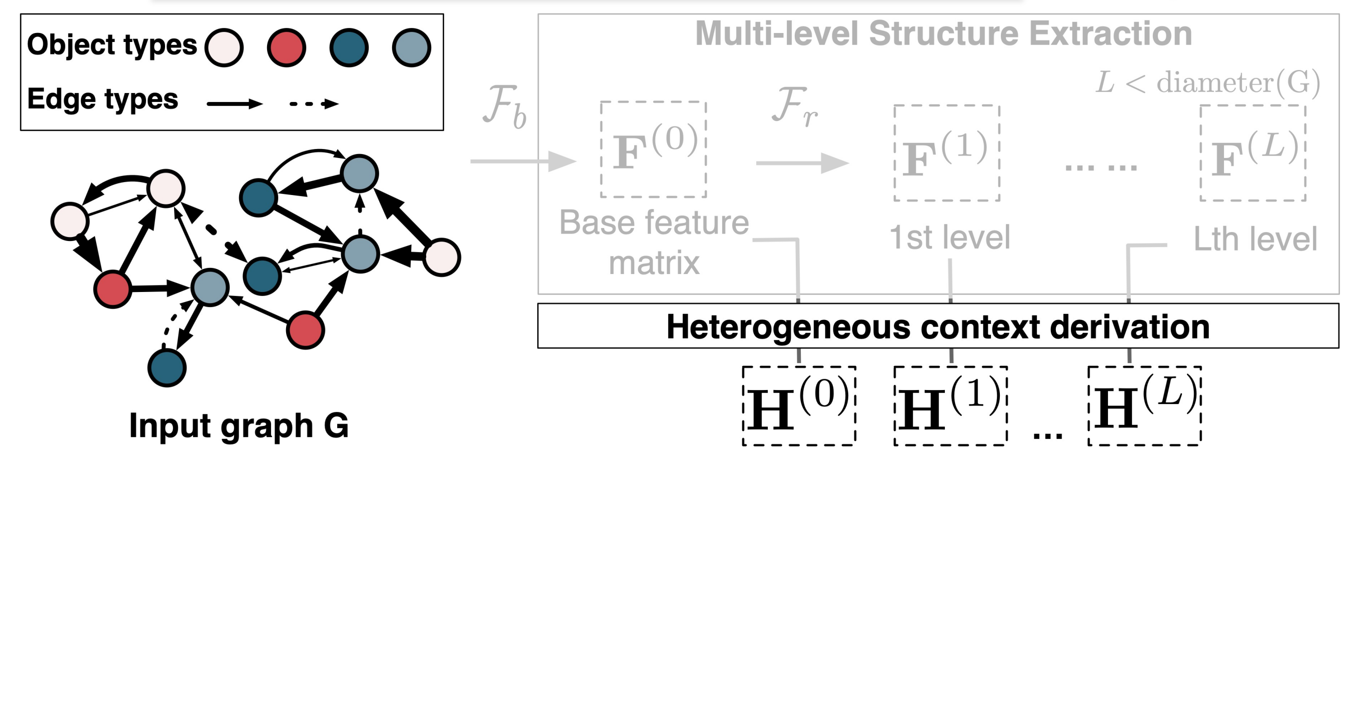
1. Relational functions to aggregate nodewise structural features automatically

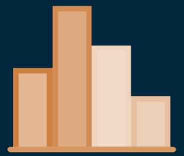


[Di Jin, Rossi et al. ACM KDD'19]

Latent Network Summarization: Overview

1. Relational functions to aggregate nodewise structural features automatically



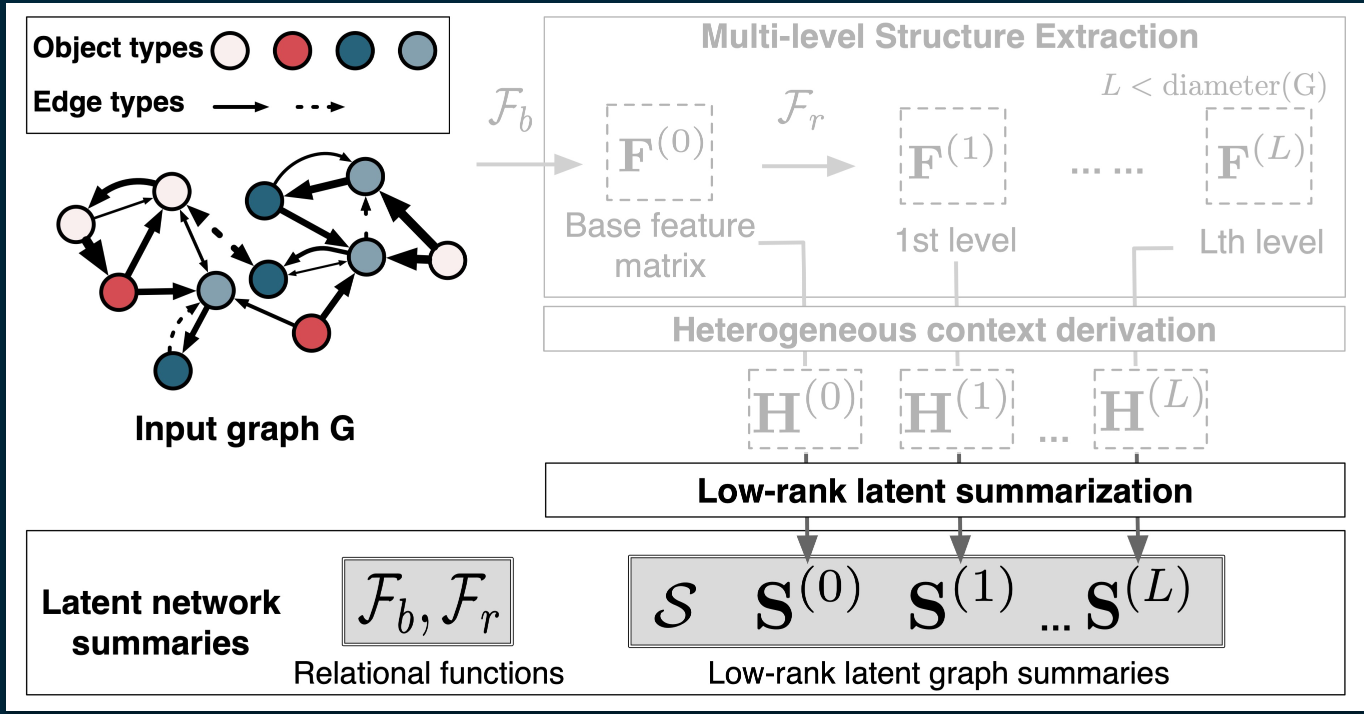


2. Histogram-based heterogeneous contexts for nodes

[Di Jin, Rossi et al. ACM KDD'19]

