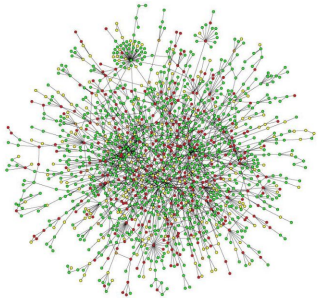
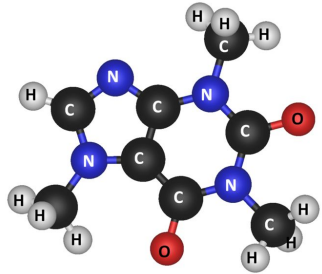
Gromov-Wasserstein Learning:
A New Machine Learning Framework for Structured Data Analysis

Hongteng Xu

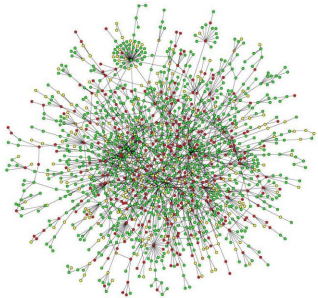
Infinia ML Inc. and Duke University

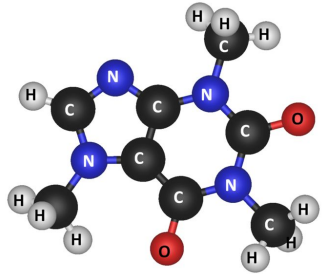
Structured data analysis





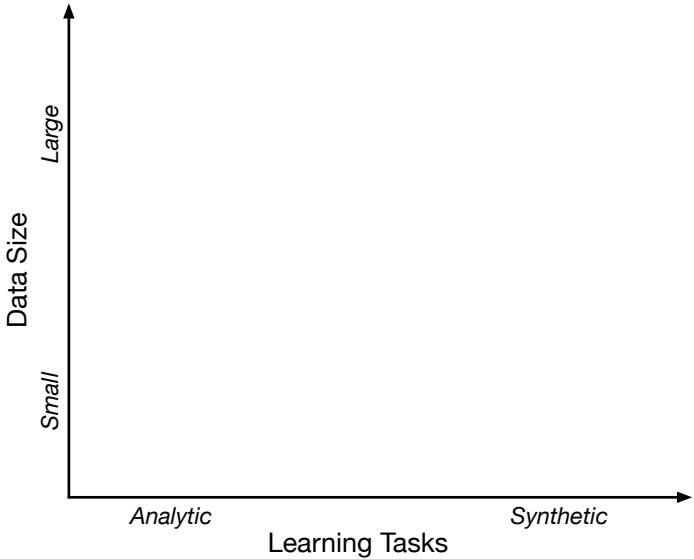
Structured data analysis



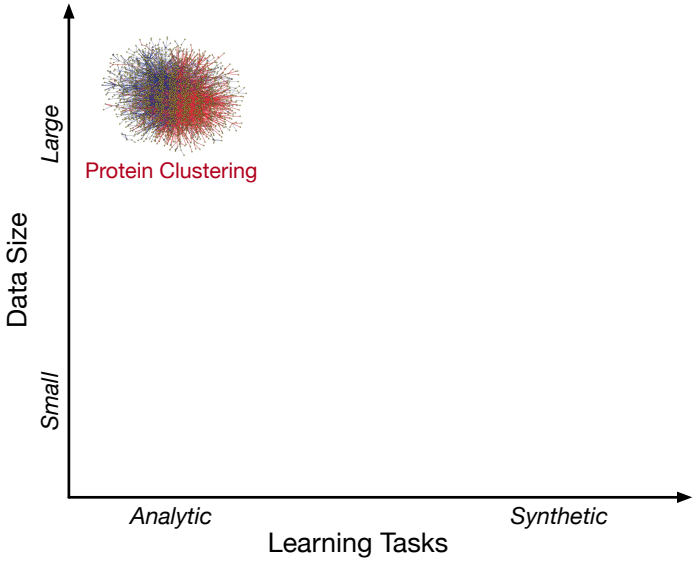


Extract knowledge from relations effectively and efficiently.

