Multilevel Network Alignment

Presented by Si Zhang (ASU)

Si Zhang Hanghang Tong Ross Maciejewski Tina Eliassi-Rad

Multiple Networks Are Prevalent!

Scenarios

social networks

transaction networks

PPI networks


yeast


elegans

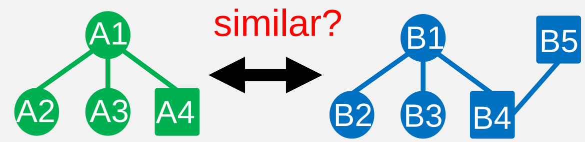

fly


mouse

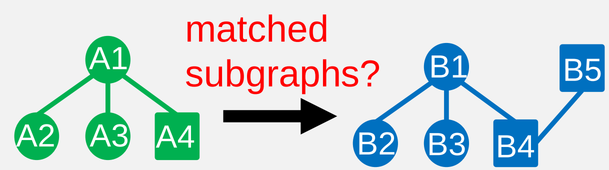
knowledge graphs

Tasks

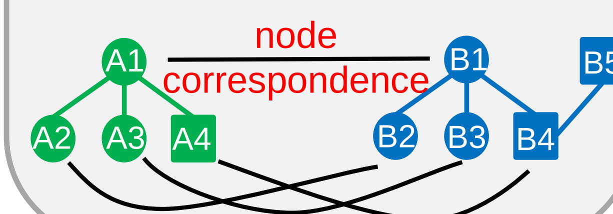
graph level



subgraph level

