Balancing Consistency and Disparity in Network Alignment
Network Alignment

- Goal: To find node correspondence across networks
- An example:

- Evolutionary relationship discovery

<table>
<thead>
<tr>
<th>Organism</th>
<th>CHIMP</th>
<th>MOUSE</th>
<th>CHICKEN</th>
<th>FRUIT FLY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gene Conservation with Humans (%)</td>
<td>99.5</td>
<td>88</td>
<td>75</td>
<td>60</td>
</tr>
</tbody>
</table>
Problem Definition

- **Given:** (1) undirected networks $\mathcal{G}_1 = \{\mathcal{V}_1, \mathcal{A}_1, \mathcal{X}_1\}$, $\mathcal{G}_2 = \{\mathcal{V}_2, \mathcal{A}_2, \mathcal{X}_2\}$; (2) a set of anchor links $\mathcal{L}$
- **Output:** alignment matrix $\mathcal{S}$
Existing Methods

- Optimization-based methods
  - Key idea: To encourage alignment consistency among neighbors
  - Example formulation (FINAL [1]):
    - Intuition: similar node pairs tend to have similar neighboring node pairs
    - Math:

\[
\min_S \sum_{a,b,x,y} \left( \frac{S(a,x)}{\sqrt{|N_1(a)||N_2(x)|}} - \frac{S(b,y)}{\sqrt{|N_1(b)||N_2(y)|}} \right)^2 \cdot A_1(a,b)A_2(x,y)
\]

Existing Methods (Con’t)

- **Embedding-based methods**
  - Key idea: To learn node embeddings w/ negative sampling
  - Example formulation [1]:
    - Intuition: Nodes that are close in embedding space are more likely to be aligned
    - Math:
      \[
      \log p(x|a) \propto \log \sigma(x^T a) + \sum_{m=1}^{K} E_{x_n \sim p_n(x)} \log \sigma(-x_n^T a)
      \]

Limitation #1: Alignment Consistency

- Alignment over-smoothness issue
  - Given an anchor link \((a, x)\), i.e., they are aligned apriori

\[
\min_S \sum_{a,b,x,y} \left[ \frac{S(a, x)}{\sqrt{\vert N_1(a)\vert \vert N_2(x)\vert}} - \frac{S(b, y)}{\sqrt{\vert N_1(b)\vert \vert N_2(y)\vert}} \right]^2 A_1(a, b)A_2(x, y)
\]

- Anchor link \((a, x)\) \(\rightarrow\) High \(S(a, x)\)
- Minimizing alignment difference \(\rightarrow\) High \(S(b, y)\) for all neighboring node pairs

- Cannot distinguish correct alignments from misleading ones
  - Equivalently, neighboring node pairs \((b, y)\) are used as positive samples of \((a, x)\)
**Limitation #2: Alignment Disparity**

- Negative sampling $\rightarrow$ disparity $\rightarrow$ reduce over-smoothness
- Competing sampling strategies

<table>
<thead>
<tr>
<th>Alignment consistency</th>
<th>Meaningful disparity</th>
<th>Example negative of anchor $(a, x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive correlation [1]</td>
<td>✔️</td>
<td>Node pair $(b, y)$</td>
</tr>
<tr>
<td>Negative correlation [2]</td>
<td>✔️</td>
<td>Node pair $(e, h)$</td>
</tr>
<tr>
<td>Degree-based sampling [3]</td>
<td>?</td>
<td>Node pair $(d, x)$</td>
</tr>
</tbody>
</table>

Balancing Consistency & Disparity

Key question:

What are the intrinsic relationships behind alignment consistency and disparity?

Q1: How to design model architecture to encode alignment consistency?

Q2: How to sample negative node pairs to distinguish correct alignments from misleading ones?

