Learning Representations for Information-rich Graphs

PhD Proposal

Zexi Huang

Committee: Ambuj Singh (Chair), Yu-Xiang Wang, Noah Friedkin, Xifeng Yan

Department of Computer Science, University of California, Santa Barbara

June 2, 2022

Representation learning: extract useful information from data

For images:

e.g., convolutional neural networks (CNNs)

For text:

e.g., skip-gram, transformers

For graphs:

- node embedding
- graph neural networks (GNNs)

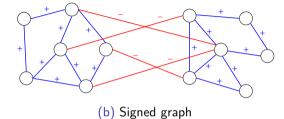
Applications: social network analysis (node classification and community detection), product recommendation (link prediction), fraud detection (anomaly detection), drug discovery (graph classification), ...



(a) Multiscale graph

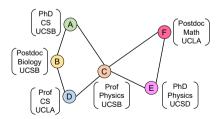


(a) Multiscale graph

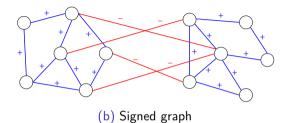




(a) Multiscale graph



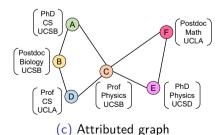
(c) Attributed graph



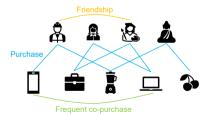
Zexi Huang



(a) Multiscale graph



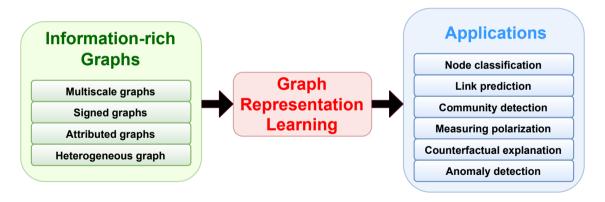
(b) Signed graph

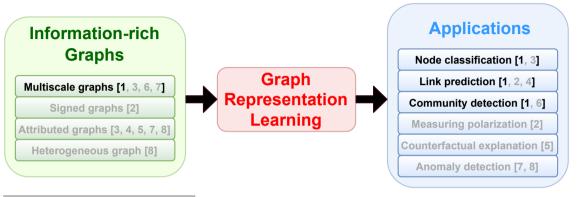


(d) Heterogeneous graph

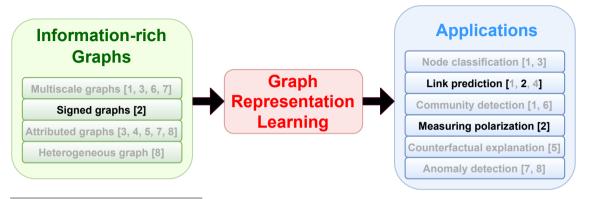
3/53

My research



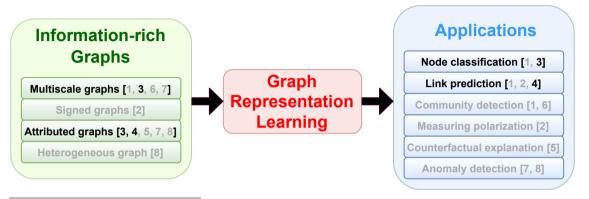


¹Huang, Silva, Singh. A broader picture of random-walk based graph embedding. KDD'21.



¹Huang, Silva, Singh. A broader picture of random-walk based graph embedding. KDD'21.

²Huang. Silva. Singh. POLE: Polarized embedding for signed networks. WSDM'22.

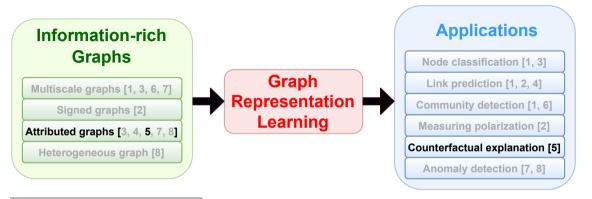


¹Huang, Silva, Singh. A broader picture of random-walk based graph embedding. KDD'21.

²Huang, Silva, Singh. POLE: Polarized embedding for signed networks. WSDM'22.

³Ye, **Huang**, Hong, Singh. Graph convolutional networks meet neural diffusions. Under review.

⁴Huang, Kosan, Silva, Singh. Link prediction without graph neural networks. Under review.



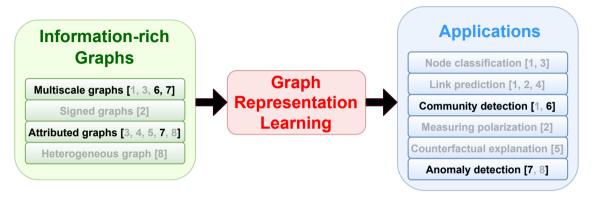
¹Huang, Silva, Singh. A broader picture of random-walk based graph embedding. KDD'21.

²Huang, Silva, Singh. POLE: Polarized embedding for signed networks. WSDM'22.

³Ye, **Huang**, Hong, Singh. Graph convolutional networks meet neural diffusions. Under review.

⁴Huang, Kosan, Silva, Singh. Link prediction without graph neural networks. Under review.

⁵Huang*, Kosan*, et al. GCFExplainer: Global counterfactual explainer for graph neural networks. Ongoing.



¹Huang, Silva, Singh. A broader picture of random-walk based graph embedding. KDD'21.

²Huang, Silva, Singh. POLE: Polarized embedding for signed networks. WSDM'22.

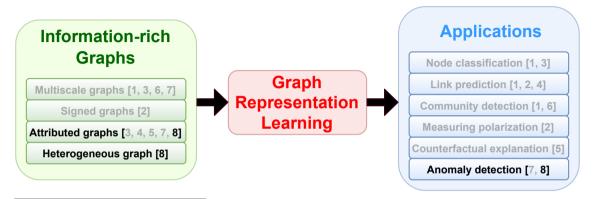
³Ye, **Huang**, Hong, Singh. Graph convolutional networks meet neural diffusions. Under review.

⁴Huang, Kosan, Silva, Singh. Link prediction without graph neural networks. Under review.

⁵Huang*, Kosan*, et al. GCFExplainer: Global counterfactual explainer for graph neural networks. Ongoing.

⁶Kondapaneni*, **Huang***, Silva, Singh. Multiscale community detection with pointwise mutual information. Ongoing.

⁷Arriola, Kosan, **Huang**, Singh. Multiscale anomaly detection with graph autoencoders. Ongoing.



¹Huang, Silva, Singh. A broader picture of random-walk based graph embedding. KDD'21.

²**Huang**, Silva, Singh. POLE: Polarized embedding for signed networks. WSDM'22.

³Ye, **Huang**, Hong, Singh. Graph convolutional networks meet neural diffusions. Under review.

⁴**Huang**, Kosan, Silva, Singh. Link prediction without graph neural networks. Under review.

⁵Huang*, Kosan*, et al. GCFExplainer: Global counterfactual explainer for graph neural networks. Ongoing.

⁶Kondapaneni*, **Huang***, Silva, Singh. Multiscale community detection with pointwise mutual information. Ongoing.

⁷Arriola, Kosan, **Huang**, Singh. Multiscale anomaly detection with graph autoencoders. Ongoing.

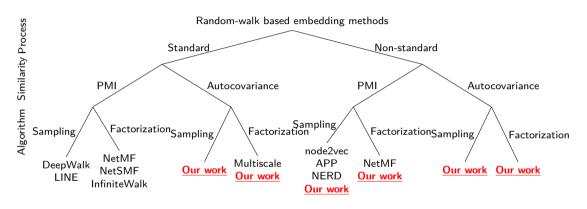
⁸Huang. Graph-based Fraud Detection in Kindle Direct Publishing. Amazon internship report.

Outline

- 2. Representation learning for multiscale graphs
- 3. Representation learning for signed graphs
- 4. Representation learning for attributed graphs
- 5. Conclusions and plan

- 2. Representation learning for multiscale graphs
- 2.1 A broader picture of random-walk based graph embedding
- 2.2 Multiscale community detection with pointwise mutual information
- 2.3 Multiscale anomaly detection with graph autoencoders
- 3. Representation learning for signed graphs
- 4. Representation learning for attributed graphs
- 5. Conclusions and plan

An unified framework for random-walk based embedding¹



- We provide key insights on how embedding captures structural scales.
- We find that Autocovariance enables state-of-the-art link prediction.

¹Huang, Silva, Singh. A broader picture of random-walk based graph embedding. KDD'21.

- 2. Representation learning for multiscale graphs
- 2.1 A broader picture of random-walk based graph embedding
- 2.2 Multiscale community detection with pointwise mutual information
- 2.3 Multiscale anomaly detection with graph autoencoders
- 3. Representation learning for signed graphs
- 4. Representation learning for attributed graphs
- 5. Conclusions and plan

PMI-based multiscale community detection²

- Stability [1, 2]: multiscale community detection based on clustered autocovariance.
- Worse node-level performance.
- ▶ Needs to know the right scale.
- Assumes a shared structural scale across different communities.

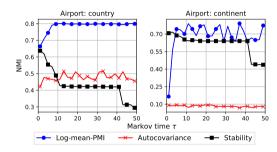


Figure: PMI embedding outperforms both methods based on autocovariance in community detection.

Zexi Huang Graph Representation Learning June 2, 2022 10/53

²Kondapaneni*, Huang*, Silva, Singh. Multiscale community detection with pointwise mutual information. Ongoing.

Preliminary finding

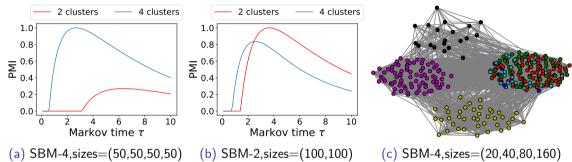


Figure: Multiscale community detection with PMI. (a) and (b) show that PMI identifies ground-truth communities and reveals their structural scales (as the best Markov time).

Ongoing work

- Developing the theory behind the scale-revealing property of PMI.
- Designing algorithms that find communities of different scales (c).