Latent Network Summarization: Overview

1. Relational functions to aggregate nodewise structural features automatically

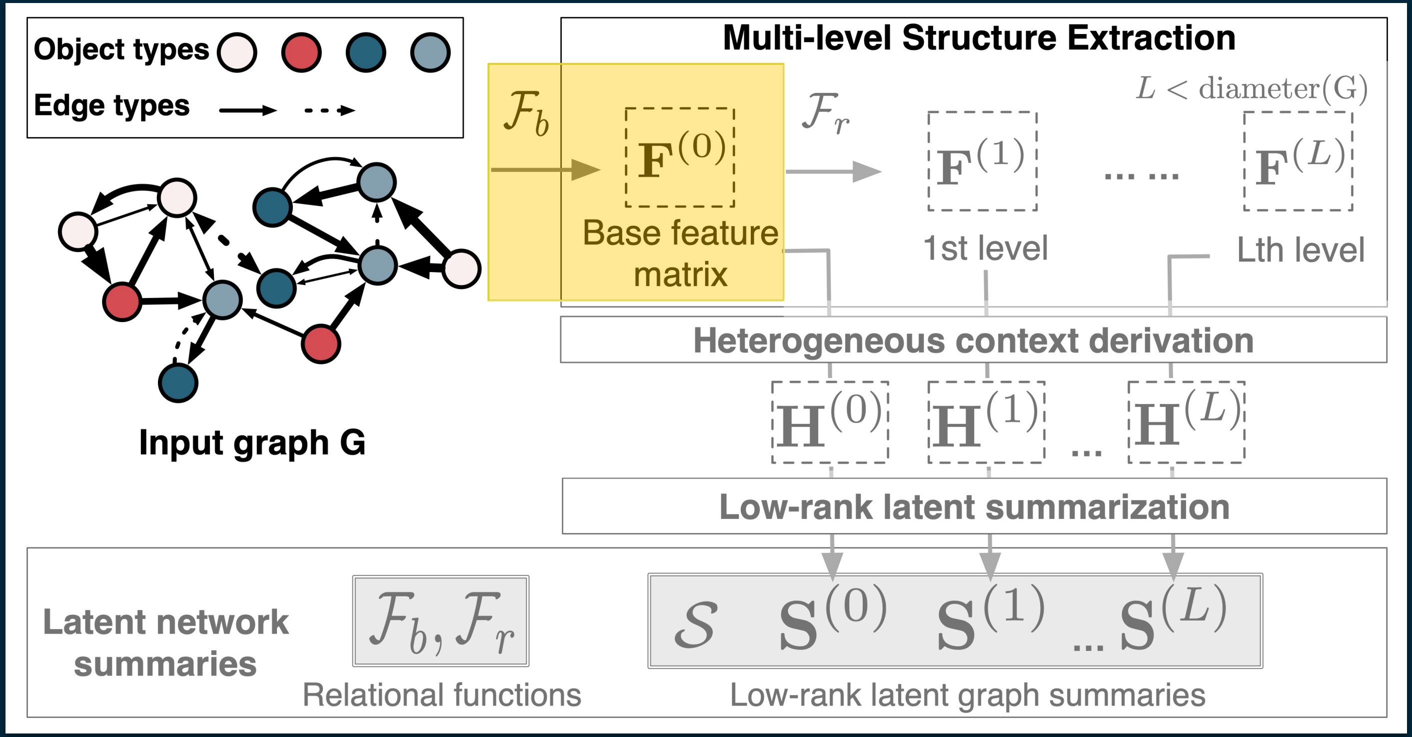


2. Histogram-based heterogeneous contexts for nodes

3. Subspace vectors from which we can derive the embeddings

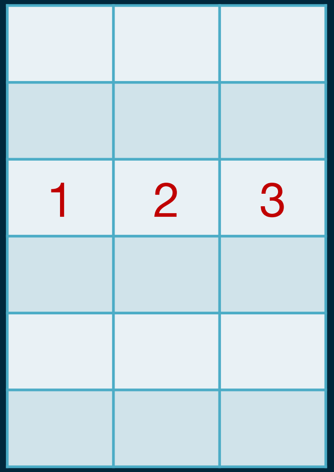
M CSE GEMS LAB [Di Jin, Rossi et al. ACM KDD'19]

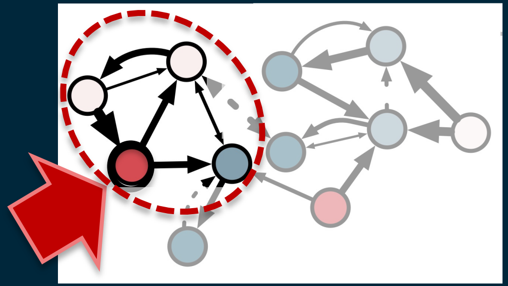
Multi-LENS: Base functions



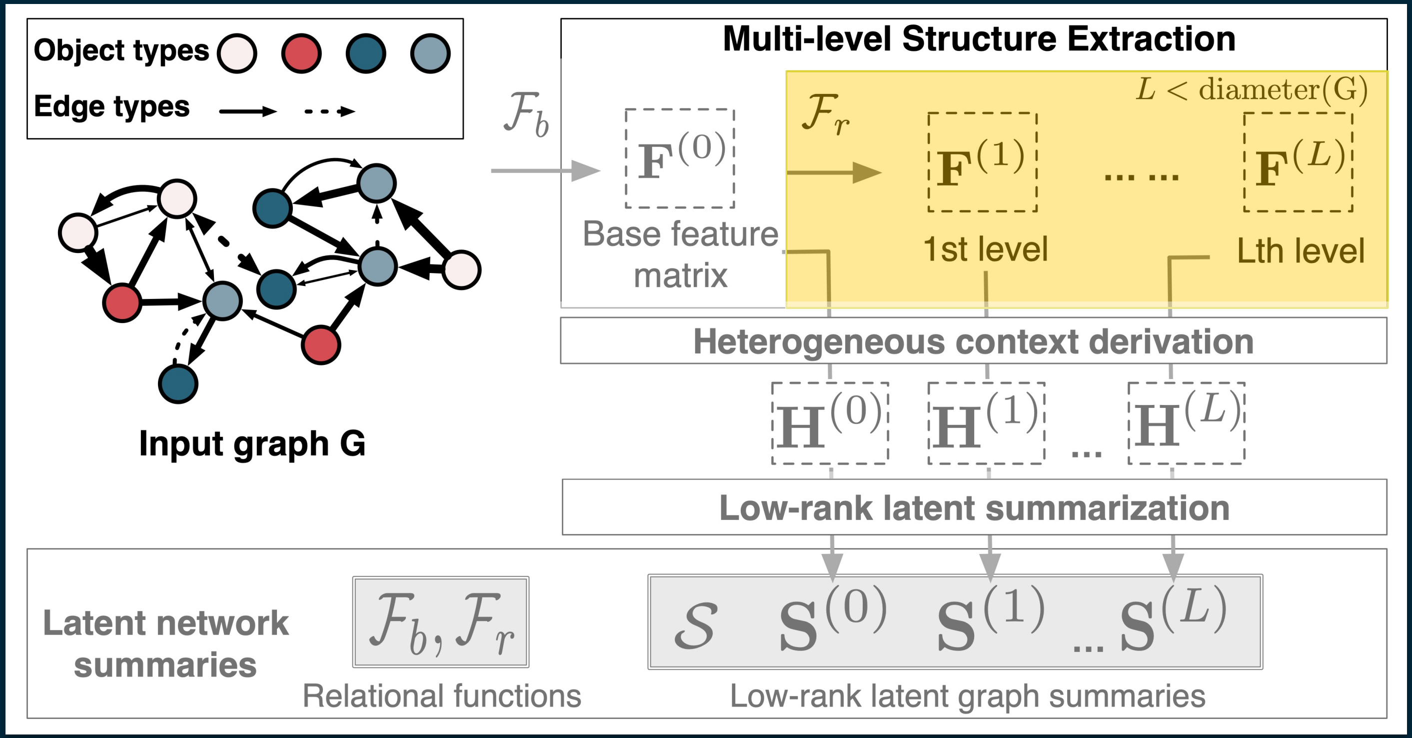
- F_b: base graph functions that operate on the adjacency matrix
  - e.g., sum Σ on egonets
  - In-/out-/total degree

