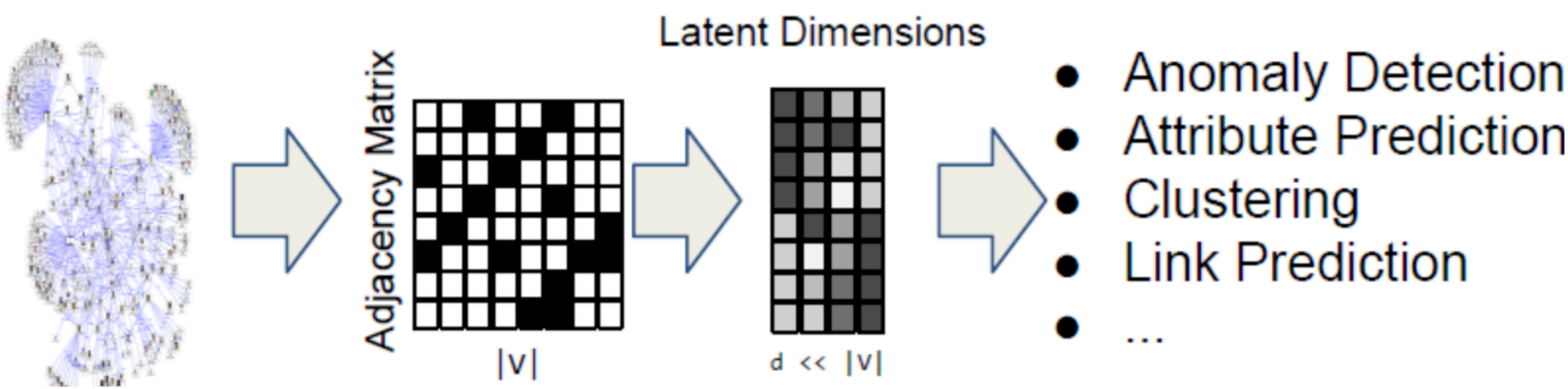
LATTE: AppLicAtion OrienTed Social NeTwork Embedding

Lin Meng, Jiyang Bai, **Jiawei Zhang**

IFM Lab,
Florida State University, FL, USA

What is Network Embedding?

- **Network Embedding: Map nodes in a network into a low-dimensional space**
  - *Distributed representation for nodes*
  - *Similarity between nodes indicate the link strength (i.e., preserve the original network structure)*
  - *Encode network information and generate node representation*



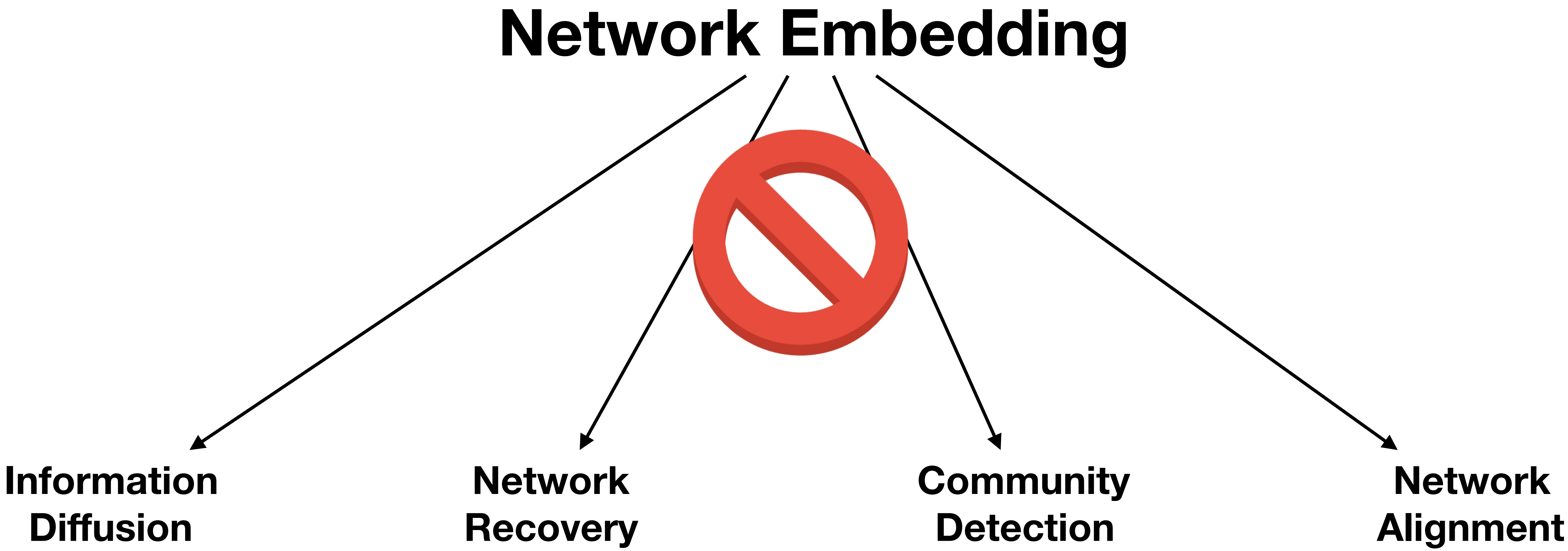
What is Network Embedding?

