FINAL: Fast Attributed Network Alignment

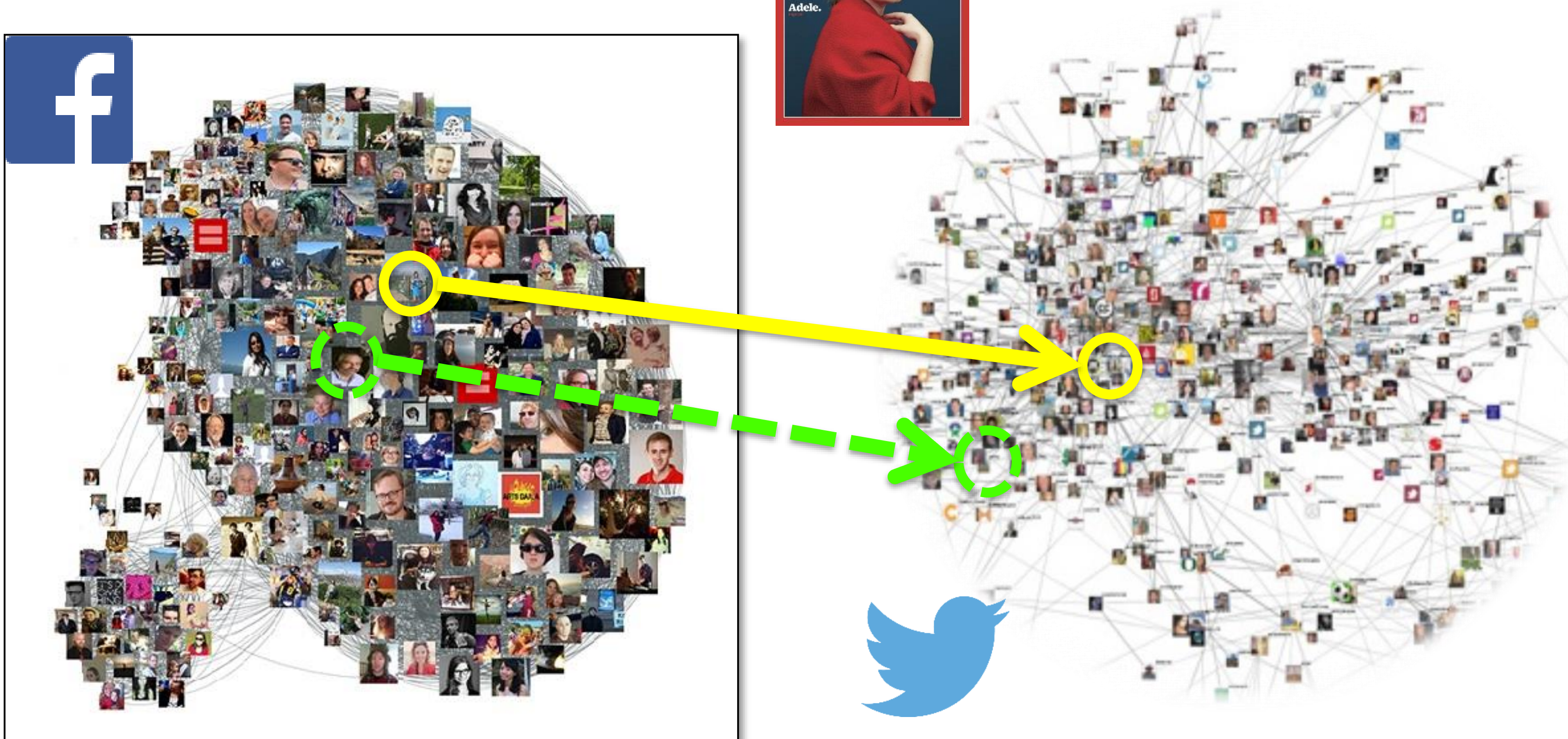
Presented by Si Zhang (ASU)

Si Zhang

Hanghang Tong

Why Network Alignment?

- Find Someone Like You

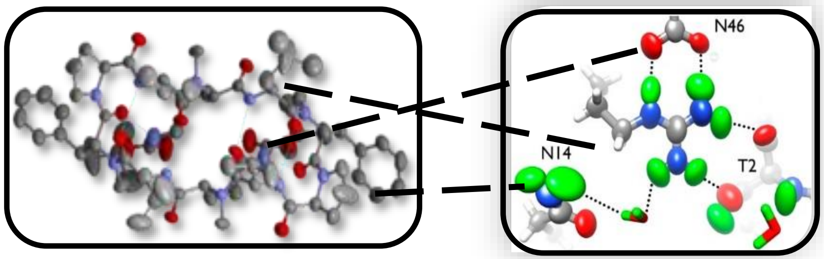


- **Q**: what if someone lives in a different universe (network)?

More Applications

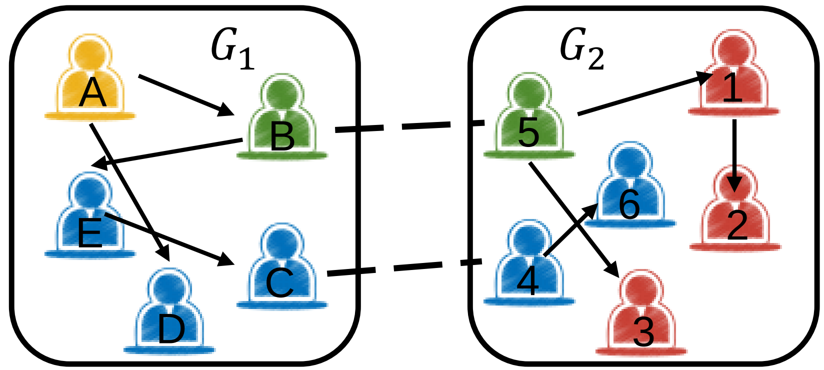
Identify Species-Specific Pathways [Singh'08]

Protein-Protein Interaction (PPI) networks



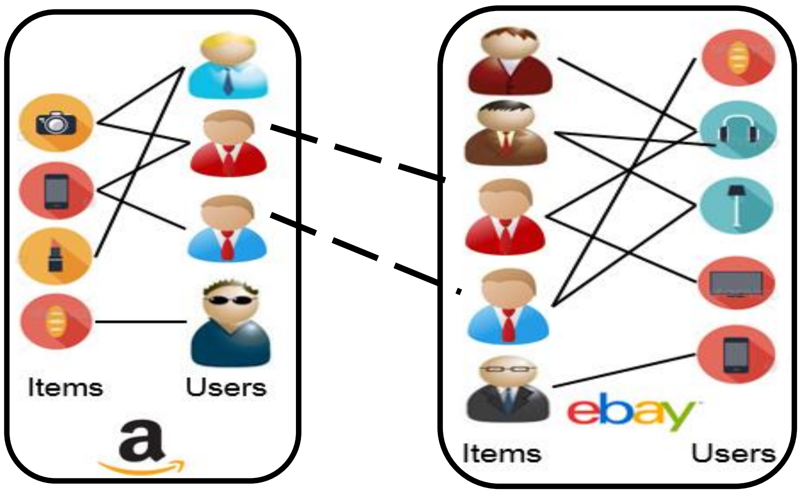
PPI network 1 PPI network 2

Cross Network Information Diffusion [Zhan'16]

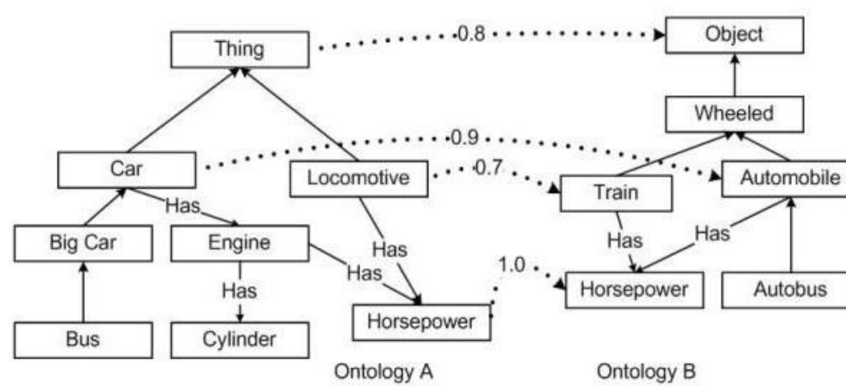


social network 1 social network 2

Cross-Site Recommendation [Zhang'14]

