RiWalk: Fast Structural Node Embedding via Role Identification

Xuewei Ma, Geng Qin, Zhiyang Qiu, Mingxin Zheng, Zhe Wang

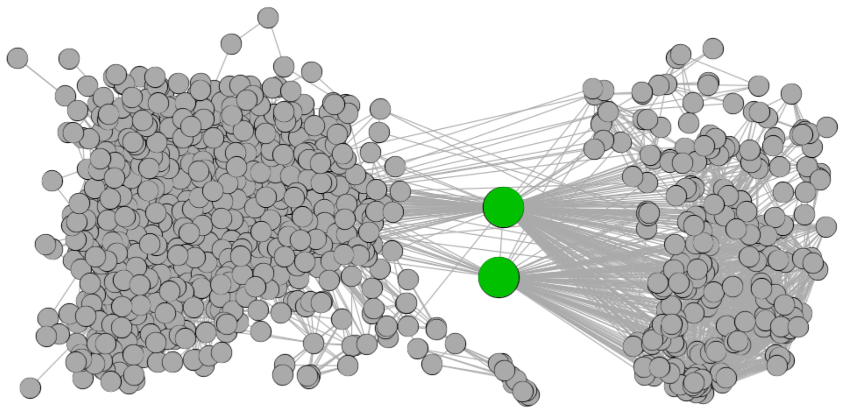
Jilin University

Roles of Nodes

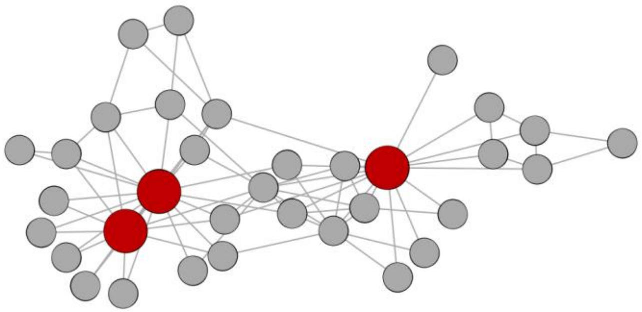
- Behavior/function ⟷ role ⟷ structure/topology
- Nodes in the same network may have similar roles
- Nodes in different networks may have similar roles

hubs/leaders

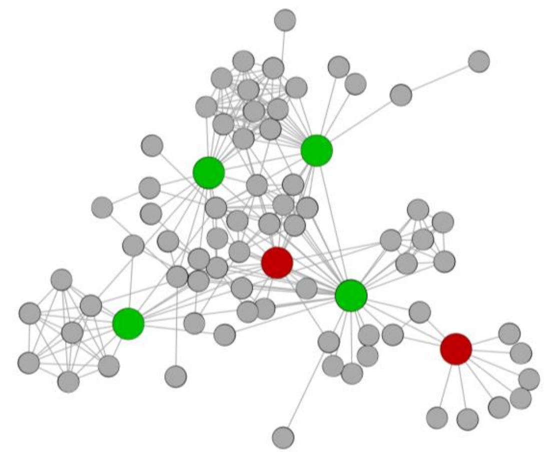
structural hole spanners



American air-traffic network



Zachary's karate network



Les Misérables coappearance network

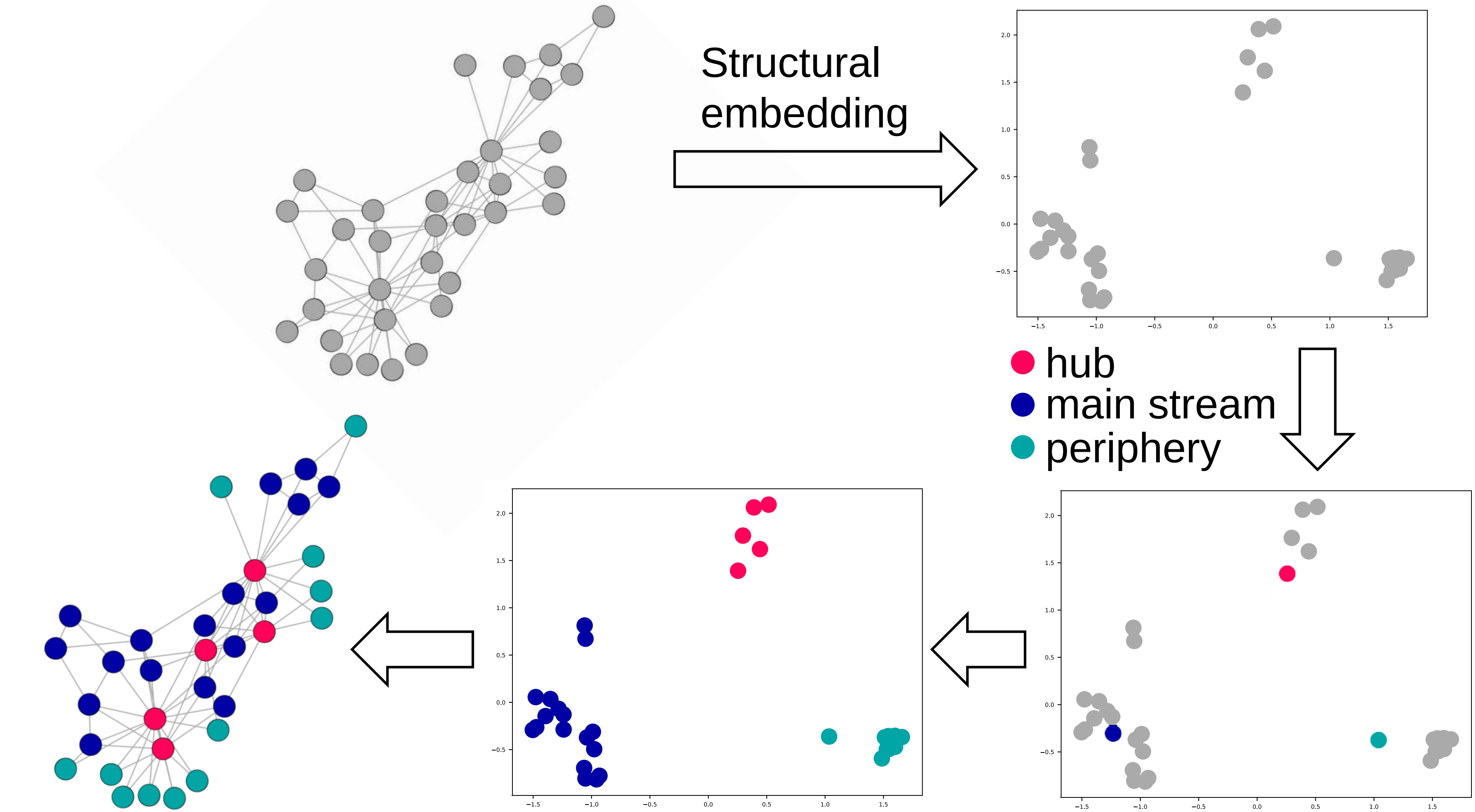
- Learning representations about roles helps to
  - predict behaviors/functions of nodes
  - understand networks
  - transfer knowledge across networks