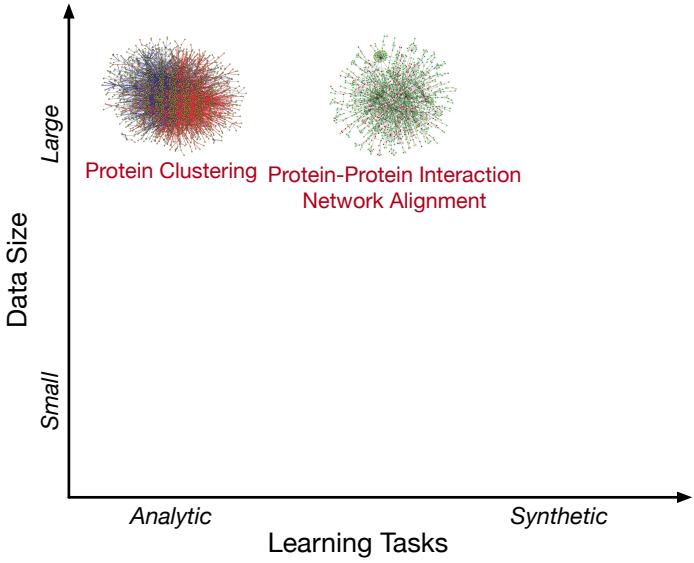
The space of problems



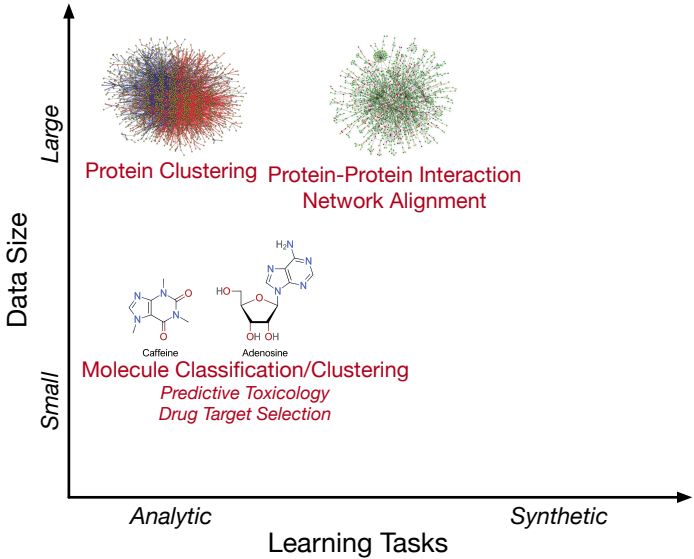
The space of problems



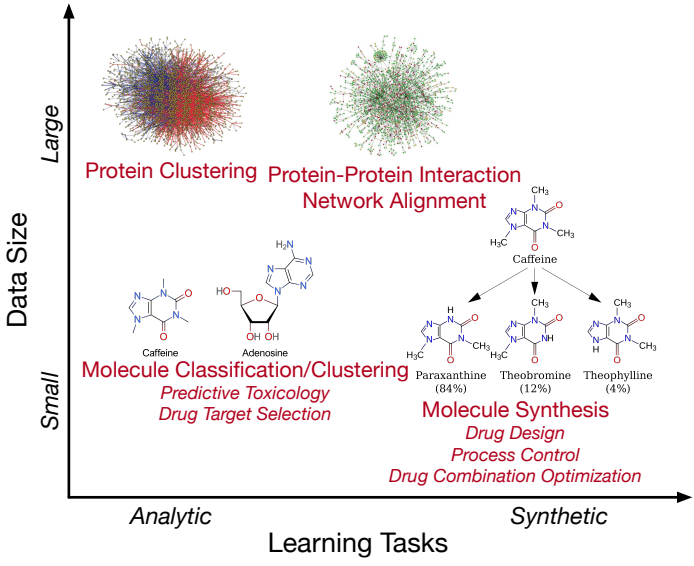
The space of problems



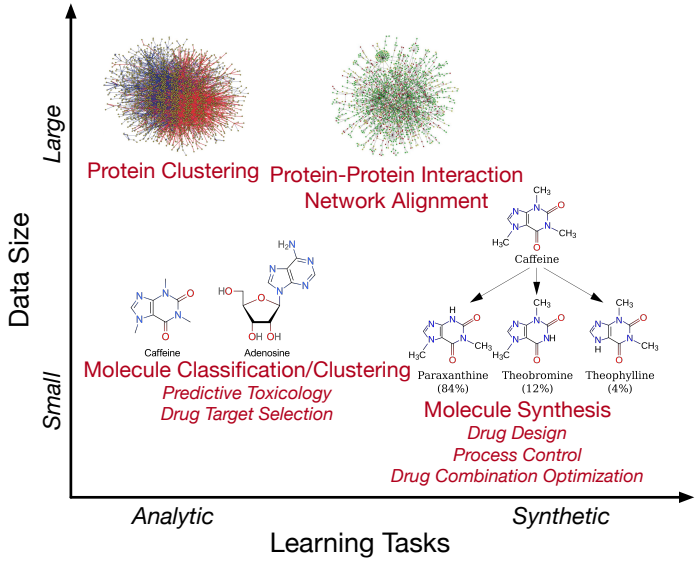
The space of problems



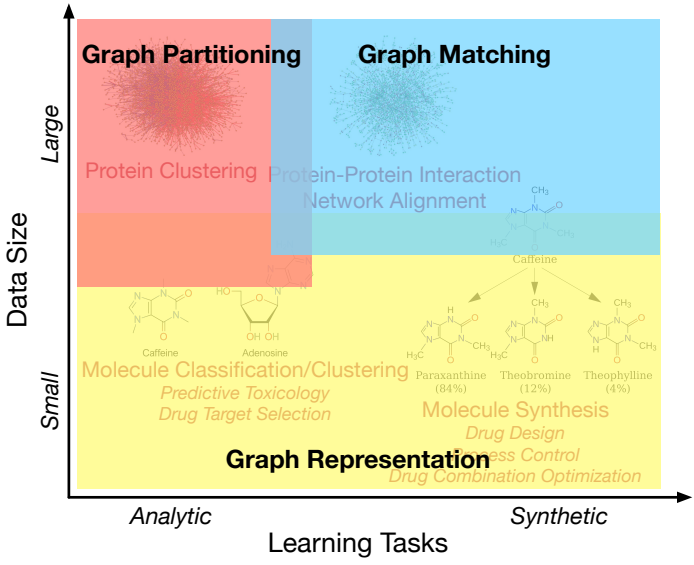
The space of problems



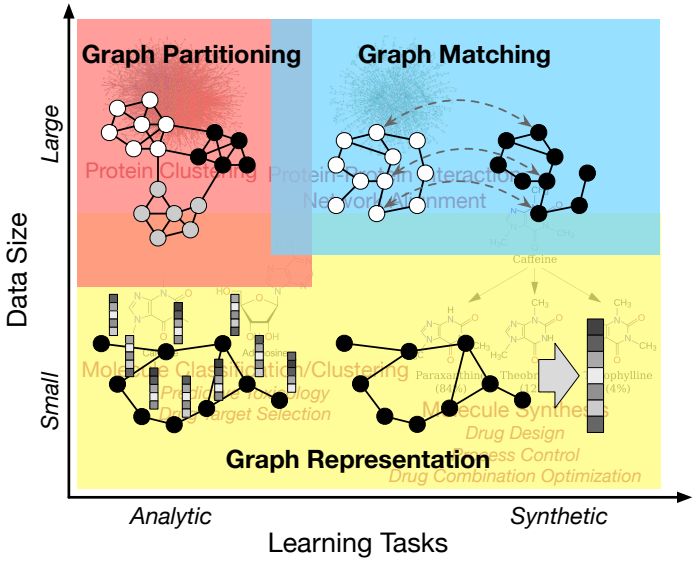
The space of problems



The space of problems



The space of problems



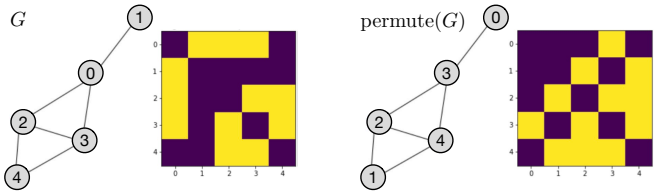
The challenges of the problems

The challenges of the problems

- NP-completeness
  - Approximation algorithms with high stability and scalability

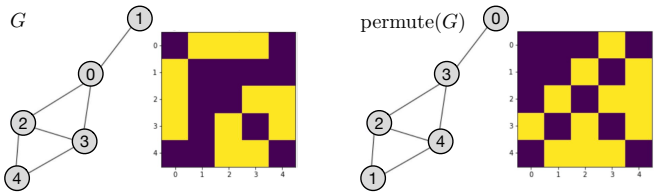
The challenges of the problems

- NP-completeness
  - Approximation algorithms with high stability and scalability
- Permutation invariance



The challenges of the problems

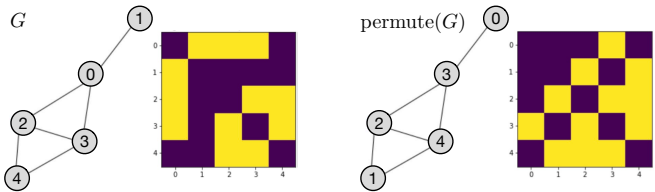
- NP-completeness
  - Approximation algorithms with high stability and scalability
- Permutation invariance



  - A permutation-invariant metric d: $d(G_X, G_Y) = d(G_X, \text{permute}(G_Y))$
  - A permutation-invariant representation model f: $f(G) = f(\text{permute}(G))$

The challenges of the problems

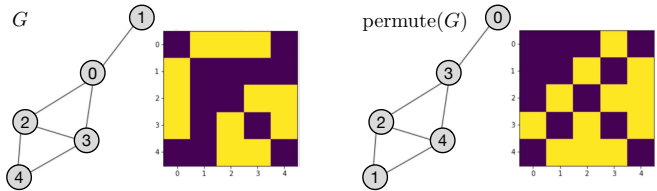
- NP-completeness
  - Approximation algorithms with high stability and scalability
- Permutation invariance



  - A permutation-invariant metric d: $d(G_X, G_Y) = d(G_X, \text{permute}(G_Y))$
  - A permutation-invariant representation model f: $f(G) = f(\text{permute}(G))$
- (Often) No labels
  - Unsupervised or semi-supervised learning

The challenges of the problems

- NP-completeness
  - Approximation algorithms with high stability and scalability
- Permutation invariance



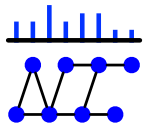
  - A permutation-invariant metric d: $d(G_X, G_Y) = d(G_X, \text{permute}(G_Y))$
  - A permutation-invariant representation model f: $f(G) = f(\text{permute}(G))$
- (Often) No labels
  - Unsupervised or semi-supervised learning

Gromov-Wasserstein Learning (GWL) provides a potential solution.

Applications: PPI network alignment, molecule clustering and classification.