- **Network Embedding: Problem Formulation**
  - Given a network *G =(V, E), V* denotes the node set, and *E* denotes the link set
  - Network Embedding problem aims at learning a mapping

$$f : V \to \mathbb{R}^d (d \ll |V|)$$

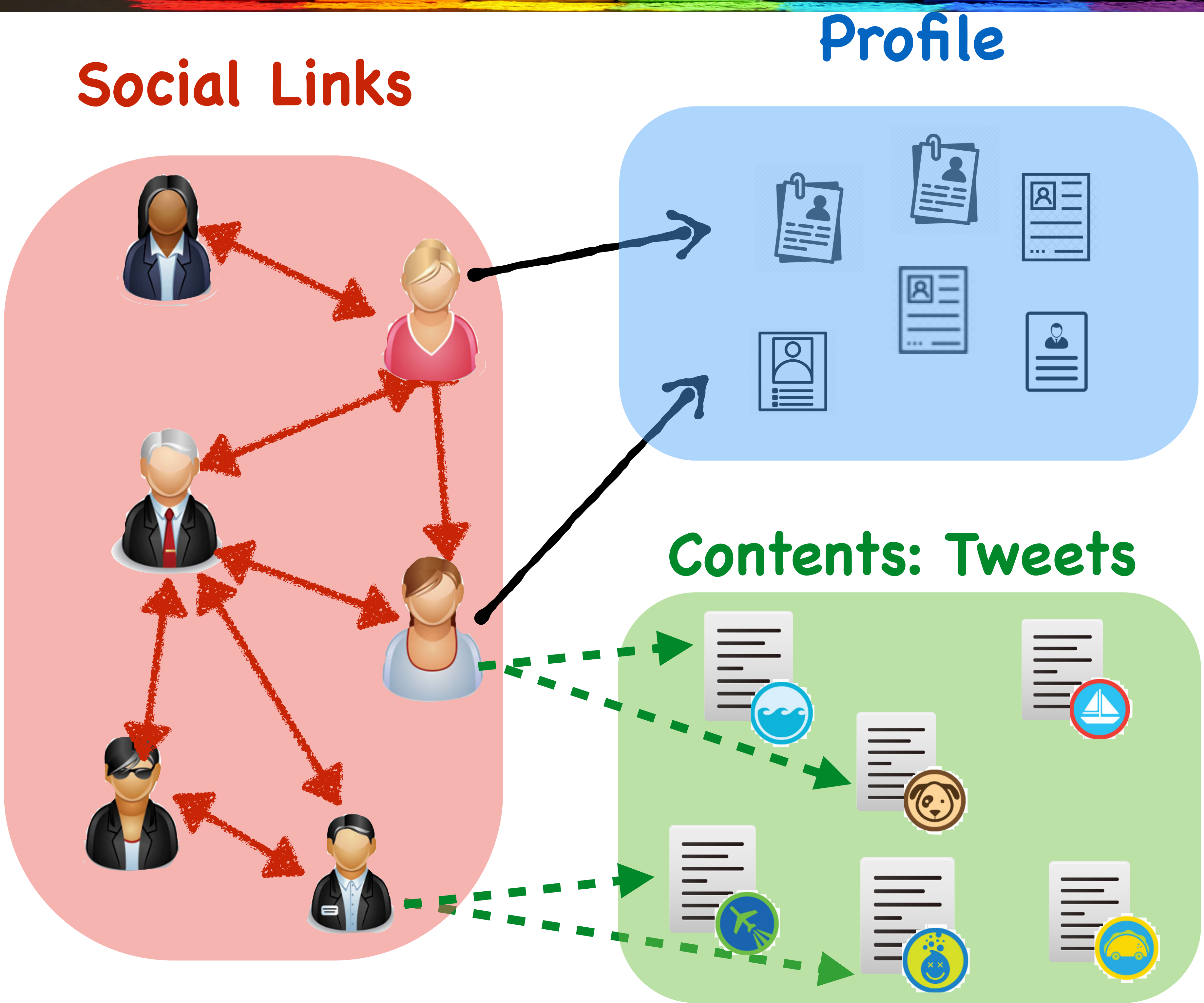
  - Requirements
    - d << |V|, storage space can be greatly saved
    - Network structure can be preserved and recovered based on the learned embedding feature vectors $\{f(v)\}_{v \in V}$
      - i.e., close nodes has close representations

One Embedding Fits All ?