- Target #1: Should not violate overall alignment consistency
- Target #2: Should learn meaningful node embeddings
Outline

- Motivations
- NeXtAlign Model
  - Model Design
  - Model Training
- Experimental Results
- Conclusions
Alignment Consistency by GCNs

- **Unsupervised FINAL [1]**

\[
\min_s \sum_{a,b,x,y} \left[ \frac{S(a,x)}{\sqrt{|\mathcal{N}_1(a)||\mathcal{N}_2(x)|}} - \frac{S(b,y)}{\sqrt{|\mathcal{N}_1(b)||\mathcal{N}_2(y)|}} \right]^2
\]

- **Relationship with GCNs**

Suppose \( S^t = (H_1^t)' H_2^t \)

\[
S^t(a, x) = (a^t)' x^t = \tilde{A}_1(a, :) S^{t-1} \tilde{A}_2(:, x)
\]

\[
S^t(a, x) = \left( \sum_{b \in \mathcal{N}_1(a)} \frac{b^{t-1}}{\sqrt{|\mathcal{N}_1(a)||\mathcal{N}_1(b)|}} \right)' \sum_{y \in \mathcal{N}_2(x)} \frac{y^{t-1}}{\sqrt{|\mathcal{N}_2(x)||\mathcal{N}_2(y)|}}
\]

- **Fixed-point solution**

\[
S^t = \tilde{A}_1 S^{t-1} \tilde{A}_2
\]

- **Update by GCN w/o parameters**

\[
\alpha^t \quad \text{Inner product} \quad x^t
\]

Alignment Consistency by GCNs (Con’t)

- Alignment consistency – semi-supervised [1]
  \[
  \min_{\alpha} \sum_{a,b,x,y} \left[ \frac{S(a, x)}{\sqrt{|N_1(a)||N_2(x)|}} - \frac{S(b, y)}{\sqrt{|N_1(b)||N_2(y)|}} \right]^2 A_1(a, b)A_2(x, y) + (1 - \alpha) \|S - L\|_F^2
  \]
  \[
  S^t = \alpha \tilde{A}_1 S^{t-1} \tilde{A}_2 + (1 - \alpha) L
  \]

- Message passing w/o parameters

\[
\begin{align*}
  u^t &= \sqrt{\alpha} \sum_{b \in N_1(u)} \frac{b^{t-1}}{\sqrt{|N_1(u)||N_1(b)|}} + \sqrt{1 - \alpha} u^{t-1} \\
  v^t &= \sqrt{\alpha} \sum_{y \in N_2(v)} \frac{y^{t-1}}{\sqrt{|N_2(v)||N_2(y)|}} + \sqrt{1 - \alpha} v^{t-1} \\
  a^t &= x^t = \sqrt{\alpha} \sum_{b \in N_1(a)} \frac{b^{t-1}}{\sqrt{|N_1(a)||N_1(b)|}} + \sqrt{1 - \alpha} x^{t-1} \\
  a^0 &= x^0 = e_i
\end{align*}
\]

Alignment consistency

- Within-network proximity

\[
\begin{align*}
  S(u, v) &= \alpha \tilde{A}_1(u, :) \tilde{A}_2(:, v) + (1 - \alpha) L(u, v) \\
  S(u, x) &= \alpha \tilde{A}_1(u, :) \tilde{A}_2(:, x) + (1 - \alpha) L(u, x) + \alpha S_1(u, a) + \sqrt{(1 - \alpha)} \frac{A_1(u, a)}{\sqrt{|N_1(u)||N_1(a)|}} \\
  S(a, x) &= 2\alpha \tilde{A}_1(a, :) \tilde{A}_2(:, x) + (1 - \alpha) L(a, x) + \alpha (S_1(a, a) + S_2(x, x))
\end{align*}
\]

Fixed-point solution

RelGCN – Relational GCN for Alignment

- Message passing w/ parameters

\[ u^t = \sqrt{\alpha} \sum_{b \in \mathcal{N}_1(u)} \frac{W_1^t b^{t-1}}{\sqrt{|\mathcal{N}_1(u)||\mathcal{N}_1(b)|}} + \sqrt{1 - \alpha} W_0^t u^{t-1} \]

\[ v^t = \sqrt{\alpha} \sum_{y \in \mathcal{N}_2(v)} \frac{W_2^t y^{t-1}}{\sqrt{|\mathcal{N}_2(v)||\mathcal{N}_2(y)|}} + \sqrt{1 - \alpha} W_0^t v^{t-1} \]

\[ x^t = a^t = \sqrt{\alpha} \sum_{b \in \mathcal{N}_1(a)} \frac{W_1^t b^{t-1}}{\sqrt{|\mathcal{N}_1(a)||\mathcal{N}_1(b)|}} + \sqrt{1 - \alpha} W_0^t x^{t-1} + \sqrt{\alpha} \sum_{y \in \mathcal{N}_2(x)} \frac{W_2^t y^{t-1}}{\sqrt{|\mathcal{N}_2(x)||\mathcal{N}_2(y)|}} \]