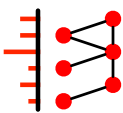
Gromov-Wasserstein distance (GWD) for structured data

Gromov-Wasserstein distance (GWD) for structured data



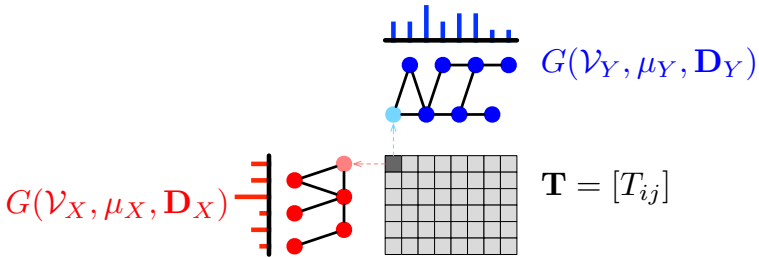
$G(\mathcal{V}_Y, \mu_Y, \mathbf{D}_Y)$

$G(\mathcal{V}_X, \mu_X, \mathbf{D}_X)$



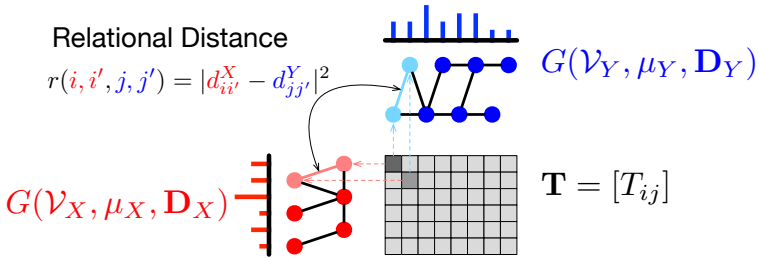
- $\mathcal{V}$: the node set
- $\boldsymbol{\mu}$: a predefined distribution of nodes
- $\boldsymbol{D} = [d_{ii'}]$: the adjacency / distance / kernel matrix

Gromov-Wasserstein distance (GWD) for structured data



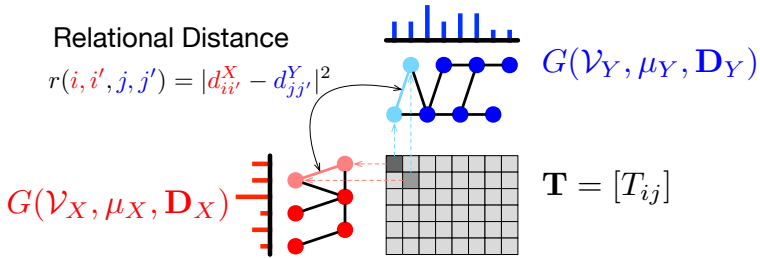
- $\boldsymbol{T} = [T_{ij}]$: a joint distribution of nodes
- $(i \in \mathcal{V}_X, j \in \mathcal{V}_j) \sim \boldsymbol{T}$.
- $\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y) = \{\boldsymbol{T} \geq \mathbf{0} \mid \boldsymbol{T}\mathbf{1} = \boldsymbol{\mu}_X,\ \boldsymbol{T}^\top \mathbf{1} = \boldsymbol{\mu}_Y\}$

Gromov-Wasserstein distance (GWD) for structured data



- $\underbrace{\boldsymbol{T} \otimes \boldsymbol{T}}_{\text{Kronecker product}}$: a joint distribution of edges.
- The pair of edges $(d^X_{ii'}, d^Y_{jj'}) \sim \boldsymbol{T} \otimes \boldsymbol{T}$.
- Relational distance $r(i,i',j,j')$: the difference between the edges.

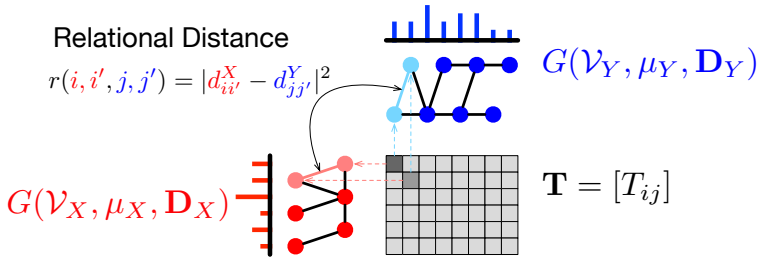
Gromov-Wasserstein distance (GWD) for structured data



The GWD is **the minimum expectation of the relational distance:**

$$
\begin{aligned}
d_{gw}(G_X, G_Y) &:= \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y)} \mathbb{E}_{(i,i',j,j') \sim \boldsymbol{T} \otimes \boldsymbol{T}} [r(i,i',j,j')] \\
&= \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y)} \sum_{i,i'} \sum_{j,j'} \underbrace{|d^X_{ii'} - d^Y_{jj'}|^2}_{\text{distance } r} \underbrace{T_{ij} T_{i'j'}}_{\text{prob}(r)} \qquad (1)
\end{aligned}
$$

Gromov-Wasserstein distance (GWD) for structured data

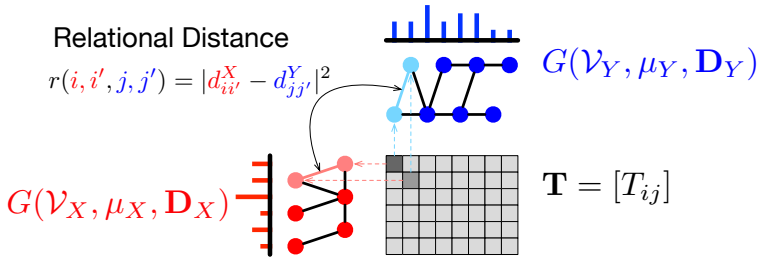


The GWD is **the minimum expectation of the relational distance:**

$$
\begin{aligned}
d_{gw}(G_X, G_Y) &:= \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y)} \mathbb{E}_{(i,i',j,j') \sim \boldsymbol{T} \otimes \boldsymbol{T}}[r(i,i',j,j')] \\
&= \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T} \rangle,
\end{aligned}
\tag{1}
$$

- $\boldsymbol{D}_{XY} = (\boldsymbol{D}_X \odot \boldsymbol{D}_X)\boldsymbol{\mu}_X \mathbf{1}^\top_{|\mathcal{V}_Y|} + \mathbf{1}_{|\mathcal{V}_X|} \boldsymbol{\mu}_Y^\top (\boldsymbol{D}_Y \odot \boldsymbol{D}_Y)^\top$.

Gromov-Wasserstein distance (GWD) for structured data

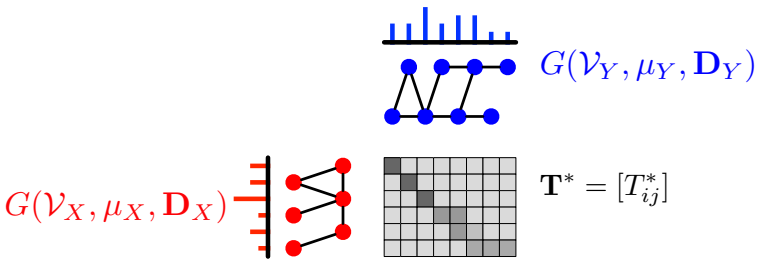


The GWD is **the minimum expectation of the relational distance:**

$$
\begin{aligned}
d_{gw}(G_X, G_Y) &:= \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y)} \mathbb{E}_{(i,i',j,j') \sim \boldsymbol{T} \otimes \boldsymbol{T}}[r(i,i',j,j')] \\
&= \min_{\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T} \rangle,
\end{aligned}
\tag{1}
$$

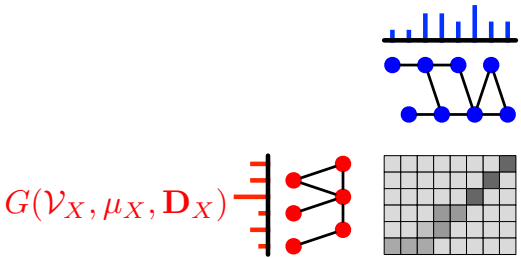
- $\boldsymbol{D}_{XY} = (\boldsymbol{D}_X \odot \boldsymbol{D}_X)\boldsymbol{\mu}_X \mathbf{1}^\top_{|\mathcal{V}_Y|} + \mathbf{1}_{|\mathcal{V}_X|} \boldsymbol{\mu}_Y^\top (\boldsymbol{D}_Y \odot \boldsymbol{D}_Y)^\top$.
- Given comparable node attributes, $\boldsymbol{D}_{XY} \leftarrow \boldsymbol{D}_{XY} + \boldsymbol{D}(\boldsymbol{F}_X, \boldsymbol{F}_Y)$

Advantages of GWD



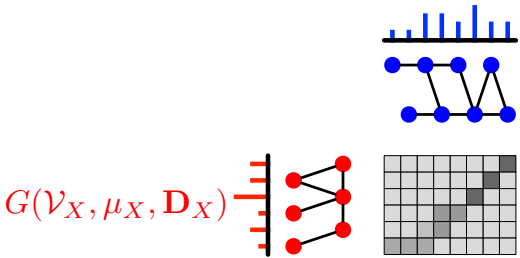
- The optimal joint distribution $\boldsymbol{T}^*$ (or called "**optimal transport**" matrix) indicates the correspondence between the two graphs.