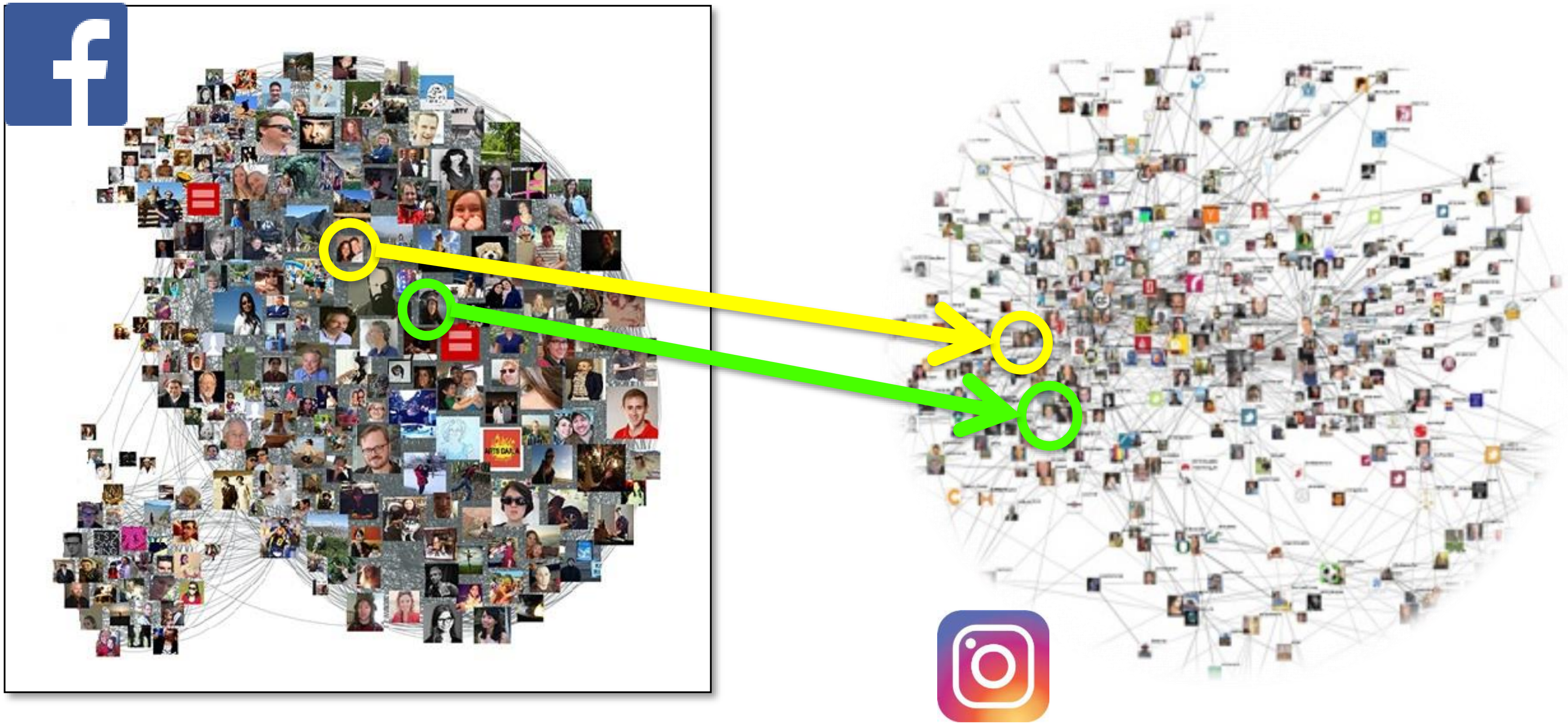


node level



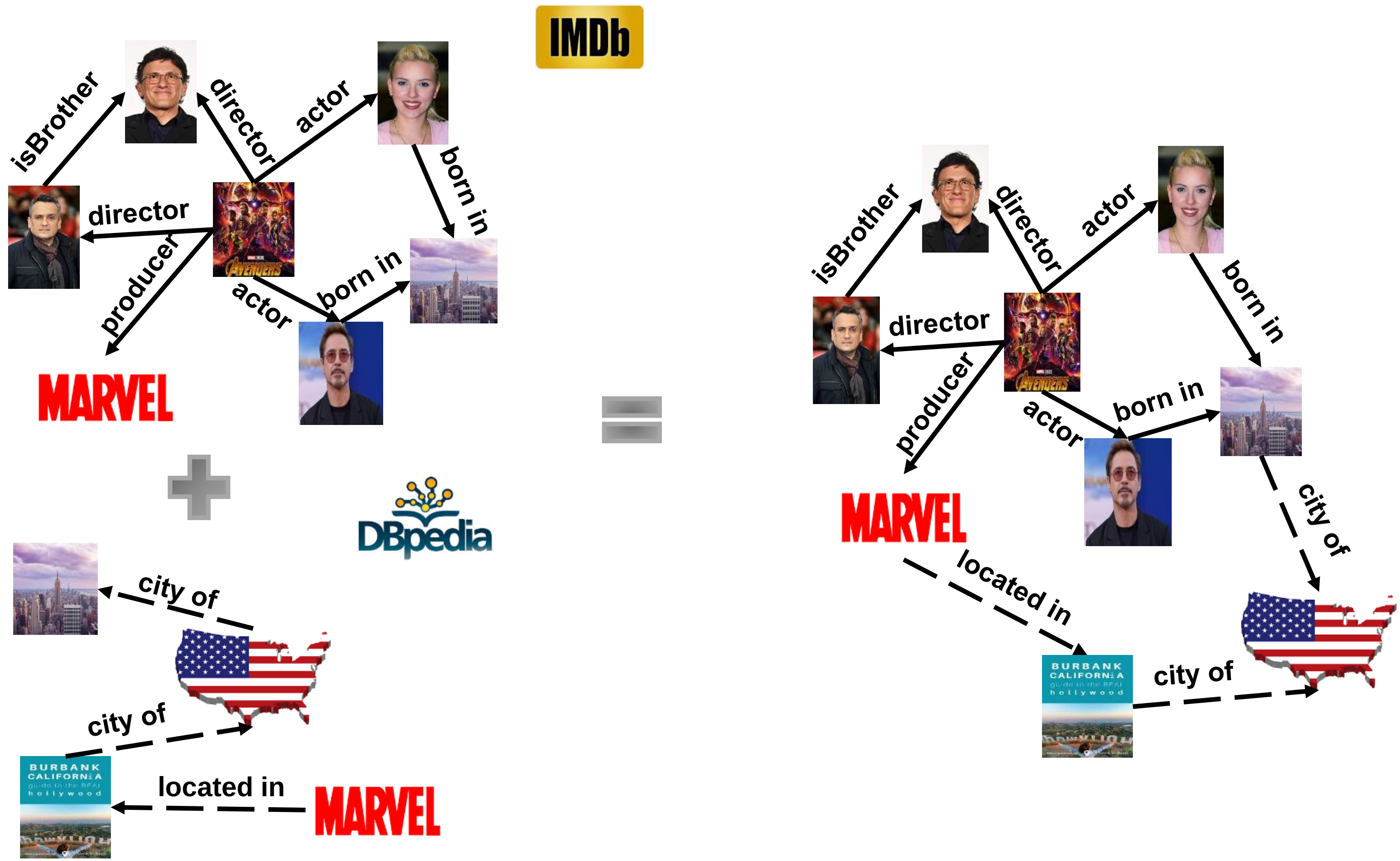
What Is Network Alignment?

- Find node correspondence across networks



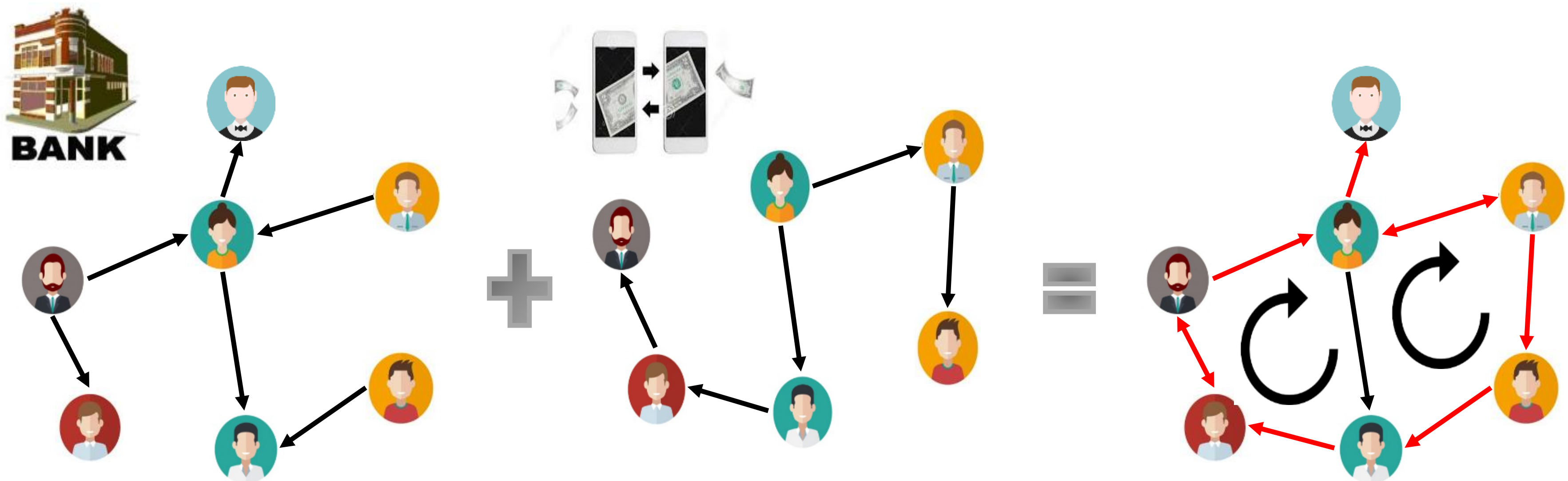
Other Applications

- Knowledge completion



Other Applications

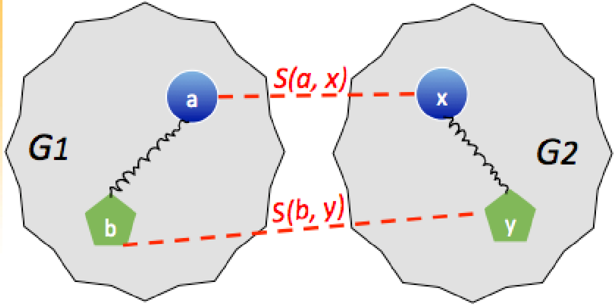
- Fraud detection



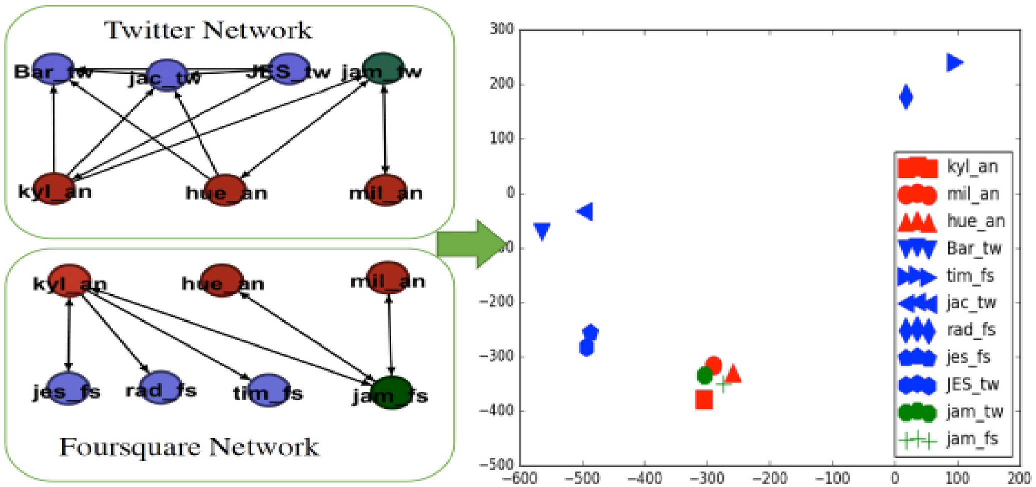
looks normal **looks normal** **money laundering?**