- 2. Representation learning for multiscale graphs
- 2.1 A broader picture of random-walk based graph embedding
- 2.2 Multiscale community detection with pointwise mutual information
- 2.3 Multiscale anomaly detection with graph autoencoders
- 3. Representation learning for signed graphs
- 4. Representation learning for attributed graphs
- 5. Conclusions and plan

Multiscale anomaly detection

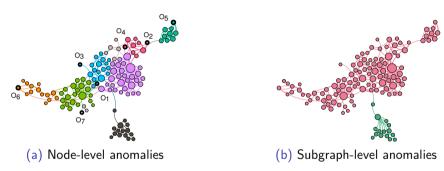
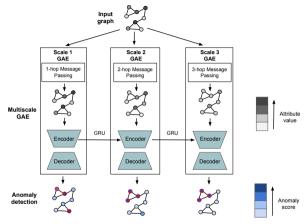


Figure: Real-world graphs have anomalies of different scales. Images from [3].

Existing work focuses on **node-level** anomaly within a particular context [4, 5] or multiscale contexts [3, 6].

Multiscale graph autoencoder³



Ongoing work

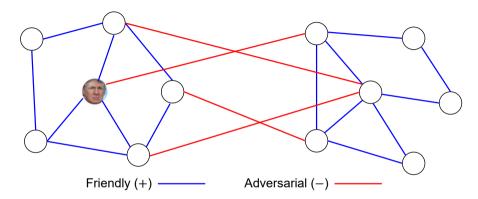
Evaluating our model against baselines on multiple datasets.

³Arriola, Kosan, **Huang**, Singh. Multiscale anomaly detection with graph autoencoders. Ongoing.

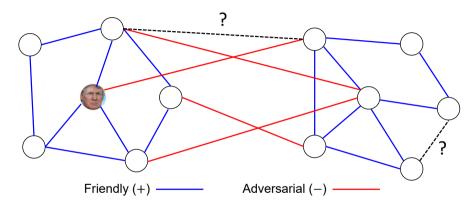
14 / 53

- 2. Representation learning for multiscale graphs
- 3. Representation learning for signed graphs
- 3.1 Polarized embedding for signed networks
- 4. Representation learning for attributed graphs
- 5. Conclusions and plan

Signed networks



Signed networks



Can we predict future conflicts in signed networks to reduce further polarization?

Signed link prediction in polarized networks

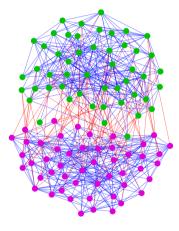
- Predicting signs of links:
 - ► Signed embedding [7, 8] (signed similarity)

Signed link prediction in polarized networks

- Predicting signs of links:
 - ► Signed embedding [7, 8] (signed similarity)
- What about predicting link existence?
 - Unsigned embedding [9, 10] (connectivity)

Signed link prediction in polarized networks

- Predicting signs of links:
 - ► Signed embedding [7, 8] (signed similarity)
- What about predicting link existence?
 - Unsigned embedding [9, 10] (connectivity)
- However, they cannot predict negative links between polarized communities!
 - ▶ Because topology and link signs are interdependent
 - ▶ Need to capture signed/unsigned similarities jointly



Intra-community: dense, positive Inter-community: sparse, negative

POLE: polarized embedding⁴

▶ Signed random-walks to capture both similarities:

$$M_{uv}(t) = \sum_{ ext{all length-} t ext{ paths } l ext{ between } u ext{ and } v} ext{Prob}(l) ext{Sign}(l)$$

▶ POLE: extends autocovariance similarity [11, 12] to signed RW

$$R(t) = M(t)^T W M(t)$$

⁴**Huang**, Silva, Singh. POLE: Polarized embedding for signed networks. WSDM'22.

Zexi Huang

POLE: polarized embedding⁴

▶ Signed random-walks to capture both similarities:

$$M_{uv}(t) = \sum_{ ext{all length-}t ext{ paths } l ext{ between } u ext{ and } v} ext{Prob}(l) ext{Sign}(l)$$

POLE: extends autocovariance similarity [11, 12] to signed RW

$$R(t) = M(t)^T W M(t)$$

- Desired properties:
 - Positive links: large positive similarity;
 - Negative links: large negative similarity;
 - Non-links: small similarity.

⁴Huang, Silva, Singh. POLE: Polarized embedding for signed networks. WSDM'22.

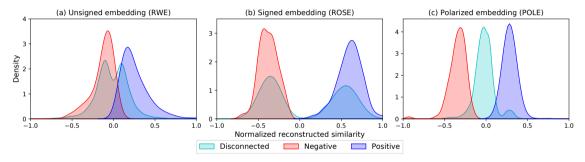


Figure: Distributions of the reconstructed similarity for different types of node pairs in a polarized graph using (a) unsigned [12], (b) signed [13], and (c) polarized embedding.

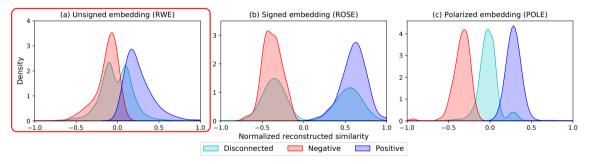


Figure: Distributions of the reconstructed similarity for different types of node pairs in a polarized graph using (a) unsigned [12], (b) signed [13], and (c) polarized embedding.

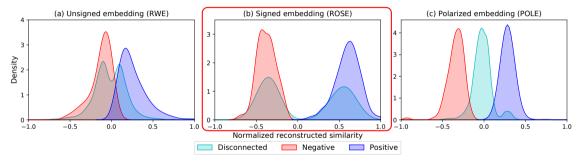


Figure: Distributions of the reconstructed similarity for different types of node pairs in a polarized graph using (a) unsigned [12], (b) signed [13], and (c) polarized embedding.

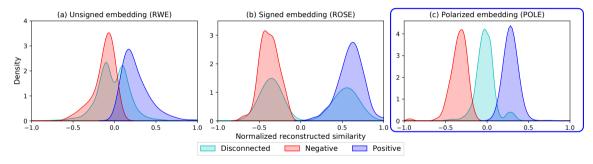


Figure: Distributions of the reconstructed similarity for different types of node pairs in a polarized graph using (a) unsigned [12], (b) signed [13], and (c) polarized embedding.

Signed link prediction results

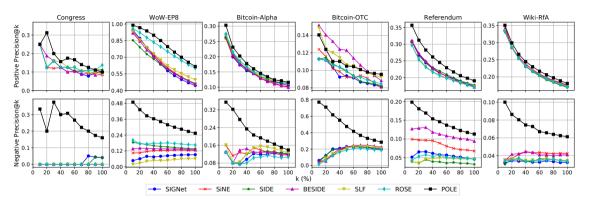
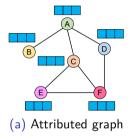
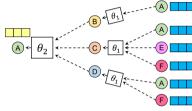


Figure: Signed link prediction with link existence information performance comparison. POLE outperforms all baselines in almost all datasets, especially for the negative links.

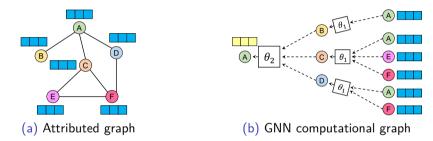
- 2. Representation learning for multiscale graphs
- 3. Representation learning for signed graphs
- 4. Representation learning for attributed graphs
- 4.1 Graph convolutional networks meet neural diffusions
- 4.2 Link prediction without graph neural networks
- 4.3 Global counterfactual explainer for graph neural networks
- 5. Conclusions and plan

Graph neural networks (GNNs) [14, 15]





Graph neural networks (GNNs) [14, 15]



How can we effectively leverage multi-hop neighborhood information in GNNs?

Graph neural diffusion network⁵

► Message-passing as graph diffusions:

$$u^{(K)} = \sum_{k=0}^{K-1} \alpha_k M^k u^{(0)}$$

 \triangleright Existing work [16, 17] adopts fixed diffusion weights α_k .

Zexi Huang Graph Representation Learning June 2, 2022

23 / 53

⁵Ye, **Huang**, Hong, Singh. Graph convolutional networks meet neural diffusions. Under review.

Graph neural diffusion network⁵

► Message-passing as graph diffusions:

$$u^{(K)} = \sum_{k=0}^{K-1} \alpha_k M^k u^{(0)}$$

- \blacktriangleright Existing work [16, 17] adopts fixed diffusion weights α_k .
- ► We propose to **learn** the diffusion weights directly from data:

$$u^{(K)} = f([u^{(0)}; Mu^{(0)}; ...; M^{K-1}u^{(0)}]; \theta)$$

➤ Key result: The learned weights are **adaptable** to different datasets, leading to better semi-supervised node classification performance.

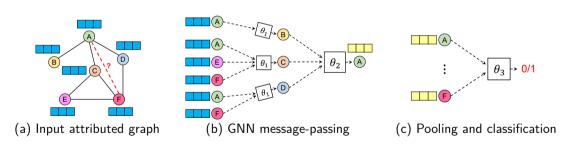
⁵Ye, **Huang**, Hong, Singh. Graph convolutional networks meet neural diffusions. Under review.

1. Introduction

- 2. Representation learning for multiscale graphs
- 3. Representation learning for signed graphs
- 4. Representation learning for attributed graphs
- 4.1 Graph convolutional networks meet neural diffusions
- 4.2 Link prediction without graph neural networks
- 4.3 Global counterfactual explainer for graph neural networks
- 5. Conclusions and plan

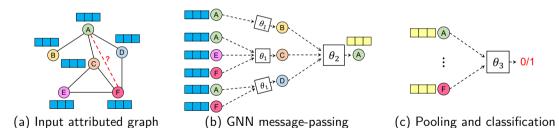
GNNs for link prediction

- ► GNNs have become the predominant tool for link prediction:
 - ► LGCN (WebConf'21), TLC-GNN (ICML'21), Neo-GNN (NeurIPS'21), NBFNet (NeurIPS'21), BScNets (AAAI'22), WalkPool (ICLR'22)



GNNs for link prediction

- ▶ GNNs have become the predominant tool for link prediction:
 - ► LGCN (WebConf'21), TLC-GNN (ICML'21), Neo-GNN (NeurIPS'21), NBFNet (NeurIPS'21), BScNets (AAAI'22), WalkPool (ICLR'22)



- ► Advantages over topological heuristics (e.g., Common Neighbors):
 - ▶ Potential to discover new heuristics via **supervised learning**.
 - Natural incorporation of node attribute information.

An imbalanced classification problem

Table: Common real-world datasets for link prediction benchmark.

	#Nodes	#Edges	Avg. degree	Density	Class ratio
Cora	2,708	5,278	3.90	0.14%	1:695
CITESEER	3,327	4,552	2.74	0.08%	1:1216
PubMed	19,717	44,324	4.50	0.02%	1:4385
Рното	7,650	119,081	31.13	0.41%	1:246
Computers	13,752	245,861	35.76	0.26%	1:385

An imbalanced classification problem

Table: Common real-world datasets for link prediction benchmark.