Advantages of GWD



- The optimal joint distribution $\boldsymbol{T}^*$ (or called "**optimal transport**" matrix) indicates the correspondence between the two graphs.
- A **permutation-invariant** (pseudo) metric
  - $d_{gw}(G_X, G_Y) = d_{gw}(G_X, \text{permute}(G_Y))$

Advantages of GWD



- The optimal joint distribution $\boldsymbol{T}^*$ (or called "**optimal transport**" matrix) indicates the correspondence between the two graphs.
- A **permutation-invariant** (pseudo) metric
  - $d_{gw}(G_X, G_Y) = d_{gw}(G_X, \text{permute}(G_Y))$
- Applicable to the graphs with different sizes, $i.e.$, $|\mathcal{V}_X| \neq |\mathcal{V}_Y|$.
- Applicable to the graphs with/without node attributes.

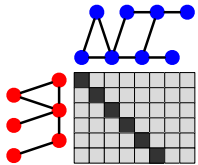
Graph matching and partitioning

Apply the Gromov-Wasserstein distance as objective functions

Matching via learning optimal transport

Quadratic assignment problem (QAP):

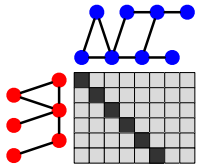
$$\max_{\boldsymbol{P}\in\mathcal{P}} \langle \boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Y^\top, \boldsymbol{P} \rangle,$$
$$\mathcal{P} = \{\boldsymbol{P} \in \{0,1\}^{|\mathcal{V}_X|\times|\mathcal{V}_Y|} \mid \boldsymbol{P}\mathbf{1} = \mathbf{1}, \boldsymbol{P}^\top\mathbf{1} \leq \mathbf{1}\}.$$



Matching via learning optimal transport

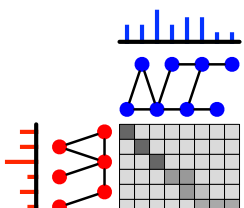
Quadratic assignment problem (QAP):

$$\max_{\boldsymbol{P}\in\mathcal{P}} \langle \boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Y^\top, \boldsymbol{P} \rangle,$$
$$\mathcal{P} = \{\boldsymbol{P} \in \{0,1\}^{|\mathcal{V}_X|\times|\mathcal{V}_Y|} \mid \boldsymbol{P}\mathbf{1} = \mathbf{1}, \boldsymbol{P}^\top\mathbf{1} \leq \mathbf{1}\}.$$



Gromov-Wasserstein distance (GWD):

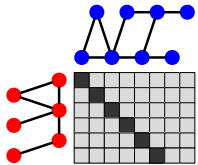
$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top, \boldsymbol{T} \rangle,$$
$$\Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y) = \{\boldsymbol{T} \geq \mathbf{0} \mid \boldsymbol{T}\mathbf{1} = \boldsymbol{\mu}_X,\ \boldsymbol{T}^\top\mathbf{1} = \boldsymbol{\mu}_Y\}$$



Matching via learning optimal transport

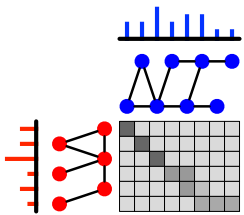
Quadratic assignment problem (QAP):

$$\max_{\boldsymbol{P}\in\mathcal{P}} \langle \boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Y^\top, \boldsymbol{P} \rangle,$$
$$\mathcal{P} = \{\boldsymbol{P} \in \{0,1\}^{|\mathcal{V}_X|\times|\mathcal{V}_Y|} \mid \boldsymbol{P}\mathbf{1} = \mathbf{1}, \boldsymbol{P}^\top\mathbf{1} \le \mathbf{1}\}.$$



Gromov-Wasserstein distance (GWD):

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top, \boldsymbol{T} \rangle,$$
$$\Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y) = \{\boldsymbol{T} \ge \mathbf{0} \mid \boldsymbol{T}\mathbf{1} = \boldsymbol{\mu}_X,\ \boldsymbol{T}^\top\mathbf{1} = \boldsymbol{\mu}_Y\}$$



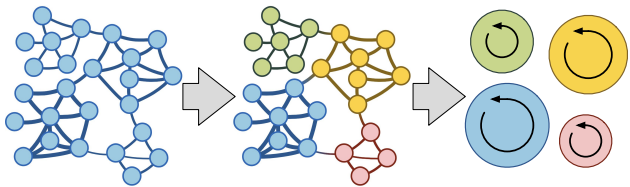
- Conditional prob. $\boldsymbol{P}^* = \frac{\boldsymbol{T}^*}{\boldsymbol{\mu}_X \mathbf{1}^\top} = [P^*(j|i)]$.
- For each node $i \in \mathcal{V}_X$, $j^* = \arg\max_j P^*(j|i)$.

Partitioning is also matching

Partitioning is also matching

Modularity maximization principle

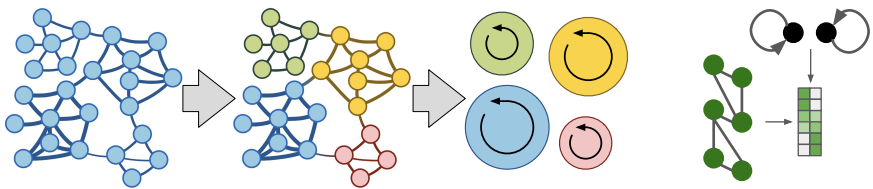
- Dense internal edges + sparse external edges.



Partitioning is also matching

Modularity maximization principle

- Dense internal edges + sparse external edges.



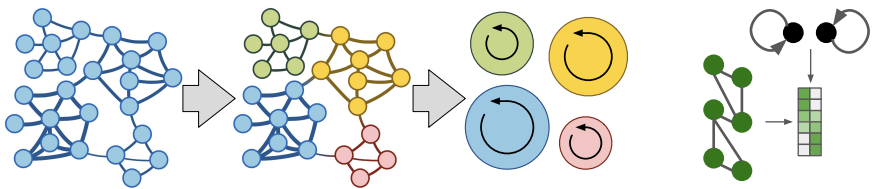
A GWD-based solution *[Xu, et al., NeurIPS 2019]*:

- $\boldsymbol{T}^* \in \mathbb{R}^{|\mathcal{V}| \times N} \leftarrow d_{gw}(G, G_{iso})$
- $G_{iso}(\mathcal{V}_{iso}, \frac{1}{N}\mathbf{1}_N, \boldsymbol{I}_{N \times N})$

Partitioning is also matching

Modularity maximization principle

- Dense internal edges + sparse external edges.



A GWD-based solution *[Xu, et al., NeurIPS 2019]*:

- $\boldsymbol{T}^* \in \mathbb{R}^{|\mathcal{V}| \times N} \leftarrow d_{gw}(G, G_{iso})$
- $G_{iso}(\mathcal{V}_{iso}, \frac{1}{N}\mathbf{1}_N, \boldsymbol{I}_{N \times N})$
- For each node $i \in G$, its cluster is $j^* = \arg\max_j T^*_{ij}$

Proposed algorithms

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T} \rangle$$

Proposed algorithms

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle$$

Proximal Gradient Algorithm *[Xu, et al., ICML 2019]*

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \underbrace{\langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T}^{(m)} \boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle}_{\text{GW term}} + \gamma \underbrace{\mathsf{KL}(\boldsymbol{T} \| \boldsymbol{T}^{(m)})}_{\text{Proximal term}}$$

Proposed algorithms

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle$$

Proximal Gradient Algorithm *[Xu, et al., ICML 2019]*

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \underbrace{\langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T}^{(m)} \boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle}_{\text{GW term}} + \gamma \underbrace{\mathsf{KL}(\boldsymbol{T}\|\boldsymbol{T}^{(m)})}_{\text{Proximal term}}$$

$$\rightarrow \underbrace{\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \underbrace{\langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T}^{(m)} \boldsymbol{D}_Y^\top - \gamma \log \boldsymbol{T}^{(m)}}_{\text{Constant}}, \boldsymbol{T}\rangle - \gamma \underbrace{\mathsf{H}(\boldsymbol{T})}_{\text{Entropy}}}_{\text{Sinkhorn distance}} \tag{2}$$