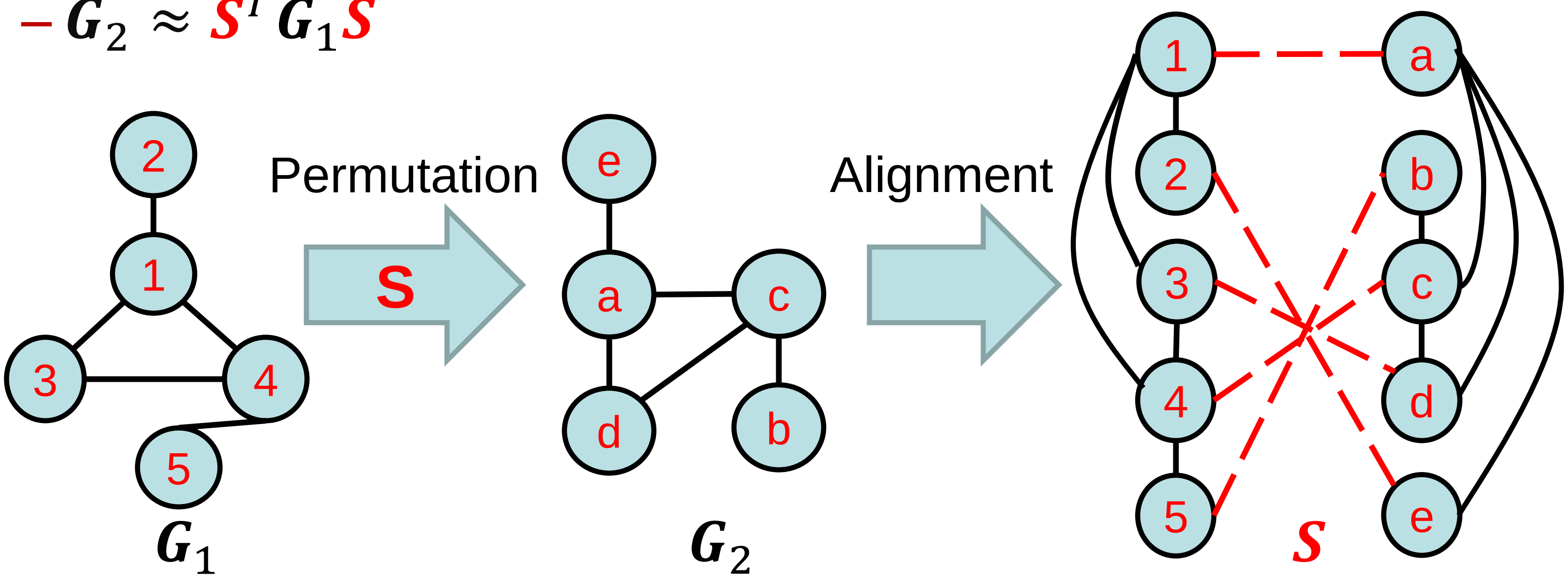


Ontology Matching on Semantic Web [Doan'02]



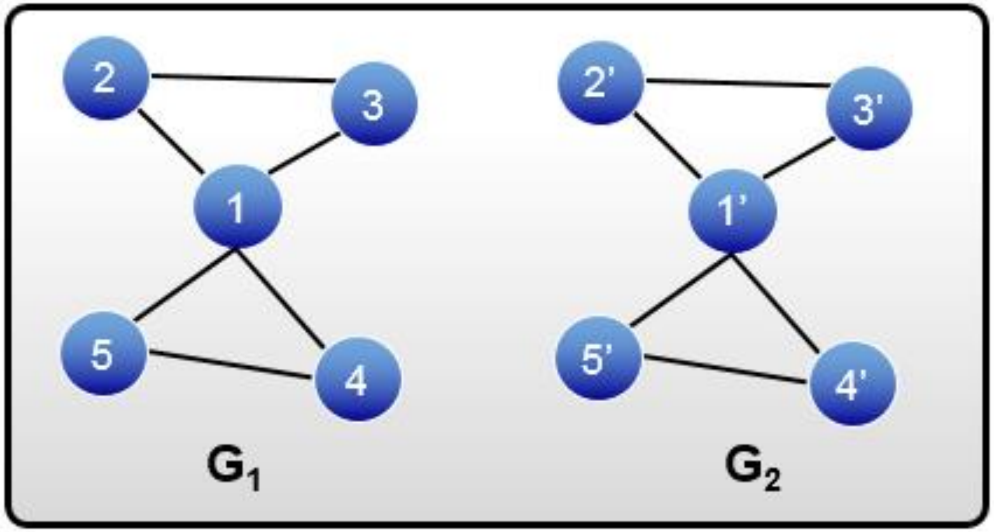
Why Network Alignment: How to

- Existing Methods
  - IsoRank [Singh'08], NetAlign [Bayati'09], BigAlign [Koutra'13], UMA [Zhang'15]
- Key Idea: topological consistency
  - Network $\boldsymbol{G}_2$ is a noisy permutation of network $\boldsymbol{G}_1$
  - $\boldsymbol{G}_2 \approx \boldsymbol{S}^T \boldsymbol{G}_1 \boldsymbol{S}$



Topology Consistency: Limitations

- Topological consistency could be easily violated
  - Same nodes may behave differently across different networks
  - Different nodes may have similar connectivity structures



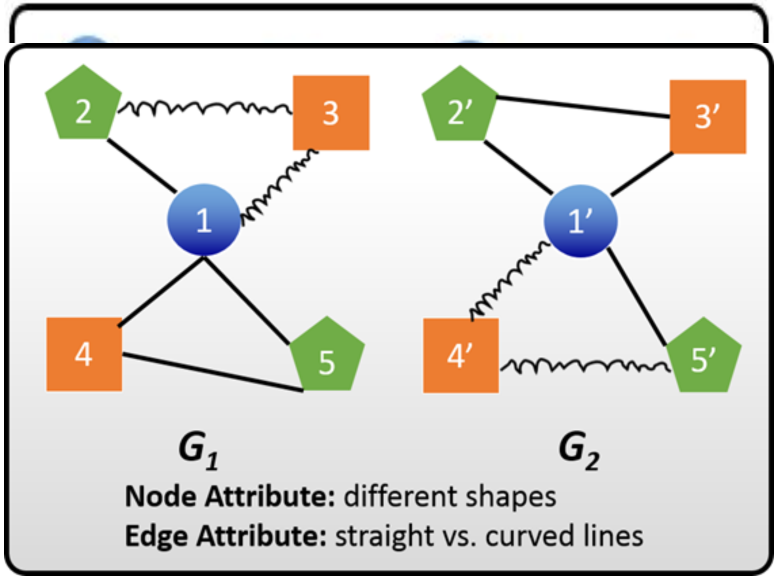
Nodes 2=3=4=5

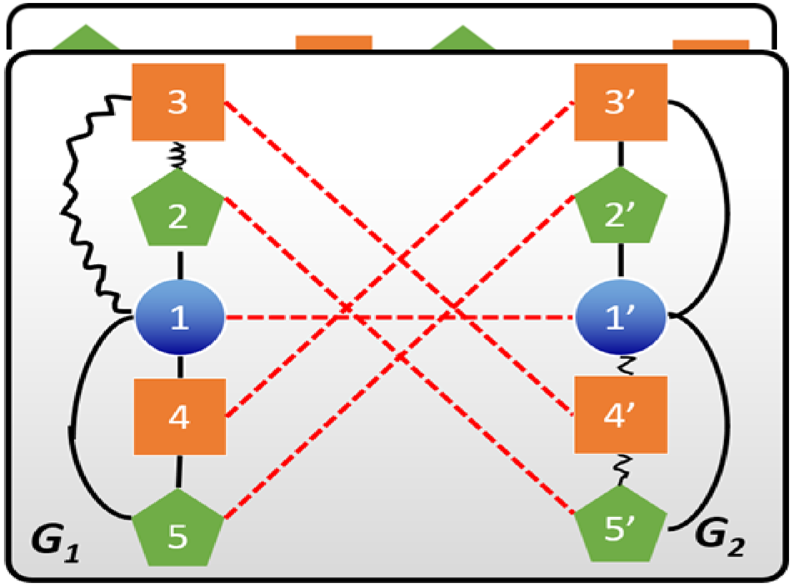
Nodes 2’=3’=4’=5’

Only topology is not enough!