	#Nodes	#Edges	Avg. degree	Density	Class ratio
Cora	2,708	5,278	3.90	0.14%	1:695
CiteSeer	3,327	4,552	2.74	0.08%	1:1216
PubMed	19,717	44,324	4.50	0.02%	1:4385
Рното	7,650	119,081	31.13	0.41%	1:246
Computers	13,752	245,861	35.76	0.26%	1:385

Have GNN-based link prediction methods properly addressed the intrinsic class imbalance?

A task for the graph topology

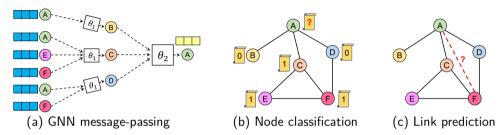


Figure: The attribute-centric message-passing mechanism is effective for tasks **on** the topology, e.g., node classification. Link prediction, however, is a task **for** the topology.

A task for the graph topology

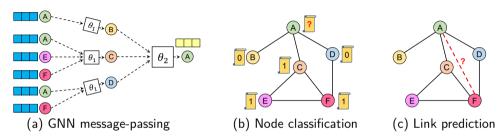


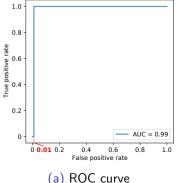
Figure: The attribute-centric message-passing mechanism is effective for tasks **on** the topology, e.g., node classification. Link prediction, however, is a task **for** the topology.

Are there better alternatives to message-passing for combining node attributes and graph topology for link prediction?

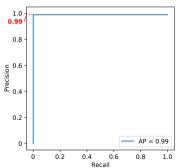
► Existing work [18, 19, 20, 21, 22, 23, 24, 25, 26] adopts AUC and AP with *biased testing* (downsampling negative/disconnected pairs).

► Existing work [18, 19, 20, 21, 22, 23, 24, 25, 26] adopts AUC and AP with *biased testing* (downsampling negative/disconnected pairs).

Example: A bad link prediction model that predicts **1M** false positives (**1k** with *biased testing*) higher than the **100k** true edges achieves **0.99** in both AUC and AP.



Zexi Huang



28 / 53

ROC curve (b) PR curve

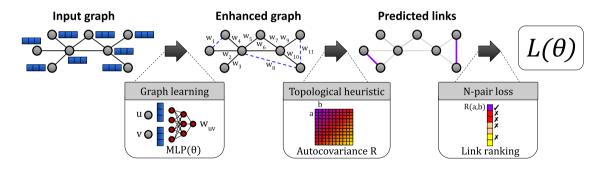
- ► Existing work [18, 19, 20, 21, 22, 23, 24, 25, 26] adopts AUC and AP with *biased testing* (downsampling negative/disconnected pairs), which pictures an **overly optimistic** view of model performance.
- We argue for the use of **precision**@k [27] with *full testing*, which has been widely applied in unsupervised link prediction [28, 29, 12] and IR.

- ► Existing work [18, 19, 20, 21, 22, 23, 24, 25, 26] adopts AUC and AP with *biased testing* (downsampling negative/disconnected pairs), which pictures an **overly optimistic** view of model performance.
- ▶ We argue for the use of **precision@***k* [27] with *full testing*, which has been widely applied in unsupervised link prediction [28, 29, 12] and IR.

Supervised link prediction training

- Existing work uses binary cross entropy loss with *biased training*.
 - lt **discards** potentially useful evidence from negative pairs.
 - lt induces the model to **overestimate** the probability of positive pairs.

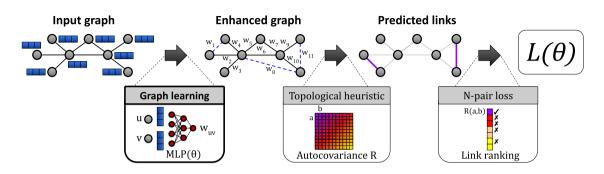
Gelato: a simpler, more effective, and faster alternative⁵

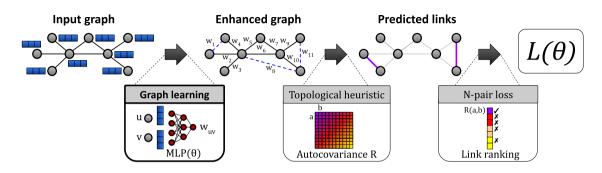


Zexi Huang Graph Representation Learning June 2, 2022

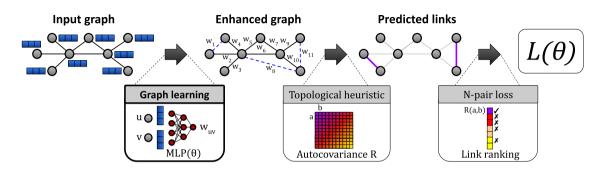
30 / 53

⁵Huang, Kosan, Silva, Singh, Link prediction without graph neural networks. Under review.

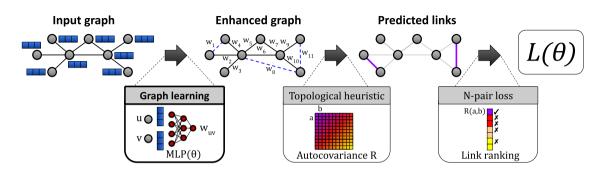




▶ Graph augmentation: $\widetilde{E} = E + \{(u, v) \mid s(x_u, x_v) > s_\eta\}$

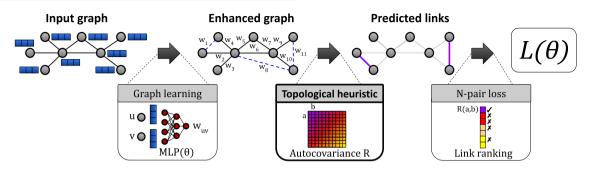


- ▶ Graph augmentation: $\widetilde{E} = E + \{(u, v) \mid s(x_u, x_v) > s_n\}$
- ► Trained weighting: $w_{uv} = \text{MLP}([x_u + x_v; |x_u x_v|]; \theta)$



- ▶ Graph augmentation: $\widetilde{E} = E + \{(u, v) \mid s(x_u, x_v) > s_\eta\}$
- ► Trained weighting: $w_{uv} = \text{MLP}([x_u + x_v; |x_u x_v|]; \theta)$
- ► Combined weights: $\widetilde{A}_{uv} = \alpha A_{uv} + (1 \alpha)(\beta w_{uv} + (1 \beta)s(x_u, x_v))$

Zexi Huang



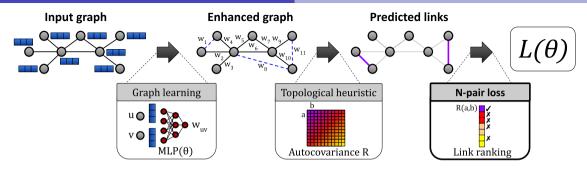
Topological heuristic

▶ Applying Autocovariance [1, 12] to the enhanced graph \widetilde{A} :

$$R = \frac{\widetilde{D}}{\operatorname{vol}(\widetilde{G})} (\widetilde{D}^{-1} \widetilde{A})^t - \frac{\widetilde{d}\widetilde{d}^T}{\operatorname{vol}^2(\widetilde{G})}$$

Zexi Huang

Graph Representation Learning



N-pair loss [30]

Contrasting each positive edge (u, v) with a set of negative pairs N(u, v) whose size equals to the class ratio (full training):

$$L(\theta) = -\sum_{(u,v)\in E} \log \frac{\exp(R_{uv})}{\exp(R_{uv}) + \sum_{(p,q)\in N(u,v)} \exp(R_{pq})}$$

Zexi Huang

Baselines

- GNN-based methods:
 - GAE [18], SEAL [19], HGCN [20], LGCN [21], TLC-GNN [23], Neo-GNN [31], NBFNet [24], BScNets [25], and WalkPool [26]
- ► Topological heuristics:
 - Common Neighbors (CN) [32], Adamic Adar (AA) [33], Resource Allocation (RA) [34], and Autocovariance (AC) [12]
- ► Two-stage methods of combining attributes and topology:
 - ► MLP, Cos, MLP+AC, Cos+AC, MLP+Cos+AC

Evaluation metric

ightharpoonup Precision@k under full testing for different values of k in terms of the ratio of testing edges

* Run only once	: as each run takes ~	·100 hrs; *** Ead	ch run takes $>\!10$	00 hrs; OOM:	Out Of Memory.	
		Cora	CITESEER	PubMed	Рното	Computers
	GAE	3.33 ± 0.53	3.93 ± 0.47	2.58 ± 0.19	12.76 ± 0.30	10.81 ± 0.14
	SEAL	6.11 ± 1.16	3.12 ± 2.03	***	17.11 ± 1.01	13.09^*
	HGCN	3.47 ± 0.42	3.56 ± 0.59	2.39 ± 0.10	7.19 ± 0.47	5.73 ± 0.27
	LGCN	4.74 ± 0.31	3.47 ± 0.54	2.73 ± 0.12	8.20 ± 0.26	4.74 ± 0.18
GNN	TLC-GNN	0.78 ± 0.45	0.70 ± 0.51	OOM	1.80 ± 0.54	OOM
	Neo-GNN	6.79 ± 1.79	5.67 ± 1.27	5.41 ± 0.46	19.09 ± 1.62	13.93*
	NBFNet	4.59 ± 0.67	2.29 ± 0.51	***	20.41 ± 1.48	***
	BScNets	0.59 ± 0.30	0.44 ± 0.33	0.58 ± 0.36	3.04 ± 0.40	1.67 ± 0.19
	WalkPool	5.29 ± 0.30	4.44 ± 0.31	4.42*	OOM	OOM