Problem: Structural Embedding

- Map nodes to low-dimensional vectors
  (usually Euclidean Space)
- Preserve structural similarities
  - Nodes with similar roles should be embedded closely

Problem: Structural Embedding

- Map nodes to low-dimensional vectors
- Preserve structural similarities

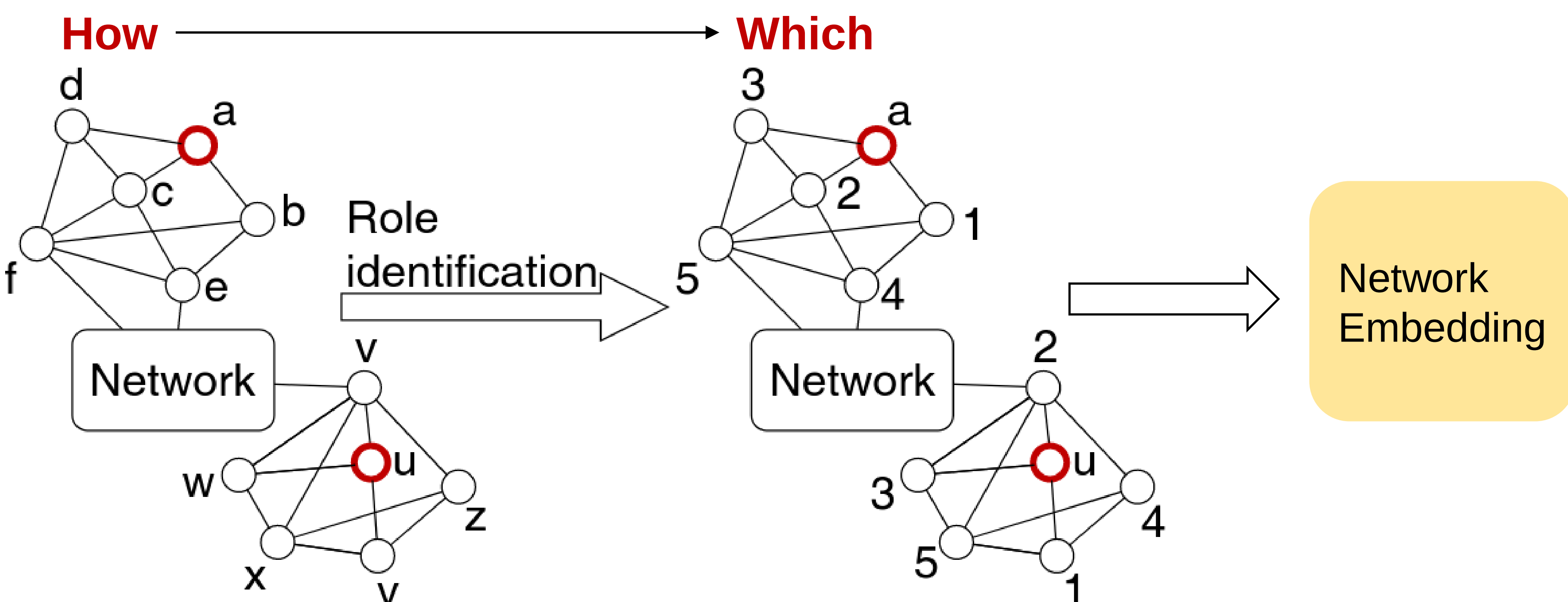


Related Works

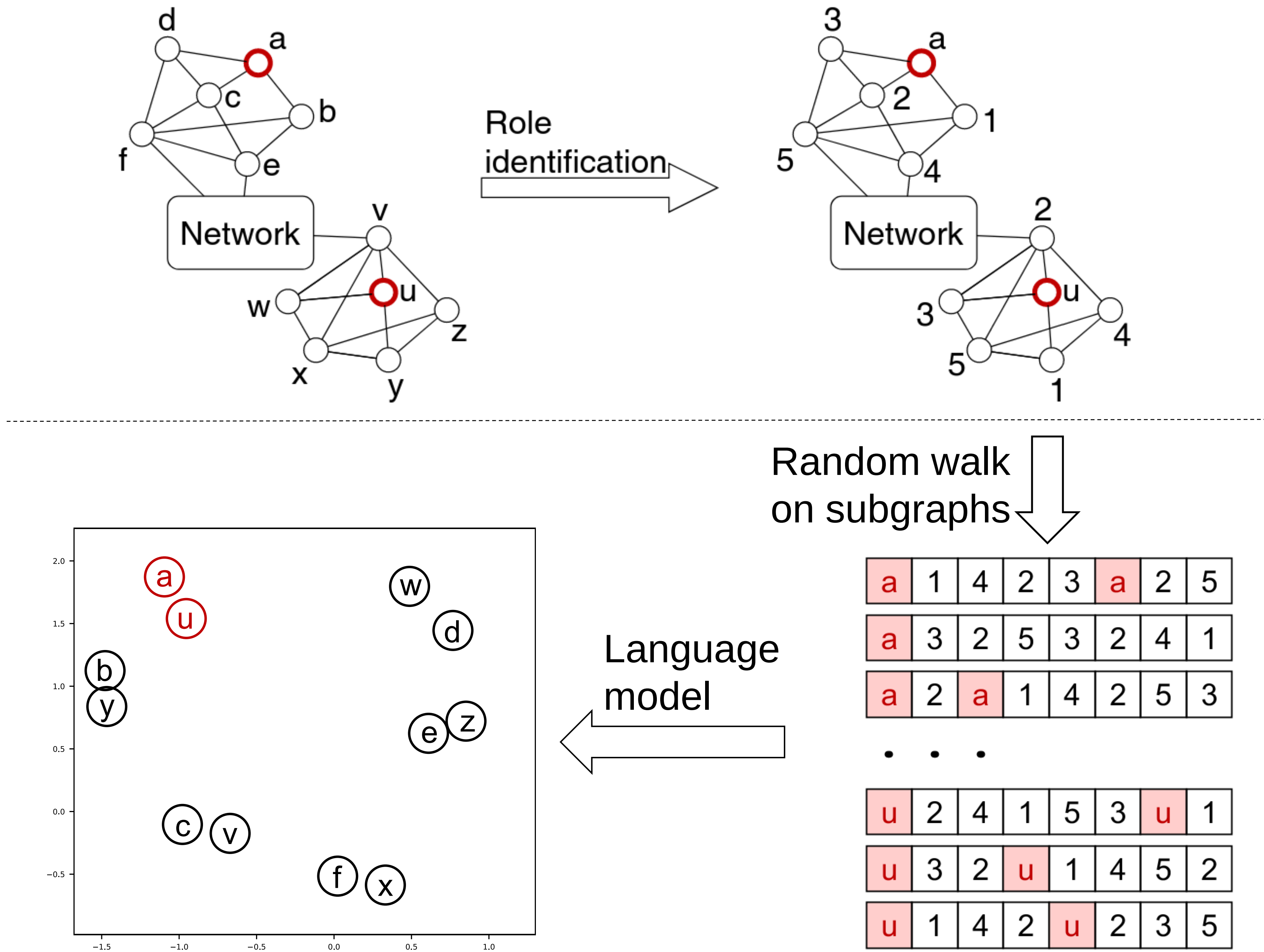
- Language Model
  - word2vec [arXiv'13]: maps words to Euclidean space by preserving linguistic contexts of words
- Network Embedding
  - DeepWalk [KDD'14]: treats random walks as sentences
  - node2vec [KDD'16]: uses biased random walk to add flexibility in neighborhood exploring