Topology Consistency: How to Rescue

- Real networks have rich attributes on nodes and/or edges





- **Q**: how to calibrate topology-based alignment by leveraging attributes?

Challenges: Attributed Network Alignment

- C1: Formulation
- C2: Optimality
- C3: Scalable Computation

C1. Formulation

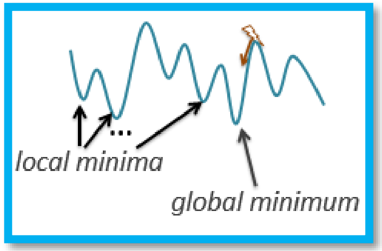
- Typical Network Alignment

NetAlign [Bayati'09]	UMA [Zhang'15]
$\underset{x}{\text{maximize}} \quad \alpha \mathrm{h}^T \mathrm{s} + \left(\frac{\beta}{2}\right) s^T W s$ $\textit{subject to} \quad As \leq 1, s_{ii'} \in \{0,1\}$	$\underset{S}{\text{minimize}} \quad \|S^T A S - B\|_F^2$ $s.t \quad S 1^{n_2 \times 1} \leq 1^{\mathrm{n}_1 \times 1}$ $S^T 1^{\mathrm{n}_1 \times 1} \leq 1^{n_2 \times 1}$

- **Obs**: only encode topological information
- **Q**: what are their attributed counterparts?

C2. Optimality

- **Obs #1**: many topology-based approaches are non-convex or even NP-hard ⟶ find the local minima



- **Obs #2**: attribute may complicate the optimization problem

$$\min_{\mathbf{S}} \; J(\mathbf{S}) = \sum_{a,b,x,y} \left[\frac{\mathbf{S}(x,a)}{\sqrt{f(x,a)}} - \frac{\mathbf{S}(y,b)}{\sqrt{f(y,b)}} \right]^2 \times \mathbf{A}_1(a,b)\mathbf{A}_2(x,y)$$

without attributes

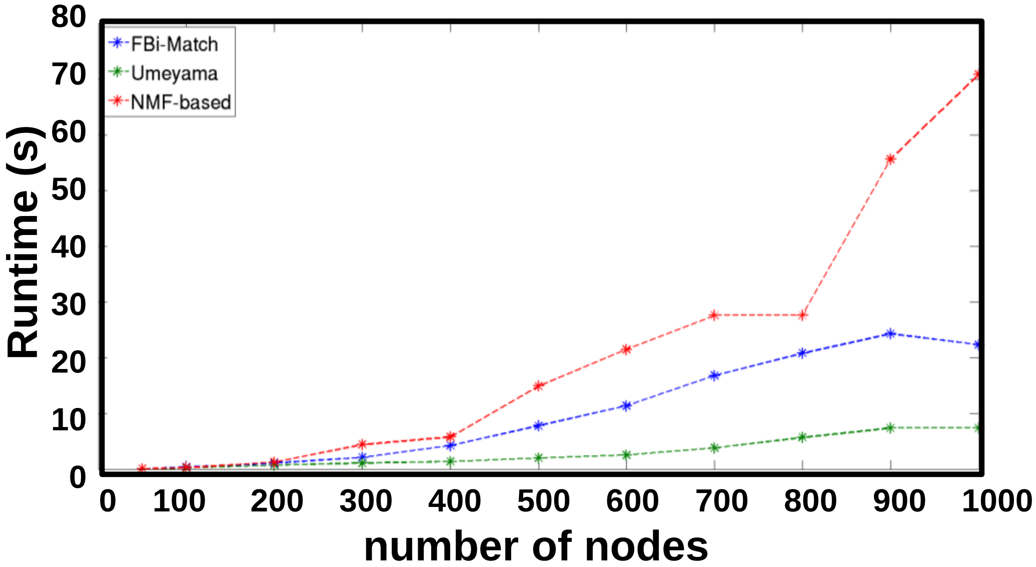
$$\min_{\mathbf{S}} \; J(\mathbf{S}) = \sum_{a,b,x,y} \left[\frac{\mathbf{S}(x,a)}{\sqrt{f(x,a)}} - \frac{\mathbf{S}(y,b)}{\sqrt{f(y,b)}} \right]^2 \mathbf{A}_1(a,b)\mathbf{A}_2(x,y) \times \mathbb{I}\big(\mathbf{N}_1(a,a) = \mathbf{N}_2(x,x)\big)\mathbb{I}\big(\mathbf{N}_1(b,b) = \mathbf{N}_2(y,y)\big) \times \mathbb{I}\big(\mathbf{E}_1(a,b) = \mathbf{E}_2(x,y)\big)$$

with attributes

- **Q #1**: what is the exact optimality of the attributed network alignment?
- **Q #2**: how to get the optimal solution, with a comparable complexity (as the topology-alone alignment)?

C3. Scalable Computation

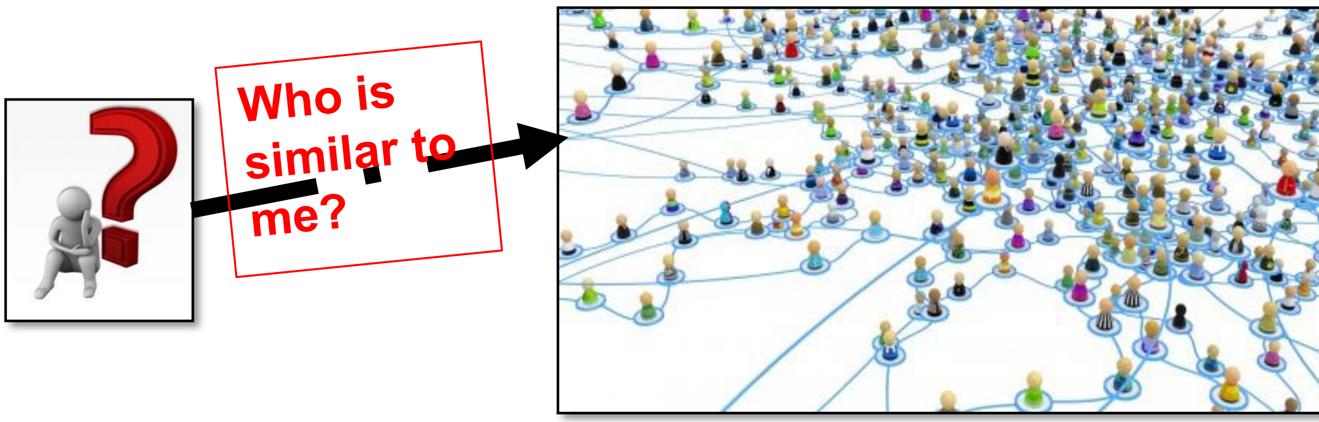
- **Obs #1**: most existing methods have an $O(mn)$ complexity [Singh'08].
- **Obs #2**: best empirical scalability is near-linear [Koutra'13].



- **Q**: how to scale up attributed network alignment?

C3. Scalable Computation

- **Obs**: cross-network search – to find similar users in one network for a given user in another network.