Run only once a	is each run takes ~	-100 hrs; *** Ea	ch run takes $>$ 10	00 hrs; OOM:	Out Of Memory.	
		Cora	CITESEER	РивМер	Рното	Computers
	GAE	3.33 ± 0.53	3.93 ± 0.47	2.58 ± 0.19	12.76 ± 0.30	10.81 ± 0.14
	SEAL	6.11 ± 1.16	3.12 ± 2.03	***	17.11 ± 1.01	13.09*
	HGCN	3.47 ± 0.42	3.56 ± 0.59	2.39 ± 0.10	7.19 ± 0.47	5.73 ± 0.27
	LGCN	4.74 ± 0.31	3.47 ± 0.54	2.73 ± 0.12	8.20 ± 0.26	4.74 ± 0.18
GNN	TLC-GNN	0.78 ± 0.45	0.70 ± 0.51	OOM	1.80 ± 0.54	OOM
0.1.1	Neo-GNN	6.79 ± 1.79	5.67 ± 1.27	5.41 ± 0.46	19.09 ± 1.62	13.93*
	NBFNet	4.59 ± 0.67	2.29 ± 0.51	***	20.41 ± 1.48	***
	BScNets	0.59 ± 0.30	0.44 ± 0.33	0.58 ± 0.36	3.04 ± 0.40	1.67 ± 0.19
	WalkPool	5.29 ± 0.30	4.44 ± 0.31	4.42*	OOM	OOM
Topological Heuristics	CN	4.36 ± 0.00	4.40 ± 0.00	2.37 ± 0.00	16.04 ± 0.00	13.39 ± 0.0
	AA	7.40 ± 0.00	4.40 ± 0.00	3.32 ± 0.00	18.37 ± 0.00	14.71 ± 0.0
	RA	7.21 ± 0.00	4.18 ± 0.00	2.10 ± 0.00	19.37 ± 0.00	15.22 ± 0.0
	AC	8.16 ± 0.00	8.35 ± 0.00	8.33 ± 0.00	24.40 ± 0.00	19.78 ± 0.0

* Run only once	as each run takes ~:	100 hrs; *** Ea	ch run takes $>\!100$	00 hrs; OOM:	Out Of Memory.	
		Cora	CITESEER	PubMed	Рното	Computers
	GAE	3.33 ± 0.53	3.93 ± 0.47	2.58 ± 0.19	12.76 ± 0.30	10.81 ± 0.14
	SEAL	6.11 ± 1.16	3.12 ± 2.03	***	17.11 ± 1.01	13.09 [*]
	HGCN	3.47 ± 0.42	3.56 ± 0.59	2.39 ± 0.10	7.19 ± 0.47	5.73 ± 0.27
	LGCN	4.74 ± 0.31	3.47 ± 0.54	2.73 ± 0.12	8.20 ± 0.26	4.74 ± 0.18
GNN	TLC-GNN	0.78 ± 0.45	0.70 ± 0.51	OOM	1.80 ± 0.54	OOM
	Neo-GNN	6.79 ± 1.79	5.67 ± 1.27	5.41 ± 0.46	19.09 ± 1.62	13.93*
	NBFNet	4.59 ± 0.67	2.29 ± 0.51	***	20.41 ± 1.48	***
	BScNets	0.59 ± 0.30	0.44 ± 0.33	0.58 ± 0.36	3.04 ± 0.40	1.67 ± 0.19
	WalkPool	5.29 ± 0.30	4.44 ± 0.31	4.42*	ООМ	ООМ
	CN	4.36 ± 0.00	4.40 ± 0.00	2.37 ± 0.00	16.04 ± 0.00	13.39 ± 0.00
Topological	AA	7.40 ± 0.00	4.40 ± 0.00	3.32 ± 0.00	18.37 ± 0.00	14.71 ± 0.00
Heuristics	RA	7.21 ± 0.00	4.18 ± 0.00	2.10 ± 0.00	19.37 ± 0.00	15.22 ± 0.00
ricuristics	AC	8.16 ± 0.00	8.35 ± 0.00	8.33 ± 0.00	24.40 ± 0.00	19.78 ± 0.00
	MLP	4.99 ± 1.10	3.45 ± 1.16	0.95 ± 0.35	3.73 ± 0.78	1.64 ± 0.76
	Cos	3.23 ± 0.00	8.35 ± 0.00	0.50 ± 0.00	0.43 ± 0.00	0.33 ± 0.00
Attributes +	MLP + AC	8.60 ± 0.25	5.71 ± 0.37	8.57 ± 0.13	23.62 ± 0.22	19.39 ± 0.17
Topology	Cos + AC	11.01 ± 0.00	12.53 ± 0.00	0.65 ± 0.00	16.94 ± 0.00	8.46 ± 0.00
	MLP+Cos+AC	8.88 ± 0.23	11.67 ± 0.79	0.81 ± 0.10	18.47 ± 0.16	9.65 ± 0.25

* Don and a second supplied to 100 hours | *** Factor to 1000 hours | OOM Out Of Management

* Run only once as each run takes \sim 100 hrs; *** Each run takes $>$ 1000 hrs; OOM: Out Of Memory.						
		Cora	CITESEER	PubMed	Рното	Computers
	GAE	3.33 ± 0.53	3.93 ± 0.47	2.58 ± 0.19	12.76 ± 0.30	10.81 ± 0.14
	SEAL	6.11 ± 1.16	3.12 ± 2.03	***	17.11 ± 1.01	13.09 [*]
	HGCN	3.47 ± 0.42	3.56 ± 0.59	2.39 ± 0.10	7.19 ± 0.47	5.73 ± 0.27
	LGCN	4.74 ± 0.31	3.47 ± 0.54	2.73 ± 0.12	8.20 ± 0.26	4.74 ± 0.18
GNN	TLC-GNN	0.78 ± 0.45	0.70 ± 0.51	OOM	1.80 ± 0.54	OOM
	Neo-GNN	6.79 ± 1.79	5.67 ± 1.27	5.41 ± 0.46	19.09 ± 1.62	13.93*
	NBFNet	4.59 ± 0.67	2.29 ± 0.51	***	20.41 ± 1.48	***
	BScNets	0.59 ± 0.30	0.44 ± 0.33	0.58 ± 0.36	3.04 ± 0.40	1.67 ± 0.19
	WalkPool	5.29 ± 0.30	4.44 ± 0.31	4.42*	OOM	ООМ
	CN	4.36 ± 0.00	4.40 ± 0.00	2.37 ± 0.00	16.04 ± 0.00	13.39 ± 0.00
Topological	AA	7.40 ± 0.00	4.40 ± 0.00	3.32 ± 0.00	18.37 ± 0.00	14.71 ± 0.00
Heuristics	RA	7.21 ± 0.00	4.18 ± 0.00	2.10 ± 0.00	19.37 ± 0.00	15.22 ± 0.00
rieuristics	AC	8.16 ± 0.00	8.35 ± 0.00	8.33 ± 0.00	24.40 ± 0.00	19.78 ± 0.00
	MLP	4.99 ± 1.10	3.45 ± 1.16	0.95 ± 0.35	3.73 ± 0.78	1.64 ± 0.76
	Cos	3.23 ± 0.00	8.35 ± 0.00	0.50 ± 0.00	0.43 ± 0.00	0.33 ± 0.00
Attributes $+$	MLP + AC	8.60 ± 0.25	5.71 ± 0.37	8.57 ± 0.13	23.62 ± 0.22	19.39 ± 0.17
Topology	Cos + AC	11.01 ± 0.00	12.53 ± 0.00	0.65 ± 0.00	16.94 ± 0.00	8.46 ± 0.00
	MLP+Cos+AC	8.88 ± 0.23	11.67 ± 0.79	0.81 ± 0.10	18.47 ± 0.16	9.65 ± 0.25
Gelato 11.		11.67 ± 0.30	13.43 ± 0.22	9.35 ± 0.16	32.13 ± 0.45	26.68 ± 0.19

Link prediction results (cont.)

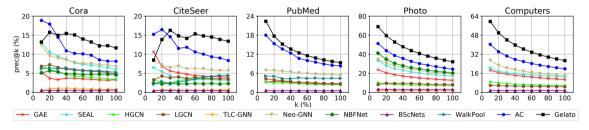
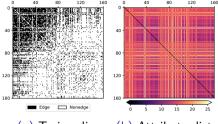


Figure: Link prediction performance in terms of precision@k with k ranging from 10% to 100%. With few exceptions, Gelato outperforms the baselines across different k.



(a) Train adj. (b) Attribute dist.

Figure: Illustration of the link prediction process of Gelato on a subgraph of Photo.

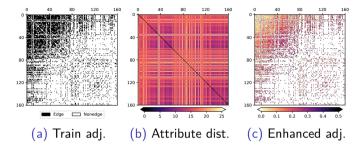


Figure: Illustration of the link prediction process of Gelato on a subgraph of Photo.

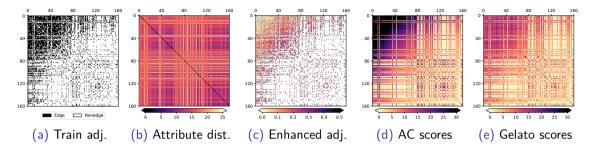


Figure: Illustration of the link prediction process of Gelato on a subgraph of Photo.

Zexi Huang

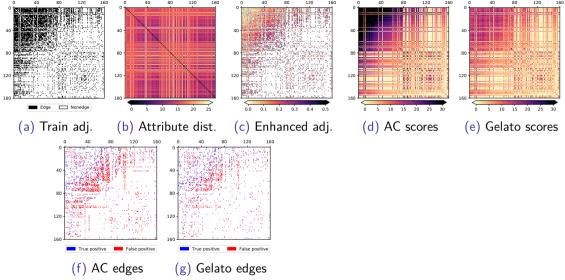


Figure: Illustration of the link prediction process of Gelato on a subgraph of Photo.

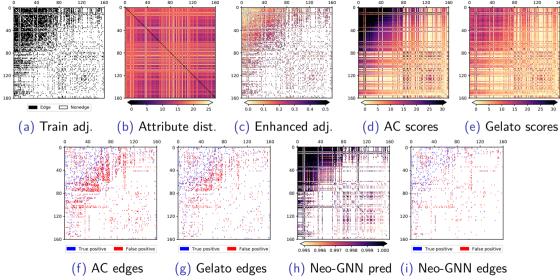


Figure: Illustration of the link prediction process of Gelato on a subgraph of Photo.

Comparing loss and training setting

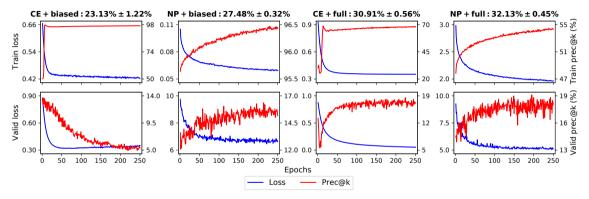


Figure: Compared with the cross entropy loss, the N-pair loss with *full training* is a more consistent proxy for precision at the top and leads to better peak performance.

