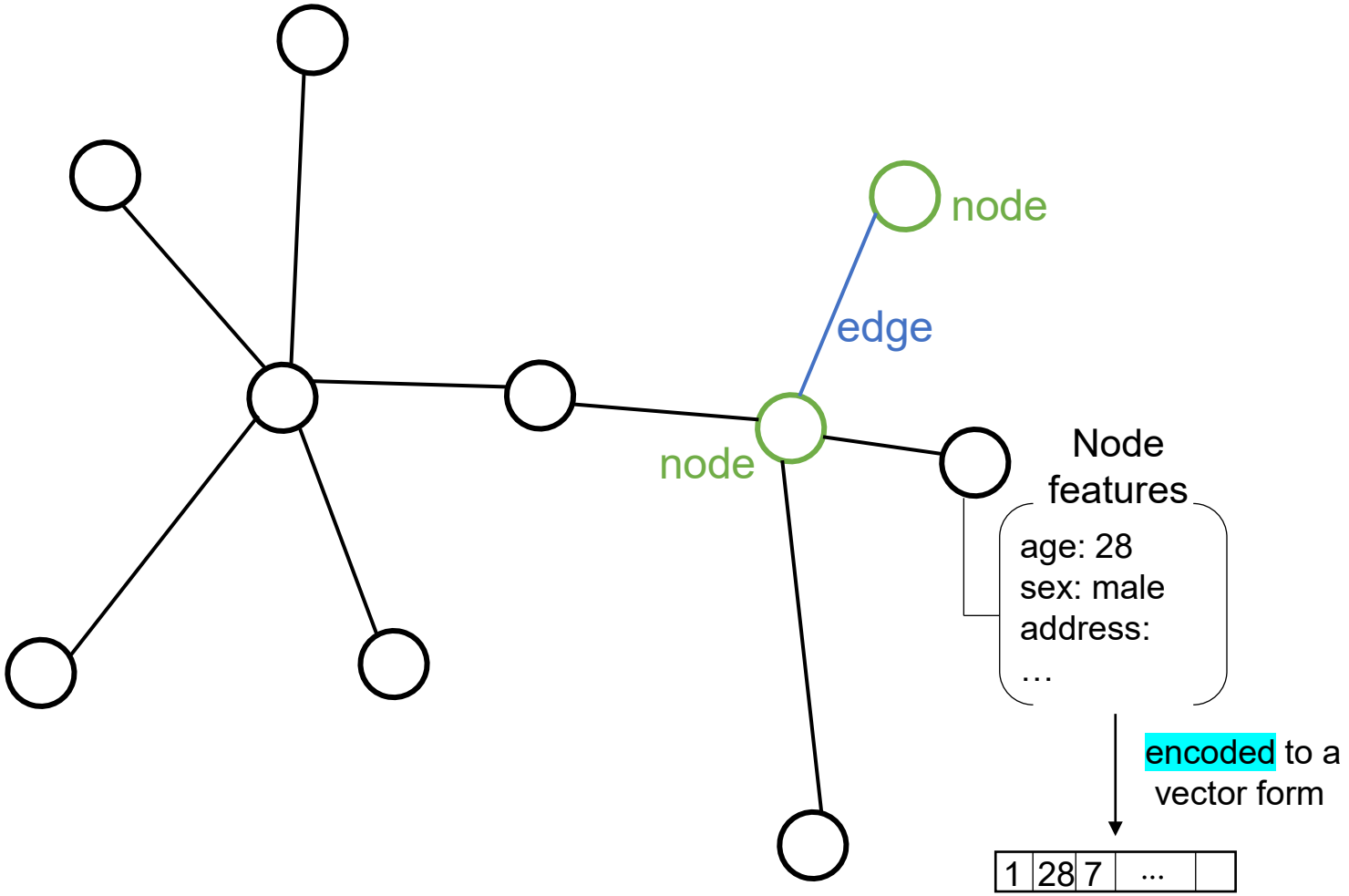
Survey on
Network Alignment Methods

Presentor: Jinduk park

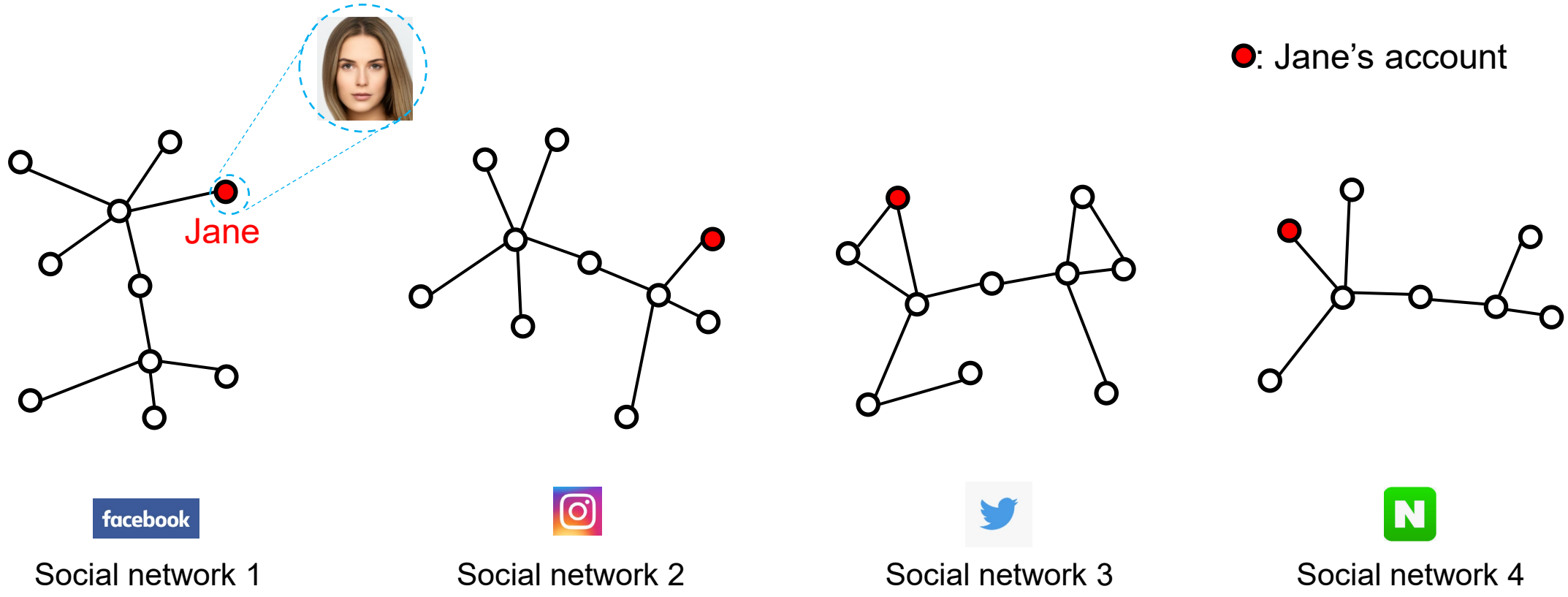
Introduction

Network data



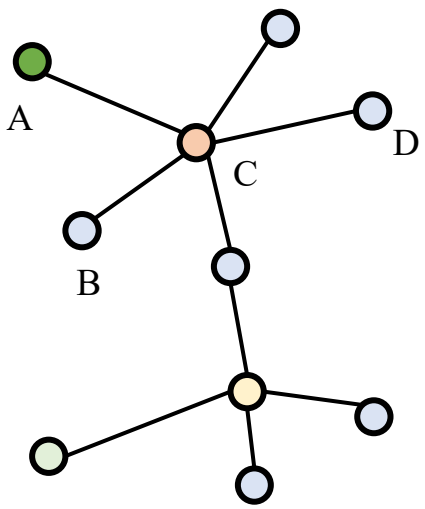
Network is a data structure that consists of nodes, edges, and its features.

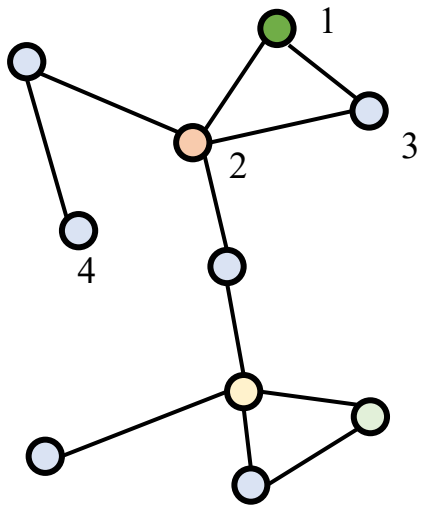
What is network alignment?



In real-world, It is natural for a same node to appear in multiple networks (e.g. user accounts across multiple platforms).