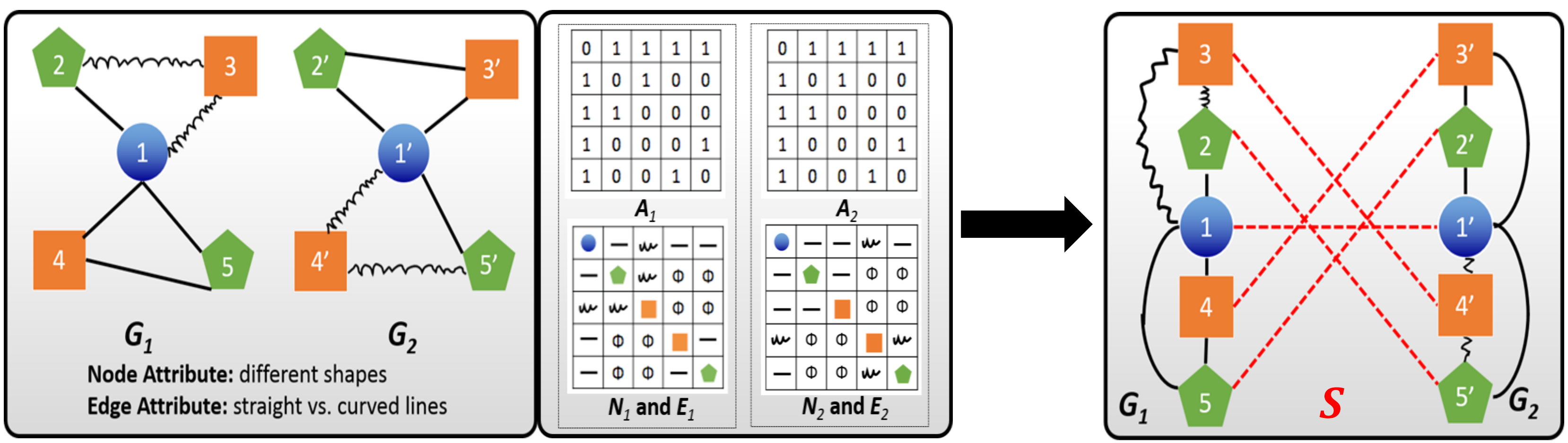
- **Q**: how to speed up on-query network alignment, without solving the full alignment problem?

Outline

- Motivations ✓
- Q1: FINAL Formulation
- Q2: FINAL Algorithms
- Q3: FINAL Speed-up Computation
- Experimental Results
- Conclusions

Prob. Def: Attributed Network Alignment

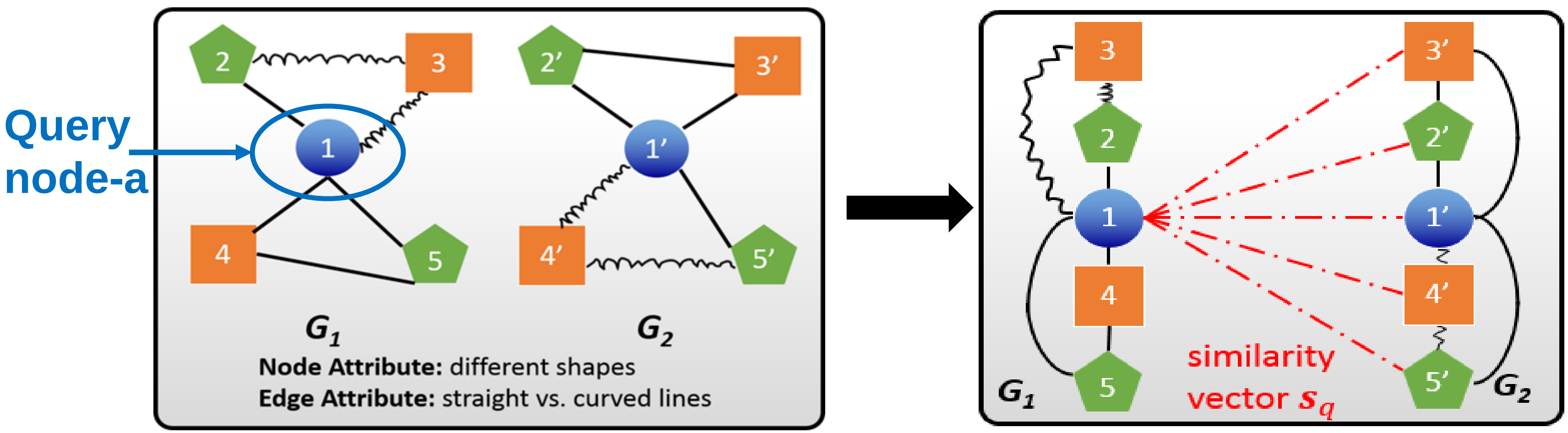
- **Given**:
  - (1) two attributed networks $G_1 = \{\boldsymbol{A}_1, \boldsymbol{N}_1, \boldsymbol{E}_1\}$ and $G_2 = \{\boldsymbol{A}_2, \boldsymbol{N}_2, \boldsymbol{E}_2\}$;
  - (2 – optional) a prior alignment preference $\boldsymbol{H}$.
- **Find**: alignment/similarity matrix $\boldsymbol{S}$
- An Illustrative Example



Given　　　　　　　　　　Find

Prob. Def: On-query Attributed Network Alignment

- **Given**:
  - (1) two attributed networks $\mathrm{G}_1 = \{\boldsymbol{A}_1, \boldsymbol{N}_1, \boldsymbol{E}_1\}$ and $G_2 = \{\boldsymbol{A}_2, \boldsymbol{N}_2, \boldsymbol{E}_2\}$;
  - (2 – optional) a prior alignment preference $\boldsymbol{H}$;
  - (3) a query node-a in G_1
- **Find**: a vector $\boldsymbol{s}_q$ (similarities of node-a vs. all nodes in G_2)

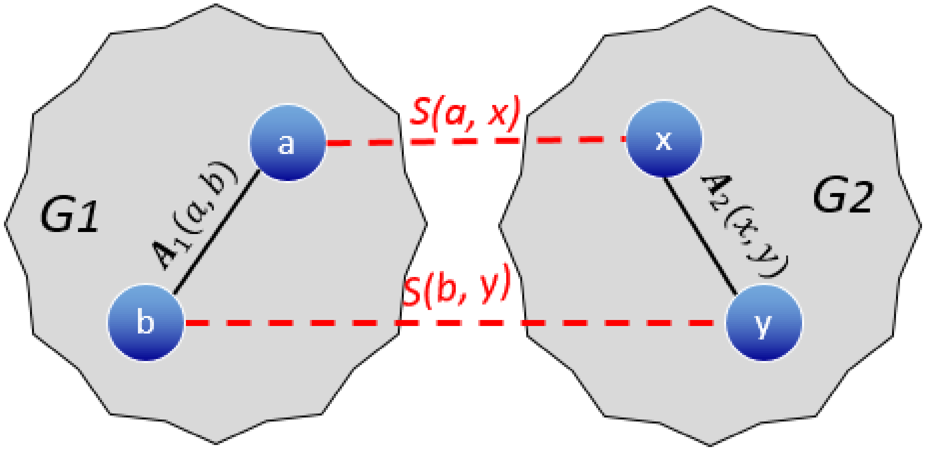


Given

Find

FINAL Formulation #1: Topological Consistency

- **Intuition**: similar node-pairs tend to have similar neighboring node-pairs



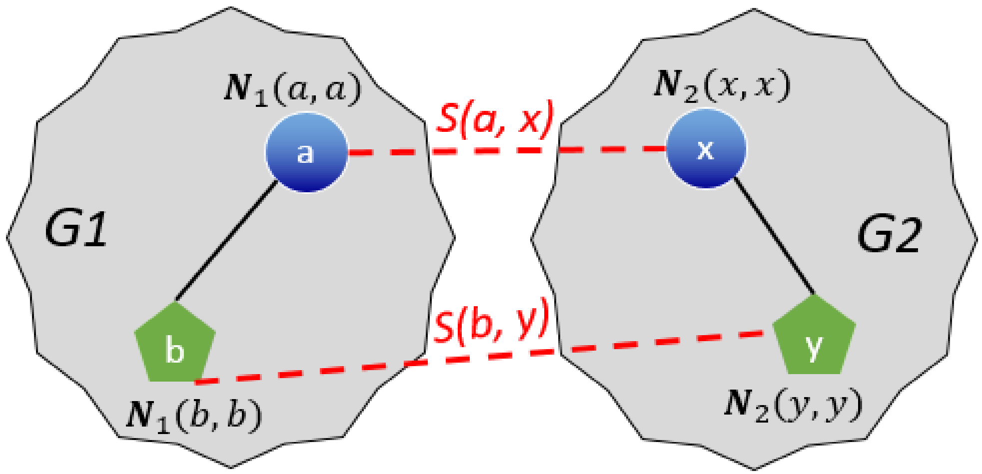
- **Example**:
  – large $\boldsymbol{S}(a,x)$
  – large $\boldsymbol{A}_1(a,b)$ and $\boldsymbol{A}_2(x,y)$