Running time comparison

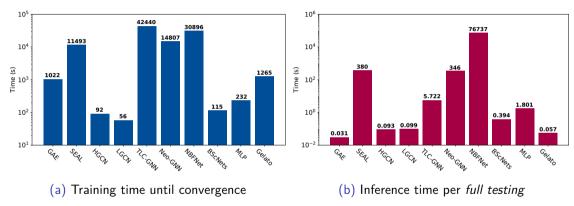


Figure: Even under full training, Gelato has competitive training time ($11 \times$ compared to Neo-GNN) and is significantly faster than most baselines for inference $(6,000\times)$.

39 / 53

This work investigates key questions regarding the training, evaluation, and ability of GNNs to effectively combine attributes and topology for link prediction.

Contributions:

- ▶ Identify limitations of existing methods in addressing class imbalance.
- Introduce a simpler, more accurate, and more efficient alternative.
- ▶ Propose the use of N-pair loss with full training for link prediction.

1. Introduction

- 2. Representation learning for multiscale graphs
- 3. Representation learning for signed graphs
- 4. Representation learning for attributed graphs
- 4.1 Graph convolutional networks meet neural diffusions
- 4.2 Link prediction without graph neural networks
- 4.3 Global counterfactual explainer for graph neural networks
- 5. Conclusions and plan

Model understanding and explanation

Deep predictive model

```
{Foreigner=True, Num_Loans=1, Missed_Payments=2} \Rightarrow Denied
```

Missed Payments

Model understanding and explanation

Deep predictive model

► Feature importance [35, 36]

```
{Foreigner=True, Num_Loans=1, Missed_Payments=2} ⇒ Denied

Foreigner
Num Loans
```

Importance

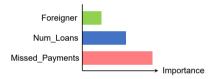
Model understanding and explanation

Deep predictive model

► Feature importance [35, 36]

► Local counterfactual [37, 38]

```
{Foreigner=True, Num_Loans=1, Missed_Payments=2} ⇒ Denied
```



{Foreigner=True, Num_Loans=1, Missed_Payments=1} \Rightarrow Approved

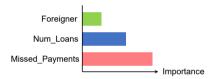
Model understanding and explanation

Deep predictive model

► Feature importance [35, 36]

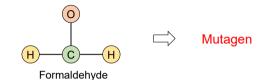
- ► Local counterfactual [37, 38]
- Global counterfactual [39]

```
{Foreigner=True, Num_Loans=1, Missed Payments=2} ⇒ Denied
```



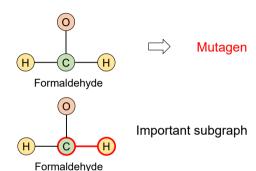
{Foreigner=True, Num_Loans=1, Missed_Payments=1}
$$\Rightarrow$$
 Approved

► Graph classification GNN



Graph classification GNN

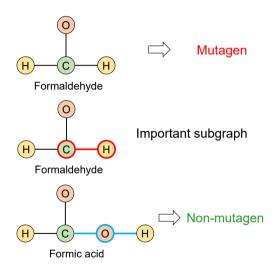
► Subgraph importance [40, 41]



► Graph classification GNN

► Subgraph importance [40, 41]

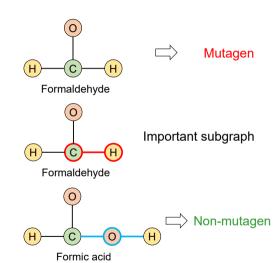
► Local counterfactual [42, 43]



► Graph classification GNN

► Subgraph importance [40, 41]

► Local counterfactual [42, 43]



Can we generate global counterfactual explanation for GNNs?

Global counterfactual explanation for GNNs⁷

- ▶ Global recourse: For any undesired graph $G \in \mathcal{G}$ (GNN(G) = 0), the explanation r should provide a recourse: $\mathbf{GNN}(r(G)) = 1$.
- \blacktriangleright Interpretable: r should be (much) easier to understand (than **GNN**).

Zexi Huang

44 / 53

⁷Huang*, Kosan*, et al. GCFExplainer: Global counterfactual explainer for graph neural networks. Ongoing.

Global counterfactual explanation for GNNs⁷

- ▶ Global recourse: For any undesired graph $G \in \mathcal{G}$ (GNN(G) = 0), the explanation r should provide a recourse: GNN(r(G)) = 1.
- ▶ Interpretable: r should be (much) easier to understand (than **GNN**).

Explanation based on counterfactual summary

- ightharpoonup Represent r with a set $\mathcal C$ of counterfactual summary graphs.
- ightharpoonup The recourse for G is given as the minimal cost summary graph:

$$r_{\mathcal{C}}(G) = \underset{C \in \mathcal{C}}{\operatorname{arg\,min\,cost}}(G, C)$$

where $cost(\cdot, \cdot)$ is any distance metric between graphs (e.g., GED).

ang Graph Representation Learning

44 / 53

⁷Huang*, Kosan*, et al. GCFExplainer: Global counterfactual explainer for graph neural networks. Ongoing.

Quantifying explanation quality

► Cost: minimize the overall recourse cost for all undesired graphs:

$$\mathbf{cost}(r_{\mathcal{C}}) = \sum_{G \in \mathcal{G}} \min_{C \in \mathcal{C}} \cot(G, C)$$

Coverage: maximize the number of undesired graphs that have an actionable recourse (i.e., within a cost budget B):

$$\mathbf{cover}(r_{\mathcal{C}}) = |\{G \in \mathcal{G} \mid \min_{C \in \mathcal{C}} \mathrm{cost}(G, C) \leq B\}|$$

Interpretability: minimize the size of the counterfactual summary:

$$size(r_{\mathcal{C}}) = |\mathcal{C}|$$

Quantifying explanation quality

► Cost: minimize the overall recourse cost for all undesired graphs:

$$\mathbf{cost}(r_{\mathcal{C}}) = \sum_{G \in \mathcal{G}} \min_{C \in \mathcal{C}} \cot(G, C)$$

► Coverage: maximize the number of undesired graphs that have an actionable recourse (i.e., within a cost budget B):

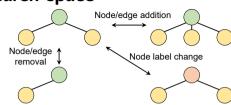
$$\mathbf{cover}(r_{\mathcal{C}}) = |\{G \in \mathcal{G} \mid \min_{C \in \mathcal{C}} \mathrm{cost}(G, C) \leq B\}|$$

▶ Interpretability: minimize the size of the counterfactual summary:

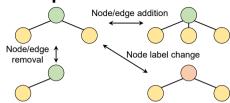
$$\mathsf{size}(r_{\mathcal{C}}) = |\mathcal{C}|$$

How can we find a good set of counterfactual summary graphs?

Graph edit map: A (meta)graph of candidate summary graphs connected by single graph edits.



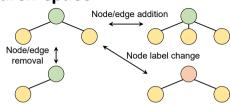
Graph edit map: A (meta)graph of candidate summary graphs connected by single graph edits.



Generating diverse counterfactual summary

▶ Vertex-reinforced random-walk [44]: $Prob(g \rightarrow g') \propto N(g')I(g')$. Converges to diverse (representative), important nodes [45, 46].

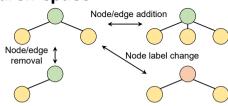
Graph edit map: A (meta)graph of candidate summary graphs connected by single graph edits.



Generating diverse counterfactual summary

- ▶ Vertex-reinforced random-walk [44]: $\operatorname{Prob}(g \to g') \propto N(g')I(g')$. Converges to diverse (representative), important nodes [45, 46].
- ▶ Importance function I(g): counterfactual probability **GNN**(g), individual cost **cost** $(r_{\{g\}})$, and individual coverage **cover** $(r_{\{g\}})$.

Graph edit map: A (meta)graph of candidate summary graphs connected by single graph edits.



Generating diverse counterfactual summary

- ▶ Vertex-reinforced random-walk [44]: $Prob(g \rightarrow g') \propto N(g')I(g')$. Converges to diverse (representative), important nodes [45, 46].
- ▶ Importance function I(g): counterfactual probability $\mathbf{GNN}(g)$, individual cost $\mathbf{cost}(r_{\{g\}})$, and individual coverage $\mathbf{cover}(r_{\{g\}})$.
- ► Teleportation to undesired graphs: manages the exponential search space, increases convergence rate with adaptive probability.

Dataset

▶ Mutagenicity [47]: A collection of 3,000 molecules classified into two categories: mutagen and non-mutagen (desired).

Baselines

- ▶ Ground-truth desired graphs in the dataset + Greedy summary
- ▶ Local counterfactuals from RCExplainer [42] + Greedy summary

Evaluation metrics

- **cost** (r_C) for different **size** (r_C)
- **cover** (r_C) for different **size** (r_C)

Comparison with summary of local counterfactuals

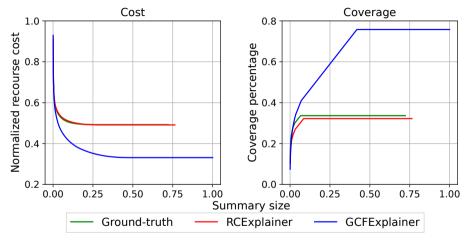


Figure: Global counterfactual quality comparison. GCFExplainer consistently outperforms the baselines with 30% less recourse cost and $2\times$ coverage percentage.

Visualizing global counterfactuals

(a) Undesired mutagenic molecules

(b) RCExplainer local counterfactuals (by removing red edges)

(c) GCFExplainer global counterfactual (by adding/replacing with a sulfur atom)

Figure: Visualization of local and global counterfactuals. The global counterfactual provides a high-level picture of the recourse for the undesired molecules.

Contributions

- We present GCFExplainer, the first global counterfactual explainer for GNNs based on vertex-reinforced random-walks.
- We demonstrate the effectiveness and usefulness of GCFExplainer in providing high-level recourse for GNN-based graph classification.

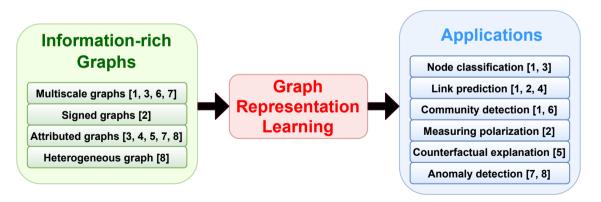
Ongoing work

- ▶ Improving the scalability of the algorithm.
- Finding a more consistent proxy of graph edit distance.
- Comparing with more baselines on more datasets.