  – Unsuspicious patterns become suspicious!
  – **Question:** How to identify the correspondences across networks?

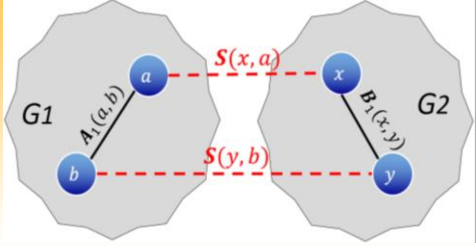
Network Alignment: How to



- Topological alignment
  - If two nodes are aligned, their neighbors are likely to be aligned
- Attributed alignment [Zhang'16]
  - Consider both topological and attribute consistency
- Embedding-based alignment [Liu'16]
  - Aligned nodes are closed in the embedding space



Network Alignment: How to (con't)



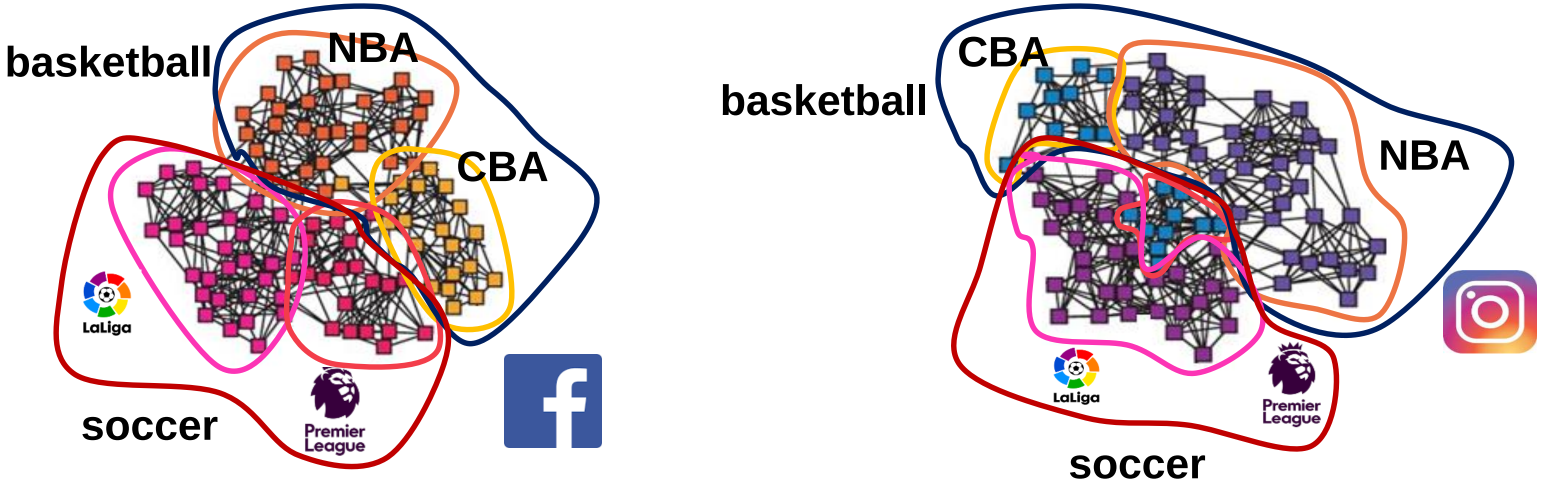
- Topological alignment: FINAL-P [Zhang'16]
  - if two nodes are likely to be aligned (i.e., similar)
  - their close neighbors are likely to be aligned (similar)
- Optimization formulation

$$\min_{\boldsymbol{s}_1} \alpha \boldsymbol{s}_1^T(\boldsymbol{I} - \boldsymbol{A}_1 \otimes \boldsymbol{B}_1)\boldsymbol{s}_1 + (1-\alpha)\|\boldsymbol{s}_1 - \boldsymbol{h}_1\|_2^2$$

  - $\boldsymbol{A}_1, \boldsymbol{B}_1$ are symmetrically normalized adjacency matrices
  - $\boldsymbol{s}_1, \mathbf{h}_1$ are the vectorization of alignment $\boldsymbol{S}_1$ and preference $\boldsymbol{H}_1$
  - convex optimization → global optimal solution
- Optimization algorithm
  - fixed point solution: $\boldsymbol{S}_1 = \alpha \boldsymbol{B}_1 \boldsymbol{S}_1 \boldsymbol{A}_1 + (1-\alpha)\boldsymbol{H}_1$