$\mathbf{F}^{(0)} =$

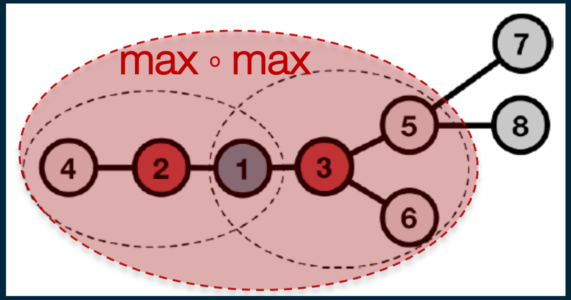




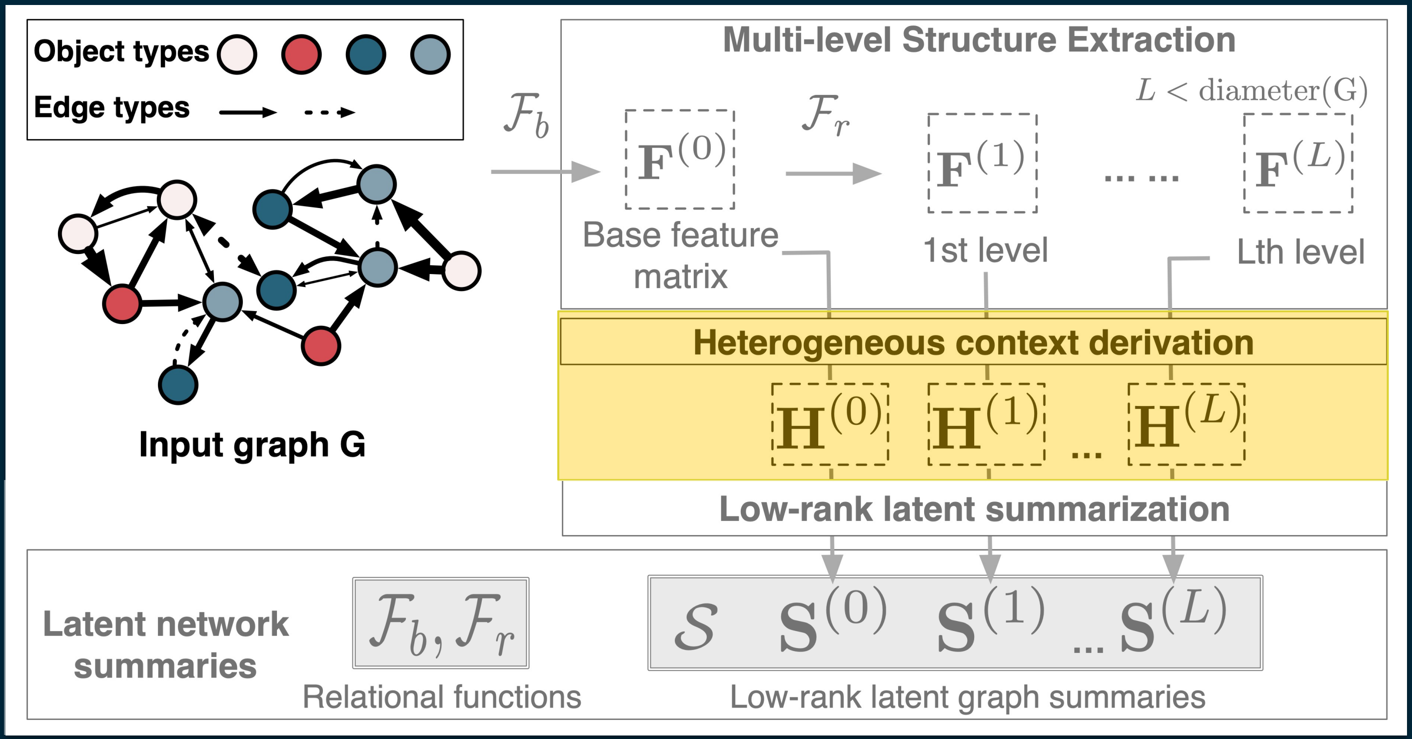
Multi-LENS: Rel Fn Compositions



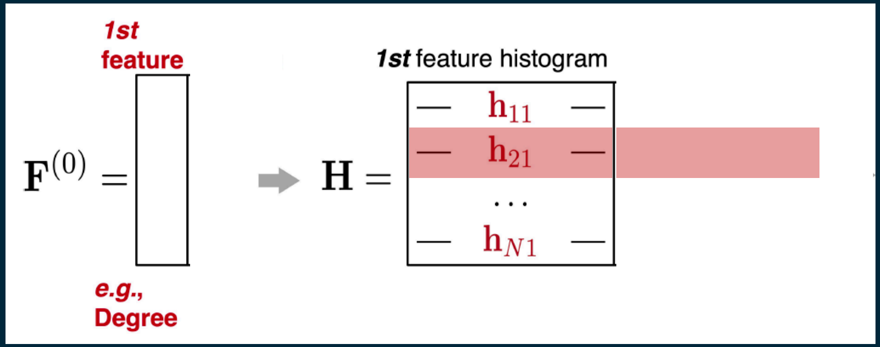
- Recursively apply the relational operators
  - *max*, *min*, *sum*, *mean*, *variance*, *l1*-dist, *l2*-dist
  - Derive complex, non-linear features automatically
- ℓ compositions over a node's neighborhood =
  higher-order structural features in the ℓ-hop neighborhoods