1. Introduction

- 2. Representation learning for multiscale graphs
- 3. Representation learning for signed graphs
- 4. Representation learning for attributed graphs
- 5. Conclusions and plan

Graph representation learning leads to new state-of-the-art results for many graph-based applications. We have overviewed our progress towards advancing it, which will form this thesis.



Plan

Sep/22: Finish paper on counterfactual explanation

Dec/22: Finish paper on PMI clustering

Mar/23: Finish paper on anomaly detection

June/23: Thesis defense

Reference I

- J-C Delvenne, Sophia N Yaliraki, and Mauricio Barahona. Stability of graph communities across time scales. PNAS, 107(29):12755–12760, 2010.
- [2] Aurelio Patelli, Andrea Gabrielli, and Giulio Cimini. Generalized markov stability of network communities. Physical Review E, 101(5):052301, 2020.
- [3] Leonardo Gutiérrez-Gómez, Alexandre Bovet, and Jean-Charles Delvenne. Multi-scale anomaly detection on attributed networks.
 In AAAI, 2020.
- [4] Ninghao Liu, Xiao Huang, and Xia Hu. Accelerated local anomaly detection via resolving attributed networks. In IJCAI, 2017.
- [5] Kaize Ding, Jundong Li, and Huan Liu. Interactive anomaly detection on attributed networks. In WSDM, 2019.

Reference II

- [6] Xuexiong Luo, Jia Wu, Amin Beheshti, Jian Yang, Xiankun Zhang, Yuan Wang, and Shan Xue. Comga: Community-aware attributed graph anomaly detection. In WSDM, 2022.
- [7] Junghwan Kim, Haekyu Park, Ji-Eun Lee, and U Kang. Side: representation learning in signed directed networks. In WebConf, 2018.
- [8] Yiqi Chen, Tieyun Qian, Huan Liu, and Ke Sun. "bridge" enhanced signed directed network embedding. In CIKM. 2018.
- [9] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In SIGKDD, 2014.
- [10] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In SIGKDD, 2016.

Reference III

- [11] Michael T Schaub, Jean-Charles Delvenne, Renaud Lambiotte, and Mauricio Barahona. Multiscale dynamical embeddings of complex networks. Physical Review E, 99(6):062308, 2019.
- [12] Zexi Huang, Arlei Silva, and Ambuj Singh. A broader picture of random-walk based graph embedding. In SIGKDD, 2021.
- [13] Amin Javari, Tyler Derr, Pouya Esmailian, Jiliang Tang, and Kevin Chen-Chuan Chang. Rose: Role-based signed network embedding. In WebConf. 2020.
- [14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In ICLR, 2017.
- [15] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In NeurIPS, 2017.

Reference IV

- [16] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In ICLR, 2018.
- [17] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. Diffusion improves graph learning. 2019.
- [18] Thomas N Kipf and Max Welling. Variational graph auto-encoders. arXiv preprint arXiv:1611.07308, 2016.
- [19] Muhan Zhang and Yixin Chen. Link prediction based on graph neural networks. In NeurIPS, 2018.
- [20] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In NeurIPS, 2019.

Reference V

- [21] Yiding Zhang, Xiao Wang, Chuan Shi, Nian Liu, and Guojie Song. Lorentzian graph convolutional networks. In WebConf. 2021.
- [22] Lei Cai, Jundong Li, Jie Wang, and Shuiwang Ji. Line graph neural networks for link prediction. IEEE TPAMI. 2021.
- [23] Zuoyu Yan, Tengfei Ma, Liangcai Gao, Zhi Tang, and Chao Chen. Link prediction with persistent homology: An interactive view. In ICML, 2021.
- [24] Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. In NeurIPS, 2021.
- [25] Yuzhou Chen, Yulia R Gel, and H Vincent Poor. Bscnets: Block simplicial complex neural networks. In AAAI, 2022.

Reference VI

- [26] Liming Pan, Cheng Shi, and Ivan Dokmanić. Neural link prediction with walk pooling. In ICLR, 2022.
- [27] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. Physica A: statistical mechanics and its applications, 390(6):1150–1170, 2011.
- [28] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In SIGKDD, 2016.
- [29] Ziwei Zhang, Peng Cui, Xiao Wang, Jian Pei, Xuanrong Yao, and Wenwu Zhu. Arbitrary-order proximity preserved network embedding. In SIGKDD, 2018.
- [30] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. NeurIPS, 2016.

Reference VII

- [31] Seongjun Yun, Seoyoon Kim, Junhyun Lee, Jaewoo Kang, and Hyunwoo J Kim. Neo-gnns: Neighborhood overlap-aware graph neural networks for link prediction. In NeurIPS, 2021.
- [32] Mark EJ Newman. Clustering and preferential attachment in growing networks. Physical review E, 64(2):025102, 2001.
- [33] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. Social networks, 25(3):211–230, 2003.
- [34] Tao Zhou, Linyuan Lü, and Yi-Cheng Zhang. Predicting missing links via local information. The European Physical Journal B, 71(4):623–630, 2009.
- [35] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. In SIGKDD, 2016.

Reference VIII

- [36] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. NeurIPS, 2017.
- [37] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. Harvard Journal of Law & Technology, 31(2), 2018.
- [38] Berk Ustun, Alexander Spangher, and Yang Liu. Actionable recourse in linear classification. In FAT. 2019
- [39] Kaivalya Rawal and Himabindu Lakkaraju. Beyond individualized recourse: Interpretable and interactive summaries of actionable recourses. NeurIPS, 2020.
- [40] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. NeurIPS, 2019.

Reference IX

- [41] Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. NeurIPS, 2020.
- [42] Mohit Bajaj, Lingyang Chu, Zi Yu Xue, Jian Pei, Lanjun Wang, Peter Cho-Ho Lam, and Yong Zhang. Robust counterfactual explanations on graph neural networks. NeurIPS, 2021.
- [43] Ana Lucic, Maartje A Ter Hoeve, Gabriele Tolomei, Maarten De Rijke, and Fabrizio Silvestri. Cf-gnnexplainer: Counterfactual explanations for graph neural networks. In AISTATS, 2022.
- [44] Robin Pemantle. Vertex-reinforced random walk. Probability Theory and Related Fields, 92(1):117–136, 1992.
- [45] Qiaozhu Mei, Jian Guo, and Dragomir Radev. Divrank: the interplay of prestige and diversity in information networks. In SIGKDD, 2010.

Reference X

- [46] Dheepikaa Natarajan and Sayan Ranu. A scalable and generic framework to mine top-k representative subgraph patterns. In ICDM, 2016.
- [47] Jeroen Kazius, Ross McGuire, and Roberta Bursi. Derivation and validation of toxicophores for mutagenicity prediction. Journal of medicinal chemistry, 48(1):312–320, 2005.
- [48] Sudhanshu Chanpuriya and Cameron Musco. Infinitewalk: Deep network embeddings as laplacian embeddings with a nonlinearity. In SIGKDD, 2020.
- [49] Jing Zhu, Xingyu Lu, Mark Heimann, and Danai Koutra. Node proximity is all you need: Unified structural and positional node and graph embedding. In SDM, 2021.
- [50] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. arXiv preprint arXiv:1709.05584, 2017.

Reference XI

- [51] Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. Machine learning on graphs: A model and comprehensive taxonomy. arXiv preprint arXiv:2005.03675, 2020.
- [52] Daixin Wang, Peng Cui, and Wenwu Zhu. Structural deep network embedding. In SIGKDD, 2016.
- [53] Giang Hoang Nguyen, John Boaz Lee, Ryan A Rossi, Nesreen K Ahmed, Eunyee Koh, and Sungchul Kim. Continuous-time dynamic network embeddings. In Companion Proceedings of the The Web Conference 2018, pages 969–976, 2018.
- [54] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In NeurIPS, pages 2177–2185, 2014.
- [55] Suhang Wang, Jiliang Tang, Charu Aggarwal, Yi Chang, and Huan Liu. Signed network embedding in social media. In SDM, 2017.

Reference XII

- [56] Mohammad Raihanul Islam, B Aditya Prakash, and Naren Ramakrishnan. Signet: Scalable embeddings for signed networks. In PAKDD, 2018.
- [57] Pinghua Xu, Wenbin Hu, Jia Wu, and Bo Du. Link prediction with signed latent factors in signed social networks. In SIGKDD, 2019.

Most Popular

Downloaded Cited

August 2016

XGBoost: A Scalable Tree Boosting System

KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on

Tiangi Chen,

Carlos Guestrin

Cited 4,504 times

August 2016 node2vec: Scalable Feature Learning for Networks

> KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on

Aditva Grover.

Jure Leskovec Cited 2,155 times

August 2003

Maximizing the spread of influence through a social network

KDD '03: Proceedings of the ninth ACM SIGKDD international conference on

David Kempe, 🚳 Jon Kleinberg,

🏙 Éva Tardos

Cited 3,740 times

July 2002 Optimizing search engines using clickthrough data

KDD '02: Proceedings of the eighth ACM SIGKDD international conference on

Thorsten Joachims

Cited 2,021 times

August 2004

Mining and summarizing customer reviews

KDD '04: Proceedings of the tenth ACM SIGKDD international conference on

Minqing Hu, Bing Liu

Cited 2,859 times

August 2008 Factorization meets the

> collaborative filtering model KDD '08: Proceedings of the 14th ACM

neighborhood: a multifaceted

SIGKDD international conference on...

Yehuda Koren

Cited 1.80% times

August 2014

DeepWalk: online learning of social representations

KDD '14: Proceedings of the 20th ACM SIGKDD international conference on

Bryan Perozzi, Rami Al-Rfou,

Steven Skiena

Cited 2,244 times

August 2016

"Why Should I Trust You?": Explaining the Predictions of Any Classifier

KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on...

Marco Tulio Ribeiro,

Sameer Singh, + 1

Cited 1,698 times

Research papers

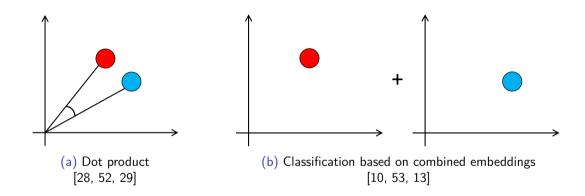
- ▶ [9] DeepWalk: online learning of social representations
- ▶ [10] node2vec: scalable feature learning for networks
- ... (with many more items omitted)
- ▶ [48] InfiniteWalk: deep network embeddings as ...
- ▶ [49] Node proximity is all you need: ...