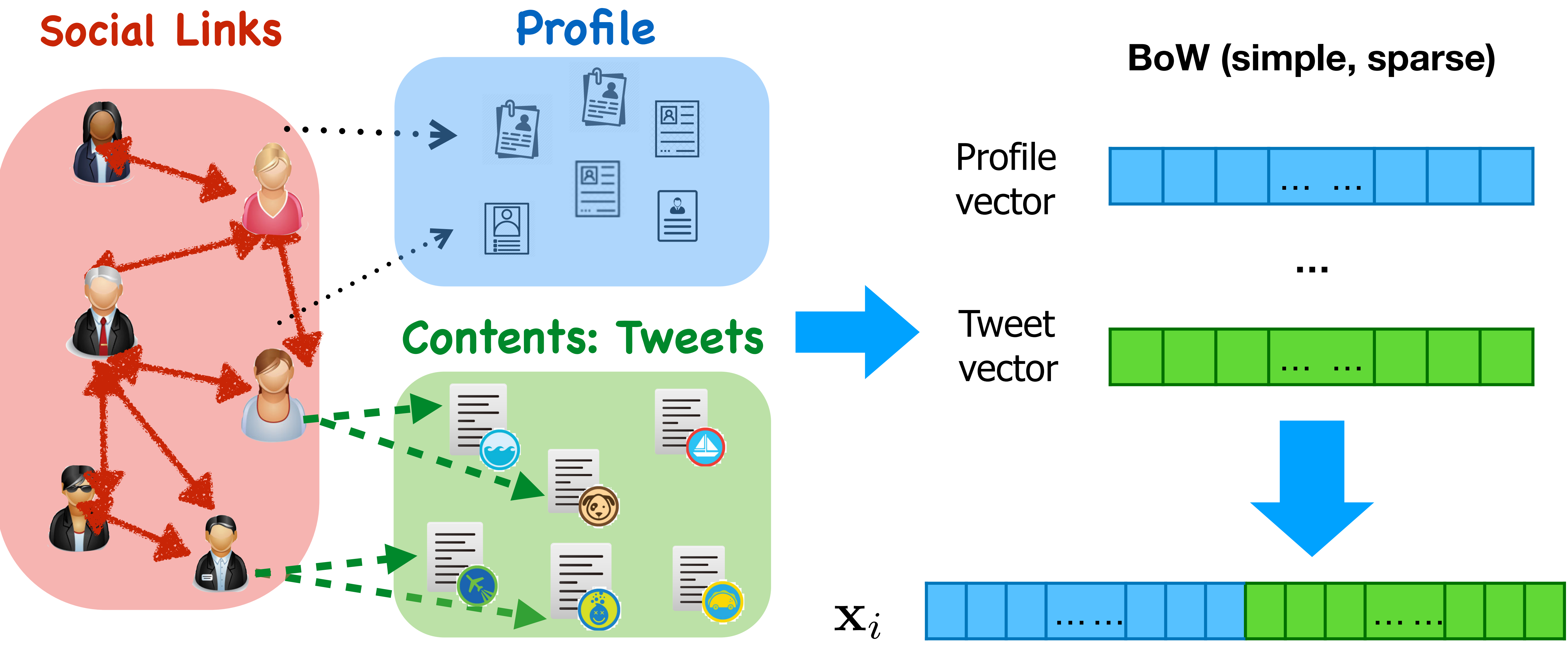
Application objectives should be involved in the embedding.