Network Alignment: Limitations

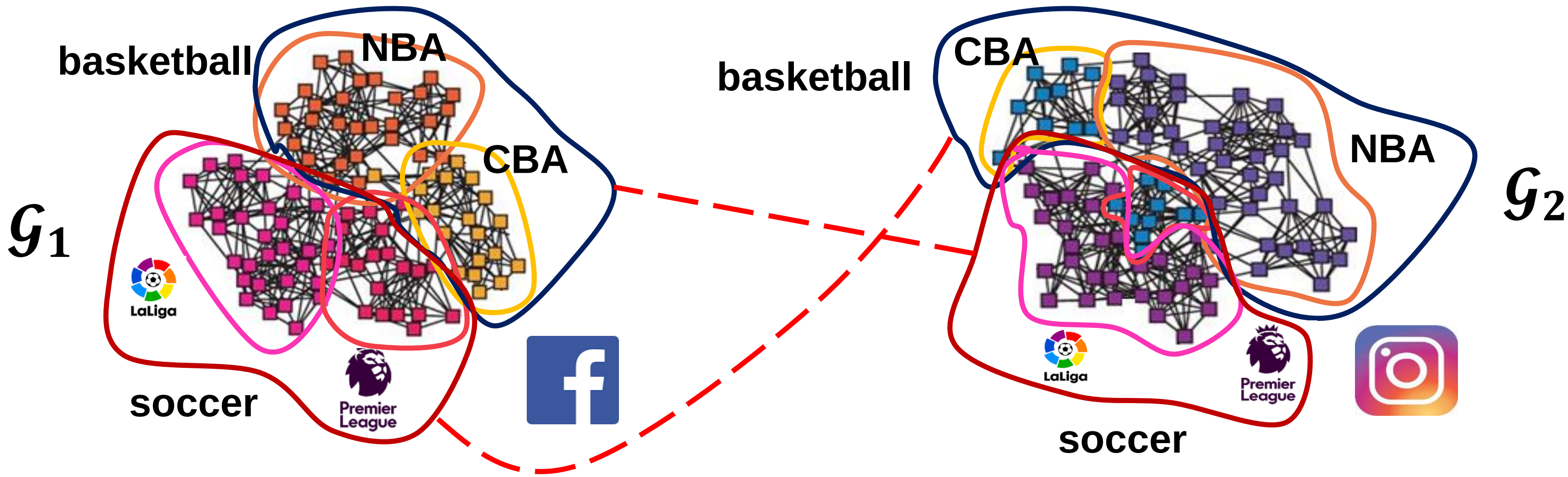
- Existing methods
  - Align networks at node level (and cluster level)
  - Have an at least quadratic computational complexity
- Rich patterns in networks
  - E.g., hierarchical cluster-within-clusters structure



- **Question:** how to align networks at different granularities?

Challenge #1: Alignment Accuracy

- Error propagation through different levels



  – If soccer in $\mathcal{G}_1$ is aligned with basketball in $\mathcal{G}_2$
  – Next cluster level: LaLiga in $\mathcal{G}_1$ cannot be aligned with LaLiga in $\mathcal{G}_2$
  – Node level: nodes in cluster LaLiga in $\mathcal{G}_1$ can't be aligned with nodes in cluster LaLiga in $\mathcal{G}_2$
- **Question:** How to mitigate error propagation?

Challenge #2: Scalability

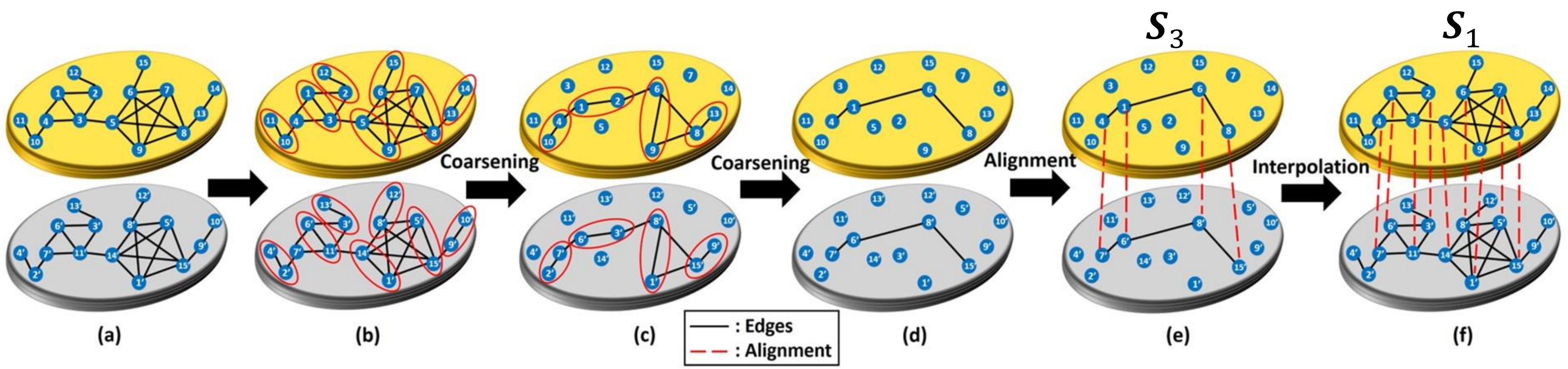
- Time complexity
  - At least $O(n^2)$ due to dense matrix multiplication
- Space complexity
  - At least $O(n^2)$ to store the dense alignment matrix
- **Question:** How can we reduce the complexity?

Outline

- Motivations ✓
- Q1: Moana Formulation
- Q2: Moana Algorithm
- Experimental Results
- Conclusions

Prob. Def: Multilevel Network Alignment

- Given:
  - (1) adjacency matrices $\overline{\boldsymbol{A}}_1, \overline{\boldsymbol{B}}_1$ of two undirected networks;
  - (2) a sparse prior alignment preference $\boldsymbol{H}_1$;
  - (3) the number of levels $L \geq 2$ of interests.
- Find: a set of alignment matrices $\boldsymbol{S}_l$ at level-l, $l = 1, \cdots, L$
  - where $\boldsymbol{S}_1$ indicates the alignment at the node level
- An illustrative example



Moana Formulation #1: Multilevel Optimization

- Generic strategy
  - coarsening → alignment → $\boxed{\text{interpolation}}$
- Alignment interpolations
  - bilinear interpolations by $\boldsymbol{P}_l \in R^{p_l \times n_1}, \boldsymbol{Q}_l \in R^{q_l \times n_2}$ $(p_l \leq n_1, q_l \leq n_2)$
  - w.l.o.g., $\boldsymbol{S}_1 = \boldsymbol{Q}_1^T \boldsymbol{S}_2 \boldsymbol{P}_1$ between level-1 & level-2
- Multilevel alignment formulation

Level-1: $\min_{\boldsymbol{s}_1} \alpha \boldsymbol{s}_1^T (\boldsymbol{I} - \boldsymbol{A}_1 \otimes \boldsymbol{B}_1) \boldsymbol{s}_1 + (1-\alpha) \|\boldsymbol{s}_1 - \boldsymbol{h}_1\|_2^2$

FINAL-P
at node level

⬇ If $\boldsymbol{P}_1 \boldsymbol{P}_1^T = \boldsymbol{I}$ and $\boldsymbol{Q}_1 \boldsymbol{Q}_1^T = \boldsymbol{I}$

Level-2: $\min_{\boldsymbol{s}_2} \alpha \boldsymbol{s}_2^T (\boldsymbol{I} - \boldsymbol{A}_2 \otimes \boldsymbol{B}_2) \boldsymbol{s}_2 + (1-\alpha) \|\boldsymbol{s}_2 - \boldsymbol{h}_2\|_2^2$