  $\longrightarrow$ large $\boldsymbol{S}(b,y)$

- **Mathematical Details**: $\min_{\boldsymbol{S}} [\boldsymbol{S}(a,x) - \boldsymbol{S}(b,y)]^2 \boldsymbol{A}_1(a,x)\boldsymbol{A}_2(b,y)$

FINAL Formulation #2: Node Attribute Consistency

- **Intuition**: similar node-pairs share same node attributes

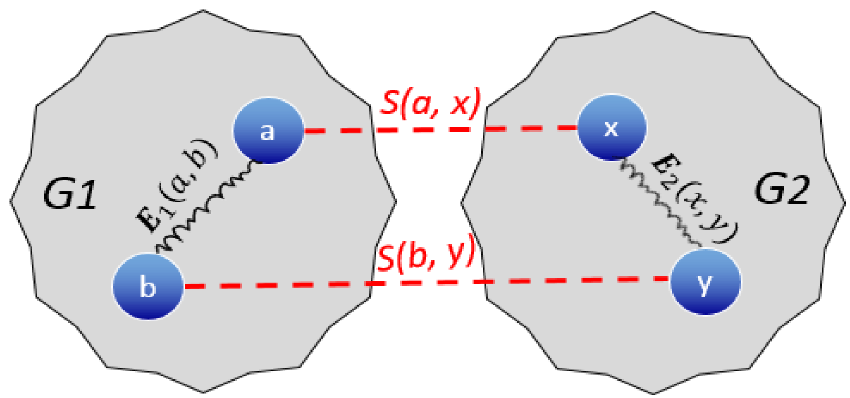


- **Example**:
  - large $\boldsymbol{S}(a,x) \longrightarrow$ node-a and node-x share same node attribute
- **Mathematical Details**: if $\boldsymbol{N}_1(a,a) = \boldsymbol{N}_2(x,x)$ and $\boldsymbol{N}_1(\mathrm{b},\mathrm{b}) = \boldsymbol{N}_2(\mathrm{y},\mathrm{y})$,

$$\min_{\boldsymbol{S}} [\boldsymbol{S}(a,x) - \boldsymbol{S}(b,y)]^2 \boldsymbol{A}_1(a,x)\boldsymbol{A}_2(b,y)$$

FINAL Formulation #3: Edge Attribute Consistency

- **Intuition**: similar node-pairs connect to their neighbor-pairs via same edge attributes



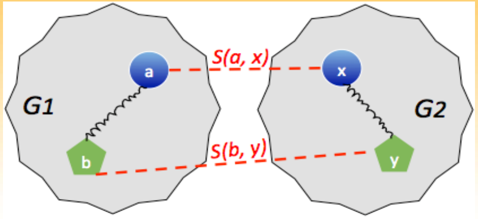
- **Example**:
  - large $\boldsymbol{S}(a,x)$
  - large $\boldsymbol{S}(b,y)$

  $\longrightarrow$ edge (a,b) and (x,y) share same attribute

- **Mathematical Details**: if $\boldsymbol{E}_1(a,b) = \boldsymbol{E}_2(x,y)$,

$$\min_{\boldsymbol{S}} [\boldsymbol{S}(a,x) - \boldsymbol{S}(b,y)]^2 \boldsymbol{A}_1(a,x)\boldsymbol{A}_2(b,y)$$

Putting everything together



- **Objective Function**:

$$\min_{\boldsymbol{S}} \ J(\boldsymbol{S}) = \sum_{a,b,x,y} \left[\frac{\boldsymbol{S}(x,a)}{\sqrt{f(x,a)}} - \frac{\boldsymbol{S}(y,b)}{\sqrt{f(y,b)}} \right]^2 \underline{\boldsymbol{A}_1(a,b)\boldsymbol{A}_2(x,y)}$$

$$\times \mathbb{I}\big(\boldsymbol{N}_1(a,a) = \boldsymbol{N}_2(x,x)\big)\mathbb{I}\big(\boldsymbol{N}_1(b,b) = \boldsymbol{N}_2(y,y)\big)$$

$$\times \mathbb{I}\big(\boldsymbol{E}_1(a,b) = \boldsymbol{E}_2(x,y)\big)$$

#1. Topology Consistency

#2. Node Attribute Consistency

#3. Edge Attribute Consistency

- $\mathbf{f}(\mathbf{x}, \boldsymbol{a})$:
  - ‘joint degree’ of node-a and node-x
  - normalization to make the optimization problem convex
- **Generalization**:
  - replacing $\mathbb{I}(\cdot)$ by an attribute similarity function
  - can handle numerical attributes on nodes and/or edges.

FINAL Formulation: Matrix Form

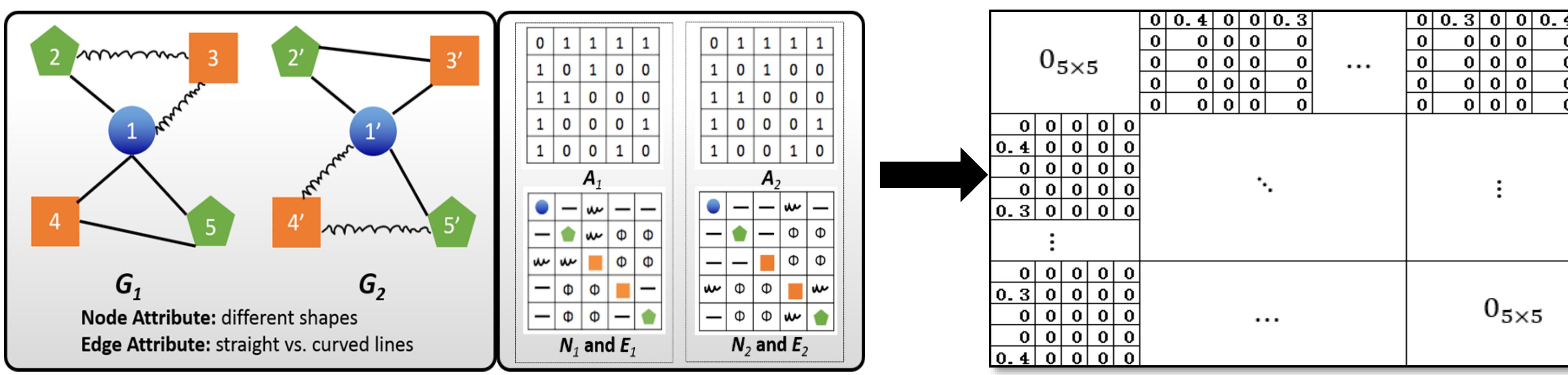
- Matrix-Form Objective Function

$$\min_{\boldsymbol{S}} J(\boldsymbol{S}) = \min_{\boldsymbol{s}} \sum_{v,w}\left[\frac{\boldsymbol{s}(v)}{\sqrt{\boldsymbol{D}(v,v)}} - \frac{\boldsymbol{s}(w)}{\sqrt{\boldsymbol{D}(w,w)}}\right]^2 \boldsymbol{W}(v,w)$$

$$\boxed{\boldsymbol{s} = \mathrm{vec}(\boldsymbol{S})} = \min_{\boldsymbol{s}} \boldsymbol{s}^T\left(\boldsymbol{I} - \widetilde{\boldsymbol{W}}\right)\boldsymbol{s}$$

  - $\boldsymbol{W} = \boldsymbol{N}[\boldsymbol{E} \odot (\boldsymbol{A}_1 \otimes \boldsymbol{A}_2)]\boldsymbol{N}$, i.e., the attributed Kronecker product
  - $\boldsymbol{D}$ is the degree matrix of $\boldsymbol{W}$

$\widetilde{\boldsymbol{W}} = \boldsymbol{D}^{-\frac{1}{2}}\mathbf{W}\boldsymbol{D}^{-\frac{1}{2}}$ is the symmetrically normalization of $\boldsymbol{W}$