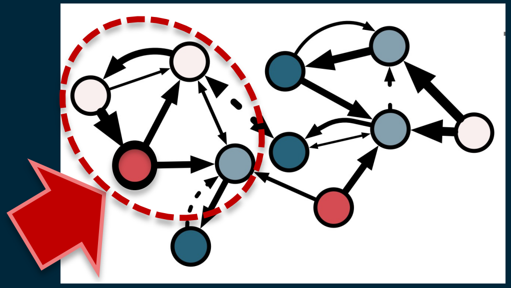


Multi-LENS: Rel Fn Compositions

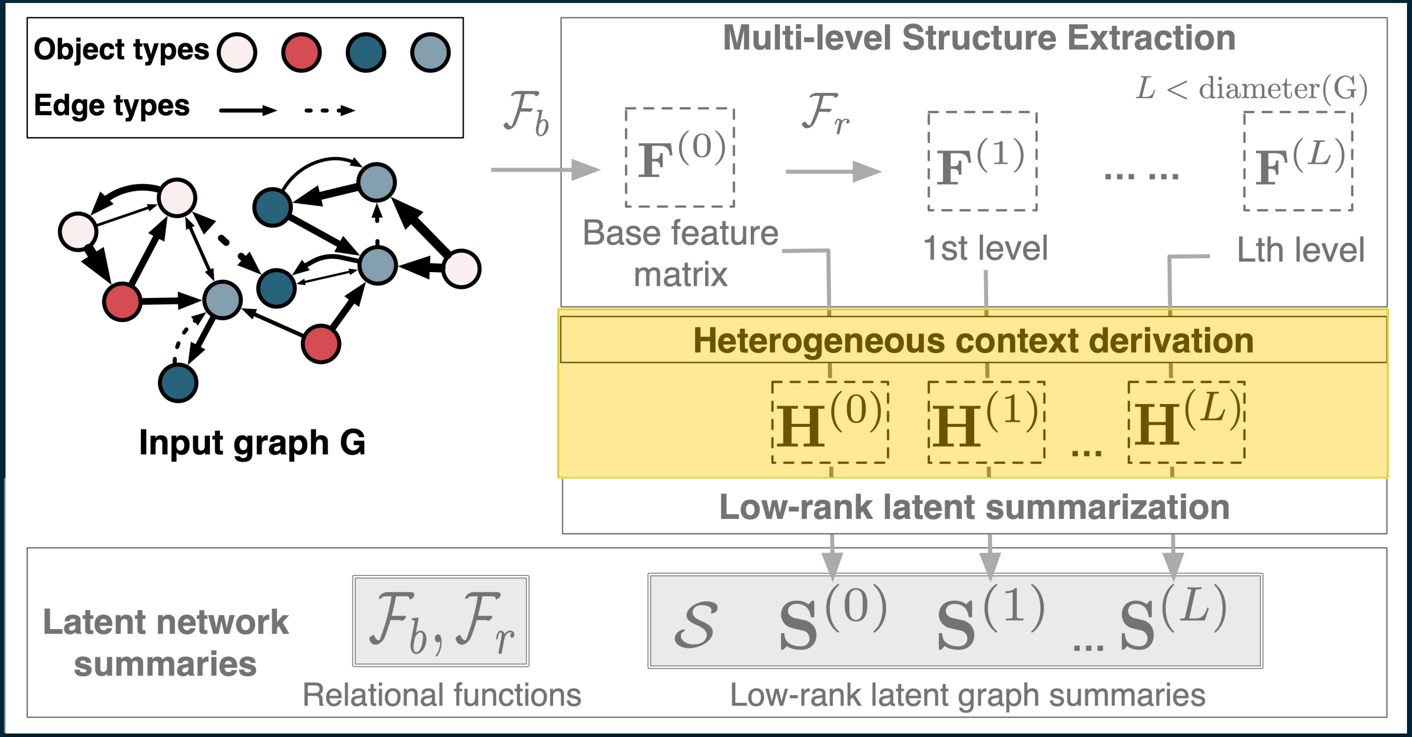


- Structural identity of node i
  - via histograms
  - log-scale for skewness

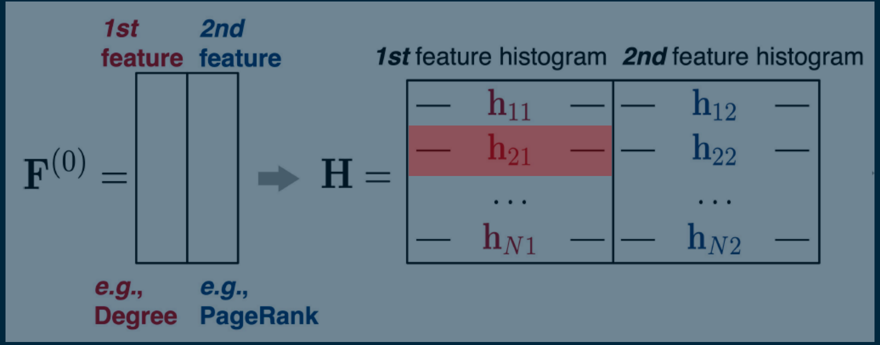




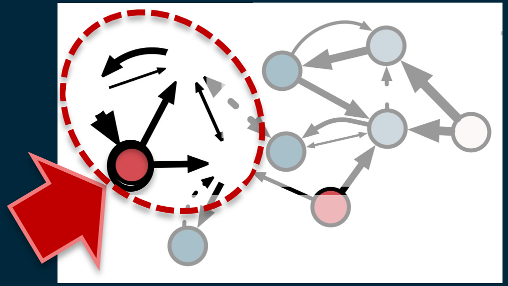
Multi-LENS: Rel Fn Compositions



- Structural identity of node i
  - via histograms
  - log-scale for skewness

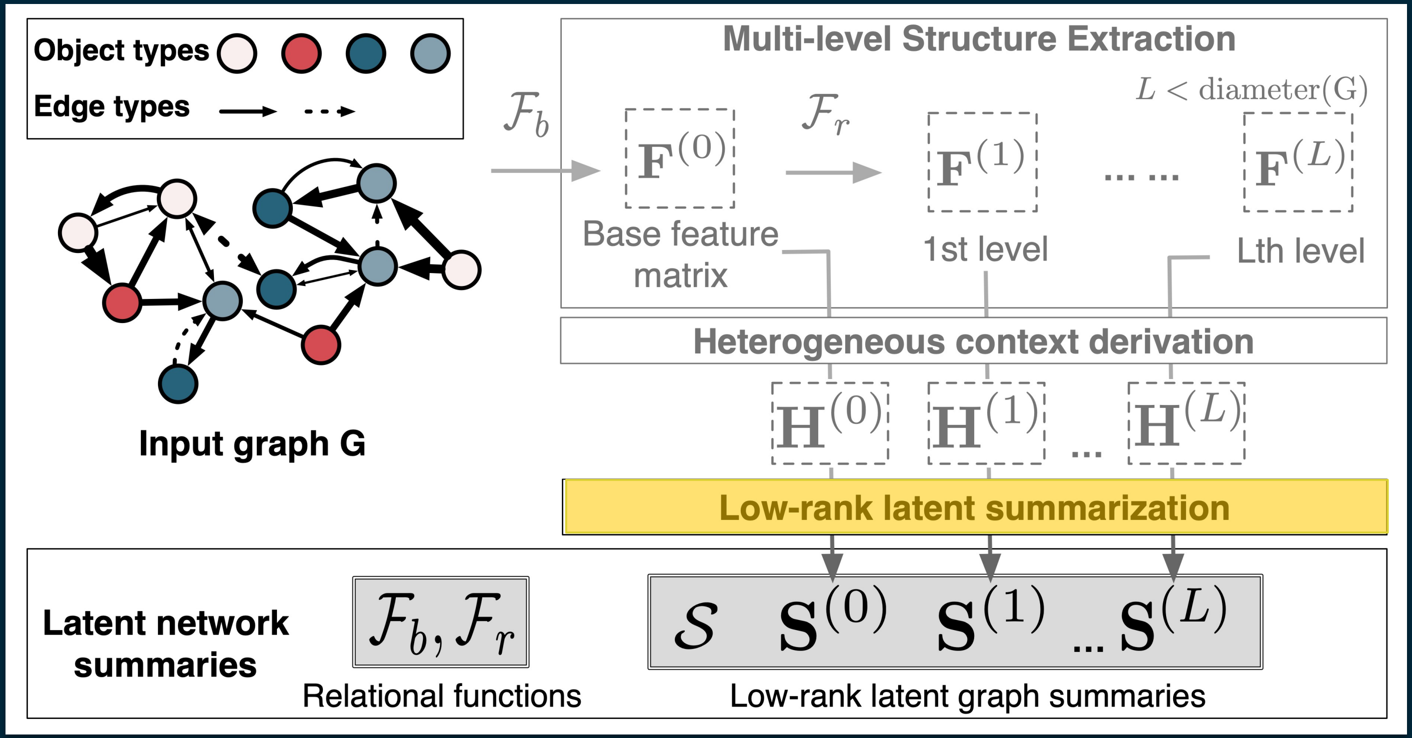


- Node/edge types & directionality
  - Histograms in different contexts
  - e.g., restricted on neighborhoods of a specific node type



[Di Jin, Rossi et al. ACM KDD'19]