Surveys

- ▶ [50]Representation learning on graphs: methods and applications
- **.**..
- ▶ [51]Machine learning on graphs: a model and comprehensive taxonomy

Difficult to compare existing methods and to design novel ones

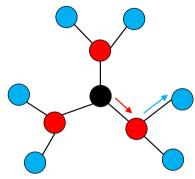
Question: link prediction



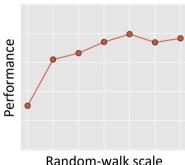
How should embeddings be used for link prediction?

Zexi Huang

Question: multiscale



(a) Random-walks capture multiple structural scales [1]



Random-wark scale

(b) Embedding performance [10] insensitive to random-walk scales

How do embeddings capture different structural scales?

Random-walk process

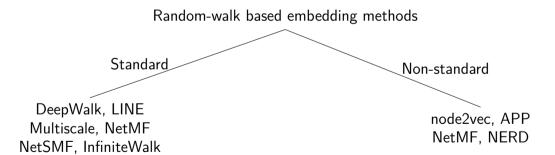
M: transition matrix

 π,Π : stationary distribution

Random-walk process

M: transition matrix

 π,Π : stationary distribution



PMI¹:
$$R = \log(\Pi M^{\tau}) - \log(\pi \pi^{T})$$

¹[54] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.

$$\mathsf{PMI}^1:\ R = \log(\Pi M^\tau) - \log(\pi \pi^T) \quad \text{ Autocovariance}^2:\ R = \Pi M^\tau - \pi \pi^T$$

¹[54] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.

²[1] Delvenne, Yalikari, and Barahona. Stability of graph communities across time scales. PNAS'10.

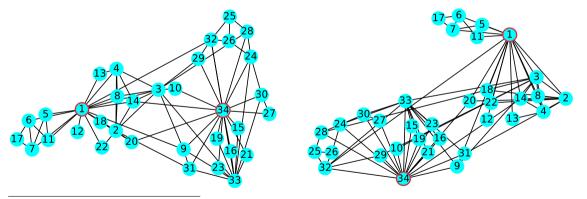
$$\mathsf{PMI}^1 \colon \, R = \log(\Pi M^\tau) - \log(\pi \pi^T) \quad \, \mathsf{Autocovariance}^2 \colon \, R = \Pi M^\tau - \pi \pi^T$$

Zexi Huang

 $^{^1}$ [54] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.

²[1] Delvenne, Yalikari, and Barahona. Stability of graph communities across time scales. PNAS'10.

$$\mathsf{PMI}^1 \colon \, R = \log(\Pi M^\tau) - \log(\pi \pi^T) \quad \text{ Autocovariance}^2 \colon \, R = \Pi M^\tau - \pi \pi^T$$



¹[54] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.

²[1] Delvenne, Yalikari, and Barahona. Stability of graph communities across time scales. PNAS'10.

$$\mathsf{PMI}^1 \colon R = \log(\Pi M^\tau) - \log(\pi \pi^T) \quad \mathsf{Autocovariance}^2 \colon R = \Pi M^\tau - \pi \pi^T$$
 Random-walk based embedding methods
$$\mathsf{Standard} \quad \mathsf{Non\text{-}standard}$$

$$\mathsf{Non\text{-}standard}$$

$$\mathsf{Non\text{-}standard}$$

$$\mathsf{PMI} \quad \mathsf{Autocovariance}$$

$$\mathsf{DeepWalk, LINE} \quad \mathsf{NetMF, NetSMF} \quad \mathsf{Multiscale} \quad \mathsf{node2vec, APP} \quad \mathsf{NetMF, NERD}$$

$$\mathsf{InfiniteWalk}$$

 2 [1] Delvenne et al. Stability of graph communities across time scales. PNAS'10.

Zexi Huang

 $^{^1}$ [54] Levy and Goldberg. Neural word embedding as implicit matrix factorization. NeurIPS'14.

Embedding algorithm

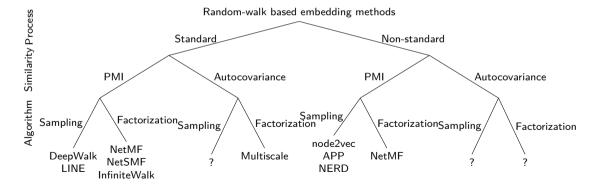
Explicit: Factorization (SVD)
$$\min \|UU^T - R\|_F^2$$

Implicit: Sampling (SGD)
$$\max \sum_{u,v} \log \Pr((u,v) \in \mathcal{D} | \mathbf{u}_u, \mathbf{v}_v)$$

Embedding algorithm

Explicit: Factorization (SVD)
$$\min \|UU^T - R\|_F^2$$

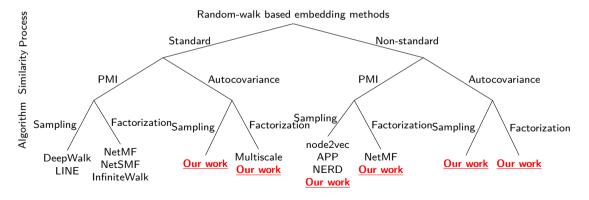
Implicit: Sampling (SGD)
$$\max \sum_{u,v} \log \Pr((u,v) \in \mathcal{D} | \mathbf{u}_u, \mathbf{v}_v)$$



Embedding algorithm

Explicit: Factorization (SVD)
$$\min \|UU^T - R\|_F^2$$

Implicit: Sampling (SGD)
$$\max \sum_{u,v} \log \Pr((u,v) \in \mathcal{D} | \mathbf{u}_u, \mathbf{v}_v)$$



Datasets

	$ \mathcal{V} $	$ \mathcal{E} $	labels
BLOGCATALOG	10,312	333,983	interests
Airport	3,158	18,606	countries/continents
Wiki-words	4,777	92,157	tags
PoliticalBlogs	1,222	16,717	ideologies

Downstream tasks

- ► Node classification (Micro/Macro-F1)
- ► Link prediction (precision@k)
- Community detection (NMI)

Embedding dimension = 128

Comparing similarity metrics: node classification

PMI:
$$R = \log(\Pi M^{\tau}) - \log(\pi \pi^T)$$
 Autocovariance: $R = \Pi M^{\tau} - \pi \pi^T$

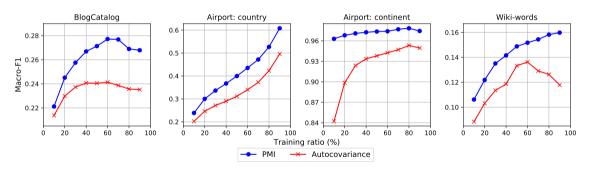


Figure: PMI consistently outperforms autocovariance in node classification.

Zexi Huang

Comparing similarity metrics: link prediction

PMI:
$$R = \log(\Pi M^{\tau}) - \log(\pi \pi^{T})$$

Autocovariance: $R = \Pi M^{\tau} - \pi \pi^{T}$

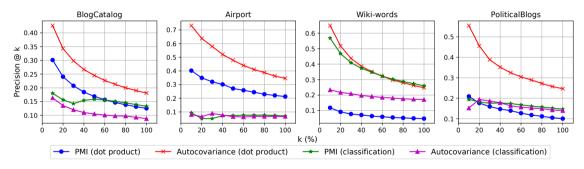


Figure: Autocovariance with dot product ranking consistently outperforms PMI (with either ranking scheme) in link prediction.

Zexi Huang

predicted degree \propto embedding norm $\|\mathbf{u}\|$

 $\text{predicted degree} \propto \text{embedding norm } \|\mathbf{u}\| \propto \begin{cases} \text{actual degree} & \text{for autocov.} \\ \text{constant} & \text{for PMI} \end{cases}$

 $\text{predicted degree} \propto \text{embedding norm } \|\mathbf{u}\| \propto \begin{cases} \text{actual degree} & \text{for autocov.} \\ \text{constant} & \text{for PMI} \end{cases}$

Autocovariance captures heterogeneous degree distribution in graphs!

$$\text{predicted degree} \propto \text{embedding norm } \|\mathbf{u}\| \propto \begin{cases} \text{actual degree} & \text{for autocov.} \\ \text{constant} & \text{for PMI} \end{cases}$$

Autocovariance captures heterogeneous degree distribution in graphs!

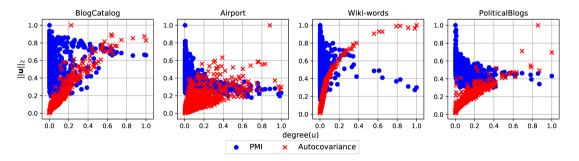


Figure: Autocovariance embedding norms correlate with actual degrees, but not PMI.

Zexi Huang Graph Representation Learning June 2, 2022 65/53

 $\text{predicted degree} \propto \text{embedding norm } \|\mathbf{u}\| \propto \begin{cases} \text{actual degree} & \text{for autocov.} \\ \text{constant} & \text{for PMI} \end{cases}$











(b) Autocovariance

Figure: Autocovariance predicts more edges connecting to the hubs than PMI.

Multiscale

PMI:
$$R = \log(\Pi M^{\tau}) - \log(\pi \pi^{T})$$
 Autocovariance: $R = \Pi M^{\tau} - \pi \pi^{T}$

Multiscale

PMI:
$$\widetilde{R} = \log(\Pi_{\tau}^{\frac{1}{\tau}} \sum_{t=1}^{\tau} M^t) - \log(\pi \pi^T)$$

Autocovariance:
$$R = \Pi M^{\tau} - \pi \pi^{T}$$

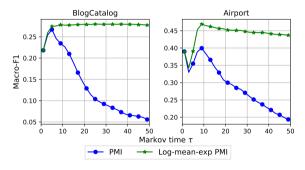
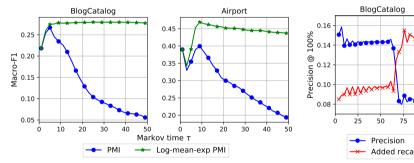


Figure: Node classification performance for PMI can be improved by smooth-averaging across multiple Markov times.

Multiscale

PMI:
$$\widetilde{R} = \log(\Pi_{\overline{\tau}}^{\frac{1}{\tau}} \sum_{t=1}^{\tau} M^t) - \log(\pi \pi^T)$$

Autocovariance:
$$R = \Pi M^{\tau} - \pi \pi^{T}$$



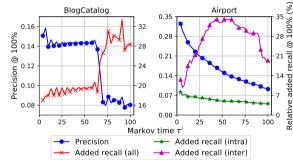


Figure: Node classification performance for PMI can be improved by smooth-averaging across multiple Markov times.