What is network alignment?





Network alignment

V_s		V_t
A	-	1
B	-	3
C	-	2
…		

Node correspondence

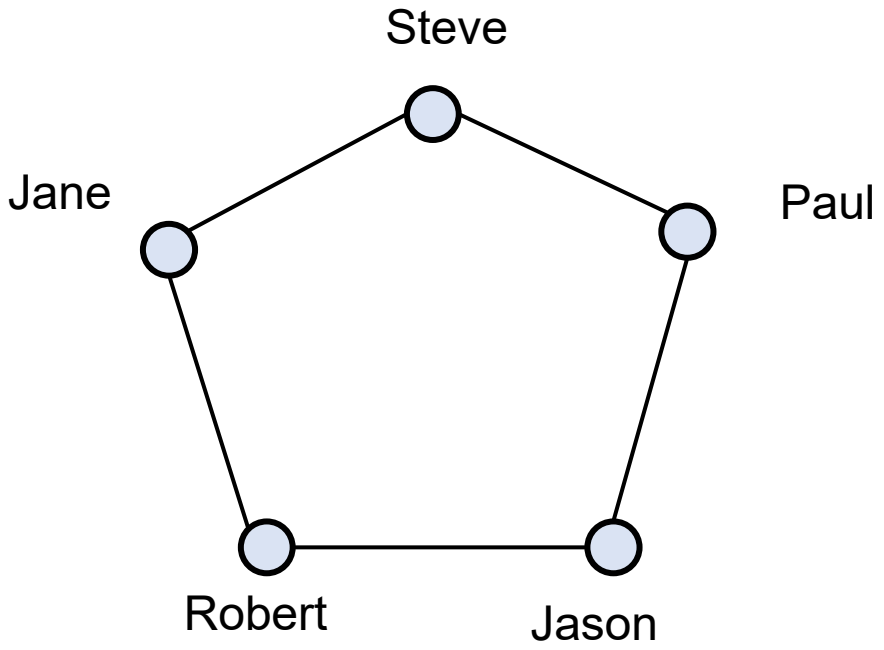
$G_s = (V_s, E_s, X_s)$

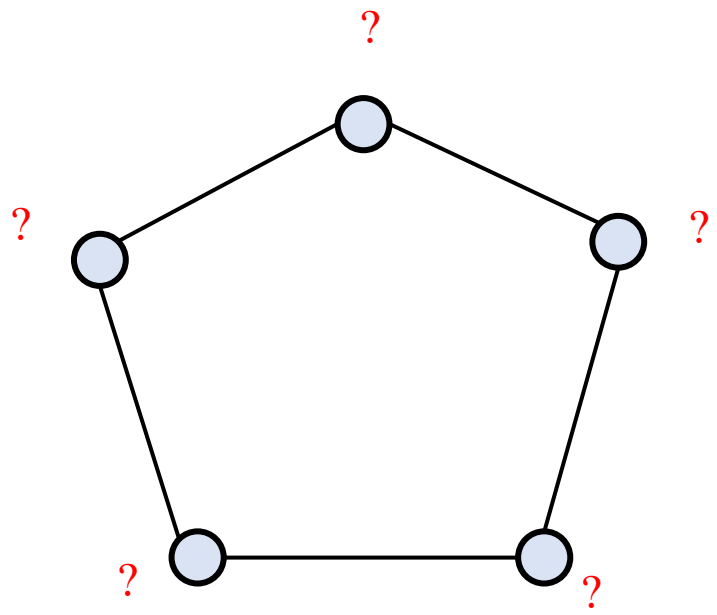
$G_t = (V_t, E_t, X_t)$

* (V, E, X) =(nodes, edges, node attributes)

Network alignment is to find node correspondence between two network data considering its topological & attribute relation of nodes.

Why NA task is needed?