FINAL Formulation: Matrix Form with Regularization

- Add a regularization term

$$\boxed{\boldsymbol{h} = \mathrm{vec}(\mathbf{H})}$$

$$\min_{\boldsymbol{s}} \ \alpha \boldsymbol{s}^T \left(\boldsymbol{I} - \widetilde{\boldsymbol{W}}\right)\boldsymbol{s} + (1-\alpha)\|\boldsymbol{s} - \boxed{\boldsymbol{h}}\|_2^2$$

  - $\boldsymbol{h}$ is default as a uniform vector
  - $\boldsymbol{h}$ encodes the prior knowledge of alignment preferences
  - $\boldsymbol{h}$ avoids trivial solution, e.g., optimal solution $\boldsymbol{s} = \mathbf{0}$ w/o $\boldsymbol{h}$
  - teleport vector in PageRank, restart vector in RWR (on the attributed Kronecker product graph)

Relationship with Existing Methods

- FINAL vs. IsoRank [Singh'08]
  - w/o attribute, FINAL = IsoRank (by a scaling factor $\boldsymbol{D}^{\frac{1}{2}}$)
- FINAL vs. Random Walk Graph Kernel (RWGK) [Vishwanathan'10]
  - $k(G_1, G_2) = \sum_i \boldsymbol{q}(i)\boldsymbol{s}(i)$, where $\boldsymbol{q}$ is the stopping probability vector
- FINAL vs. SimRank (Node Proximity) [Jeh'02]
  - $G_1 = G_2$ and w/o attribute, FINAL = SimRank by a scaling factor $\boldsymbol{D}^{\frac{1}{2}}$
- FINAL vs. Random Walk with Restart (RWR) [Tong'06]
  - $\boldsymbol{s}$ = RWR vector (defined on the attributed Kronecker graph)

Outline

- Motivations ✓
- Q1: FINAL Formulation ✓
- Q2: FINAL Algorithms
- Q3: FINAL Speed-up Computation
- Experimental Results
- Conclusions

FINAL Solutions

$$\boldsymbol{s} = \alpha \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{N} \mathrm{vec}\left(\sum_{l=1}^{L} (\boldsymbol{E}_2^l \odot \boldsymbol{A}_2) \boldsymbol{Q} (\boldsymbol{E}_1^l \odot \boldsymbol{A}_1)^T \right) + (1-\alpha)\boldsymbol{h}$$

$$\min_{\boldsymbol{s}} \ \alpha \boldsymbol{s}^T (\boldsymbol{I} - \widetilde{\boldsymbol{W}}) \boldsymbol{s} + (1-\alpha) \|\boldsymbol{s} - \boldsymbol{h}\|_2^2$$

- **Obs**: a convex optimization problem
- **Benefits**: a fixed-point solution converging to the global optimal solution

  $\boldsymbol{s} = a\widetilde{\boldsymbol{W}}\boldsymbol{s} + (1-\alpha)\boldsymbol{h} \Rightarrow \boldsymbol{s} = (1-\alpha)(\boldsymbol{I} - \alpha\widetilde{\boldsymbol{W}})^{-1}\boldsymbol{h}$ (closed form)
- **Intuition**: a similarity propagation to neighboring node-pairs, which is additionally filtered by node/edge attributes
- **Challenges**: computationally VERY expensive
  - Iterative solution: $O(m^2 t_{\max})$ (due to Kronecker product)
  - Closed form solution: $O(m^6)$ (due to matrix inversion)
- **Q**: how to scale up and speed up?

DATA Lab

Outline

- Motivations ✓
- Q1: FINAL Formulation ✓
- Q2: FINAL Algorithms ✓
- Q3: FINAL Speed-up Computation
- Experimental Results
- Conclusions

FINAL — Speed-up Full Alignment

- **Obs**: FINAL vs. RWGK and RWR
- **Solution**: leverage the existing fast solutions for RWGK and/or RWR [Kang 2012]
- **An Example**: only consider node attributes

$$\boldsymbol{s} = (1-\alpha)\left(\boldsymbol{I} - \alpha \boldsymbol{D}_N^{-\frac{1}{2}} \boldsymbol{N}(\boldsymbol{A}_1 \otimes \boldsymbol{A}_2)\boldsymbol{N}\boldsymbol{D}_N^{-\frac{1}{2}}\right)^{-1} \boldsymbol{h}$$

- **Key Idea:** low rank approximation of $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$

$$\boldsymbol{A}_1 \approx \boldsymbol{U}_1 \boldsymbol{\Lambda}_1 \boldsymbol{U}_1^T$$
$$\boldsymbol{A}_2 \approx \boldsymbol{U}_2 \boldsymbol{\Lambda}_2 \boldsymbol{U}_2^T$$

Sherman-Morrison Lemma

$$\boldsymbol{s} \approx (1-\alpha)\left(\boldsymbol{I} + \alpha \boldsymbol{D}_N^{-\frac{1}{2}} \boldsymbol{N}\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{N}\boldsymbol{D}_N^{-\frac{1}{2}}\right)\boldsymbol{h}$$

where $\boldsymbol{U} = \boldsymbol{U}_1 \otimes \boldsymbol{U}_2$

$$\boldsymbol{\Lambda} = [(\boldsymbol{\Lambda}_1 \otimes \boldsymbol{\Lambda}_2)^{-1} - \alpha \boldsymbol{U}^T \boldsymbol{N}\boldsymbol{D}_N^{-1}\boldsymbol{N}\boldsymbol{U}]^{-1}$$

- **Complexity**: $O(n^2 r^4)$
- **Challenge**: it is still $O(n^2)$. Can we do better?

FINAL — Speed-up On-query Alignment

- **Obs**: only need one column, or one segment of $\boldsymbol{S}$
- **Key Ideas**:
  - Low-rank approximation (same as for the full alignment)
  - Relax the degree matrix $\boldsymbol{D}_N = \boldsymbol{D}_1 \otimes \boldsymbol{D}_2$
- **Details**: $\boldsymbol{s}_a = (1-\alpha)\left[\boldsymbol{H}(:,a) + \alpha(\boldsymbol{D}_1(a,a)\boldsymbol{D}_2)^{-\frac{1}{2}}\boldsymbol{N}_a\right]$

$$\times \left[\underbrace{\boxed{(\boldsymbol{U}_1(a,:) \otimes \boldsymbol{U}_2)}}_{O(nr^2)} \boxed{\widehat{\boldsymbol{\Lambda}}} \underbrace{\boxed{\boldsymbol{U}^T\boldsymbol{N}\,(\boldsymbol{D}_1 \otimes \boldsymbol{D}_2)^{-\frac{1}{2}}\boldsymbol{h}}}_{O(n^2r^2)} \right]$$

(1) same trick as for full alignment

$$\boldsymbol{g} = \boldsymbol{U}^T\boldsymbol{N}(\boldsymbol{D}_1 \otimes \boldsymbol{D}_2)^{-\frac{1}{2}}\boldsymbol{h} = \sum_{i=1}^{p}\sum_{k=1}^{K}\sigma_i \left(\underbrace{\boxed{\boldsymbol{U}_1^T\boldsymbol{N}_1^k\boldsymbol{D}_1^{-\frac{1}{2}}\boldsymbol{v}_i}}_{O(nr)}\right) \otimes \left(\underbrace{\boxed{\boldsymbol{U}_2^T\boldsymbol{N}_2^k\boldsymbol{D}_2^{-\frac{1}{2}}\boldsymbol{u}_i}}_{O(nr)}\right)$$

(2) SVD on matrix $\boldsymbol{H} = \sum_{i=1}^{p}\sigma_i\boldsymbol{u}_i\boldsymbol{v}_i^T$