- Structural Embedding
  - RolX [KDD'12]: factorizes node feature matrix to get node representaions
  - struc2vec [KDD'17]: builds a hierarchy to measure structural similarity
  - GraphWave [KDD'18]: uses empirical characteristic function to embed wavelet distributions

Key idea

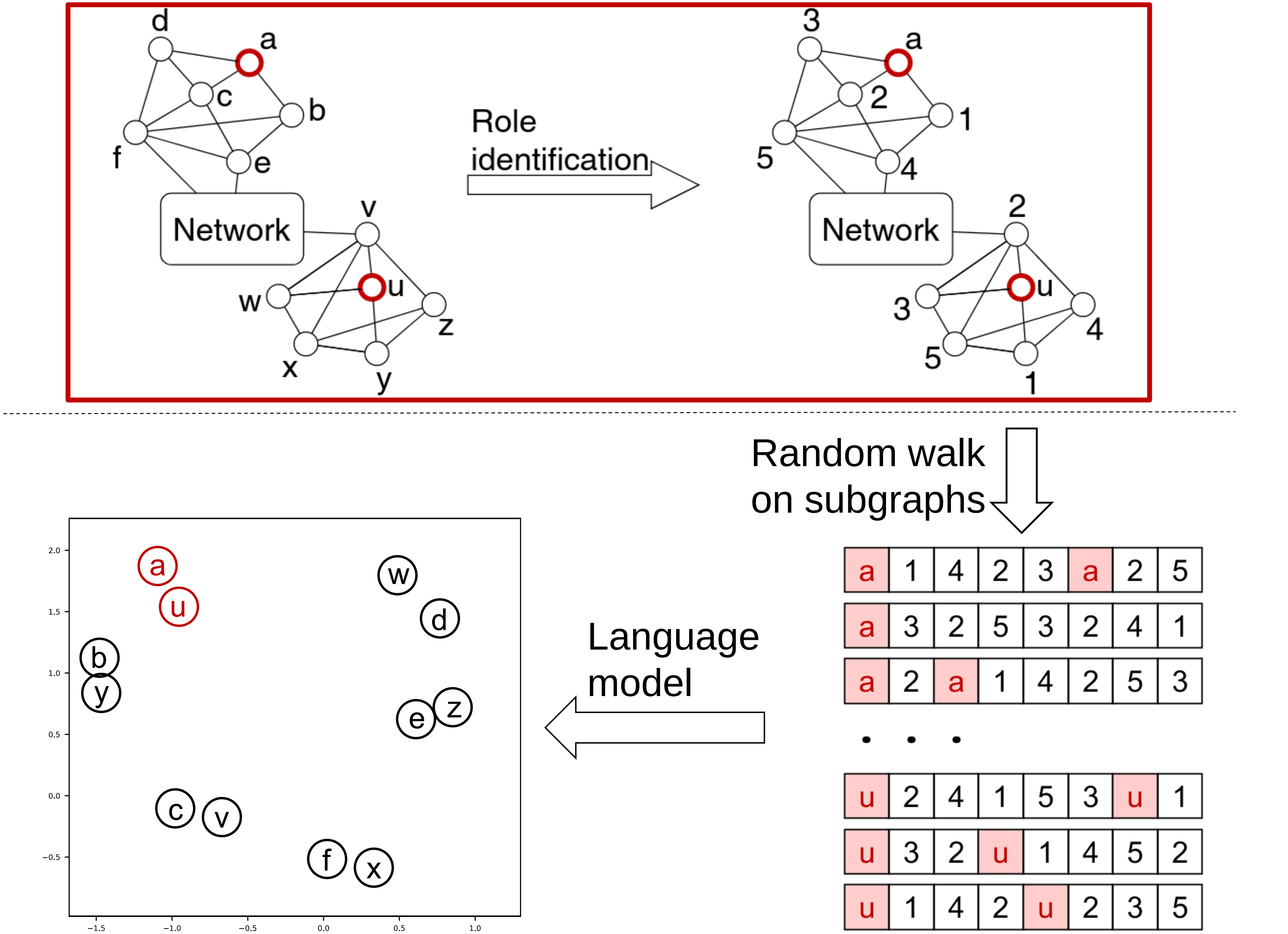
- Typical network embedding
  - Nodes sharing many context nodes are embedded closely
  - Focuses on which node one node connects to
- Structural embedding
  - Nodes having similar local structures are embedded closely
  - Focuses on how one node connects to its context nodes