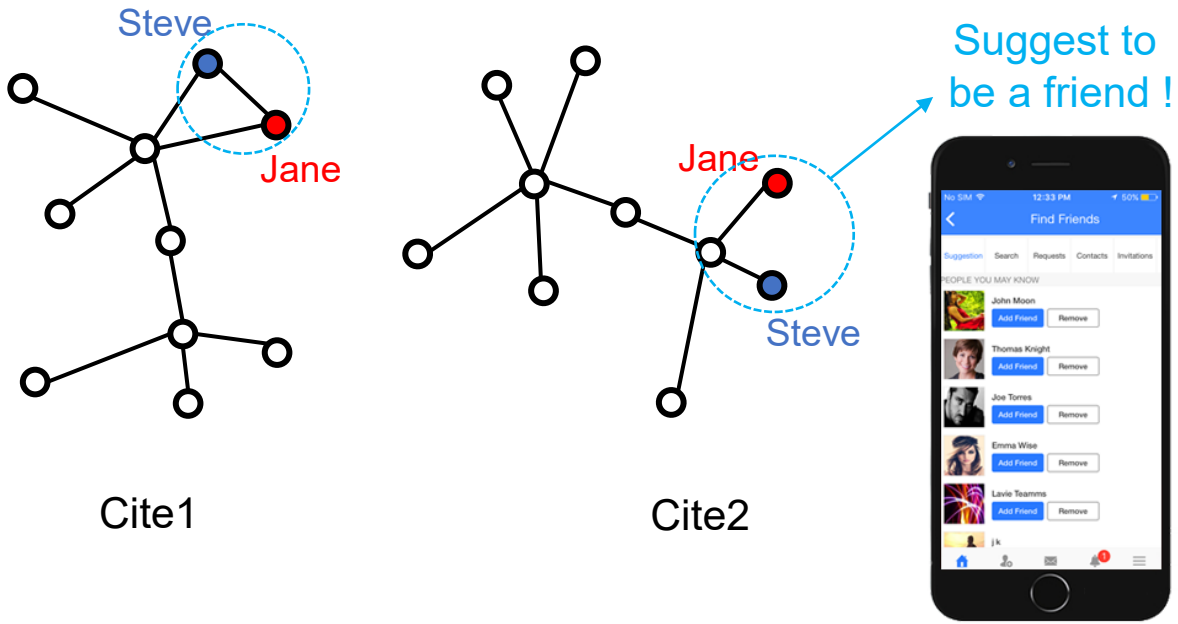


Can you find which node is about Jane?

Graph or network data is not in a spatial domain.

What is network alignment?

Network alignment is often the very first task to solve the tasks on multiple networks.



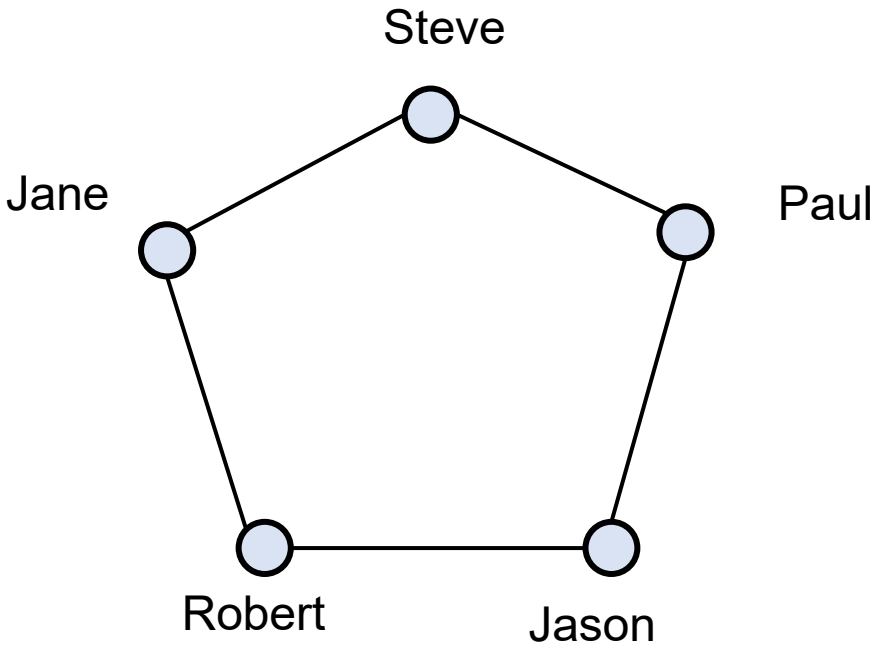
[Tasks / applications]

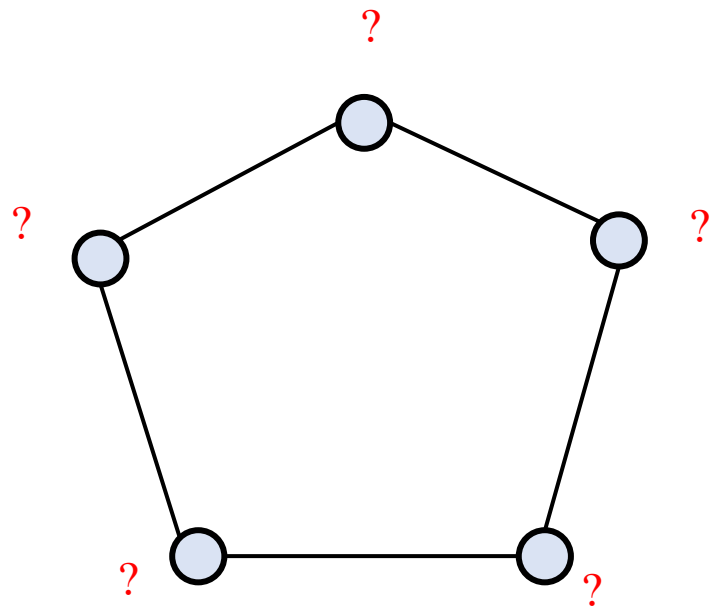
- Cross-site item recommendation
- Friend suggestion
- Advertisement
- Bioinformatics
- Discovering protein-protein interactions
- …

IMG source: https://socialapps.tech/content/new-feature-friend-suggestions-ios-android-mobile-apps-socialengine

Prior information in network alignment (a.k.a prior anchor links)

Find one-to-one correspondence of nodes is difficult in some conditions
(when each node is not distinctive both in network structure and node attributes).