Proposed algorithms

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle$$

Proximal Gradient Algorithm *[Xu, et al., ICML 2019]*

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \underbrace{\langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T}^{(m)} \boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle}_{\text{GW term}} + \gamma \underbrace{\mathsf{KL}(\boldsymbol{T} \| \boldsymbol{T}^{(m)})}_{\text{Proximal term}}$$

$$\rightarrow \underbrace{\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \underbrace{\langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T}^{(m)} \boldsymbol{D}_Y^\top - \gamma \log \boldsymbol{T}^{(m)}}_{\text{Constant}}, \boldsymbol{T}\rangle - \gamma \underbrace{\mathsf{H}(\boldsymbol{T})}_{\text{Entropy}}}_{\text{Sinkhorn distance}} \tag{2}$$

Bregman ADMM *[Xu, AAAI 2020]*

$$\underbrace{\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot),\ \boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\mu}_Y),\ \boldsymbol{T}=\boldsymbol{S}}}_{\text{Decoupling constraints by Aux.}} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{S} \boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle$$

Proposed algorithms

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle$$

Proximal Gradient Algorithm *[Xu, et al., ICML 2019]*

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \underbrace{\langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T}^{(m)} \boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle}_{\text{GW term}} + \gamma \underbrace{\mathsf{KL}(\boldsymbol{T}\|\boldsymbol{T}^{(m)})}_{\text{Proximal term}}$$

$$\rightarrow \underbrace{\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \underbrace{\boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T}^{(m)} \boldsymbol{D}_Y^\top - \gamma \log \boldsymbol{T}^{(m)}}_{\text{Constant}}, \boldsymbol{T}\rangle - \gamma \underbrace{\mathsf{H}(\boldsymbol{T})}_{\text{Entropy}}}_{\text{Sinkhorn distance}} \tag{2}$$

Bregman ADMM *[Xu, AAAI 2020]*

$$\underbrace{\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot),\ \boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\mu}_Y),\ \boldsymbol{T}=\boldsymbol{S}}}_{\text{Decoupling constraints by Aux.}} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{S} \boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle$$

$$\rightarrow \min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot),\boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\mu}_Y),\boldsymbol{Z}} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{S} \boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle + \underbrace{\overbrace{\langle \boldsymbol{Z}, \boldsymbol{T}-\boldsymbol{S}\rangle}^{\text{Dual}} + \overbrace{\gamma \mathsf{B}(\boldsymbol{T},\boldsymbol{S})}^{\text{Bregman Div.}}}_{\text{Augmented Lagrangian}} \tag{3}$$

Further analysis

Convergence

- $\lim_{m\to\infty} \boldsymbol{T}^{(m)}$ is a stationary point.
- Linear convergence.

Further analysis

Convergence

- $\lim_{m\to\infty} \boldsymbol{T}^{(m)}$ is a stationary point.
- Linear convergence.
- PGA requires fewer iterations.
- B-ADMM works better on directed graphs.

Further analysis

Convergence

- $\lim_{m\to\infty} \boldsymbol{T}^{(m)}$ is a stationary point.
- Linear convergence.
- PGA requires fewer iterations.
- B-ADMM works better on directed graphs.

Computational complexity per iteration

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle$$

- $\boldsymbol{D}_X$, $\boldsymbol{D}_Y$ are dense distance/kernel matrices: $\mathcal{O}(V^3)$.

Further analysis

Convergence

- $\lim_{m\to\infty} \boldsymbol{T}^{(m)}$ is a stationary point.
- Linear convergence.
- PGA requires fewer iterations.
- B-ADMM works better on directed graphs.

Computational complexity per iteration

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle$$

- $\boldsymbol{D}_X$, $\boldsymbol{D}_Y$ are dense distance/kernel matrices: $\mathcal{O}(V^3)$.
- $\boldsymbol{D}_X$, $\boldsymbol{D}_Y$ are adjacency matrices: $\mathcal{O}(VE)$.

Further analysis

Convergence

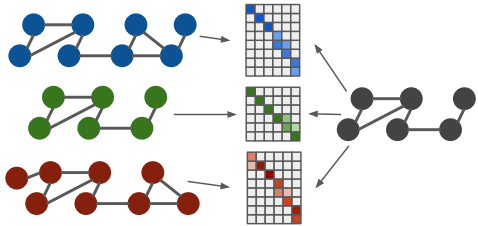
- $\lim_{m\to\infty} \boldsymbol{T}^{(m)}$ is a stationary point.
- Linear convergence.
- PGA requires fewer iterations.
- B-ADMM works better on directed graphs.

Computational complexity per iteration

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle$$

- $\boldsymbol{D}_X$, $\boldsymbol{D}_Y$ are dense distance/kernel matrices: $\mathcal{O}(V^3)$.
- $\boldsymbol{D}_X$, $\boldsymbol{D}_Y$ are adjacency matrices: $\mathcal{O}(VE)$.
- When $V = |\mathcal{V}_X| \gg |\mathcal{V}_Y| = N$ (graph partitioning): $\mathcal{O}(N(E+V))$.

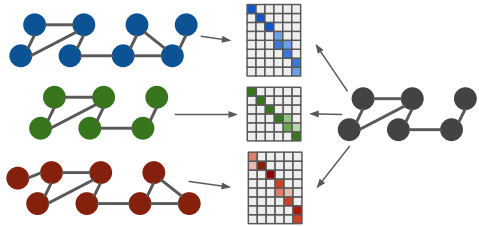
Large-scale matching based on GW barycenters



Given $\{G_k\}_{k=1}^K$, $K \geq 2$, their GW barycenter is defined as

$$\underbrace{B_{gw}(\bar{\mathcal{V}}, \bar{\boldsymbol{\mu}}, \boldsymbol{B}^*)}_{\text{Barycenter graph}}, \underbrace{\{\boldsymbol{T}_k^*\}_{k=1}^K}_{\text{OT matrices}} := \arg\min_B \sum_{k=1}^K \lambda_k d_{gw}(B, G_k), \tag{4}$$

Large-scale matching based on GW barycenters



Given $\{G_k\}_{k=1}^K$, $K \geq 2$, their GW barycenter is defined as

$$\underbrace{B_{gw}(\bar{\mathcal{V}}, \bar{\boldsymbol{\mu}}, \boldsymbol{B}^*)}_{\text{Barycenter graph}}, \underbrace{\{\boldsymbol{T}_k^*\}_{k=1}^K}_{\text{OT matrices}} := \arg\min_B \sum_{k=1}^K \lambda_k d_{gw}(B, G_k), \tag{4}$$

Learn $\{\boldsymbol{T}_k^*\}_{k=1}^K$ and $\boldsymbol{B}^*$ via **alternating optimization**.

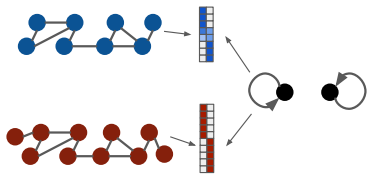
$$\boldsymbol{B}^* = \frac{1}{\bar{\boldsymbol{\mu}}\bar{\boldsymbol{\mu}}^\top} \sum_{k=1}^K \lambda_k (\boldsymbol{T}_k^*)^\top \boldsymbol{D}_k \boldsymbol{T}_k^* \tag{5}$$

Large-scale matching based on GW barycenters

Co-partition two graphs:

$$\boldsymbol{B}^*, \boldsymbol{T}_X^*, \boldsymbol{T}_Y^* = \arg\min \frac{|\mathcal{V}_X|}{|\mathcal{V}_X| + |\mathcal{V}_Y|} d_{gw}(B, G_X) + \frac{|\mathcal{V}_Y|}{|\mathcal{V}_X| + |\mathcal{V}_Y|} d_{gw}(B, G_Y)$$

Initialize the barycenter graph by a disconnected graph.