- **Benefits**: linear complexity $O((Kr^2 + pKr + p^2)n + mr + m_H p + r^6)$

Outline

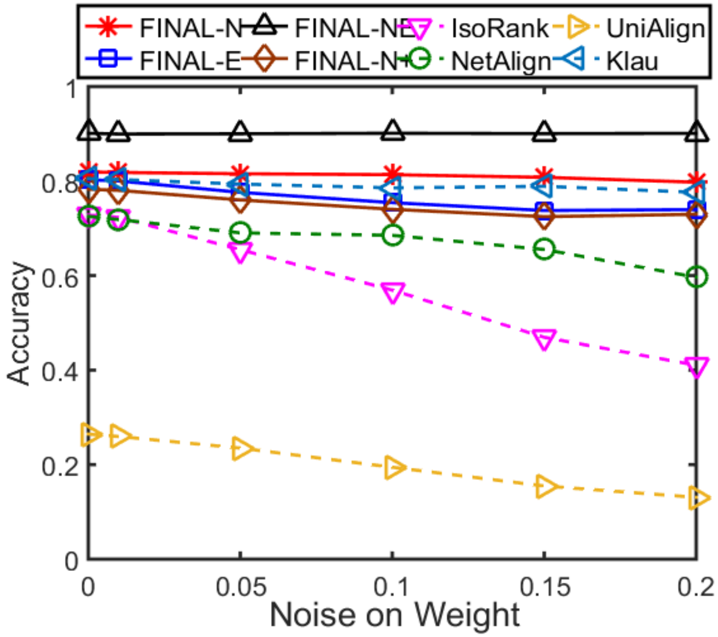
- Motivations ✓
- Q1: FINAL Formulation ✓
- Q2: FINAL Algorithms ✓
- Q3: FINAL Speed-up Computation ✓
- Experimental Results
- Conclusions

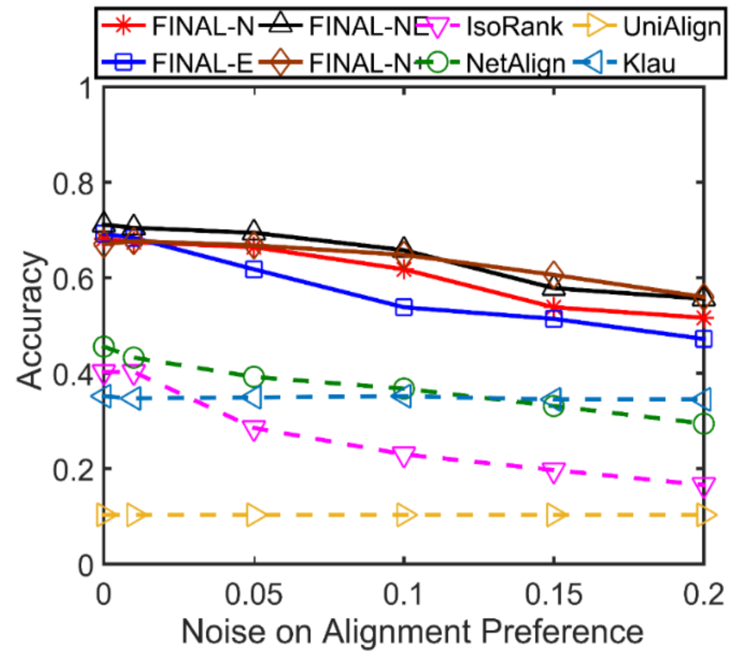
Experimental Setup

- **Datasets**:
  - DBLP co-author networks (nodes: 9,143 vs. 9,143)
  - Douban online & offline networks (nodes: 3,906 vs. 1,118)
  - Flickr & Last.fm networks (nodes: 12,974 vs. 15,436)
  - Flickr & Myspace networks (nodes: 6,714 vs. 10,733)
- **Evaluation Objectives**:
  - Effectiveness: one-to-one alignment accuracy
  - Efficiency: running time
- **Comparison Methods:**

FINAL (Our Methods)	**Baseline Methods**
☐ FINAL-NE (with node & edge attributes) ☐ FINAL-N (with node attributes) ☐ FINAL-E (with edge attributes) ☐ FINAL-N+ (speed-up FINAL-N)	☐ IsoRank [Singh'08] ☐ NetAlign [Bayati'09] ☐ UniAlign [Koutra'13] ☐ Klau's Algorithm [Klau'09]