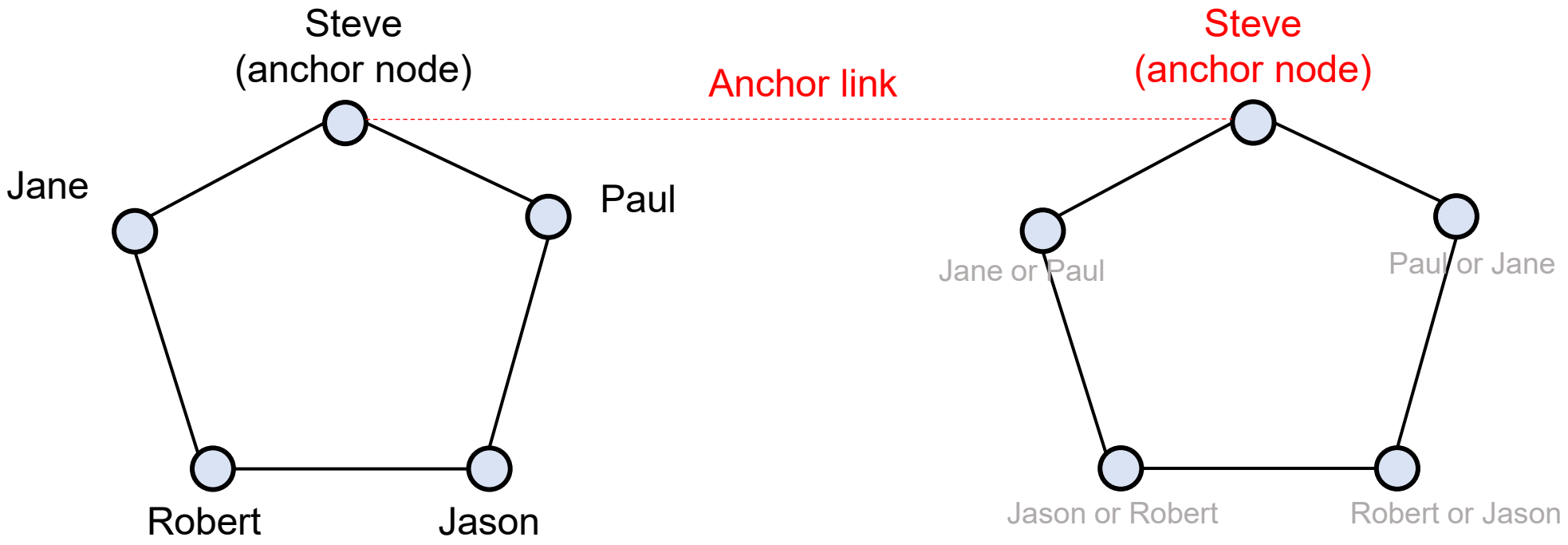




* For this example, there are 10 possible alignment cases
if we consider topological similarity of nodes.

Prior information in network alignment

Small number of prior matching information (also known as anchor links and anchor nodes) is very helpful finding correspondence of other nodes.



* What if we know 1 node correspondence beforehand?

: Now the candidate cases are greatly reduced from 10 to only 2 case !

Prior information in network alignment

Practically, such prior information is available also in real-world.

: e.g.) The accounts that have extremely distinguishable feature
(# of follower, official mark, etc.)
or we already know who he/she is beforehand.

*celebrity accounts

NN-based NA Approaches

Classical approach

Stemmed from graph matching, where only topological information is available for the graph pair, the problem can be formulated as:

Finding permutation matrix $\mathbf{P}$ that

$$\min_{\mathbf{P}} ||\mathbf{PAP}^T - \mathbf{B}||_F^2,$$

$\mathbf{A}$: Adjacency matrix for the source graph

$\mathbf{B}$: Adjacency matrix for the target graph

Classical approach

However, permutation matrix have some properties.

$$P_\pi = \begin{bmatrix} \mathbf{e}_{\pi(1)} \\ \mathbf{e}_{\pi(2)} \\ \mathbf{e}_{\pi(3)} \\ \mathbf{e}_{\pi(4)} \\ \mathbf{e}_{\pi(5)} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_4 \\ \mathbf{e}_2 \\ \mathbf{e}_5 \\ \mathbf{e}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}.$$

- Each col & row summed up to 1
- Binary discrete element (0, 1)

⬇

Only one element should be 1 otherwise 0 at every rows & columns.

Classical approach

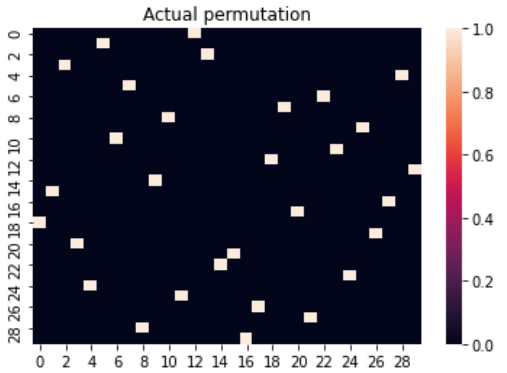
If we are to apply deep-learning,
how can we give such constraints on the output prediction?

A_1

A_2

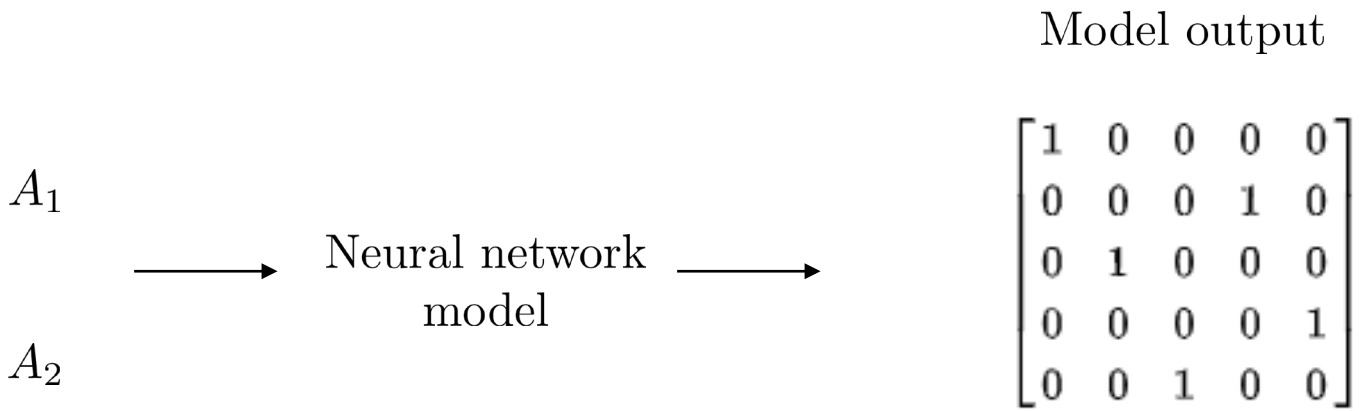
→ Neural network model →

Model output



Classical approach

In this descrete model output, we cannot compute gradient,
thus we cannot update parameters in the DNN.