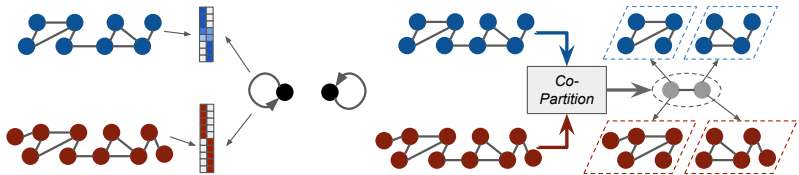


Large-scale matching based on GW barycenters

Co-partition two graphs:

$$\boldsymbol{B}^*, \boldsymbol{T}_X^*, \boldsymbol{T}_Y^* = \arg\min \frac{|\mathcal{V}_X|}{|\mathcal{V}_X| + |\mathcal{V}_Y|} d_{gw}(B, G_X) + \frac{|\mathcal{V}_Y|}{|\mathcal{V}_X| + |\mathcal{V}_Y|} d_{gw}(B, G_Y)$$

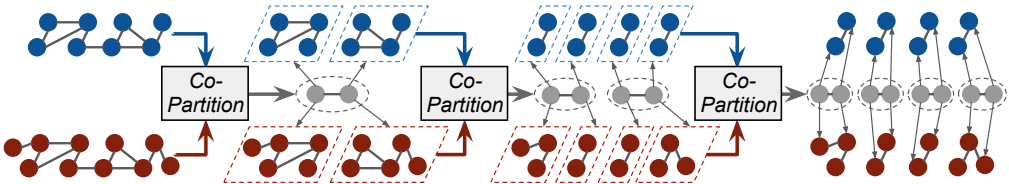
Initialize the barycenter graph by a disconnected graph.



Computational complexity: $\mathcal{O}(2(V + E))$

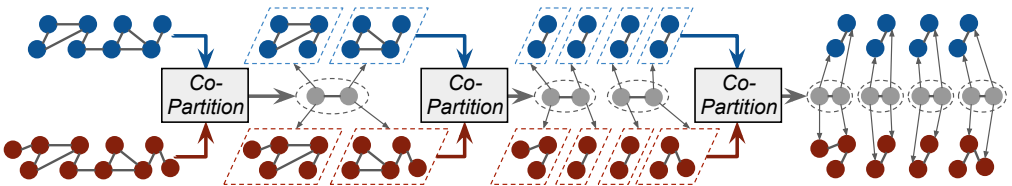
Large-scale matching based on GW barycenters

A "Divide and Conquer" strategy based on recursive co-partitioning *[Xu, et al., NeurIPS 2019]*



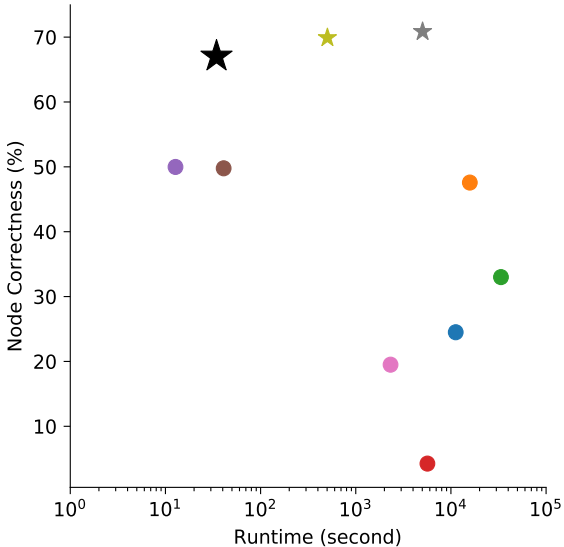
Large-scale matching based on GW barycenters

A "Divide and Conquer" strategy based on recursive co-partitioning *[Xu, et al., NeurIPS 2019]*