The RiWalk Framework



The RiWalk Framework



Role Identification: RiWalk-SP

- $\psi_i(v_j) = h(\delta_i) \oplus h(\delta_j) \oplus s_{ij},$ for each $v_j \in \mathcal{N}_i^k \setminus \{v_i\}$

- $h(x) = \lfloor \log_2(x+1) \rfloor$

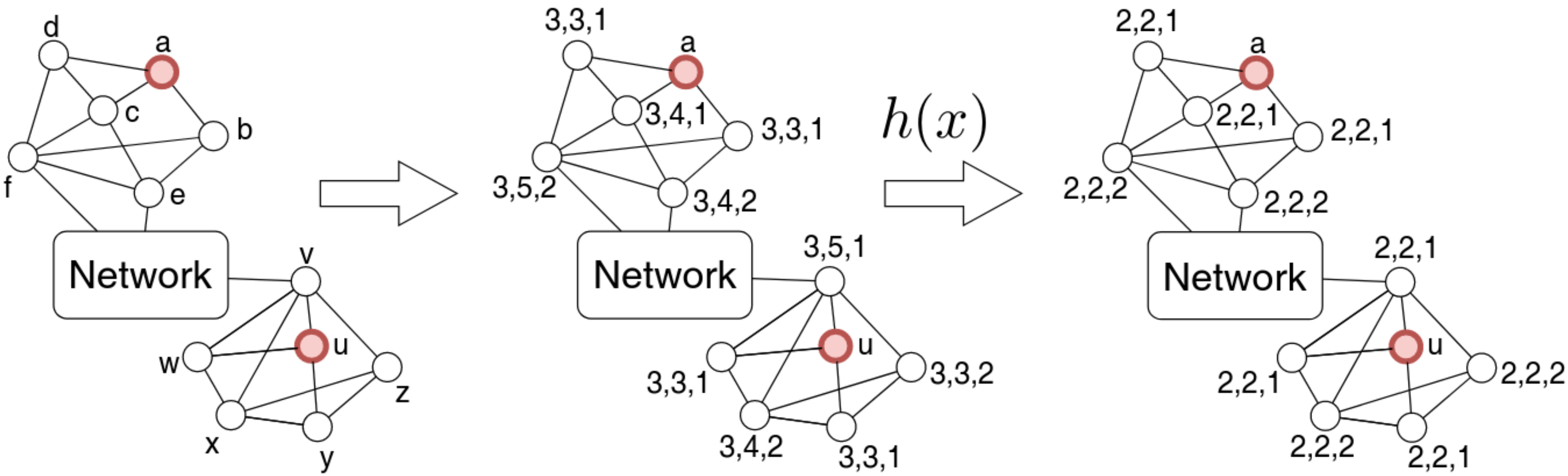
- v_i : Anchor node
- $\mathcal{N}_i^k$: Neighbors within k hops from v_i
- s_{ij} : Shortest path length between v_i and v_j
- δ_i : Degree of v_i

K. M. Borgwardt and H.-P. Kriegel, “Shortest-path kernels on graphs” [ICDM’05]

Role Identification: RiWalk-SP

- $\psi_i(v_j) = h(\delta_i) \oplus h(\delta_j) \oplus s_{ij},$ $\quad$ for each $v_j \in \mathcal{N}_i^k \setminus \{v_i\}$
- $h(x) = \lfloor \log_2(x+1) \rfloor$

- v_i : Anchor node
- $\mathcal{N}_i^k$: Neighbors within k hops from v_i
- s_{ij} : Shortest path length between v_i and v_j
- δ_i : Degree of v_i



K. M. Borgwardt and H.-P. Kriegel, "Shortest-path kernels on graphs" [ICDM'05]

Role Identification: RiWalk-WL

- $\mathbf{x}_{ij}^{(n)} = \left| \{ v_l \in \mathcal{N}_j \mid s_{il} = n \} \right|$

  $\text{for } n \in \{0, 1, \ldots, k\}$
- $\psi_i(v_j) = h(\mathbf{x}_{ii}) \oplus h(\mathbf{x}_{ij}) \oplus s_{ij},$

  $\text{for each } v_j \in \mathcal{N}_i^k \setminus \{v_i\}$
- $h(x) = \lfloor \log_2(x+1) \rfloor$