  - where $\boldsymbol{A}_2 = \boldsymbol{P}_1 \boldsymbol{A}_1 \boldsymbol{P}_1^T, \boldsymbol{B}_2 = \boldsymbol{Q}_1 \boldsymbol{B}_1 \boldsymbol{Q}_1^T$ and $\boldsymbol{H}_2 = \boldsymbol{Q}_1 \boldsymbol{H}_1 \boldsymbol{P}_1^T$
  - same properties (e.g., convexity) and algorithm as FINAL-P
  - 'good' (semi-) orthogonal $\boldsymbol{P}_1, \boldsymbol{Q}_1$ can make $\boldsymbol{A}_2, \boldsymbol{B}_2$ well-represented

Moana Formulation #2: Perfect Interpolation

- Alignment error propagation
  - imperfect interpolations bring errors to $\boldsymbol{S}_l$ even from optimal $\boldsymbol{S}_{l+1}^*$
  - mathematically, $\boldsymbol{S}_l^* \neq \boldsymbol{Q}_l^T \boldsymbol{S}_{l+1}^* \boldsymbol{P}_l$ if $\boldsymbol{P}_l, \boldsymbol{Q}_l$ are not well-chosen
  - errors can be propagated or diverged to level-$(l-1)$
- Perfect interpolation
  - if $\boldsymbol{P}_l, \boldsymbol{Q}_l$ $(l = 1, \cdots, L-1)$ are orthogonal
  - then $\boldsymbol{S}_l^* = \boldsymbol{Q}_l^T \boldsymbol{S}_{l+1}^* \boldsymbol{P}_l$ where $\boldsymbol{S}_l^*, \boldsymbol{S}_{l+1}^*$ are optimal solutions at level-l and level-$(l+1)$
  - proof in the paper

Outline

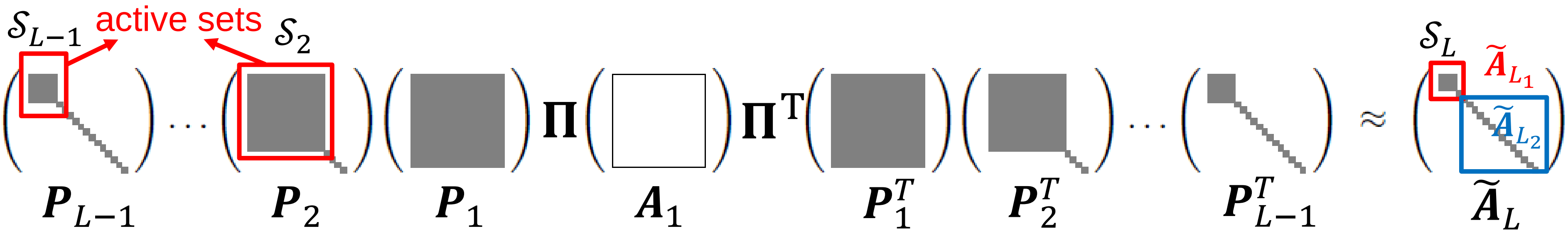
- Motivations
- Q1: Moana Formulation
- Q2: Moana Algorithm
- Experimental Results
- Conclusions

Moana Algorithm #1: Coarsening

- Generic strategy
  - $\boxed{\text{coarsening}}$ → alignment → interpolation
- Network coarsening by $\boldsymbol{P}_l, \boldsymbol{Q}_l$
  - $\boldsymbol{A}_{l+1} = \boldsymbol{P}_l \boldsymbol{A}_l \boldsymbol{P}_l^T, \boldsymbol{B}_{l+1} = \boldsymbol{Q}_l \boldsymbol{B}_l \boldsymbol{Q}_l^T$
- Requirements on $\boldsymbol{P}_l, \boldsymbol{Q}_l$
  - **perfect interpolation:** they are orthogonal matrix
  - **efficient computation:** they are sparse matrix
  - **informative coarsening:** they can uncover hierarchical cluster-within-clusters structures

Moana Algorithm #1: Coarsening (Con't)

- Multiresolution matrix factorization [Kondor'14]



  - $\boldsymbol{\Pi}$ is to reorder for visualization (no need to calculate)
  - $\boldsymbol{P}_l$ contains: (1) a rotation matrix block, (2) an identity matrix block
  - active set $\mathcal{S}_l$ indicates nodes at the l-th granularity (i.e., clusters)
- Coarsening procedure
  - $\boldsymbol{P}_{L-1} \cdots \boldsymbol{P}_2 \boldsymbol{P}_1 \boldsymbol{A}_1 \boldsymbol{P}_1^T \boldsymbol{P}_2^T \cdots \boldsymbol{P}_{L-1}^T = \boldsymbol{A}_L \rightarrow \widetilde{\boldsymbol{A}}_L$
  - $\boldsymbol{Q}_{L-1} \cdots \boldsymbol{Q}_2 \boldsymbol{Q}_1 \boldsymbol{B}_1 \boldsymbol{Q}_1^T \boldsymbol{Q}_2^T \cdots \boldsymbol{Q}_{L-1}^T = \boldsymbol{B}_L \rightarrow \widetilde{\boldsymbol{B}}_L$

- Orthogonality ✓
- Sparsity ✓
- Informativeness ✓

- Remark: $\boldsymbol{S}(\mathcal{S}_{B_l}, \mathcal{S}_{A_l})$ indicates the alignment among clusters at the l-th granularity

Moana Algorithm #2: Alignment

- Generic strategy
  - coarsening → $\boxed{\text{alignment}}$ → interpolation

$$\tilde{\mathbf{A}}_L = \mathbf{\Pi}_A \begin{bmatrix} \tilde{\mathbf{A}}_{L_1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{A}}_{L_2} \end{bmatrix} \mathbf{\Pi}_A^T$$
$$\tilde{\mathbf{B}}_L = \mathbf{\Pi}_B \begin{bmatrix} \tilde{\mathbf{B}}_{L_1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{B}}_{L_2} \end{bmatrix} \mathbf{\Pi}_B^T$$

- Alignment across the coarsest networks

$$\tilde{\boldsymbol{S}}_L = \alpha \begin{bmatrix} \tilde{\boldsymbol{B}}_{L_1} & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{B}}_{L_2} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{S}}_{L_1} & \tilde{\boldsymbol{S}}_{L_2} \\ \tilde{\boldsymbol{S}}_{L_3} & \tilde{\boldsymbol{S}}_{L_4} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{A}}_{L_1} & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{A}}_{L_2} \end{bmatrix} + (1-\alpha) \begin{bmatrix} \tilde{\boldsymbol{H}}_{L_1} & \tilde{\boldsymbol{H}}_{L_2} \\ \tilde{\boldsymbol{H}}_{L_3} & \tilde{\boldsymbol{H}}_{L_4} \end{bmatrix}$$