Multi-LENS: Summarization



Node context at different neighborhoods

$$\mathbf{H}^{(l)} = \mathbf{U}^{(l)}\Sigma^{(l)}\mathbf{V}^{(l)}$$

Level-ℓ summarized representation

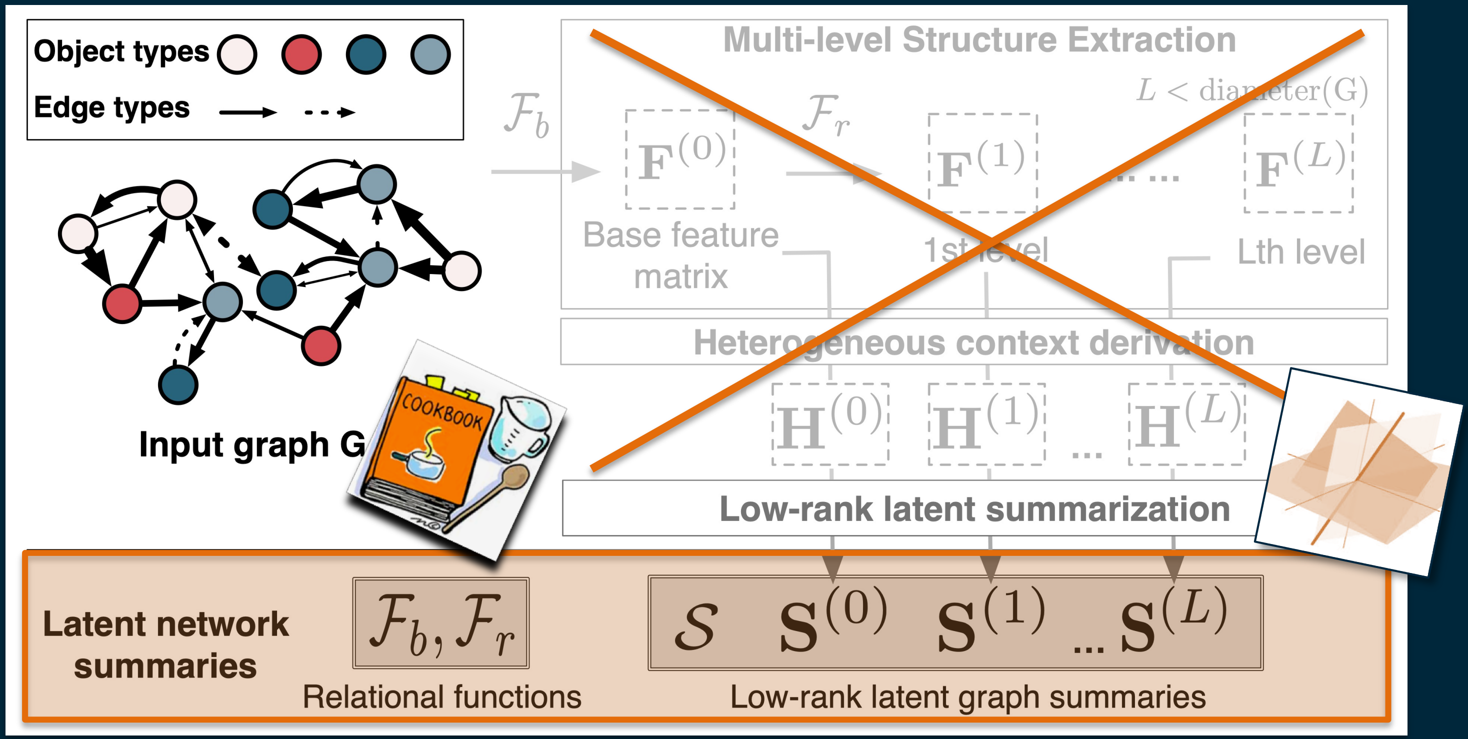
$$\mathbf{S}^{(\ell)} = \sqrt{\Sigma^{(\ell)}}\mathbf{V}^{(\ell)T}$$

= *$K^{(\ell)}$-dim subspace for node context*

Level-ℓ node embeddings (not stored)

$$\mathbf{Y}^{(\ell)} = \mathbf{U}^{(\ell)}\sqrt{\Sigma^{(\ell)}}$$

Multi-LENS: Summarization



Node context at different neighborhoods

$$\mathbf{H}^{(l)} = \mathbf{U}^{(l)}\Sigma^{(l)}\mathbf{V}^{(l)}$$

Level-ℓ summarized representation

$$\boxed{\mathbf{S}^{(\ell)}} = \sqrt{\Sigma^{(\ell)}}\mathbf{V}^{(\ell)T}$$

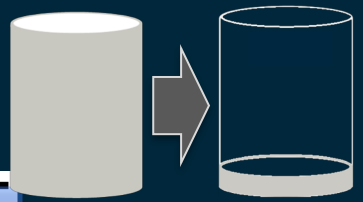
= $K^{(\ell)}$-dim subspace for node context

Level-ℓ node embeddings (not stored)

$$\mathbf{Y}^{(\ell)} = \mathbf{U}^{(\ell)}\sqrt{\Sigma^{(\ell)}}$$

- Higher-order features based on graph structure; independent of IDs
  $\rightarrow$ they generalize across networks
- Inductively learn node embeddings in unseen G’: $\mathbf{Y}'^{(\ell)} = \mathbf{H}'^{(\ell)}\boxed{(\mathbf{S}^{(\ell)})^{\dagger}}$

Space comparison



Data	SE	LINE	n2vec	DW	m2vec	AspEm	G2G	ML (MB)
facebook								**0.58**
yahoo								**0.62**
dbpedia								**0.81**
digg								**0.54**
bibson.								**0.75**

Multi-LENS requires 4-2152x less output storage space than the other embedding methods.

Data	#Nodes	#Edges	#Node Types	Graph Type
facebook	4 039	88 234	1	unweighted
yahoo-msg	100 058	1 057 050	2	weighted
dbpedia	495 936	921 710	4	unweighted
digg	283 183	4 742 055	2	unweighted
bibsonomy	977 914	3 754 828	3	weighted