Challenge 1: Diverse Information in Social Networks



Raw Feature Embedding via Masked Autoencoder

Raw feature extraction

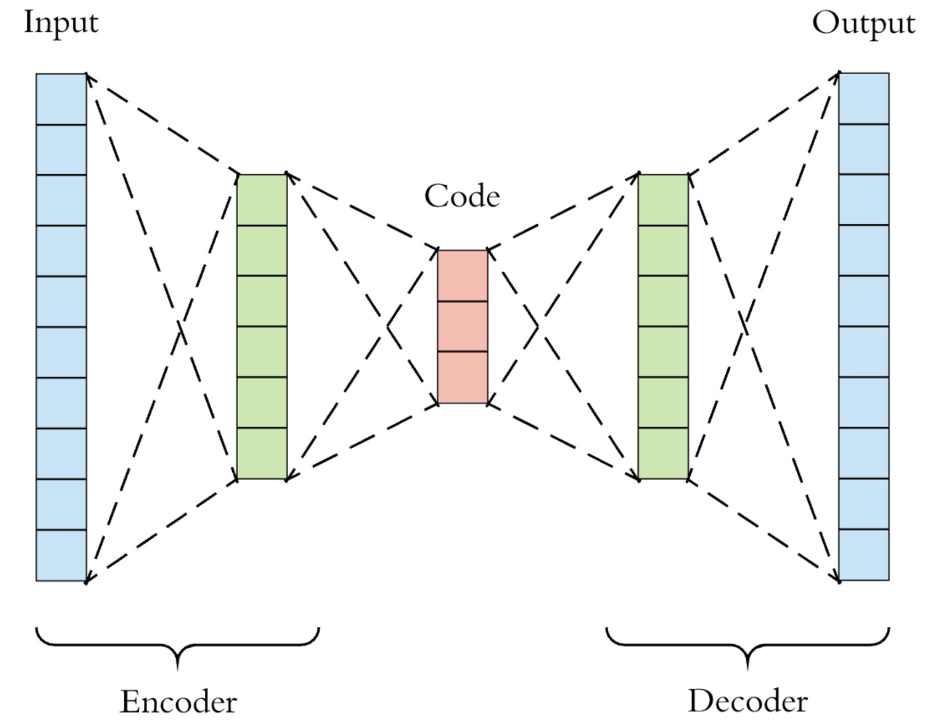


Raw Feature Embedding via Masked Autoencoder

Autoencoder Model

Encoder:
$$\begin{cases} \mathbf{y}_i^1 &= \sigma(\mathbf{W}^1 \mathbf{x}_i + \mathbf{b}^1), \\ \mathbf{y}_i^k &= \sigma(\mathbf{W}^k \mathbf{y}_i^{k-1} + \mathbf{b}^k), \forall k \in \{2, 3, \cdots, o\} \\ \mathbf{z}_i &= \sigma(\mathbf{W}^{o+1} \mathbf{y}_i^o + \mathbf{b}^{o+1}). \end{cases}$$

Decoder:
$$\begin{cases} \hat{\mathbf{y}}_i^o &= \sigma(\hat{\mathbf{W}}^{o+1} \mathbf{z}_i + \hat{\mathbf{b}}^{o+1}), \\ \hat{\mathbf{y}}_i^{k-1} &= \sigma(\hat{\mathbf{W}}^k \hat{\mathbf{y}}_i^k + \hat{\mathbf{b}}^k), \forall k \in \{2, 3, \cdots, o\} \\ \hat{\mathbf{x}}_i &= \sigma(\hat{\mathbf{W}}^1 \hat{\mathbf{y}}_i^1 + \hat{\mathbf{b}}^1). \end{cases}$$