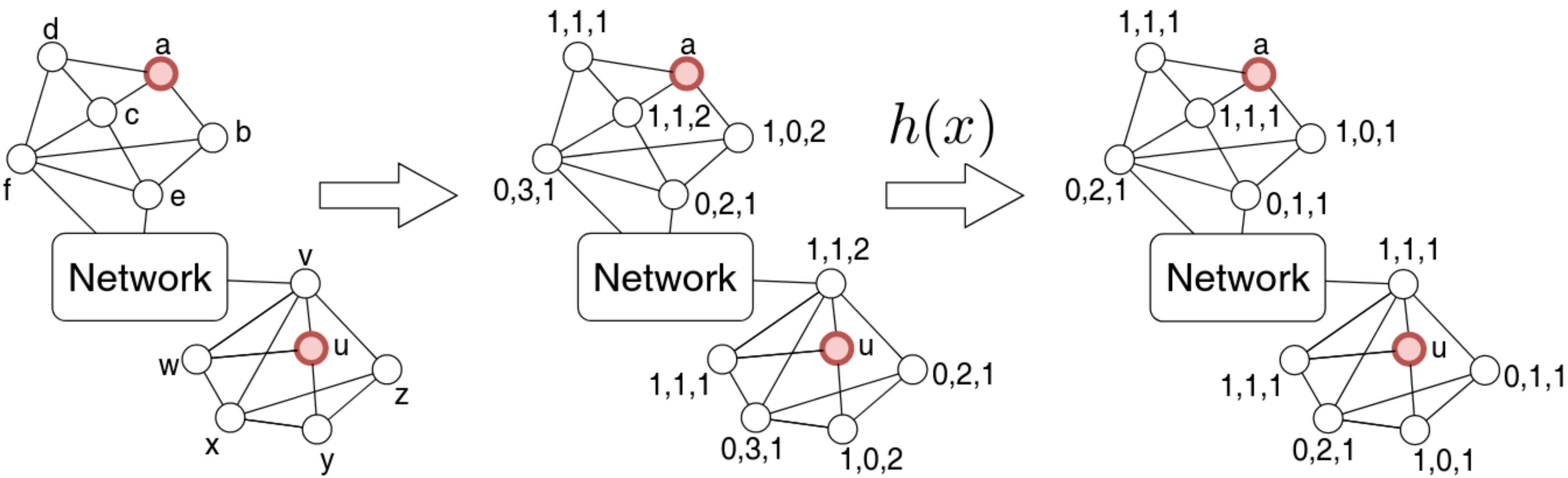
- v_i : Anchor node
- $\mathcal{N}_i^k$: Neighbors within k hops from v_i
- s_{ij} : Shortest path length between v_i and v_j
- δ_i : Degree of v_i

N. Shervashidze and K. Borgwardt, "Fast subtree kernels on graphs" [NeurIPS'09]

Role Identification: RiWalk-WL

- $\mathbf{x}_{ij}^{(n)} = \left| \{ v_l \in \mathcal{N}_j \mid s_{il} = n \} \right|$
  for $n \in \{0, 1, \ldots, k\}$
- $\psi_i(v_j) = h(\mathbf{x}_{ii}) \oplus h(\mathbf{x}_{ij}) \oplus s_{ij},$
  for each $v_j \in \mathcal{N}_i^k \setminus \{v_i\}$
- $h(x) = \lfloor \log_2(x+1) \rfloor$

- v_i : Anchor node
- $\mathcal{N}_i^k$: Neighbors within k hops from v_i
- s_{ij} : Shortest path length between v_i and v_j
- δ_i : Degree of v_i

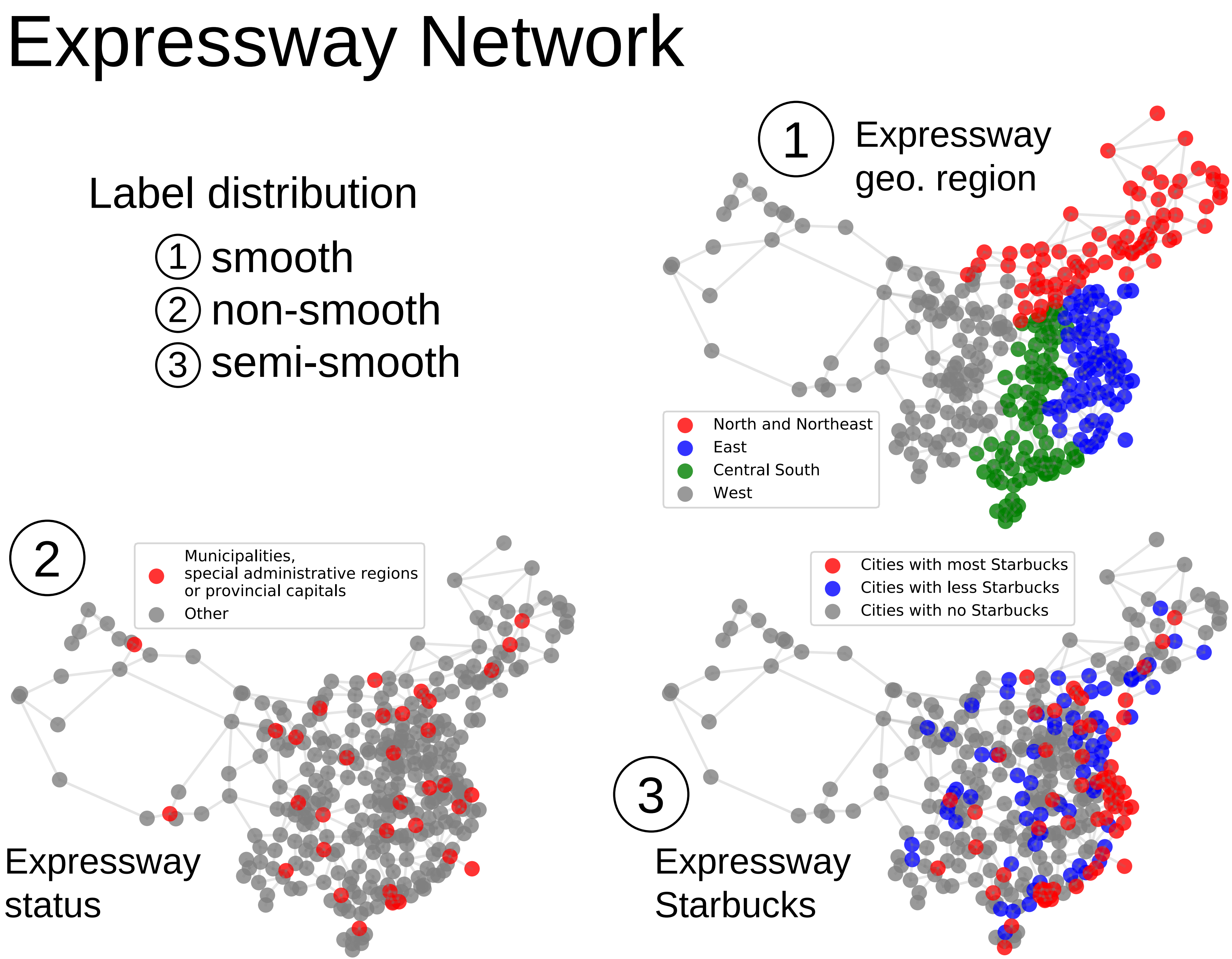


N. Shervashidze and K. Borgwardt, “Fast subtree kernels on graphs” [NeurIPS’09]

Expressway Network

- Nodes: cities
- Edges: expressways

Collected from “https://wikipedia.org/wiki/Category:Chinese_national-level expressways”