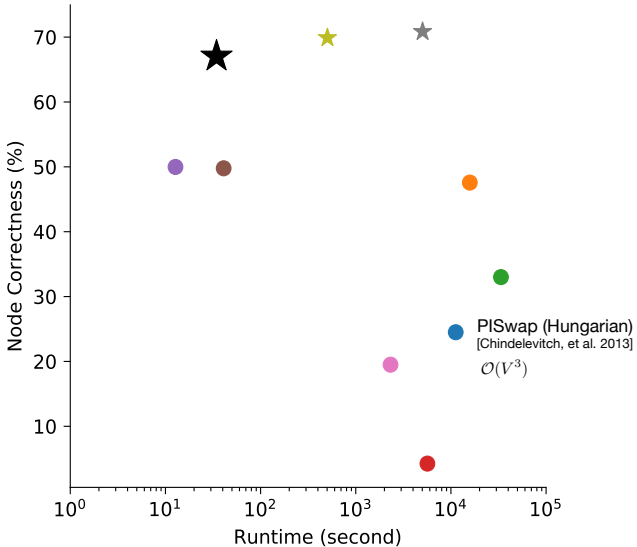


Computational complexity: $\mathcal{O}((E+V)\log V)$

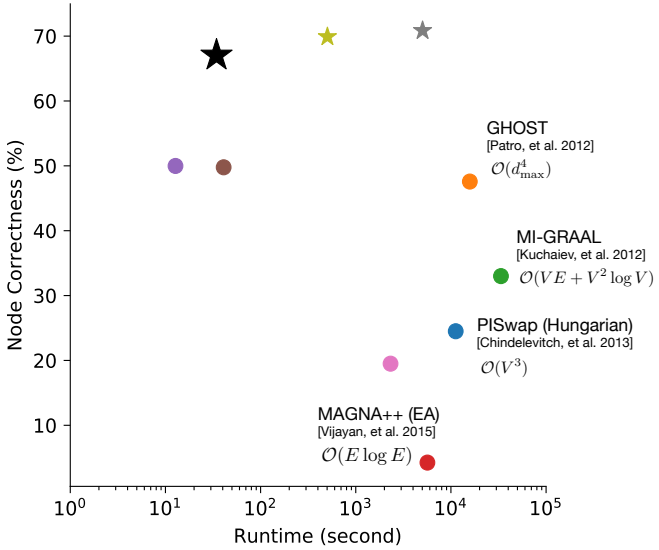
Matching synthetic graphs



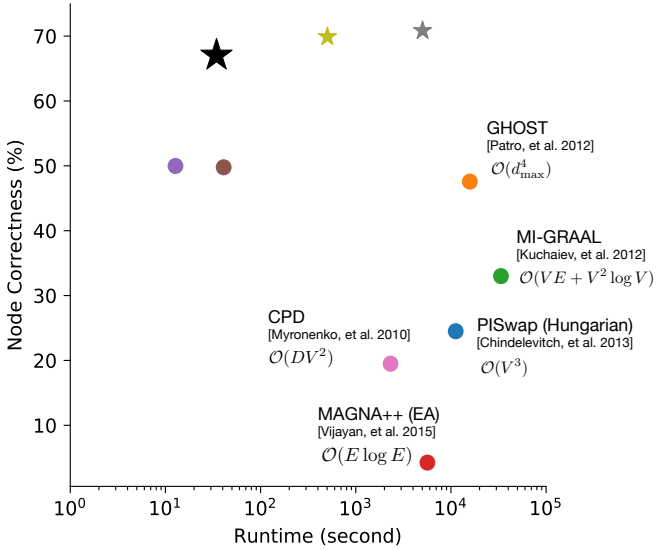
Matching synthetic graphs



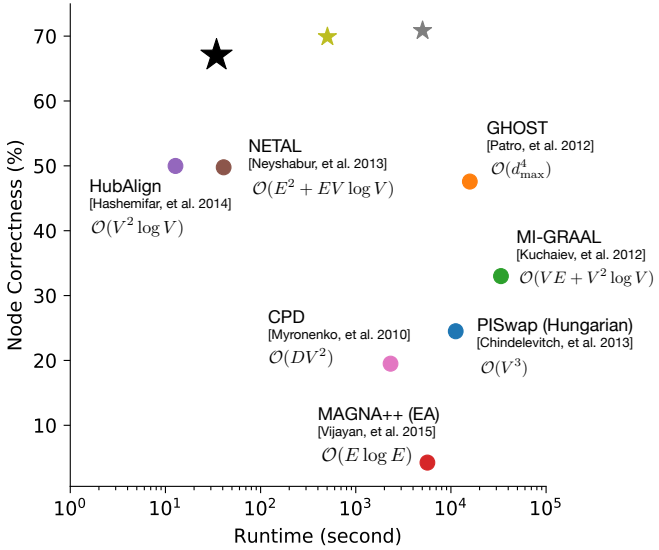
Matching synthetic graphs



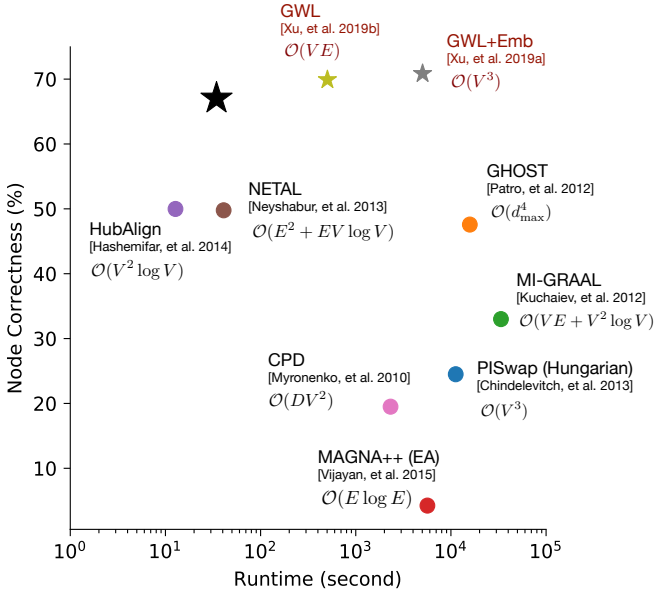
Matching synthetic graphs



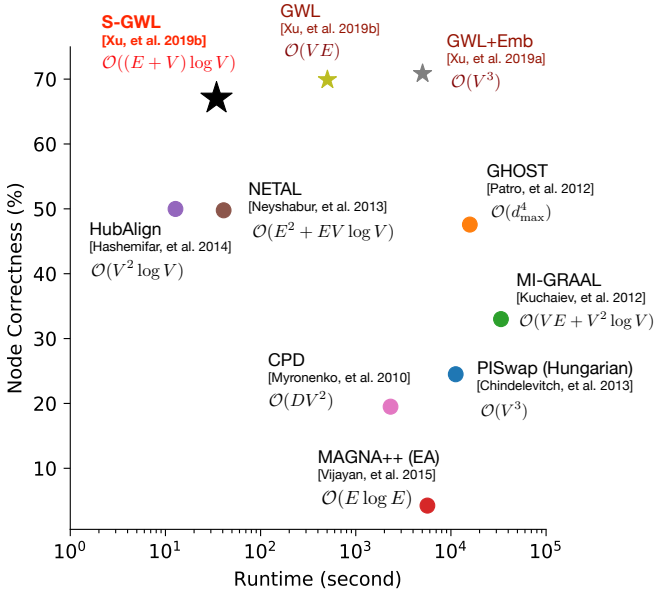
Matching synthetic graphs



Matching synthetic graphs

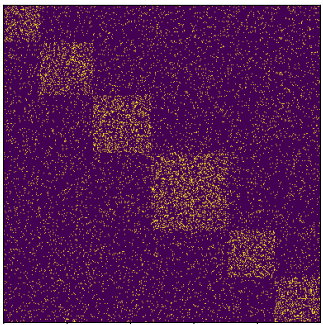


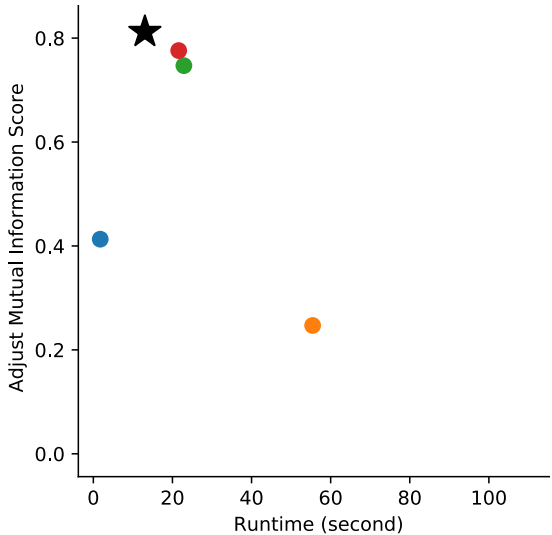
Matching synthetic graphs



Partitioning synthetic graphs

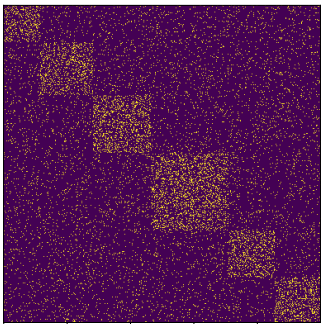
$V = 4,000$
$p_{\text{within}} = 0.2$
$p_{\text{across}} = 0.05$

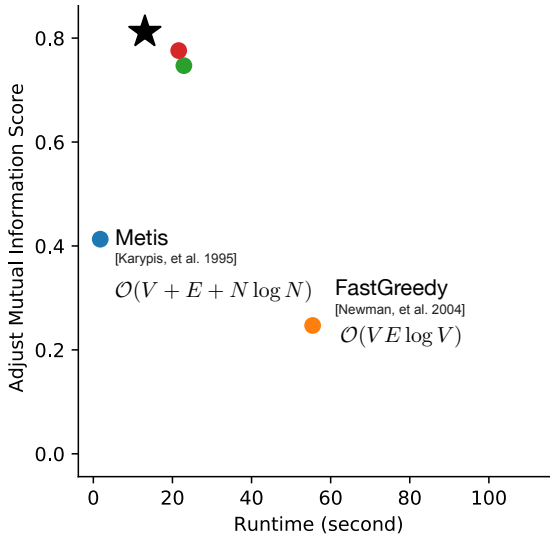




Partitioning synthetic graphs

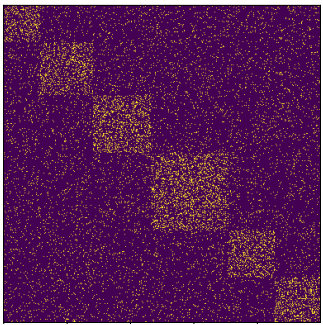
$V = 4,000$
$p_{\text{within}} = 0.2$
$p_{\text{across}} = 0.05$

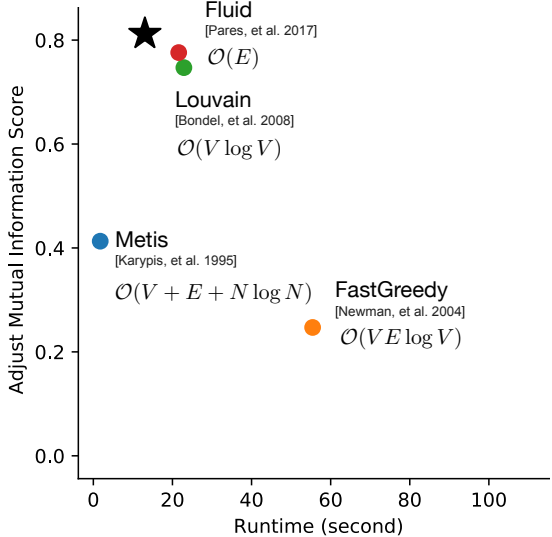




Partitioning synthetic graphs

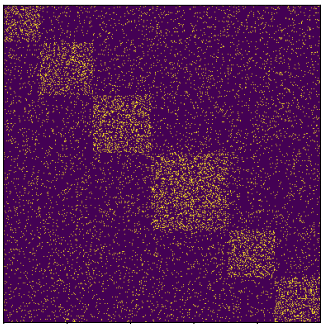
$V = 4,000$
$p_{\text{within}} = 0.2$
$p_{\text{across}} = 0.05$