Figure: Prediction of edges of specific structural scales can be improved with different Markov times for autocovariance

Problems:

- 1. How can one compare existing methods and to advance the SOTA?
- 2. How should embeddings be used for link prediction?
- 3. How do embeddings capture different structural scales?

Contributions:

- 1. A unified view of different processes, similarities, and algorithms.
- 2. Autocovariance embedding is significantly better for link prediction.
- 3. Ways to exploit multiscale similarity for optimized performance.

Social polarization





TODAY WE MAKE AMERICA GREAT AGAIN!





The Fake News is working overtime. Just reported that, despite the tremendous success we are having with the economy & all things else, 91% of the Network News about me is negative (Fake). Why do we work so hard in working with the media when it is corrupt? Take away credentials?



These are the things and events that happen when a sacred landslide election victory is so unceremoniously & viciously stripped away from great patriots who have been badly & unfairly treated for so long. Go home with love & in peace. Remember this day forever!

This claim of election fraud is disputed, and this Tweet can't be replied to. Retweeted or liked due to a risk of violence



Replying to @realDonaldTrump

These damned fake news outlets need to be taken down! If they had done this to Ohama, it would've been the end of the world!

We are with you. Mr President!



Replying to @realDonaldTrump

The IDI party will cease to exist once it's all exposed. FAKE NEWS can no longer control [dampen] public awareness of the TRUTH. DARK TO LIGHT.



I love seeing Trump supporters CRY, it's my daily medicine, my weekly energy, my monthly inspiration and my yearly motivation. Their loss is the only reason i'm still alive, i was born to love and enjoy the failure that they have achieved.



Signed random walk

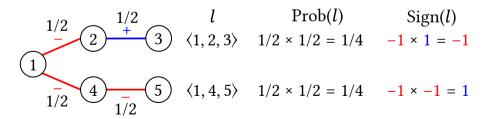
- \blacktriangleright Unsigned RW: $|M|_{uv}(t) = \sum_{\text{all length-}t \text{ paths }l \text{ between }u \text{ and }v} \mathsf{Prob}(l)$
 - ightharpoonup Prob(l) captures the unsigned similarity.

Signed random walk

- ▶ Unsigned RW: $|M|_{uv}(t) = \sum_{\text{all length-}t \text{ paths } l \text{ between } u \text{ and } v} \mathsf{Prob}(l)$
 - ightharpoonup Prob(l) captures the unsigned similarity.
- **Signed RW**: $M_{uv}(t) = \sum_{\mathsf{all\ length}-t\ \mathsf{paths}\ l\ \mathsf{between}\ u\ \mathsf{and}\ v} \mathsf{Prob}(l) \mathsf{Sign}(l)$
 - ightharpoonup Sign(l) based on the social balance theory captures the signed similarity.

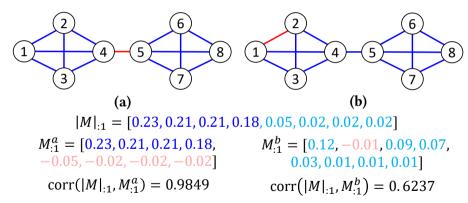
Signed random walk

- ▶ Unsigned RW: $|M|_{uv}(t) = \sum_{\text{all length-}t \text{ paths } l \text{ between } u \text{ and } v} \mathsf{Prob}(l)$
 - ightharpoonup Prob(l) captures the unsigned similarity.
- lacksquare Signed RW: $M_{uv}(t) = \sum_{\mathsf{all\ length}-t\ \mathsf{paths}\ l\ \mathsf{between}\ u\ \mathsf{and}\ v} \mathsf{Prob}(l) \mathsf{Sign}(l)$
 - ightharpoonup Sign(l) based on the social balance theory captures the signed similarity.



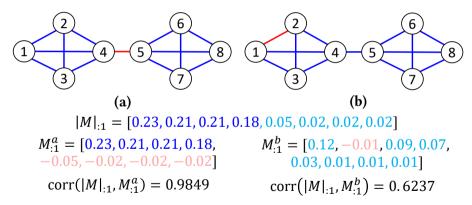
Measuring polarization

 $ightharpoonup |M|_{:u}(t)$ and $M_{:u}(t)$ are highly correlated if u is polarized.



Measuring polarization

 $ightharpoonup |M|_{:u}(t)$ and $M_{:u}(t)$ are highly correlated if u is polarized.



 $ightharpoonup \operatorname{Pol}(u;t) = \operatorname{corr}(|M|_{\cdot u}(t), M_{\cdot u}(t)), \operatorname{Pol}(G;t) = \operatorname{mean}_{u \in G}(\operatorname{Pol}(u;t)).$

Zexi Huang

71 / 53

Datasets:

	$ \mathcal{V} $	$ \mathcal{E} $	$ \mathcal{E}_{-} / \mathcal{E} $
Congress WoW-EP8 BITCOIN-ALPHA BITCOIN-OTC REFERENDUM	219 789 3,772 5,872 10,864	523 116,009 14,077 21,431 251,396	20.46% 18.63% 9.31% 14.71% 5.09%
Wiki-RfA	11,275	169,925	22.04%

Baselines: SiNE [55], SIGNet [56], SIDE [7], BESIDE [8], SLF [57], ROSE [13] **Signed link prediction setting:**

- POLE: compute dot product similarity
- Baselines: train two classifiers (positive/negative vs non-links)
- ► Evaluation metric: positive/negative precision@k

Signed link prediction without link existence information

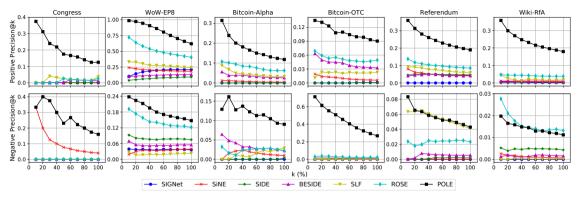


Figure: Signed link prediction performance comparison between POLE and baselines without link existence information. POLE outperforms all baselines in almost all datasets, especially for the polarized ones.

Interaction between unsigned and signed similarity

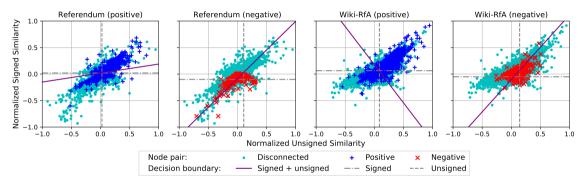


Figure: Scatter plot of the reconstructed signed and unsigned similarity for different node pairs in signed link prediction. Combining signed and unsigned similarity improves prediction for negative links but has a negligible effect on predicting positive links.

Measuring polarization

Table: Ten least polarized congresspeople by our RW-based polarization measure.

Congressperson	State	Party	Score
Henry Cuellar*	Texas	D	-0.6542
Jane Harman [†]	California	D	-0.5376
Curt Weldon	Pennsylvania	R	-0.4381
Dutch Ruppersberger	Maryland	D	-0.4318
Jim Moran	Virginia	D	-0.3832
Dave Obey	Wisconsin	D	-0.3588
Wayne Gilchrest	Maryland	R	-0.3503
Duke Cunningham	California	R	-0.3248
Al Edwards	Texas	D	-0.3063
Lincoln Davis	Tennessee	D	-0.2901

^{* &}quot;Voted with President Trump 75% of time" — 538

^{† &}quot;Best Republican in the Democratic Party" — LA Times

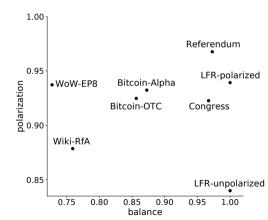
Measuring polarization

Table: Ten least polarized congresspeople by our RW-based polarization measure.

Congressperson	State	Party	Score
Henry Cuellar*	Texas	D	-0.6542
Jane Harman [†]	California	D	-0.5376
Curt Weldon	Pennsylvania	R	-0.4381
Dutch Ruppersberger	Maryland	D	-0.4318
Jim Moran	Virginia	D	-0.3832
Dave Obey	Wisconsin	D	-0.3588
Wayne Gilchrest	Maryland	R	-0.3503
Duke Cunningham	California	R	-0.3248
AI Edwards	Texas	D	-0.3063
Lincoln Davis	Tennessee	D	-0.2901

^{* &}quot;Voted with President Trump 75% of time" — 538

Figure: Polarization and social balance of real-world graphs and LFR benchmarks.



^{† &}quot;Best Republican in the Democratic Party" — LA Times

Simple Graph Convolution¹:

$$H^{(k)} = \sigma(\widetilde{M}^k X \Theta)$$

 $H^{(k)}$: kth order embedding, X: node attributes, Θ : trainable weights \widetilde{M} : random-walk transition with added self-loops

Simple Graph Convolution¹:

$$H^{(k)} = \sigma(\widetilde{M}^k X \Theta)$$

 $H^{(k)}$: kth order embedding, X: node attributes, Θ : trainable weights \widetilde{M} : random-walk transition with added self-loops

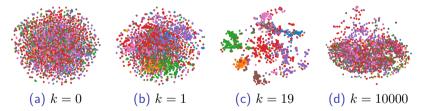


Figure: Visualization of embedding without training for Cora. Colors denote actual classes. A good choice of neighborhood helps reveal the cluster structure.

¹Wu et al. Simplifying graph convolutional networks. ICML'19.

Node classification results

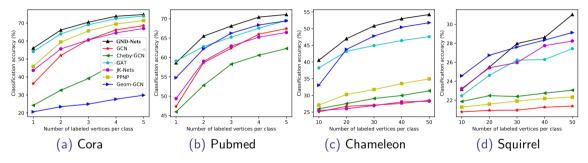


Figure: Semi-supervised node classification performance. Our method (GND-Nets) outperforms baselines for various number of labeled nodes across the datasets.

Learned diffusion weights

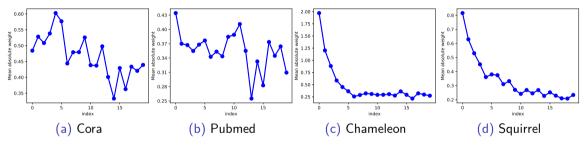


Figure: Mean diffusion weights (over 30 splits) across different datasets. Note that Cora and Pubmed have larger weights for high-order neighborhoods compared to Chameleon and Squirrel.