Expressway Network
Label distribution
1 smooth
2 non-smooth
3 semi-smooth
1 Expressway geo. region
North and Northeast
East
Central South
West
2
Municipalities, special administrative regions or provincial capitals
Other
Expressway status
3
Cities with most Starbucks
Cities with less Starbucks
Cities with no Starbucks
Expressway Starbucks

Expressway Network

RESULTS OF CLASSIFICATION TASKS ON EXPRESSWAY NETWORKS. (MACRO-F_1 (%))

Algorithm	Dataset smooth Expressway geo. region	 non-smooth Expressway status	 semi-smooth Expressway Starbucks
node2vec	**96.13**	50.09	**52.75**
struc2vec	39.78	**54.44**	38.60
GraphWave	**51.39**	51.28	42.88
RiWalk-SP	50.05	53.05	44.26
RiWalk-WL	**51.39**	**56.62**	**44.70**
Majority	11.41	47.37	25.37

- Smooth label distribution ⇨ network embedding
- Non-smooth & role related ⇨ structural emedding
- Semi-smooth ⇨ combining both?

Within-network Node Classification

	Europe	USA	Film	Actor
# Vertices	399	1190	27312	7779
# Edges	5995	13599	122514	26752
# Classes	4	4	4	4

MICRO-F_1(%) SCORES OF WITHIN-NETWORK ROLE CLASSIFICATION.

		Labeled Nodes (%)									Time and Memory Usage		
Dataset	Method	10	20	30	40	50	60	70	80	90	Mem (M)	Real (s)	User (s)
USA	*node2vec*	54.86	58.84	61.03	61.78	62.79	63.44	63.74	63.86	64.18			
	struc2vec	54.39	58.06	60.23	60.93	61.86	62.73	63.17	64.38	65.75	82	94	863
	GraphWave	**60.30**	**61.30**	**62.45**	62.90	62.38	62.98	62.36	63.25	64.67	127	6	74
	RiWalk-SP	58.62	60.35	61.21	63.03	63.69	63.58	64.47	65.83	64.60	13	4	19
	RiWalk-WL	58.25	60.82	62.39	**63.04**	**64.34**	**64.38**	**65.92**	**66.17**	**66.25**	42	17	146
Film	*node2vec*	44.04	45.36	45.91	46.10	46.36	46.33	46.46	46.75	46.68			
	struc2vec	54.14	55.59	56.10	56.24	56.37	56.54	56.46	56.70	56.43	1027	1972	18236
	GraphWave	—	—	—	—	—	—	—	—	—	—	—	—
	RiWalk-SP	**60.26**	**61.08**	**61.40**	**61.52**	**61.61**	**61.63**	**61.65**	**61.44**	**61.56**	111	179	1148
	RiWalk-WL	59.15	60.23	60.48	60.71	60.67	60.82	60.76	60.88	60.98	113	600	5404
Actor	*node2vec*	31.24	33.34	34.88	35.74	36.04	36.83	36.61	37.14	37.82			
	struc2vec	42.46	**44.72**	**45.43**	**45.99**	**46.51**	**46.56**	**47.05**	**47.48**	**47.56**	284	379	3459
	GraphWave	—	—	—	—	—	—	—	—	—	—	—	—
	RiWalk-SP	**43.27**	44.61	45.05	45.60	45.31	45.78	46.56	46.05	45.13	65	30	177
	RiWalk-WL	41.60	43.43	44.25	44.16	44.69	45.27	45.39	45.23	46.54	64	62	545

- RiWalk achieves comparable performance with other baselines while being an order of magnitude more efficient (time & space).

Within-network Node Classification

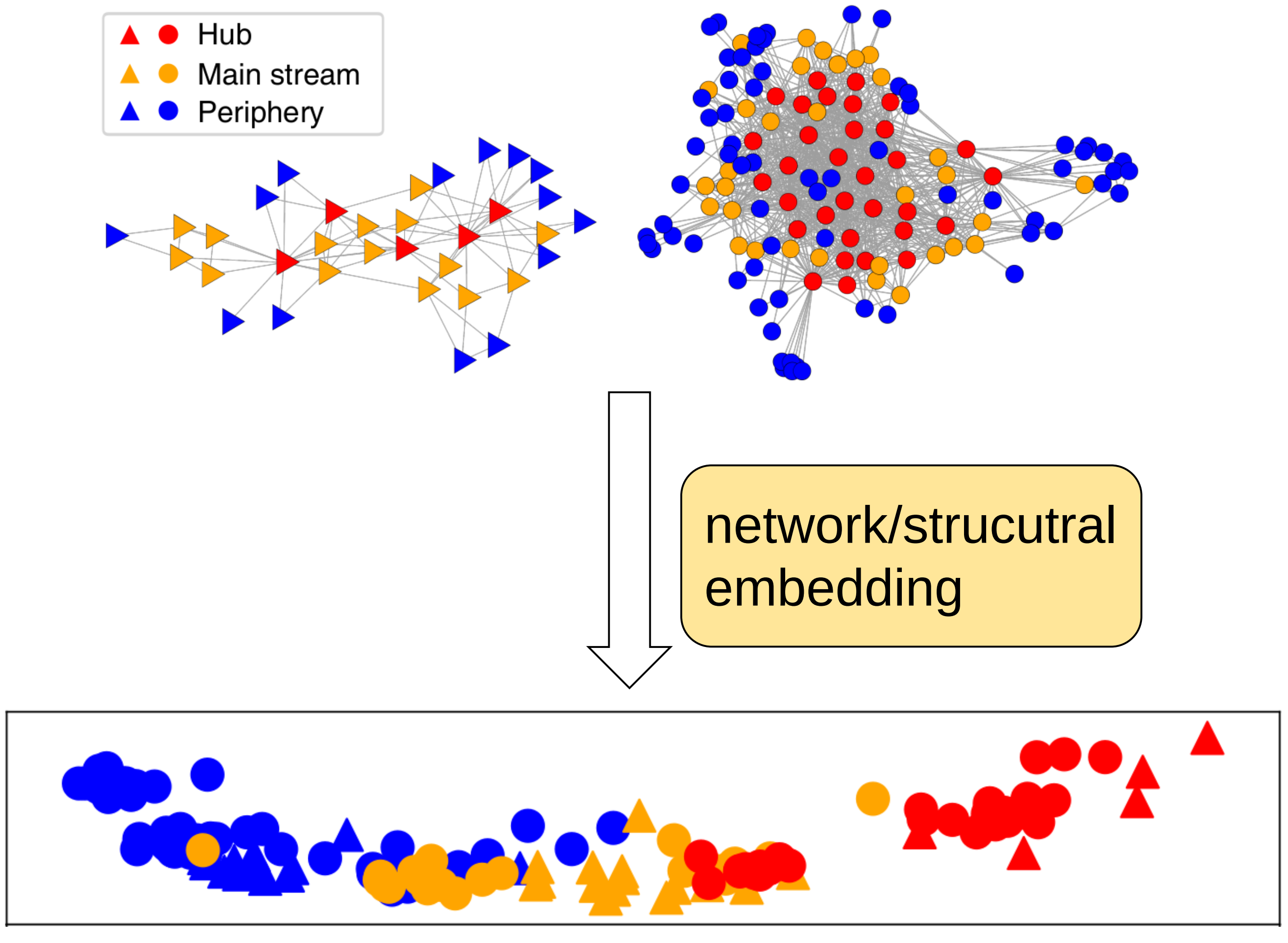
MICRO-F_1(%) SCORES OF WITHIN-NETWORK ROLE CLASSIFICATION.