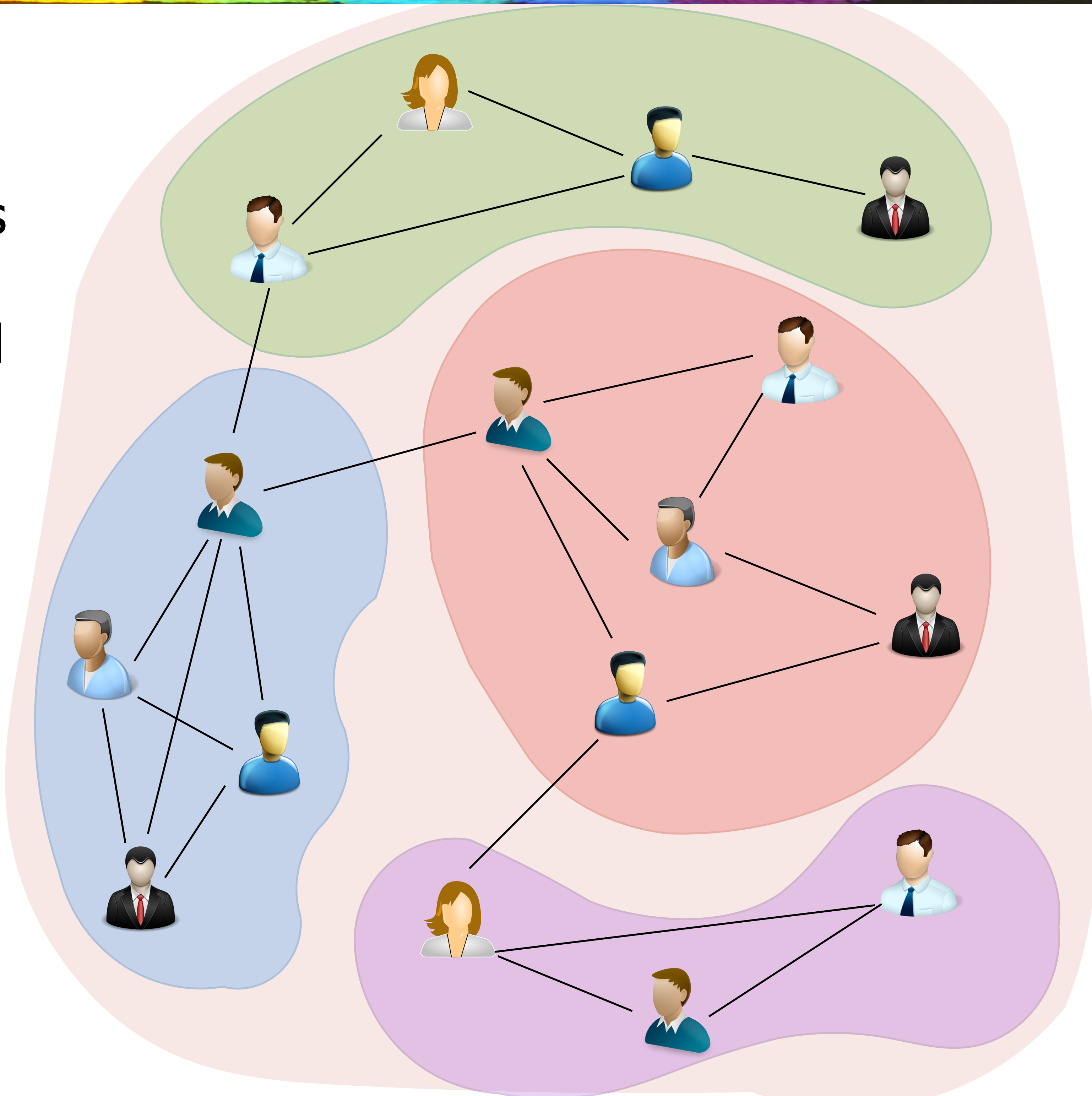
- Loss term for Autoencoder model

$$\mathcal{L}_a(G) = \sum_{v_i \in \mathcal{V}} \left\| (\mathbf{x}_i - \hat{\mathbf{x}}_i) \odot \mathbf{c}_i \right\|_2^2$$

To avoid the trivial solutions, we use a mask vector $\mathbf{c}_i$.

Challenge 2: Local and Global Network Structure Preservation

- Graph representation models usually captures local structures
- Global network structure should be also preserved



How to preserve both local and global network structure in a unified framework?

Graph Structure Embedding via Proximity Preservation

Diffusive Network Proximity: normalized transition matrix B

$$\mathbf{B} = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} (\mathbf{D}^{\top})^{-\frac{1}{2}} \qquad \mathbf{B}_n = \sum_{i=1}^{n} \omega_i \mathbf{B}^i$$

- $\mathbf{A}$ is the adjacency matrix of the input network
- $\mathbf{D}$ is the diagonal matrix of $\mathbf{A}$

$\mathbf{B}_n$ will be used as the network local and global proximity matrix.

Loss term with Diffusive Network Proximity:

$$\mathcal{L}_n(G) = -\sum_{v_i \in \mathcal{V}} \sum_{v_j \in \mathcal{V}, v_i \neq v_j} B_n(i,j) \log p(v_i, v_j)$$

$$p(v_i, v_j) = \frac{1}{1 + \exp(-\mathbf{z}_i^{\top} \cdot \mathbf{z}_j)}.$$

Graph Structure Embedding via Proximity Preservation

Joint Objective function for Embeddings

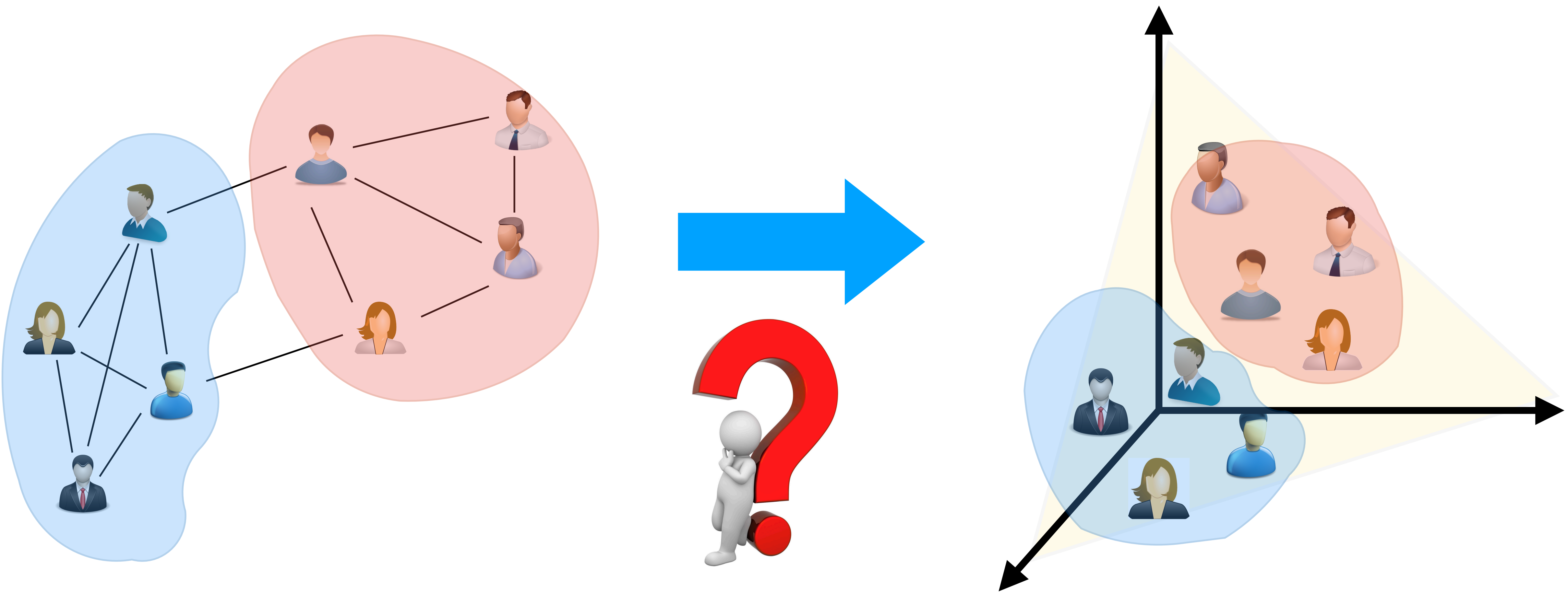
$$\mathcal{L}_e(G) = \mathcal{L}_a(G) + \mathcal{L}_n(G) + \theta \cdot \mathcal{L}_{reg}$$