- \( W_0^t, W_1^t, W_2^t \): parameters at the \( t \)-th layer
- RelGCN-U: variant w/o parameters
NeXtAlign – Model Design

- **Key idea:**
  - Use RelGCNs to compute relative positions w.r.t. anchor nodes
  - Feed to a linear layer to compute final embeddings

- **Model architecture**
Model Design Details

- **Goal:** To use RelGCN-U to encode alignment consistency
- **Pre-positioning:**
  - Anchor nodes: \( a^0 = x^0 = e_i \)
  - Non-anchor nodes: RWR scores w.r.t. anchor nodes [1,2]

- **Goal:** To mitigate over-smoothness of RelGCN-U
  - RelGCN w/ attention to rescale positions

\[
\begin{align*}
    c_{ua} &= \frac{\exp(w'_c[\hat{u}||\hat{a}])}{\sum_{b \in \mathcal{L}_1} \exp(w'_c[\hat{u}||\hat{b}])}
\end{align*}
\]

---


Outline

- Motivations
- NeXtAlign Model
  - Model Design
  - Model Training
- Experimental Results
- Conclusions
NeXtAlign – Model Training

- **Loss functions**

  \[ J_a = -\sum_{b \in V_1} [p_d(b|a) \log \sigma(b'|a) + k p_n(b|a) \log \sigma(-b'|a)] \]
  \[ J_x = -\sum_{y \in V_2} [p_a(y|x) \log \sigma(y'|x) + k p_n(y|x) \log \sigma(-y'|x)] \]
  \[ J_{ax} = -\sum_{b \in V_1} [p_{dc}(b|x) \log \sigma(b'|x) + k p_{nc}(b|x) \log \sigma(-b'|x)] \]
  \[ -\sum_{y \in V_2} [p_{dc}(y|a) \log \sigma(y'|a) + k p_{nc}(y|a) \log \sigma(-y'|a)] \]
  \[ J = \sum_{(a,x) \in \mathcal{L}} J_{a,x} = \sum_{(a,x) \in \mathcal{L}} J_a + J_x + J_{ax} \]

- \( p_d, p_n \): within-network positive, negative sampling distributions
- \( p_{dc}, p_{nc} \): cross-network positive, negative sampling distributions

**Question: How to design sampling distributions?**
Sampling Strategy

- An intuitive design
  - \( p_d \): similar nodes are likely to co-occur in the context [1]
  - \( p_n \): samples distant/dissimilar nodes [2]
  - \( p_{dc} \): high-similarity node pairs preserve alignment consistency
  - \( p_{nc} \): high-similarity node pairs \( \rightarrow \) hard negative alignment pairs [3] \( \rightarrow \) alignment disparity

\[ \text{Lemma} \]
Denote \( \Delta \theta_b = \theta_b^B - \theta_b^* \) and \( \Delta \theta_y = \theta_y^B - \theta_y^* \). The mean square errors for nodes \( b \in L_1 \) and \( y \in L_2 \) can be formulated by

\[
\begin{align*}
\mathbb{E}[\Delta \theta_b^2] &= \frac{1}{B} \left[ \frac{1}{p_d(b|a) + p_{dc}(b|x)} + \frac{1}{k p_n(b|a) + k p_{nc}(b|x)} - C \right], \\
\mathbb{E}[\Delta \theta_y^2] &= \frac{1}{B} \left[ \frac{1}{p_d(y|x) + p_{dc}(y|a)} + \frac{1}{k p_n(y|x) + k p_{nc}(y|a)} - C \right].
\end{align*}
\]