Link Prediction

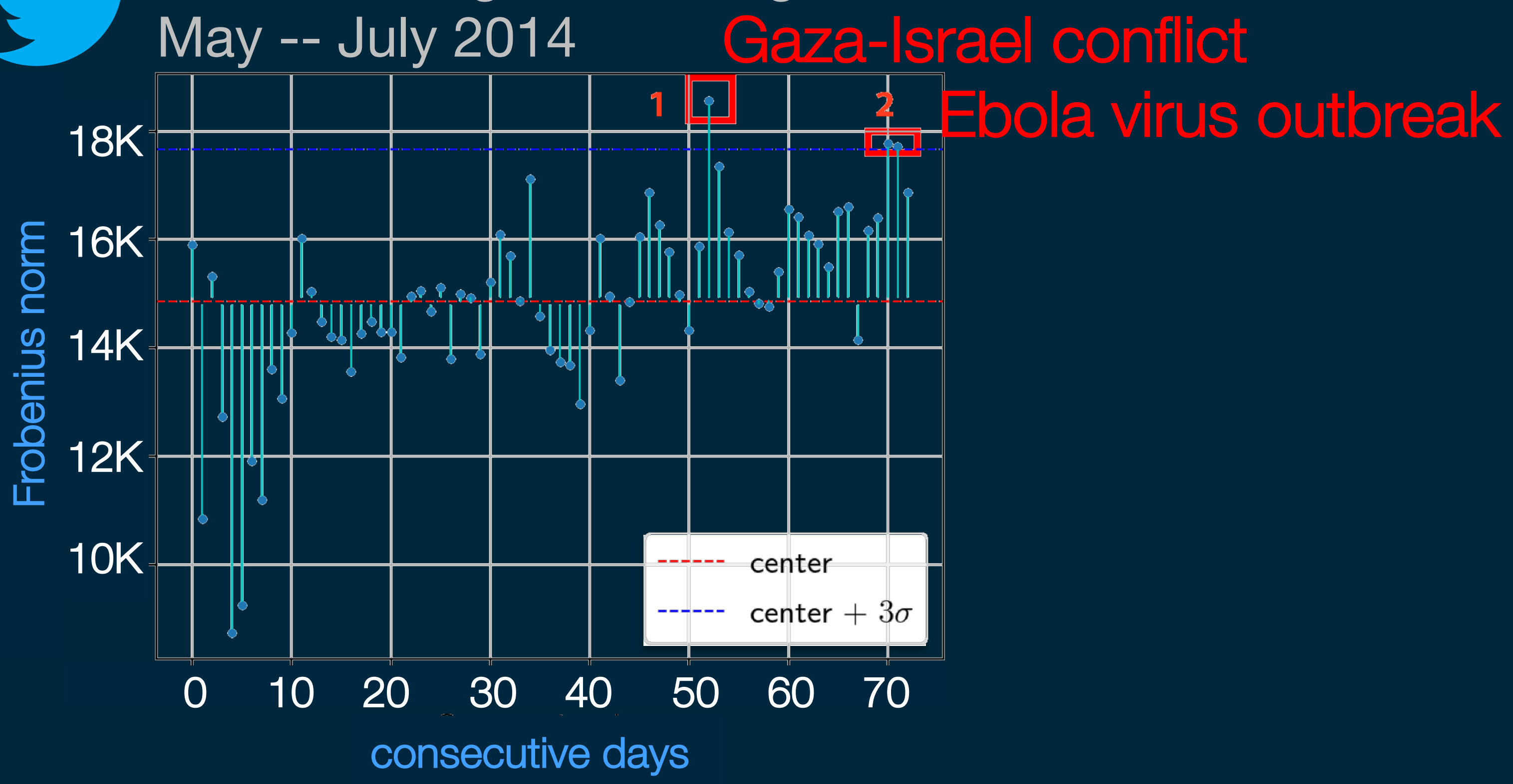
Data	Metric	NA	SE	LINE	DW	n2vec	GR	s2vec	DNGR	m2vec	AspEm	G2G	ML(L = 1)	ML(L = 2)
facebook	**AUC**	0.6213	0.6717	0.7948	0.7396	0.7428	0.8157	0.8155	0.7894	0.7495	0.5886	0.7968	0.8703	**0.8709***
	ACC	0.5545	0.5995	0.7210	0.6460	0.6544	0.7368	0.7388	0.7062	0.7051	0.5628	0.7274	**0.7920***	0.7904
	F1 macro	0.5544	0.5716	0.7210	0.6296	0.6478	0.7367	0.7387	0.7060	0.7041	0.5628	0.7273	**0.7920***	0.7905
yahoo-msg	**AUC**	0.7189	0.5375	0.6745	0.7715	0.7830	0.7535	OOT	OOM	0.6708	0.5587	0.6988	0.8443	**0.8446***
	ACC	0.2811	0.5224	0.6269	0.6927	0.7036	0.6825			0.6164	0.5379	0.6564	**0.7587***	**0.7587***
	F1 macro	0.2343	0.5221	0.6265	0.6897	0.7016	0.6821			0.6145	0.5377	0.6562	**0.7577***	**0.7577***
dbpedia	**AUC**	0.6002	0.5211	0.9632	0.8739	0.8774	OOM	OOT	OOM	OOT	0.6364	0.7384	**0.9820***	0.9809
	ACC	0.3998	0.5399	0.9111	0.8436	0.8436					0.5869	0.6625	**0.9186**	0.9151
	F1 macro	0.2968	0.4539	0.9110	0.8402	0.8402					0.5860	0.6613	**0.9186**	0.9150
digg	**AUC**	0.7199	0.6625	0.9405	0.9664	0.9681	OOM	OOT	OOM	0.9552	0.5644	0.8978	**0.9894***	0.9893
	ACC	0.2801	0.6512	0.8709	0.9023	0.9049				0.8891	0.5459	0.8492	**0.9596***	0.9590
	F1 macro	0.2660	0.6223	0.8709	0.9019	0.9046				0.8890	0.5459	0.8492	**0.9595***	0.9590
bibsonomy	**AUC**	0.7836	0.6694	0.9750	0.6172	0.6173	OOM	OOT	OOM	OOT	0.6127	OOM	**0.9909***	0.9909
	ACC	0.2164	0.6532	0.9350	0.5814	0.5816					0.5790		**0.9485***	0.9466
	F1 macro	0.2070	0.6064	0.9349	0.5781	0.5782					0.5772		**0.9485***	0.9466

The Multi-LENS node embeddings outperform all the baselines by 3.5–34.3% in AUC.

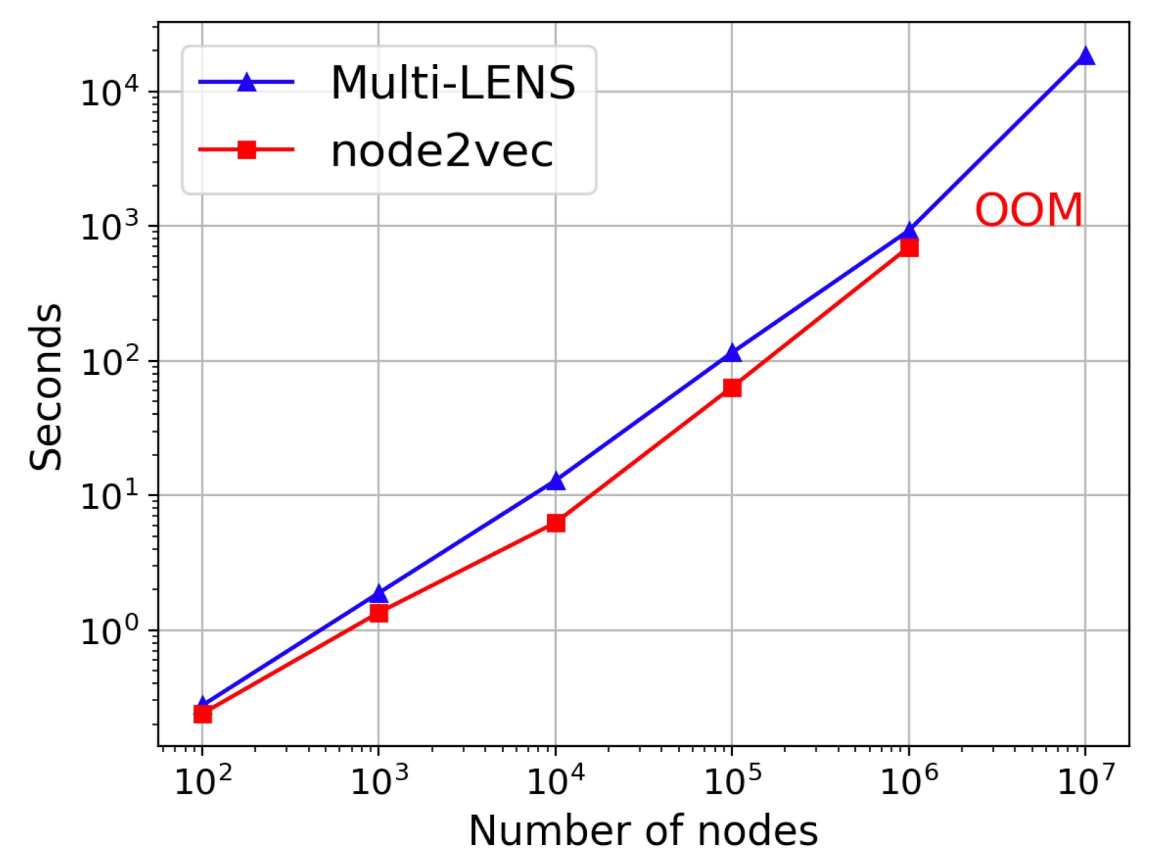
Inductive Anomaly Detection

- Learn summary of G_{t-1}, apply to G_t
- Compute the distance between the embeddings at *t-1* and *t*