- θ denotes the weight for the regularization term

Regularization term

$$\mathcal{L}_{reg} = \sum_{k=1}^{o} \left(\left\| \mathbf{W}^k \right\|_F^2 + \left\| \hat{\mathbf{W}}^k \right\|_F^2 \right)$$

Challenge 3: External Application Incorporation



How to effectively incorporate the external application tasks in the network representation learning is still an open question by this context so far.

Task Oriented Network Embedding Objective Function

$$\min c \cdot \mathcal{L}_e(G) + (1-c) \cdot \mathcal{L}_t(G)$$

The loss term of specific application tasks

The objective function of specific application tasks

- **Network Alignment**
- **Community detection**
- **Information Diffusion**

$$\begin{aligned}
\mathcal{L}_t(G) = \min_{\mathbf{w},\mathbf{y}} \ & \frac{1}{2} \cdot \|\mathbf{w}\|_2^2 + \frac{k}{2} \cdot \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2, \\
s.t. \quad & \mathbf{y} \in \{0,1\}^{|\mathcal{L}|}, \ y_{i,j} = 1, \forall (u_i^{(1)}, u_j^{(2)}) \in \mathcal{A}^{(1,2)} \\
& 0 \le \sum_{u_i^{(1)} \in \mathcal{U}^{(1)}} y_{i,j} \le 1, \forall u_j^{(2)} \in \mathcal{U}^{(2)}, \\
& 0 \le \sum_{u_j^{(2)} \in \mathcal{U}^{(2)}} y_{i,j} \le 1, \forall u_i^{(1)} \in \mathcal{U}^{(1)},
\end{aligned}$$

$$\mathcal{L}_t(G) = \frac{1}{2}\mathrm{Tr}(\mathbf{Z}^\top \mathbf{L}_{\mathbf{A}_u} \mathbf{Z}) + \beta \cdot \left\|\mathbf{Z}^\top \mathbf{Z} - \mathbf{I}\right\|_F^2$$

$$\mathcal{L}_t(G) = \sum_{o_i \in \mathcal{O}} \sum_{u_j, u_k \in \mathcal{U}} s_{j,k}^{o_i} \|\mathbf{z}_j - \mathbf{z}_k\|_2^2$$

Experimental Dataset

Experiment Dataset

	property	Twitter
# node	user	5,223
	tweet	9,490,707
	location	297,182
# link	friend/follow	164,920
	write	9,490,707
	locate	615,515

Experimental Settings

- **We only show the results on community detection based network embedding in the slides**