For nodes \( b \in L_1 \) and \( y \in L_2 \), the mean square error is computed by

\[
\begin{align*}
\mathbb{E}[\Delta \theta_b^2] &= \mathbb{E}[\Delta \theta_y^2] = \frac{1}{B} \left[ \frac{1}{p_1} + \frac{1}{k p_2} - C \right],
\end{align*}
\]

Competing objectives

- High-probability node pairs
- Large \( p_{nc} \)
- High \( p_d, p_{dc} \)
- Low \( p_n \)
Sampling Strategy (Con’t)

- Denote $b = [b_1 \parallel b_2], x = [x_1 \parallel x_2]$
  - $b_1$: captures local information of node-$b$ in $G_1$
  - $b_2$: captures how node-$b$ posits in $G_2$

- A new scoring function $\rightarrow$ instead of plain inner product

$$b \ast x = w_1 b'_1 x_1 + w_2 b'_1 x_2 + w_3 b'_2 x_1 + w_4 b'_2 x_2$$

$a = x$

Intra-network proximity

Node interaction similar as recommendation

Node interaction in the context of $G_2$

$p_{da}, p_n$

$p_{nc}$

$p_{dc}$

[Diagram showing the interaction and loss function]
Outline

- Motivations ✓
- NeXtAlign Model
  - Model Design ✓
  - Model Training ✓
- Experimental Results
- Conclusions
Experimental Setup

- Evaluation objectives
  - How accurate is NeXtAlign for network alignment?
  - Effectiveness of different components

- Datasets

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Networks</th>
<th># of nodes</th>
<th># of edges</th>
<th># of attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>ACM</td>
<td>9,872</td>
<td>39,561</td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>DBLP</td>
<td>9,916</td>
<td>44,808</td>
<td>17</td>
</tr>
<tr>
<td>S2</td>
<td>Foursquare</td>
<td>5,313</td>
<td>54,233</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Twitter</td>
<td>5,120</td>
<td>130,575</td>
<td>0</td>
</tr>
<tr>
<td>S3</td>
<td>Phone</td>
<td>1,000</td>
<td>41,191</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td>1,003</td>
<td>4,627</td>
<td>0</td>
</tr>
</tbody>
</table>

- Baseline methods
  - Bright [1], NetTrans [2], FINAL [3], IONE [4], CrossMNA [5]

Experimental Results #1

Results with 20% training data w/o node attributes.

<table>
<thead>
<tr>
<th></th>
<th>ACM-DBLP</th>
<th></th>
<th>Foursquare-Twitter</th>
<th></th>
<th>Phone-Email</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hits@10</td>
<td>Hits@30</td>
<td>Hits@10</td>
<td>Hits@30</td>
<td>Hits@10</td>
<td>Hits@30</td>
</tr>
<tr>
<td>NeXtAlign</td>
<td>0.8417±0.0032</td>
<td>0.9011±0.0081</td>
<td>0.2956±0.0096</td>
<td>0.4174±0.0066</td>
<td>0.3926±0.0168</td>
<td>0.6748±0.0105</td>
</tr>
<tr>
<td>Bright</td>
<td>0.7904±0.0041</td>
<td>0.8669±0.0041</td>
<td>0.2500±0.0154</td>
<td>0.3206±0.0097</td>
<td>0.2570±0.0091</td>
<td>0.5344±0.0086</td>
</tr>
<tr>
<td>NetTrans</td>
<td>0.7925±0.0065</td>
<td>0.8356±0.0082</td>
<td>0.2468±0.0036</td>
<td>0.3458±0.0098</td>
<td>0.2650±0.0025</td>
<td>0.5325±0.0075</td>
</tr>
<tr>
<td>FINAL</td>
<td>0.6768±0.0080</td>
<td>0.8237±0.0098</td>
<td>0.2357±0.0091</td>
<td>0.3457±0.0091</td>
<td>0.2203±0.0151</td>
<td>0.4586±0.0184</td>
</tr>
<tr>
<td>IONE</td>
<td>0.7476±0.0125</td>
<td>0.8453±0.0097</td>
<td>0.1624±0.0109</td>
<td>0.2918±0.0209</td>
<td>0.3779±0.0131</td>
<td>0.6444±0.0084</td>
</tr>
<tr>
<td>CrossMNA</td>
<td>0.6532±0.0042</td>
<td>0.7900±0.0041</td>
<td>0.0236±0.0172</td>
<td>0.0751±0.0384</td>
<td>0.1542±0.0041</td>
<td>0.4045±0.0115</td>
</tr>
</tbody>
</table>