308K hashtags, 2.6M edges
May -- July 2014



Scalability of Multi-LENS



Erdos-Renyi networks; d_{avg}=10

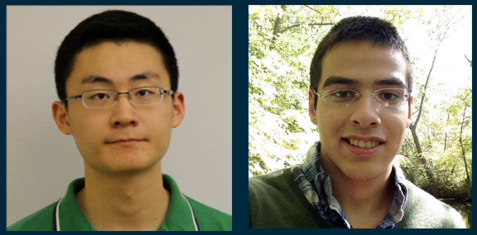
Multi-LENS is scalable to large graphs.

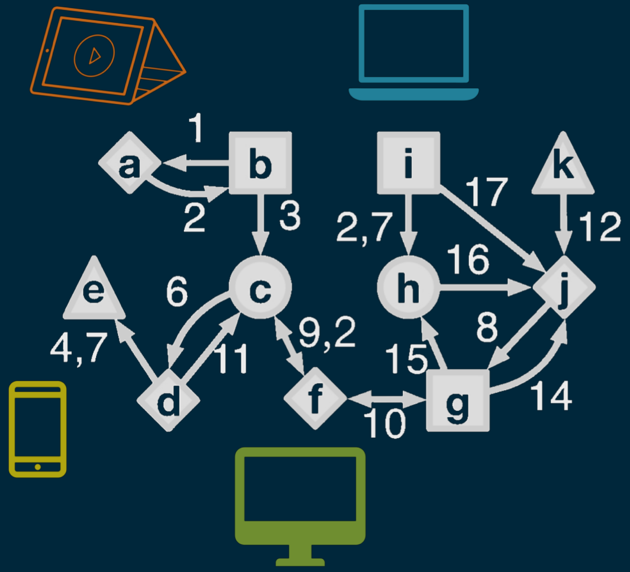
[Di Jin, Rossi et al. ACM KDD'19]

https://github.com/GemsLab/MultiLENS

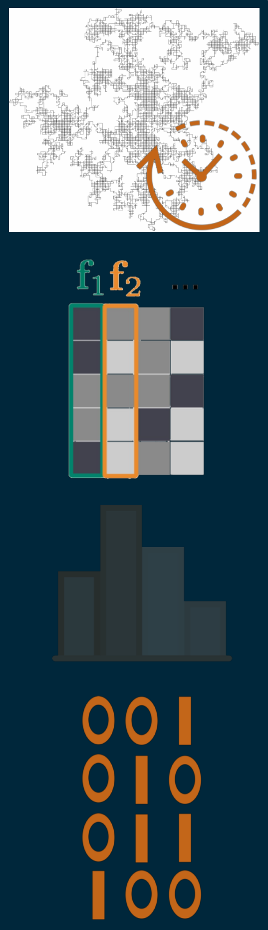
Can we capture
structural roles
with random walks?