		Labeled Nodes (%)									Time and Memory Usage		
Dataset	Method	10	20	30	40	50	60	70	80	90	Mem (M)	Real (s)	User (s)
USA	*node2vec*	54.86	58.84	61.03	61.78	62.79	63.44	63.74	63.86	64.18			
	struc2vec	54.39	58.06	60.23	60.93	61.86	62.73	63.17	64.38	65.75	82	94	863
	GraphWave	**60.30**	**61.30**	**62.45**	62.90	62.38	62.98	62.36	63.25	64.67	127	6	74
	RiWalk-SP	58.62	60.35	61.21	63.03	63.69	63.58	64.47	65.83	64.60	13	4	19
	RiWalk-WL	58.25	60.82	62.39	**63.04**	**64.34**	**64.38**	**65.92**	**66.17**	**66.25**	42	17	146
Film	*node2vec*	44.04	45.36	45.91	46.10	46.36	46.33	46.46	46.75	46.68			
	struc2vec	54.14	55.59	56.10	56.24	56.37	56.54	56.46	56.70	56.43	1027	1972	18236
	GraphWave	—	—	—	—	—	—	—	—	—	—	—	—
	RiWalk-SP	**60.26**	**61.08**	**61.40**	**61.52**	**61.61**	**61.63**	**61.65**	**61.44**	**61.56**	111	179	1148
	RiWalk-WL	59.15	60.23	60.48	60.71	60.67	60.82	60.76	60.88	60.98	113	600	5404
Actor	*node2vec*	31.24	33.34	34.88	35.74	36.04	36.83	36.61	37.14	37.82			
	struc2vec	42.46	**44.72**	**45.43**	**45.99**	**46.51**	**46.56**	**47.05**	**47.48**	**47.56**	284	379	3459
	GraphWave	—	—	—	—	—	—	—	—	—	—	—	—
	RiWalk-SP	**43.27**	44.61	45.05	45.60	45.31	45.78	46.56	46.05	45.13	65	30	177
	RiWalk-WL	41.60	43.43	44.25	44.16	44.69	45.27	45.39	45.23	46.54	64	62	545

- RiWalk achieves comparable performance with other baselines while being an order of magnitude more efficient (time & space).
- RiWalk performs well when labels are sparse.

Across-network Node Classification

- Merge two networks into one, feed it to embedding methods
- One network as training data, the other one as test.
- Train a classifier on the training data
  to predict labels of nodes in the other network



Across-network Node Classification

MACRO-F_1(%)) SCORES OF ACROSS-NETWORK ROLE CLASSIFICATION.

Algorithm	Dataset			
	USA:Europe	Europe:USA	Actor:USA	USA:Actor
node2vec	42.92	45.99	46.91	42.88
struc2vec	78.87	**79.74**	**80.13**	57.48
GraphWave	**86.17**	73.98	—	—
RiWalk-SP	**81.98**	**80.07**	78.95	**73.97**
RiWalk-WL	81.95	78.99	**80.90**	**67.34**
Majority	42.91	42.87	42.87	42.86

- Structural embedding can transfer across networks.