There comes Sinkhorn operation
for the back-propagation.

Sinkhorn operator

$$
\begin{aligned}
S^0(X) &= \exp(X), \\
S^l(X) &= \mathcal{T}_c\left(\mathcal{T}_r(S^{l-1}(X))\right) \\
S(X) &= \lim_{l\to\infty} S^l(X).
\end{aligned}
$$

Where $\mathcal{T}_c, \mathcal{T}_r$ represent column-wise, row-wise normalization, respectively.

* softmax: differentiable

Theorem 1. *For a doubly-stochastic matrix P, define its entropy as $h(P) = -\sum_{i,j} P_{i,j} \log(P_{i,j})$. Then, one has,*

$$S(X/\tau) = \arg\max_{P\in\mathcal{B}_N} \langle P, X\rangle_F + \tau h(P). \tag{3}$$

Now, assume also the entries of X are drawn independently from a distribution that is absolutely continuous with respect to the Lebesgue measure in $\mathbb{R}$. Then, almost surely, the following convergence holds:

$$M(X) = \lim_{\tau\to 0^+} S(X/\tau). \tag{4}$$

↓ If you are interested in the proof:

Mena, Gonzalo, et al. "Learning latent permutations with gumbel-sinkhorn networks." ICLR(2018).

Sinkhorn operator

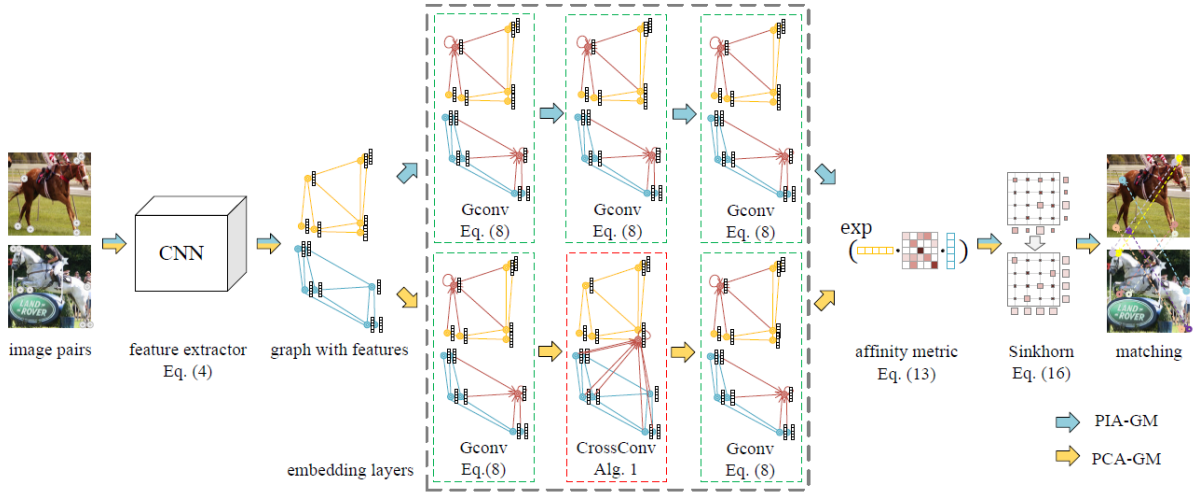
Limitation of the method : Scalability

Direct inference of n by n permutation matrix lacks scalability (NP-hard problem)

* nondeterministic polynomial time

if # of nodes is higher than few hundreds, performance drops significantly.

: usually applied for video domain, which dosen't need many vertices.



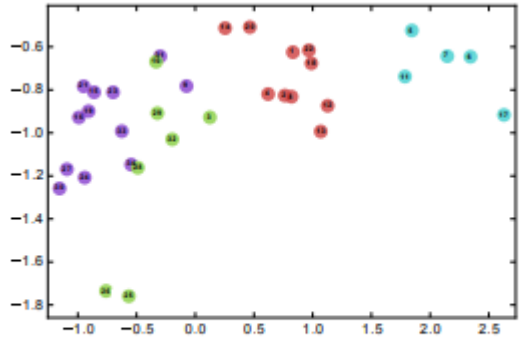
Wang, Runzhong, Junchi Yan, and Xiaokang Yang. "Learning combinatorial embedding networks for deep graph matching." *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019.

Break

Network embedding (NE)

(a) Input: Karate Graph

Network embedding



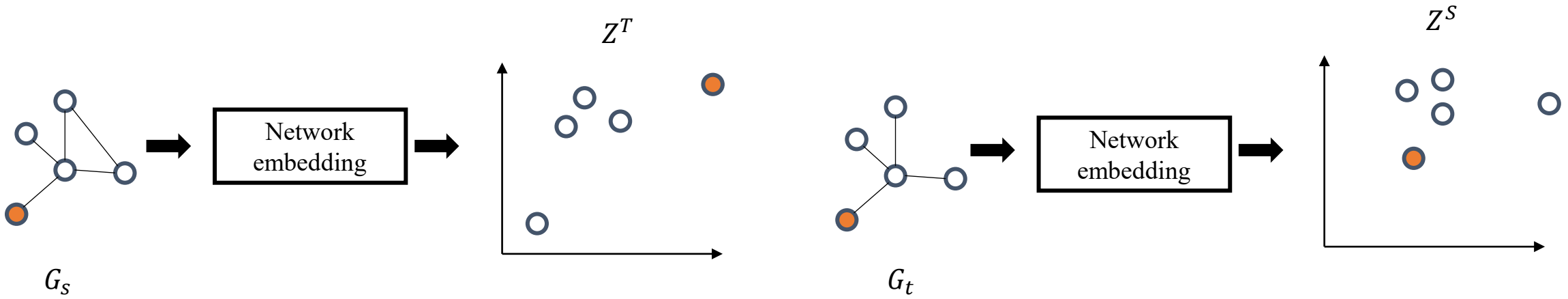
(b) Output: Representation

We can maps a graph(network) into a low dimensional representation in euclidean space preserving the network topology or properties.

→ Overcome scalability!

Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014.