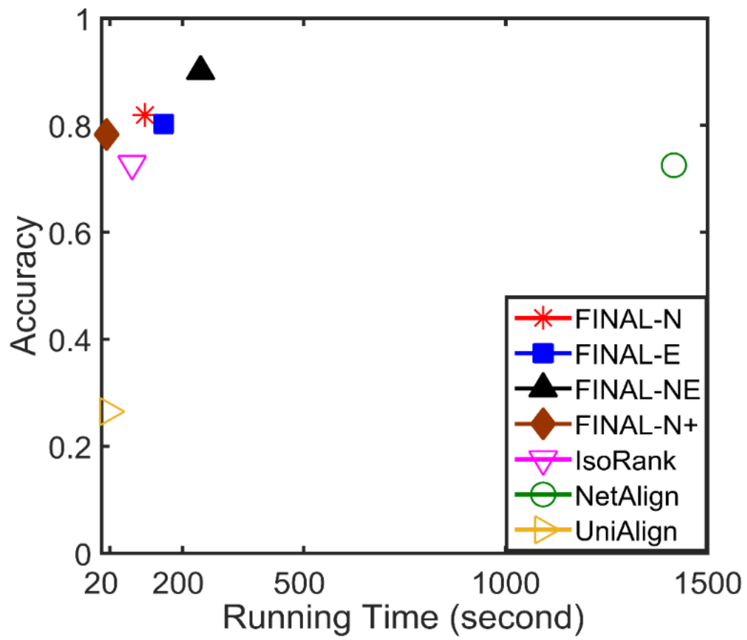
R1. Effectiveness Results

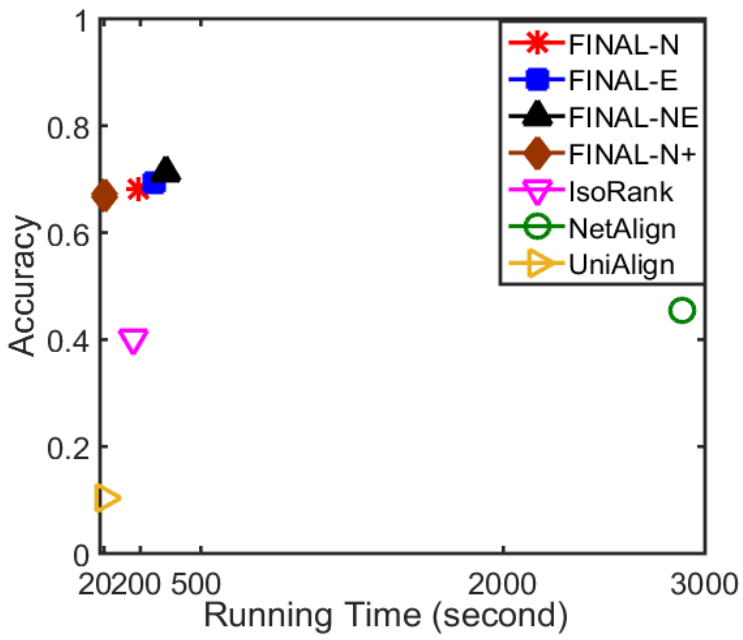




Obs: attributes help improve network alignment

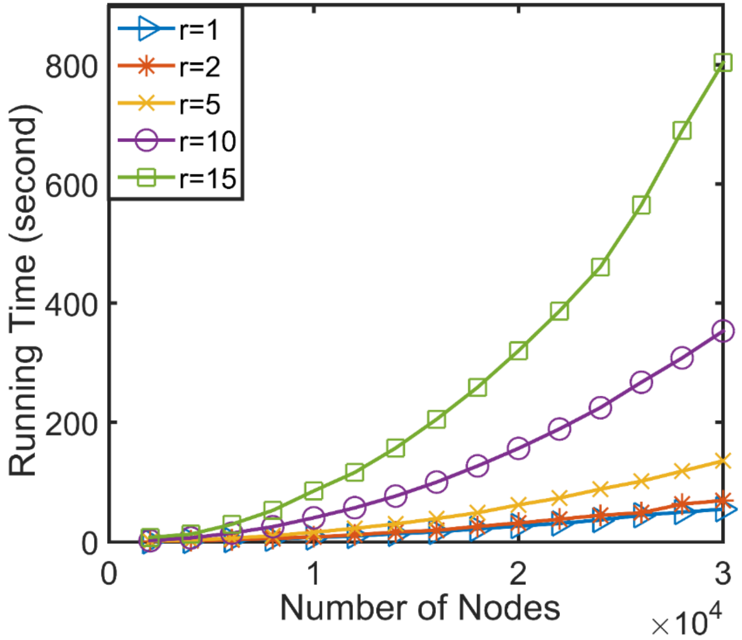
R2. Quality-Speed Balance





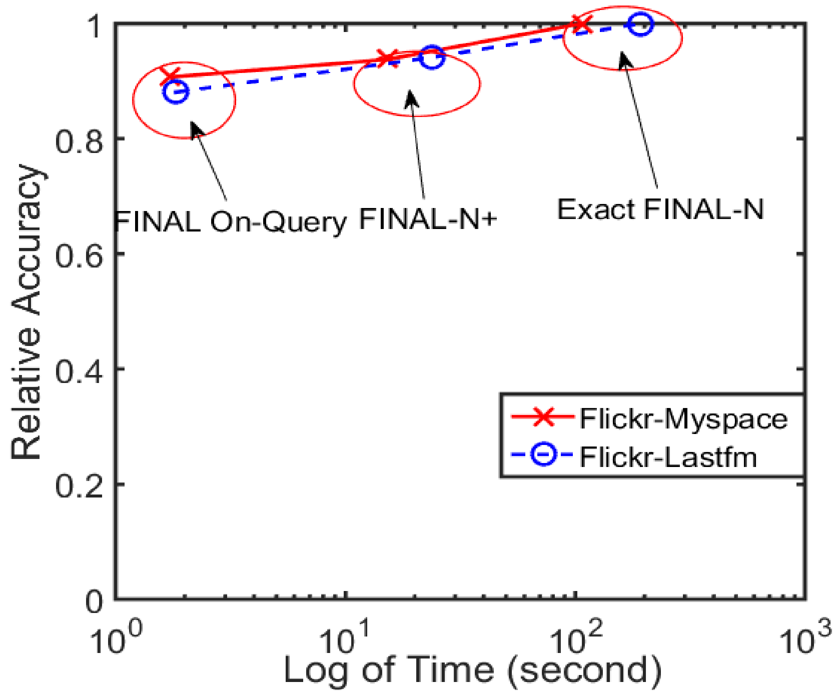
Obs: FINAL gain a better quality-speed balance.

R3. Scalability of FINAL-N+



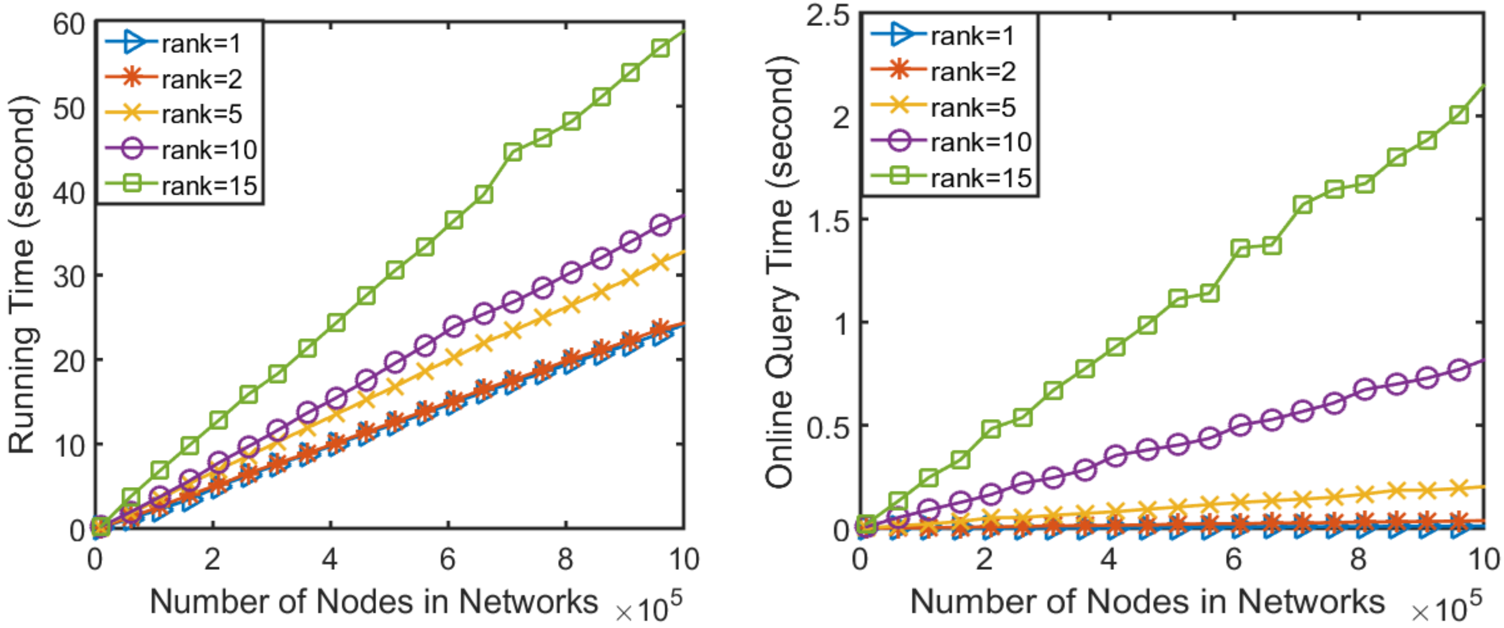
Obs: FINAL-N+ has a quadratic time complexity w.r.t the number of nodes.

R4. Quality-Speed of FINAL On-Query



Obs: FINAL On-Query gains around 90% accuracy relative to exact FINAL-N, but more than 100 times faster.

R5. Scalability of FINAL On-Query



Obs: FINAL On-Query has a **linear** time complexity

Outline

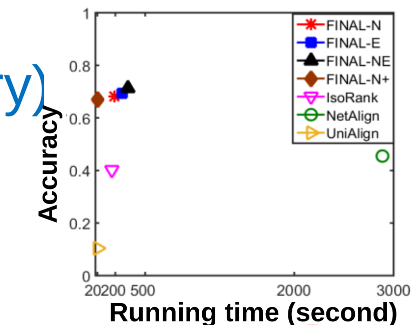
- Motivations ✓
- Q1: FINAL Formulation ✓
- Q2: FINAL Algorithms ✓
- Q3: FINAL Speed-up Computation ✓
- Experimental Results ✓
- Conclusions

Conclusions

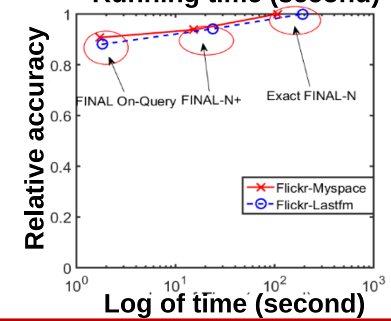
- Attributed Network Alignment
  - **Q1:** Formulation
  - **A1:** FINAL family →

$$\min_{\mathbf{S}} \; J(\mathbf{S}) = \sum_{a,b,x,y} \left[\frac{\mathbf{S}(x,a)}{\sqrt{f(x,a)}} - \frac{\mathbf{S}(y,b)}{\sqrt{f(y,b)}} \right]^2 \mathbf{A}_1(a,b)\mathbf{A}_2(x,y)$$
$$\times \mathbb{I}\big(\mathbf{N}_1(a,a) = \mathbf{N}_2(x,x)\big)\mathbb{I}\big(\mathbf{N}_1(b,b) = \mathbf{N}_2(y,y)\big)$$
$$\times \mathbb{I}\big(\mathbf{E}_1(a,b) = \mathbf{E}_2(x,y)\big)$$

  - **Q2:** Optimality
  - **A2:** Convex optimization problem → global optimal solution
  - **Q3:** Scalable computation
  - **A3:** Fast algorithms (FINAL-N+ & FINAL On-Query)



- Results
  - FINAL outperform other baseline methods
  - FINAL On-Query linear complexity



- More in Paper
  - Proof of optimality & more experimental results