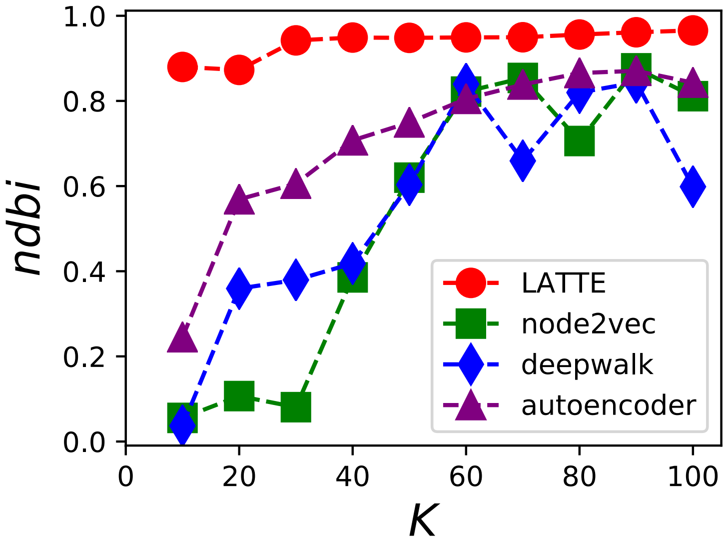
Comparison Methods

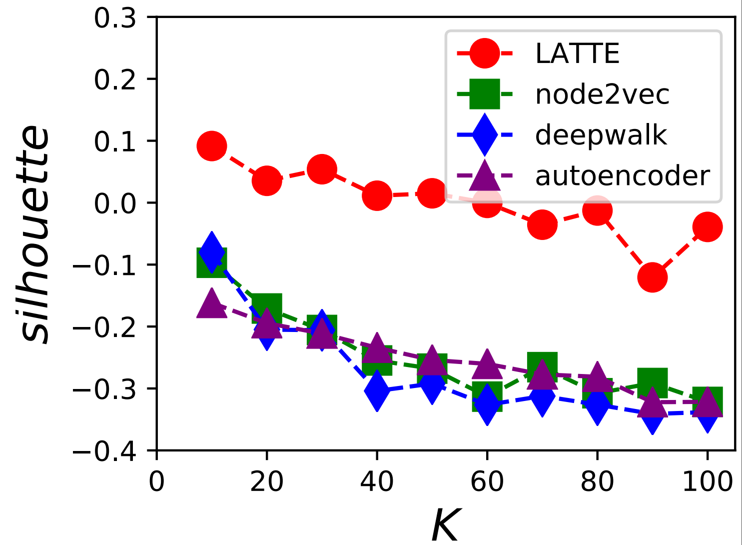
Task Oriented Model	LATTE
Classic Network Embedding Model	Node2vec, Deepwalk, Autoencoder

Evaluation Task & Metrics

	Evaluation Metrics			
Community Detection	NDBI	Silhouette	Density	Entropy

Experimental Result: Community Detection Task



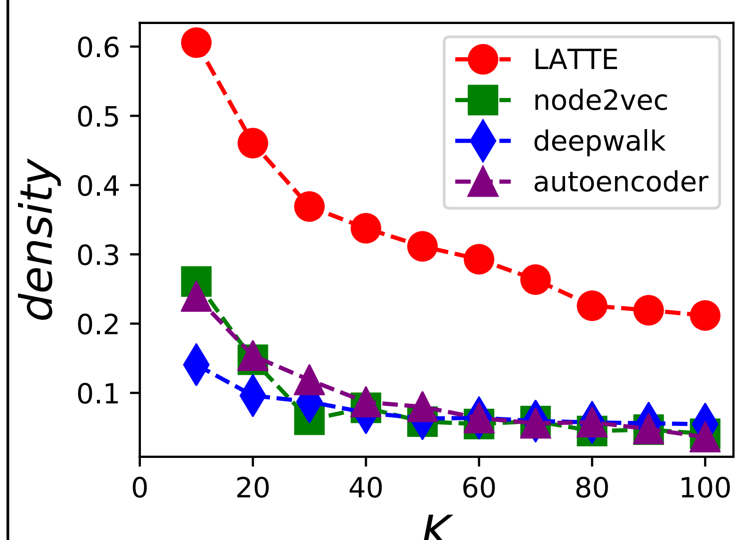


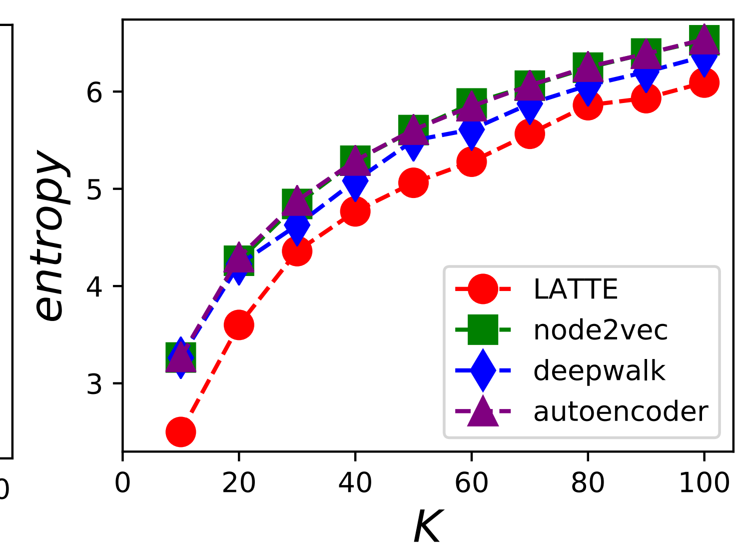
Observations:

1. LATTE incorporating the community detection objective in the framework can outperform the other pure network embedding models with great advantages
2. The density obtained by LATTE is almost one time greater than those obtained by the other baseline methods

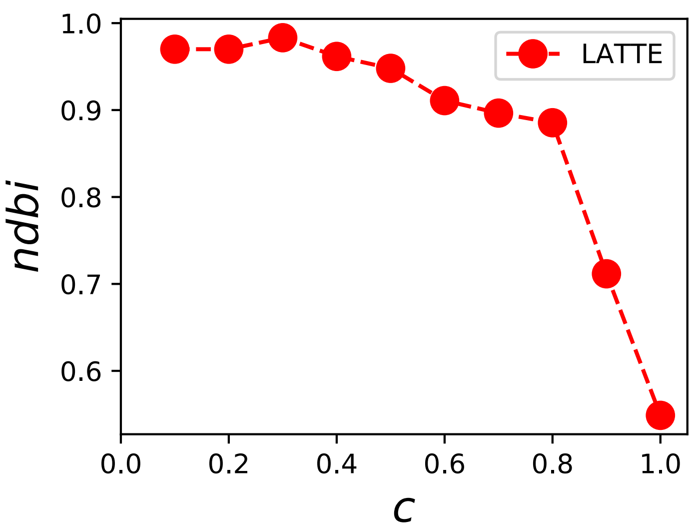
Explanation:

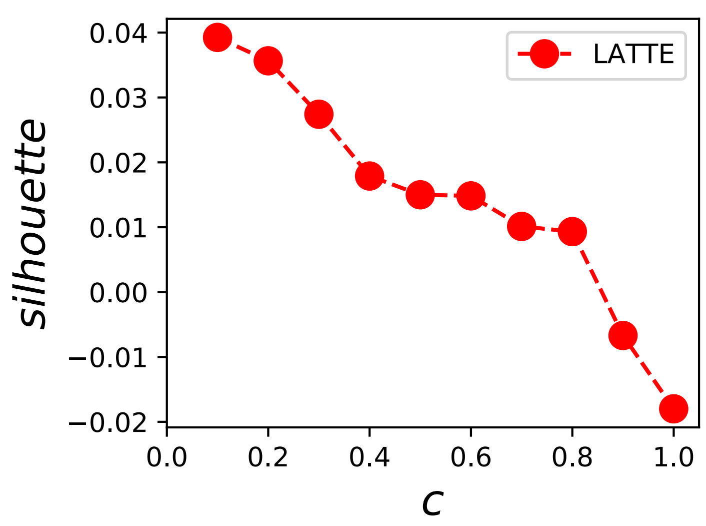
1. The community structures obtained by LATTE seems to be more balanced and reasonable compared with the community structures detected by the other methods.
2. LATTE consider the edge cut loss in the embedding process, and the learned representation feature vectors can effective indicate the optimal community partition results of the network

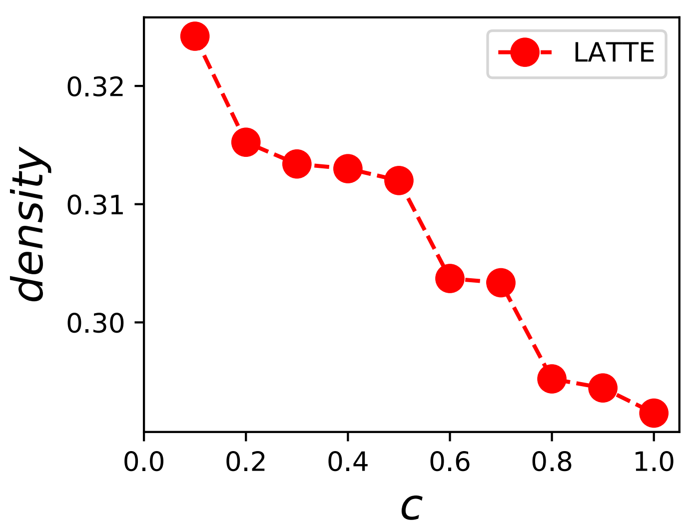


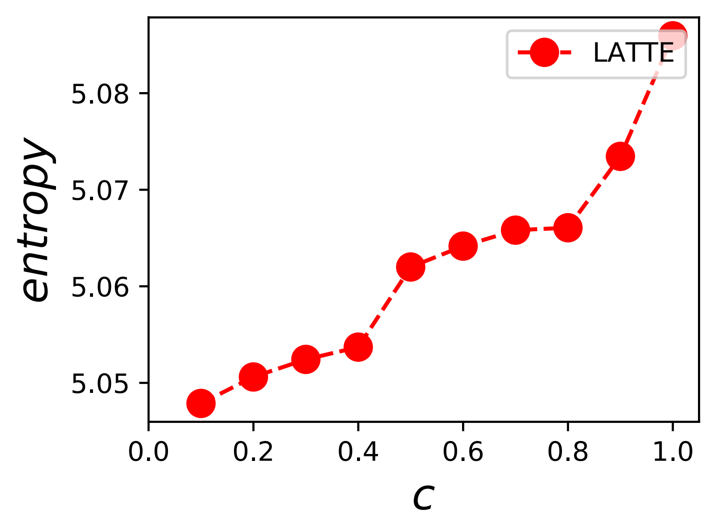


Experimental Result: Parameter Sensitivity Analysis about c on Community Detection Task









$$\min c \cdot \mathcal{L}_e(G) + (1 - c) \cdot \mathcal{L}_t(G)$$

Observations:

1. *As c value increases, the performance of LATTE will generally degrade steadily*
2. *when c = 1.0, the performance of LATTE can still outperform the other baseline methods (with k = 50) on ndbi*

Explanations:

1. *As c increase, the framework will aim optimizing the embedding component instead of the application task, and the learned embedding feature vectors can mainly reflect the embedding objective instead.*
2. *When c = 1.0, the performance of utilizing the heterogeneous information for the network representation learning in LATTE can helpfully capture better social community structure than the other network embedding methods.*

LATTE: Application Oriented Social Network Embedding

Q&A

Jiawei Zhang
jzhang@cs.fsu.edu