Network embedding (NE) : challenges for multi-network embeddings

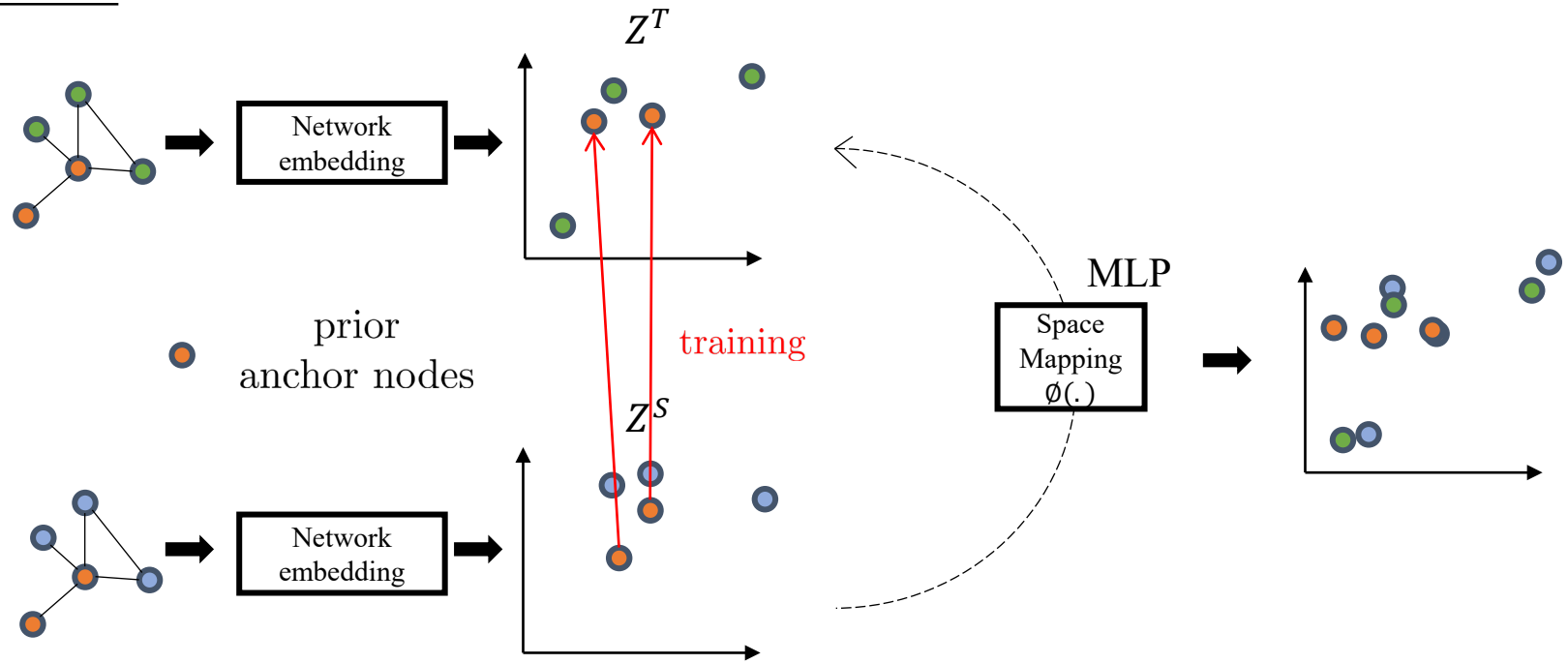


Some typical NE including LINE, DeepWalk, cannot generate embeddings consistently.

↓

Even though the two network has almost similar or same topology!

Find mappings via MLP



$$L_m(\phi, Z^s, Z^t, T) = \sum_{(v_l^s, u_n^t) \in T} ||\phi(z_l^s; \Theta) - z_n^t||_F,$$

Use prior anchor nodes as training set and others as inference.

* Man, Tong, et al. "Predict anchor links across social networks via an embedding approach." *Ijcai*. Vol. 16. 2016.

Find mappings via MLP : its limitations

1. Unsatisfactory performance

Hard to approximate patterns of the two embeddings
(relations between node pairs are often not consistent: low accuracy at inference step)

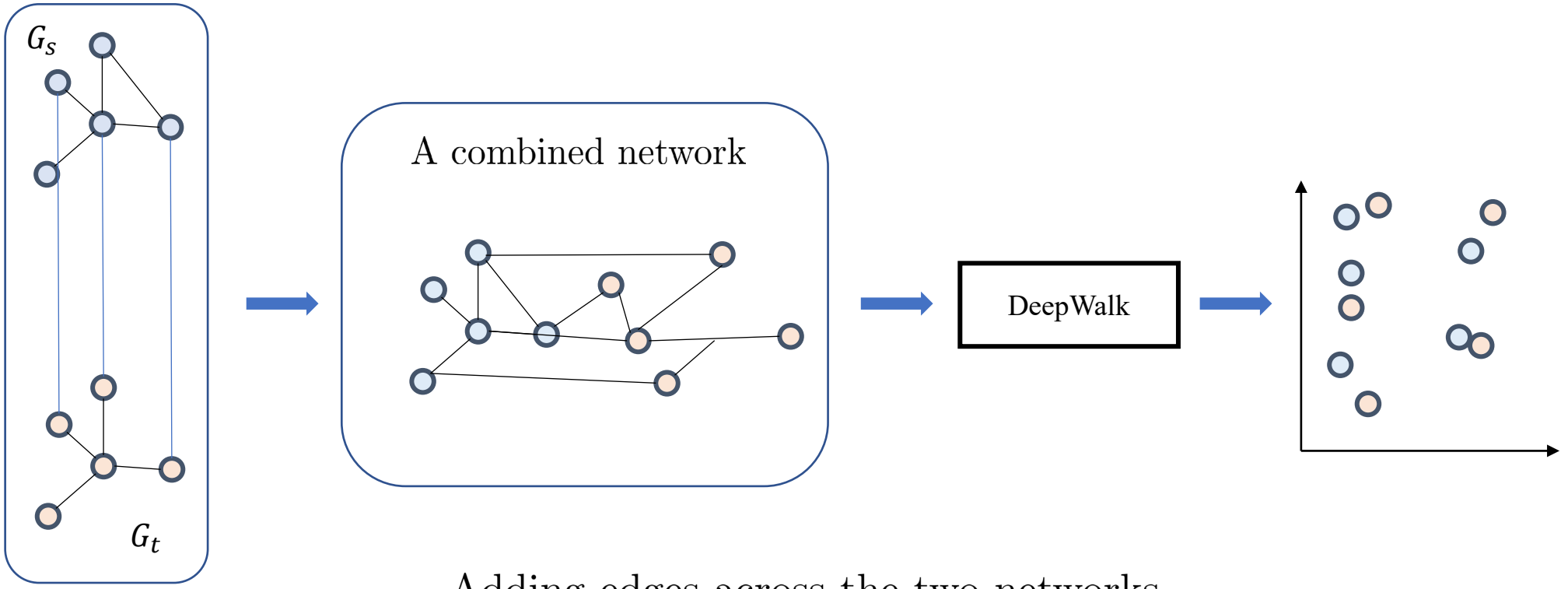
2. Can't utilize attribute info.