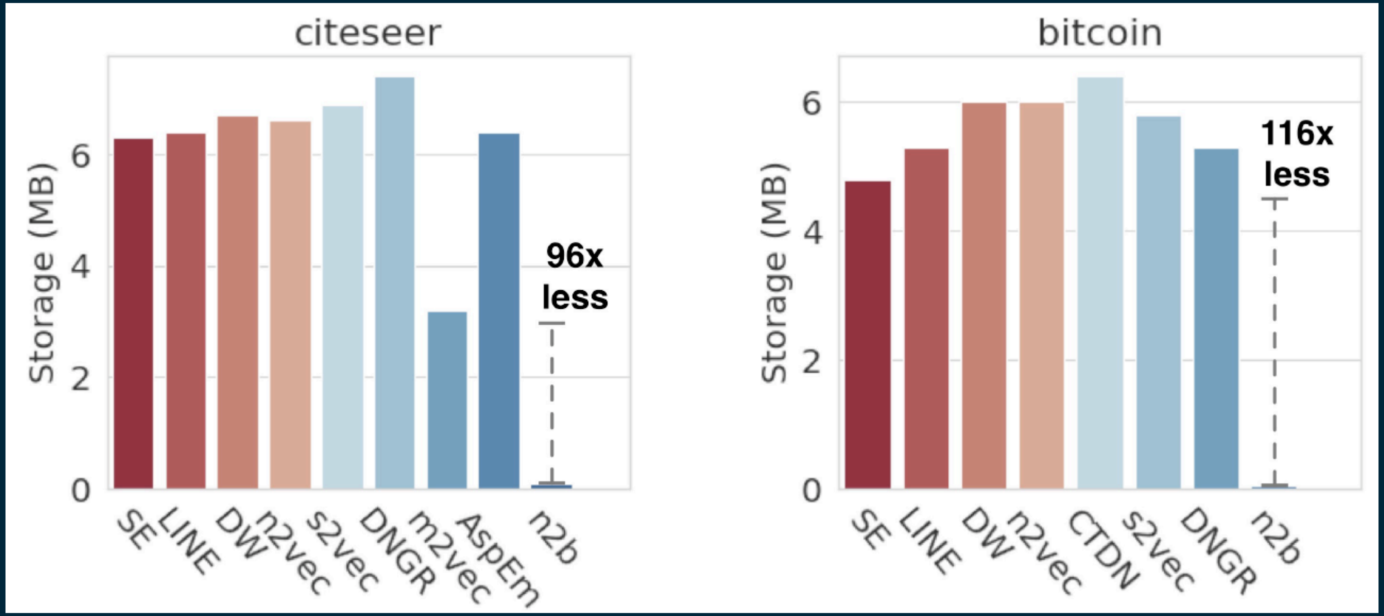
Binary embeddings





Ideas



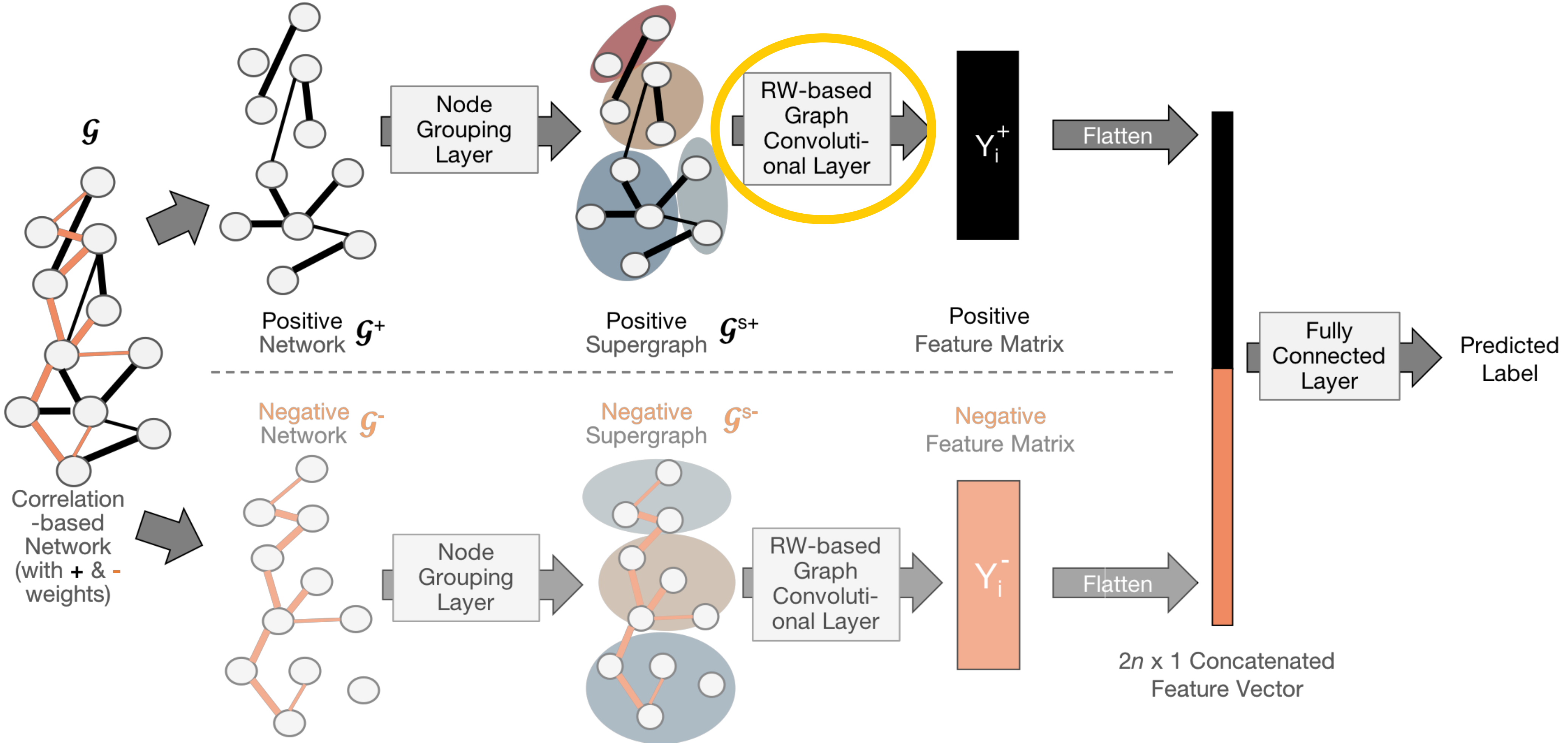


GEMS LAB

[Di Jin, Mark Heimann+, arXiv:1904.08572]

https://commons.wikimedia.org/wiki/File:Random_walk_25000.gif

GroupINN Architecture for Network Classification

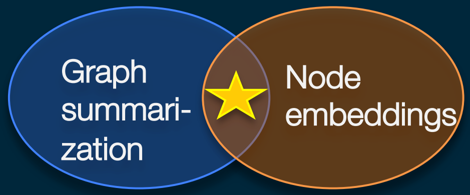


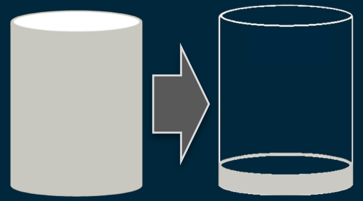
Take-away Messages

- Structural embeddings are less studied, but are appropriate / necessary in several tasks

- Histograms are powerful in capturing the graph structure
    - ✧ flexible, versatile (heterogeneity, attributes, directionality, weights…), less info loss

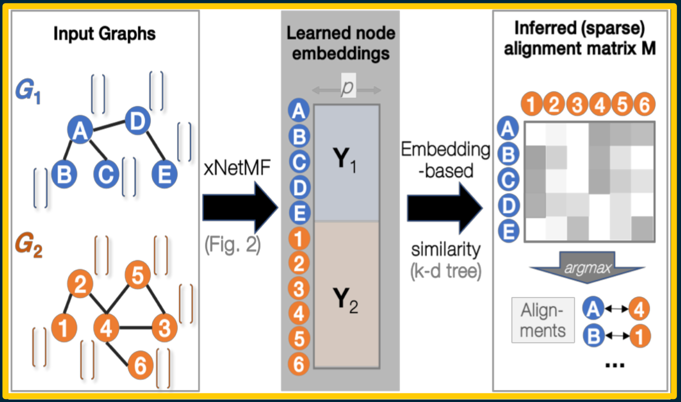
- Implicit matrix factorization allows for speed
- Summarization for greater space efficiency
    - ✧ Global and local structures (graph summarization)
    - ✧ Individual element encoding (node embeddings)

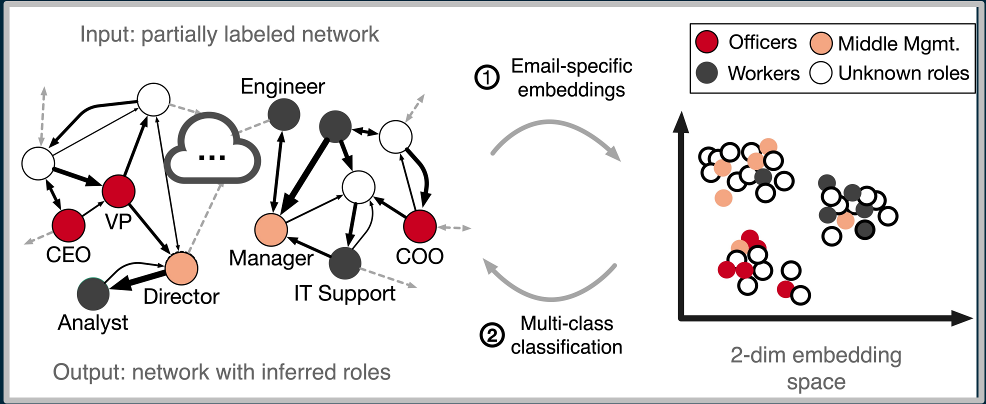


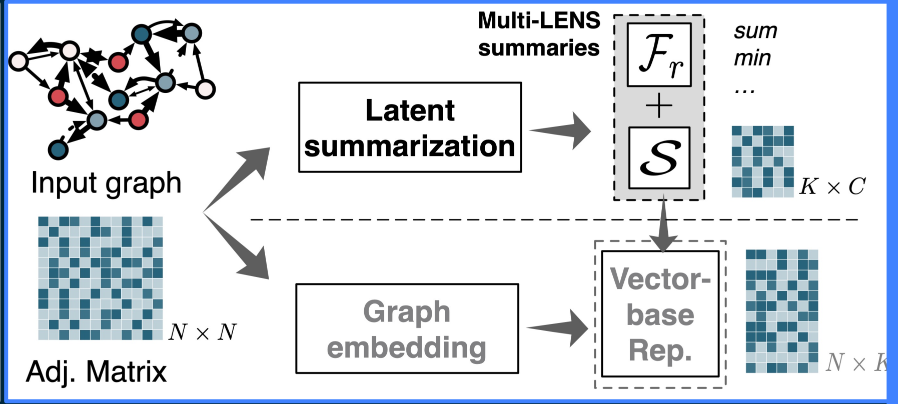


Thank you! Questions?

Structural Embeddings in Large-scale Networks







https://github.com/GemsLab/REGAL

</>

https://github.com/GemsLab/MultiLENS

https://github.com/GemsLab/EMBER

http://danaikoutra.com

dkoutra@umich.edu