⬇ block-wise computation

$$\tilde{\boldsymbol{S}}_{L_1} = \alpha \tilde{\boldsymbol{B}}_{L_1} \tilde{\boldsymbol{S}}_{L_1} \tilde{\boldsymbol{A}}_{L_1} + (1-\alpha) \tilde{\boldsymbol{H}}_{L_1}$$
$$\tilde{\boldsymbol{S}}_{L_2} = \alpha \tilde{\boldsymbol{B}}_{L_1} \tilde{\boldsymbol{S}}_{L_2} \tilde{\boldsymbol{A}}_{L_2} + (1-\alpha) \tilde{\boldsymbol{H}}_{L_2}$$
$$\tilde{\boldsymbol{S}}_{L_3} = \alpha \tilde{\boldsymbol{B}}_{L_2} \tilde{\boldsymbol{S}}_{L_3} \tilde{\boldsymbol{A}}_{L_1} + (1-\alpha) \tilde{\boldsymbol{H}}_{L_3}$$
$$\tilde{\boldsymbol{s}}_{L_4} = (1-\alpha)\left(\boldsymbol{I} - \alpha \tilde{\boldsymbol{A}}_{L_2} \otimes \tilde{\boldsymbol{B}}_{L_3}\right)^{-1} \tilde{\boldsymbol{h}}_{L_4}$$

  - matrix composition: e.g., $\boldsymbol{S}_L(\mathcal{S}_{B_L}, \mathcal{S}_{A_L}) = \tilde{\boldsymbol{S}}_{L_1}$, $\boldsymbol{S}_L(\bar{\mathcal{S}}_{B_L}, \bar{\mathcal{S}}_{A_L}) = \tilde{\boldsymbol{S}}_{L_4}$

- Alignment at finer levels
  - perfect interpolations: $\boldsymbol{S}_l = \boldsymbol{Q}_l^T \boldsymbol{S}_{l+1} \boldsymbol{P}_l$

Moana Algorithm: Analysis

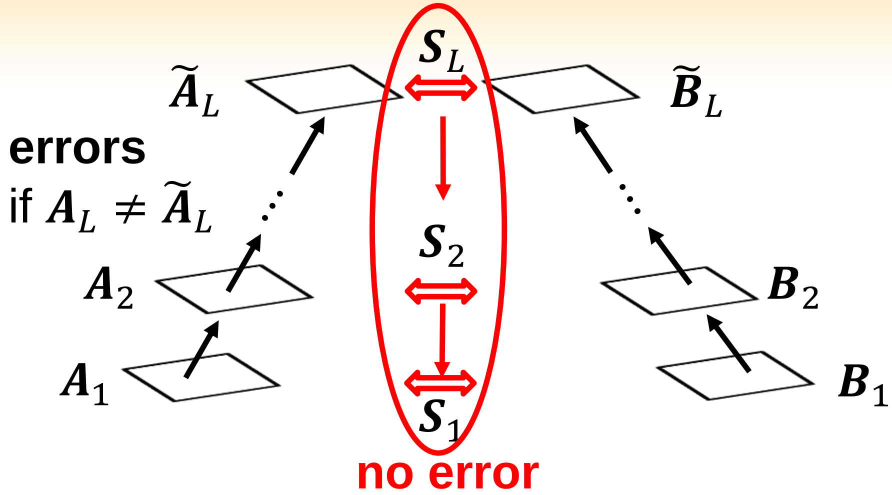
- Alignment error bound

$$\frac{\|\boldsymbol{S}_l^* - \boldsymbol{S}_l\|_F}{\|\boldsymbol{S}_l\|_F} \leq \frac{2\epsilon\kappa}{1-\epsilon\kappa}, \forall l = 1, \cdots, L$$

where $\epsilon = \sqrt{\frac{\alpha}{2n}}(\delta_1 r_2 + \delta_2 r_1 + \delta_1\delta_2)$,

$\delta_1 = \left\|\boldsymbol{A}_L - \widetilde{\boldsymbol{A}}_L\right\|_F, \delta_2 = \left\|\boldsymbol{B}_L - \widetilde{\boldsymbol{B}}_L\right\|_F$,

κ is condition number, r_1, r_2 are ranks



- Complexity analysis
  - linear time and space complexity

Outline

- Motivations ✓
- Q1: Moana Formulation ✓
- Q2: Moana Algorithm ✓
- Experimental Results
- Conclusions