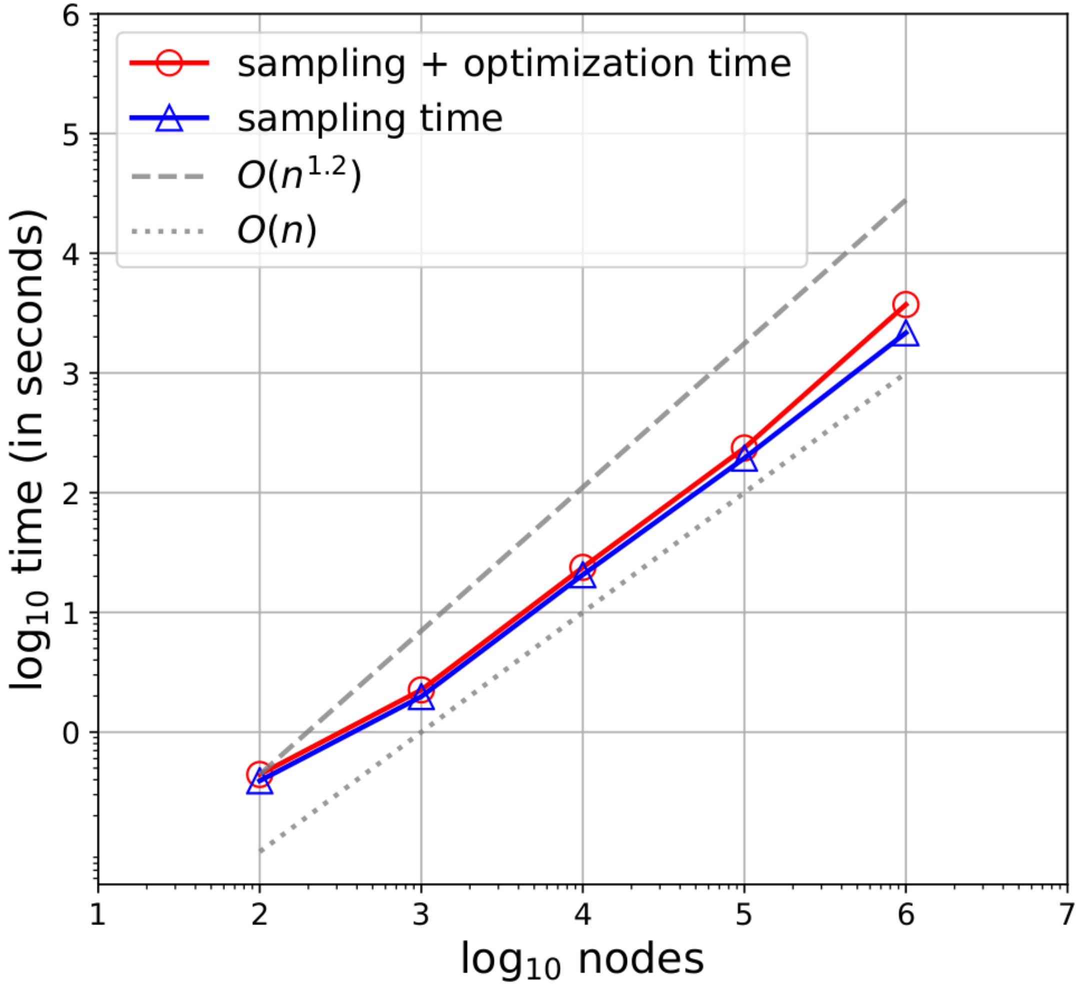
Across-network Node Classification

MACRO-F_1(%)) SCORES OF ACROSS-NETWORK ROLE CLASSIFICATION.

Algorithm	Dataset			
	USA:Europe	Europe:USA	Actor:USA	USA:Actor
node2vec	42.92	45.99	46.91	42.88
struc2vec	78.87	**79.74**	**80.13**	57.48
GraphWave	**86.17**	73.98	—	—
RiWalk-SP	**81.98**	**80.07**	78.95	**73.97**
RiWalk-WL	81.95	78.99	**80.90**	**67.34**
Majority	42.91	42.87	42.87	42.86

- Structural embedding can transfer across networks.
- RiWalk is robust when transferring from small networks to large networks

Scalability

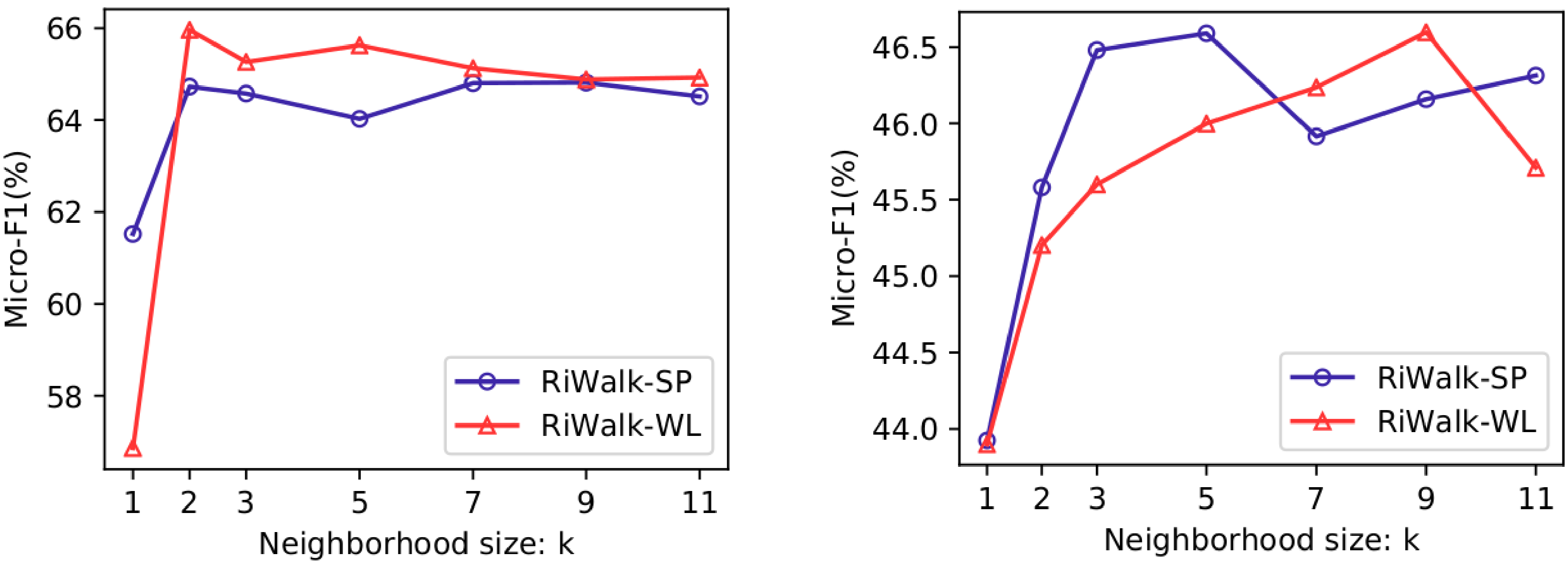


Running time on Erdos-Renyi graphs (constant degree of 10)

Thanks

Xuewei Ma: xuew.ma@gmail.com

Code: github.com/maxuewei2/RiWalk



(a) USA (b) Actor

Performance *w.r.t.* neighboorhood size k.