Network attribute info. is another key point for finding matchings.

3. Scalability

Many optimization processes.

Another approach with cross-edge construction



Adding edges across the two networks

by computing node similarities

1) Degree similarity
2) Node attribute similarity

Du, Xingbo, Junchi Yan, and Hongyuan Zha. "Joint Link Prediction and Network Alignment via Cross-graph Embedding." *IJCAI*. 2019.

Another approach with cross-edge construction : its limitations

1. Large computation cost

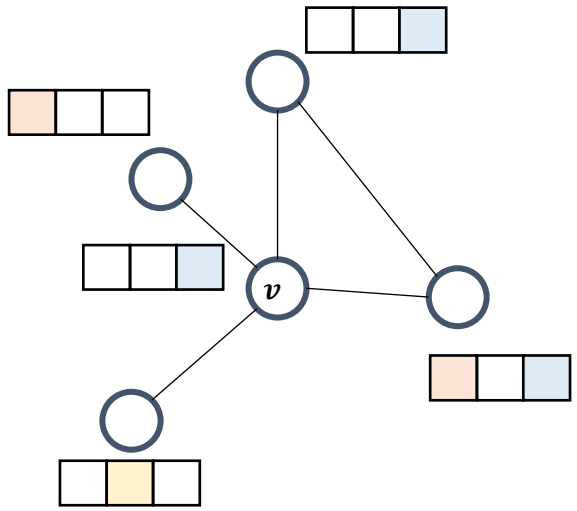
Needs at least $O(n^2)$ computation complexity
for constructing cross-network edges

2. Can't fully utilize attribute info.

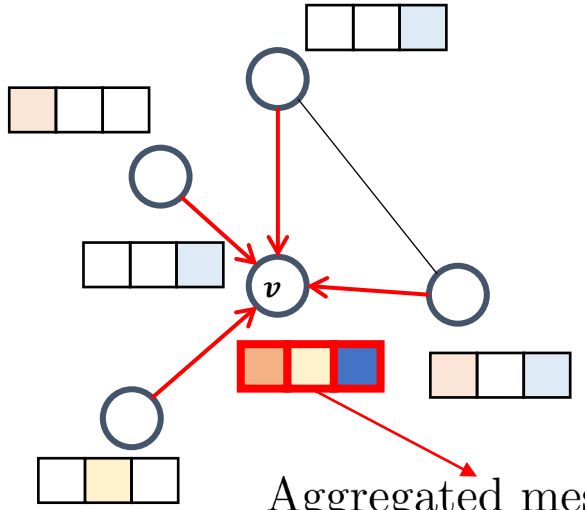
Still uses the network embedding schemes that don't consider attributes

Graph Neural Network (GNN)

0. Given network



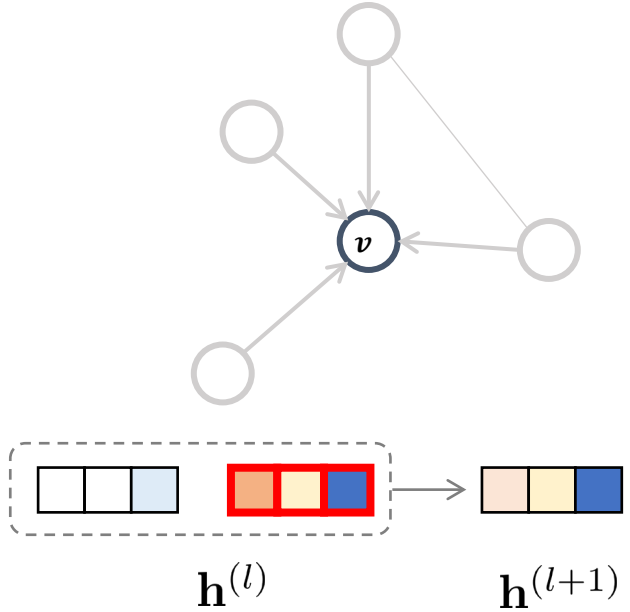
1. Message passing & aggregation



Aggregated message

(Summing up, taking average, max-pooling, ...)

2. Update representation

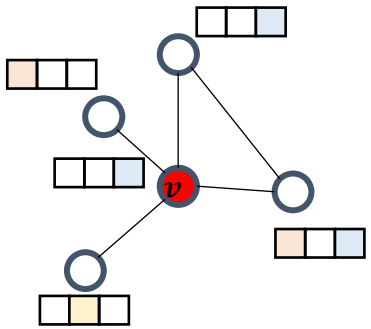


Usually by mulitplying trainable weight parameters (tough not all the case).

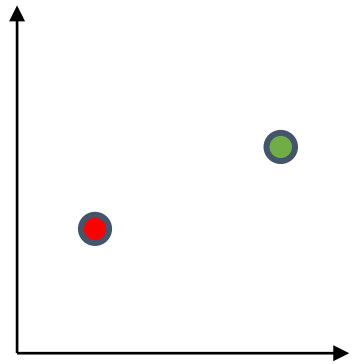
Consistent embedding with GNN

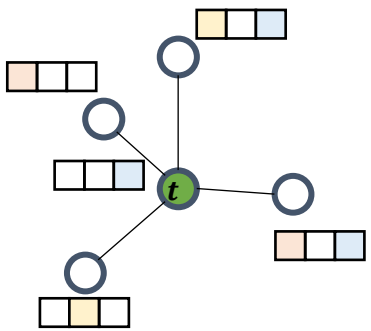
For example, in GraphSAGE layer with mean aggregator,

$$\mathbf{h}^{(l+1)} = \mathbf{W_1}\mathbf{h}^{(l)} + \mathbf{W_2} \cdot mean_{j \in \mathcal{N}_i} \mathbf{h}_j$$

G_s 

$$\longrightarrow \mathbf{v}^{(l+1)} = \mathbf{W_1}\mathbf{v}^{(l)} + \mathbf{W_2} \cdot mean_{j \in \mathcal{N}_i} \mathbf{v}_j \longrightarrow$$