Observations:

- Our method NeXtAlign significantly outperforms other baseline methods.
- More improvements on Foursquare-Twitter and Phone-Email whose network structures are disparate (i.e., consistency may not work well).
Experimental Results #2

Results with node attributes.

<table>
<thead>
<tr>
<th></th>
<th>10% training data</th>
<th>20% training data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hits@10</td>
<td>Hits@30</td>
</tr>
<tr>
<td>NeXtAlign</td>
<td>0.785±0.010</td>
<td>0.871±0.009</td>
</tr>
<tr>
<td>Bright</td>
<td>0.781±0.004</td>
<td>0.862±0.003</td>
</tr>
<tr>
<td>NetTrans</td>
<td>0.708±0.004</td>
<td>0.846±0.009</td>
</tr>
<tr>
<td>FINAL</td>
<td>0.651±0.013</td>
<td>0.817±0.009</td>
</tr>
</tbody>
</table>

**Observation:** Our method NeXtAlign still outperforms other baseline methods.
Experimental Results #3

- Ablation study on model design
  - (1) RWR scores, (2) RelGCN-U: uses output of RelGCN-U,
  - (3) RelGCN-C: uses re-scaled relative positions

**Observation:** All components are necessary to achieve the best performance.
Experimental Results #4

- Ablation study on negative sampling strategies

<table>
<thead>
<tr>
<th></th>
<th>ACM-DBLP</th>
<th>Foursquare-Twitter</th>
<th>Phone-Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>NeXtAlign</td>
<td>0.9277</td>
<td>0.4103</td>
<td>0.6813</td>
</tr>
<tr>
<td>Uniform</td>
<td>0.8975</td>
<td>0.3924</td>
<td>0.6525</td>
</tr>
<tr>
<td>Degree</td>
<td>0.9093</td>
<td>0.3923</td>
<td>0.6637</td>
</tr>
<tr>
<td>Positive</td>
<td>0.9097</td>
<td>0.4040</td>
<td>0.6650</td>
</tr>
</tbody>
</table>

**Observation:** The proposed negative sampling method achieves better performance than sampling hard negatives.
Outline

▪ Motivations ✓
▪ NeXtAlign Model ✓
  ▪ Model Design
  ▪ Model Training
▪ Experimental Results ✓
▪ Conclusions
Conclusions

- **Goal:** To strike a balance of alignment consistency and disparity in semi-supervised network alignment

- **Method:**
  - Model design
    - Connect GCNs with FINAL
  - RelGCN for alignment consistency
  - Model training
    - New sampling method for disparity

- **Results**
  - NeXtAlign significantly outperforms baseline methods
  - The proposed sampling method achieves better performance
Thank you
Embedding Mean Square Errors

- Empirical risk $J^B(a,x)$
  - Sample $B$ nodes by $p_d, p_n, p_{dc}, p_{nc}$
- Denote $\theta = [b'_1 x, \ldots, b'_{n_1} x, y'_1 x, \ldots, y'_{n_2} x]$
- $\theta^*, \theta^B$: optimal embedding to $J(a,x), J^B(a,x)$

\[
J^B_{(a,x)} = -\frac{1}{B} \sum_{i_1,i_2,j_1,j_2} \left( \log \sigma(b'_{i_1} x) + \log \sigma(b'_{i_2} x) \\
+ \log \sigma(y'_{j_1} x) + \log \sigma(y'_{j_2} x) \right) \\
- \frac{1}{B} \sum_{i_3,i_4,j_3,j_4} \left( \log \sigma(-b'_{i_3} x) + \log \sigma(-b'_{i_4} x) \\
+ \log \sigma(-y'_{j_3} x) + \log \sigma(-y'_{j_4} x) \right)
\]