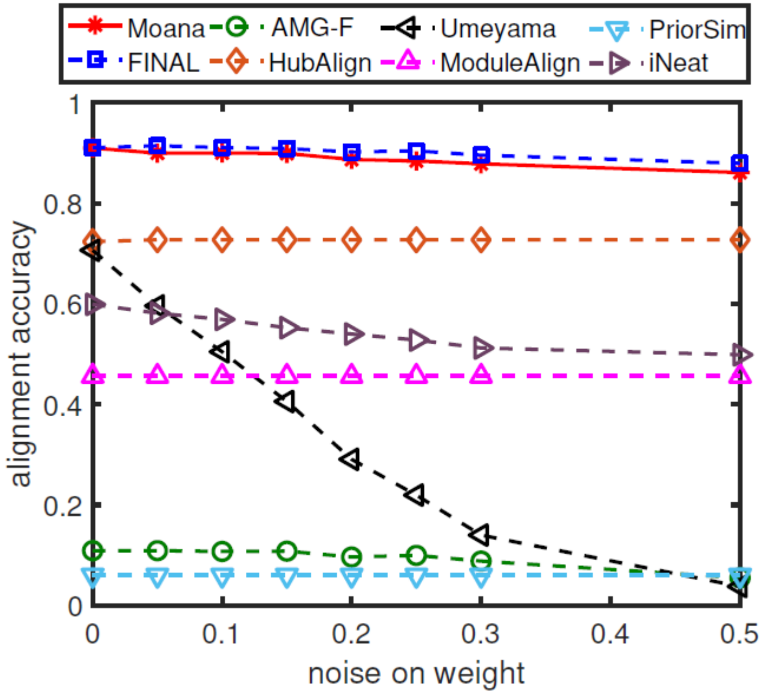
Experimental Setup

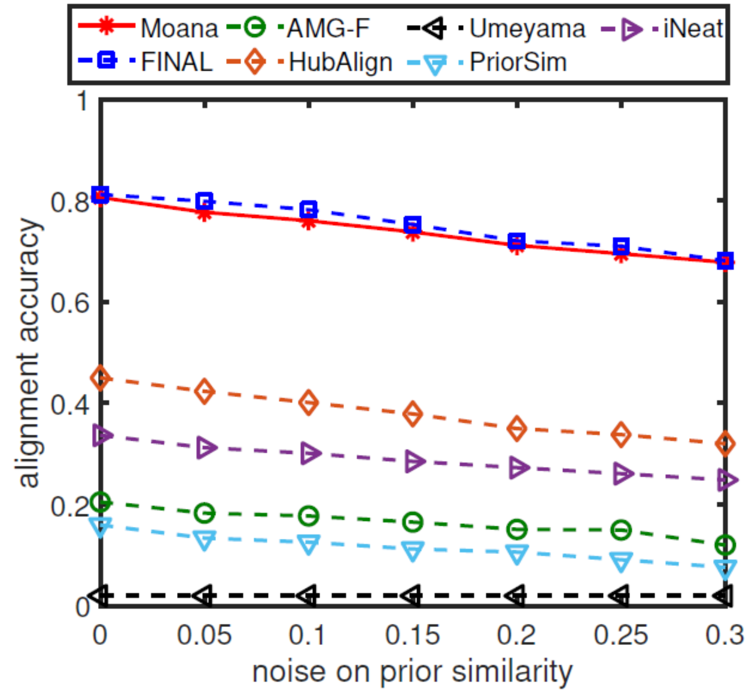
- Datasets
  - Gr-Qc network vs. its permutation (nodes: 5,241 vs. 5,241)
  - Google+ network vs. its permutation (nodes: 23,628 vs. 23,628)
  - Amazon co-purchasing networks (nodes: 74,596 vs. 66,951)
  - ACM vs DBLP coauthor networks (nodes: 9,872 vs. 9,916)
- Evaluation objectives
  - Effectiveness: how accurate is our algorithm in aligning networks?
  - Efficiency: how fast and scalable is our algorithm?
- Comparison methods

Moana	AMG-FINAL	Umeyama	PriorSim
FINAL-P	HubAlign	ModuleAlign	iNeat

R1: Effectiveness Results

- Effectiveness in node-level alignment

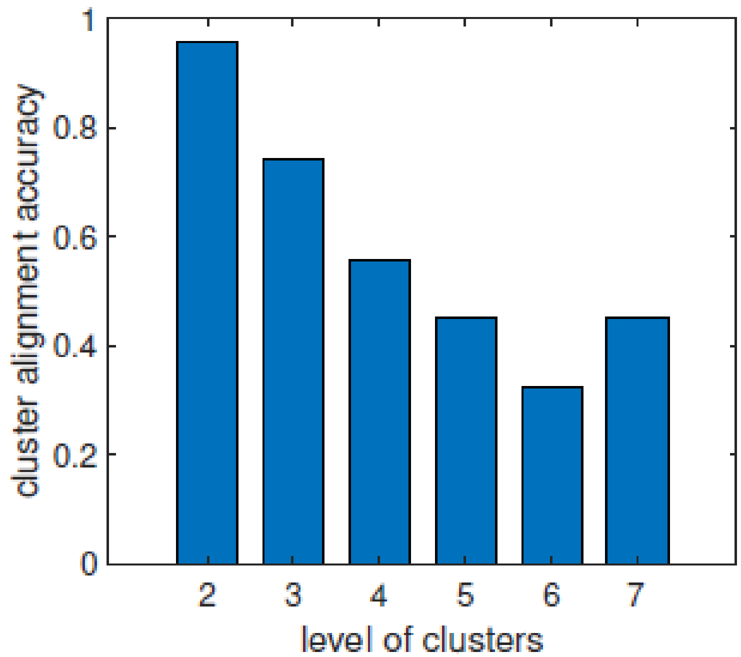




Observations:
(1) the performance of Moana is close to FINAL-P;
(2) Moana outperforms all other methods.

R2: Effectiveness Results

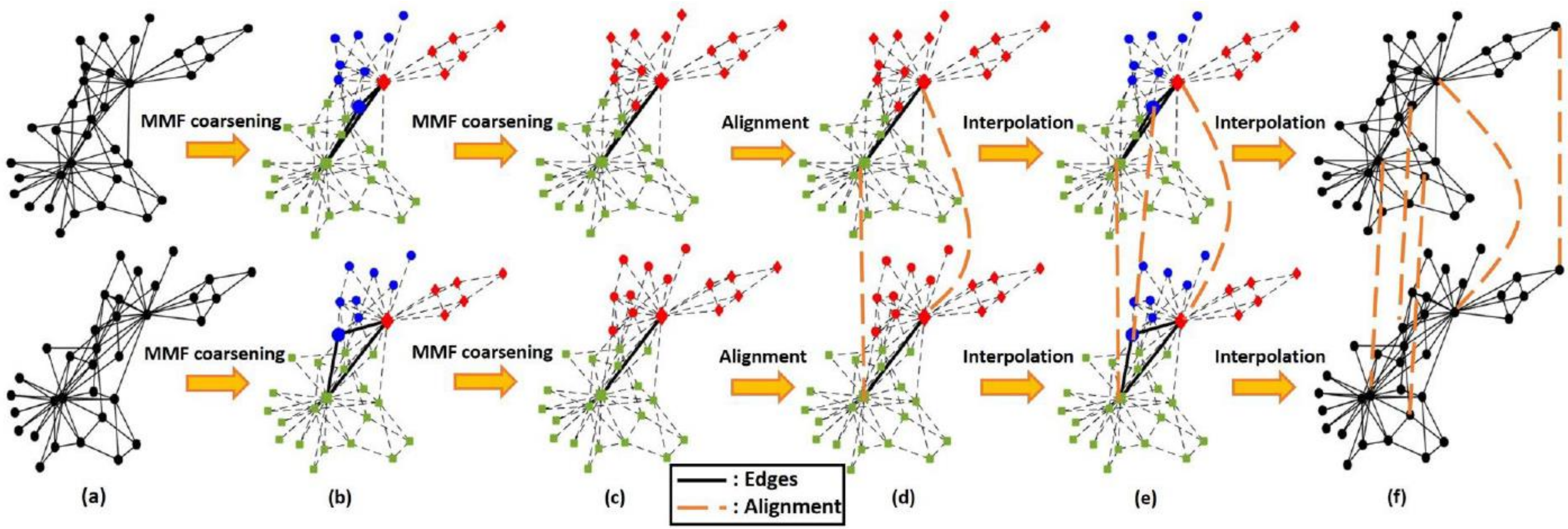
- Effectiveness in cluster-level alignment



Observations: Moana achieves a good performance in cluster alignment at different levels.