G_t 

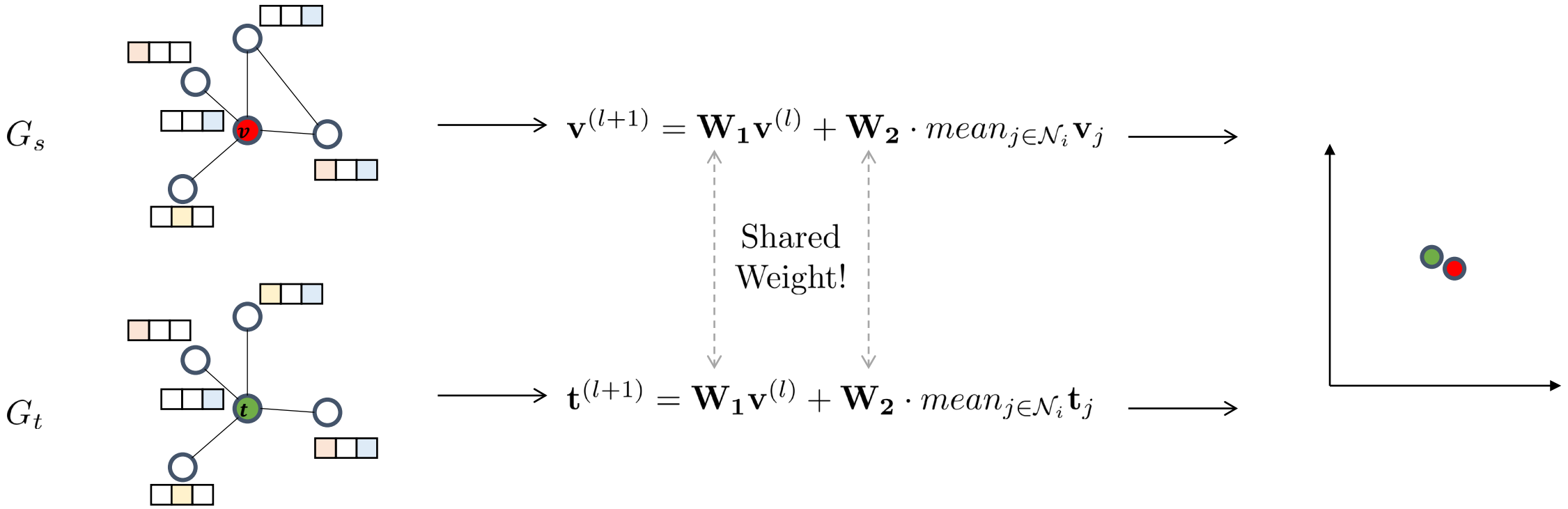
$$\longrightarrow \mathbf{t}^{(l+1)} = \mathbf{W}'_{\mathbf{1}}\mathbf{v}^{(l)} + \mathbf{W}'_{\mathbf{2}} \cdot mean_{j \in \mathcal{N}_i} \mathbf{t}_j \longrightarrow$$

Even though the two nodes have similar toplogy&attributes,
the embedding is generated far apart.

Hamilton, William L., Rex Ying, and Jure Leskovec. "Inductive representation learning on large graphs." *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017.

Consistent embedding with GNN

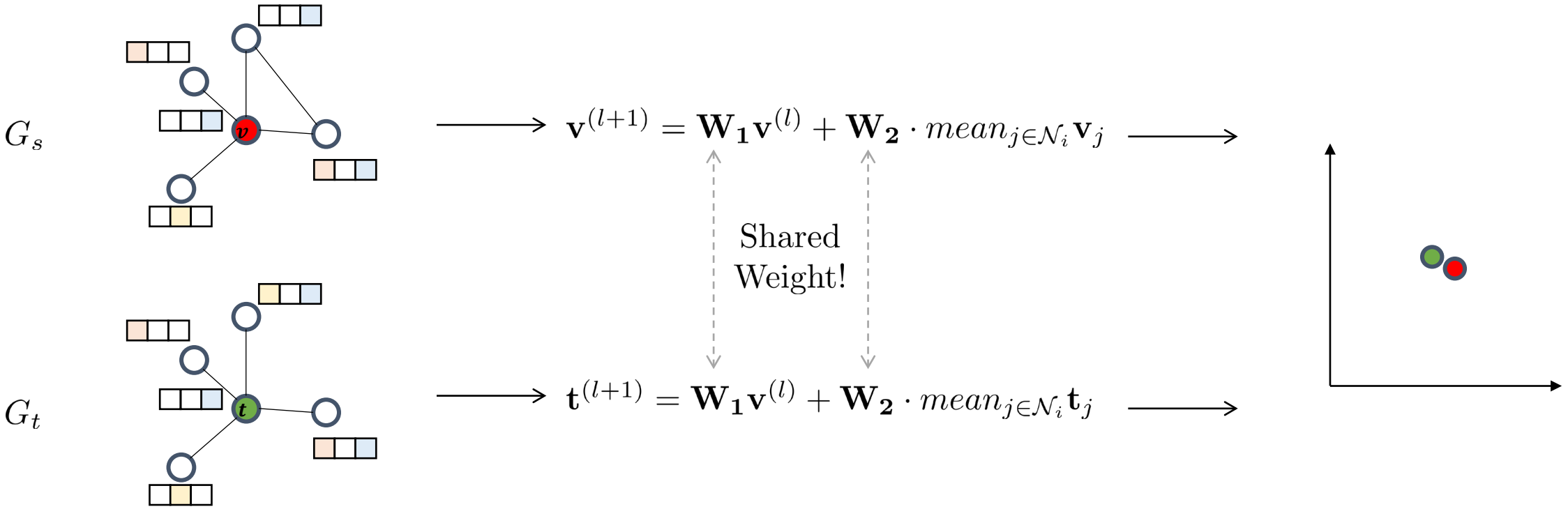
What if the two networks share same GNN parameters?



If two nodes has similar neighbor structure and attribute information,

now the representations of both become also similar!

Consistent embedding with GNN



1. Effective !

in that GNN is a powerful feature extractor for attributed network.

2. Scalable !

Low-dimensional reduction

Conclusion

Summary and conclusion

1.
With the recent development of neural network model,
a lot of progress has been made in neural network-based network alignment.

2.
Because of its effectiveness to represent proximity of nodes and scalability,
GNN has been emerged as an useful approach for solving NA task.

Thanks you for the listening.