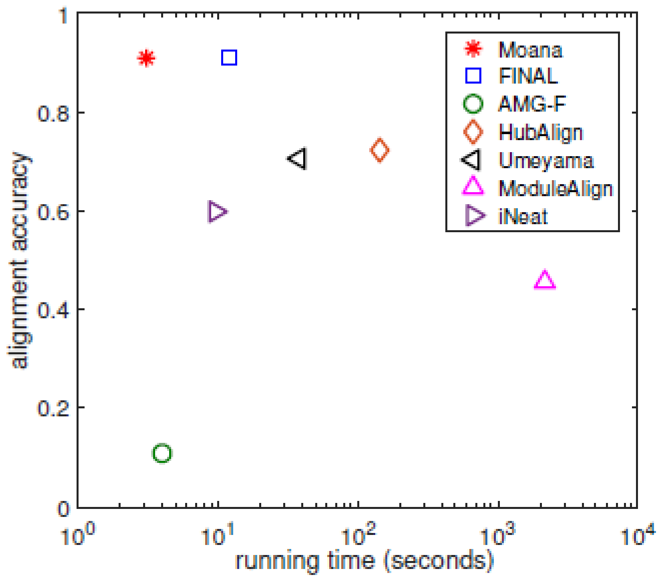
R3: Case Study on Multilevel Alignment

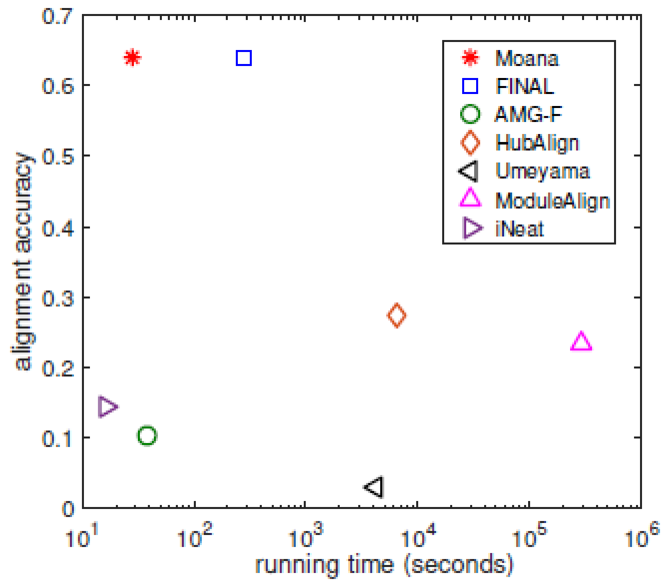
- A case study on Zachary's Karate networks



Observations: Moana can unveil meaningful alignment of clusters at different granularities.

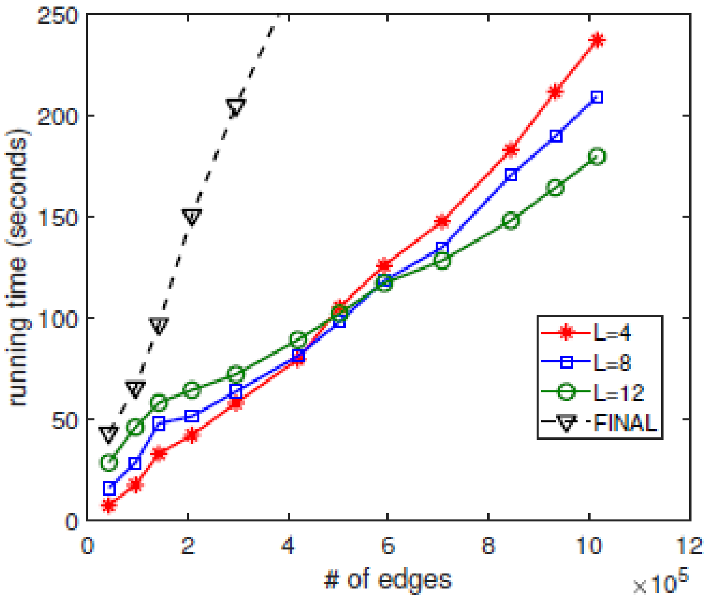
R4: Quality-Speed Balance

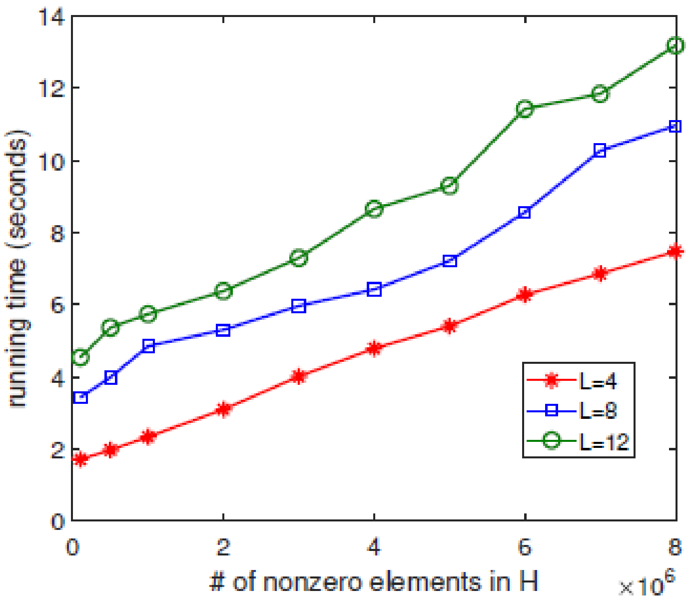




Observations: Moana can achieve a better quality-speed balance.

R5: Scalability





Observations:

(1) Moana scales linearly w.r.t. the number of edges;

(2) Moana scales linearly w.r.t. the number of nonzero elements in $\boldsymbol{H}_1$.

Outline

- Motivations ✓
- Q1: Moana Formulation ✓
- Q2: Moana Algorithm ✓
- Experimental Results ✓
- Conclusions

Conclusions

- Multilevel network alignment
  - Q1: Formulation
  - A1: Multilevel optimization + perfect interpolation
  - Q2: Scalability
  - A2: Moana algorithm
- Results
  - Moana outperforms most baseline methods in node alignment
  - Moana achieves good performance in cluster alignment
  - Moana has linear complexity
- More in paper
  - Proof of algorithm analysis & more experimental results