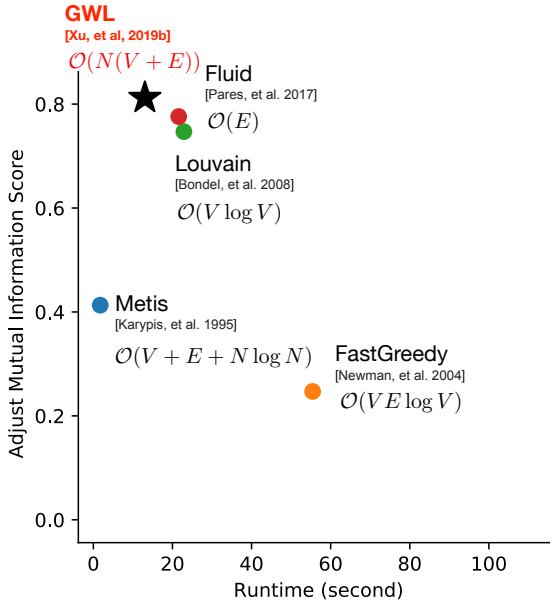




Partitioning synthetic graphs

$V = 4,000$
$p_{\text{within}} = 0.2$
$p_{\text{across}} = 0.05$



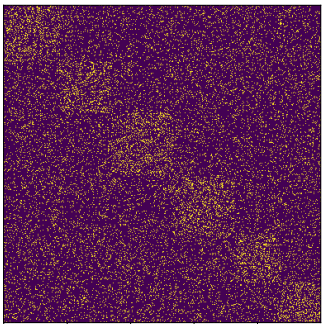


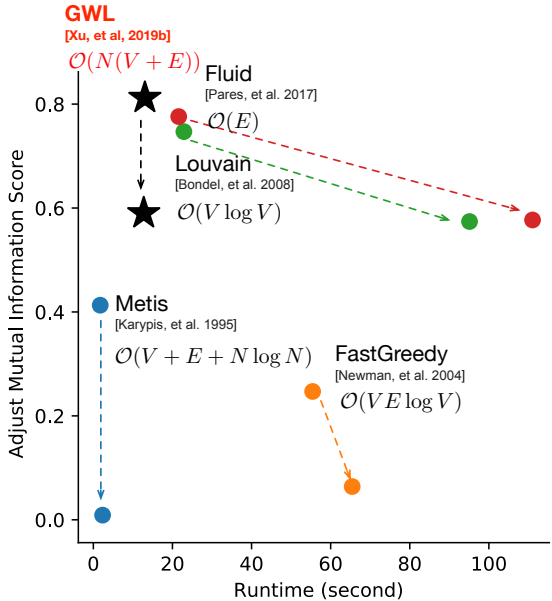
Partitioning synthetic graphs

$V = 4,000$

$p_{\text{within}} = 0.2$

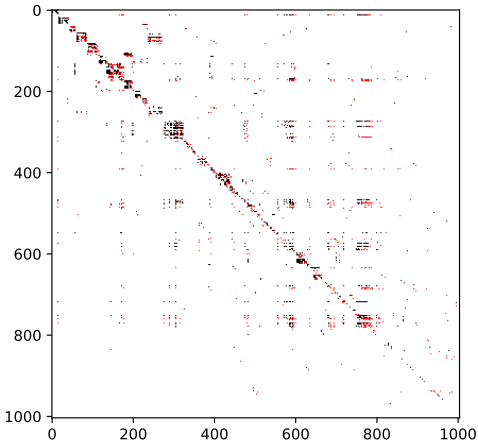
$p_{\text{across}} = 0.1$





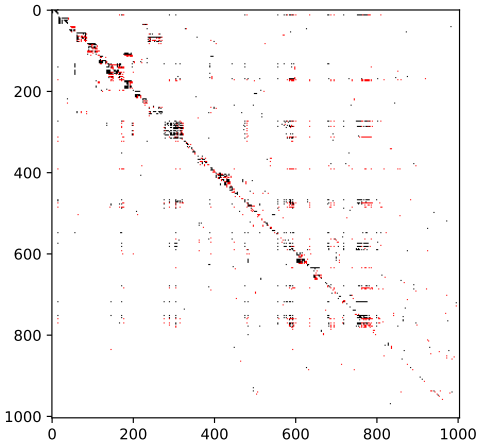
Real-world PPI network alignment

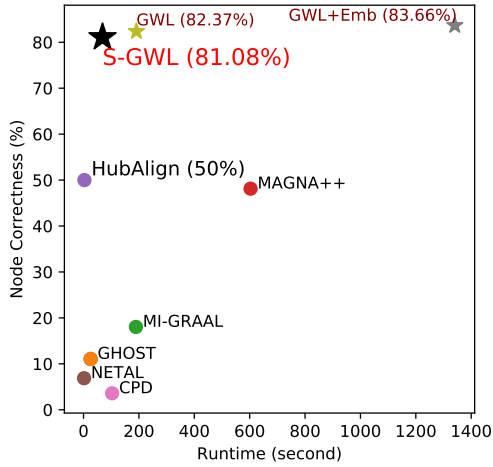
Yeast PPI $\leftrightarrow$ Yeast PPI + 5% LC edges



Real-world PPI network alignment

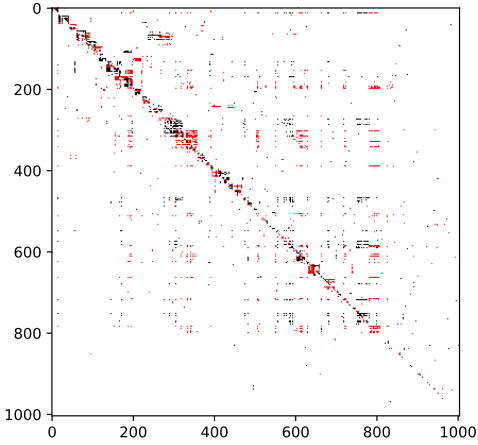
Yeast PPI $\leftrightarrow$ Yeast PPI + 5% LC edges

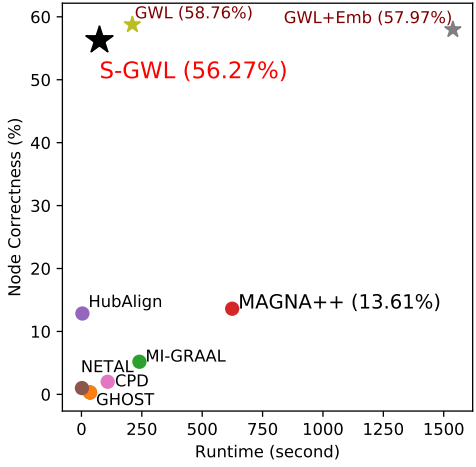




Real-world PPI network alignment

Yeast PPI $\leftrightarrow$ Yeast PPI + **25% LC edges**

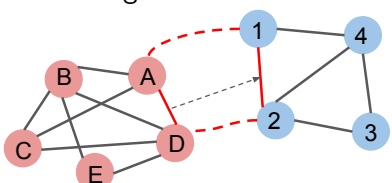




Matching real-world PPI networks

Matching the PPI network of yeast (2,340 proteins) to that of human (9,141 proteins)

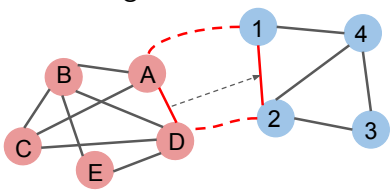
Edge correctness:

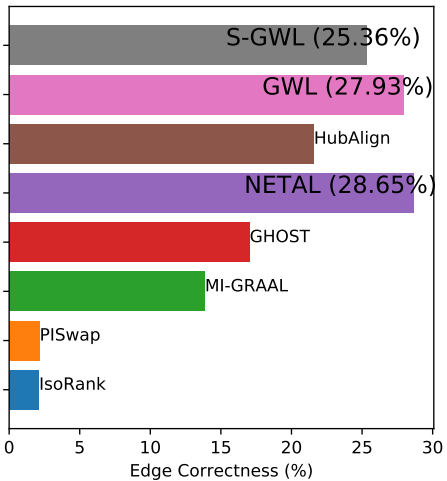


Matching real-world PPI networks

Matching the PPI network of yeast (2,340 proteins) to that of human (9,141 proteins)

Edge correctness:

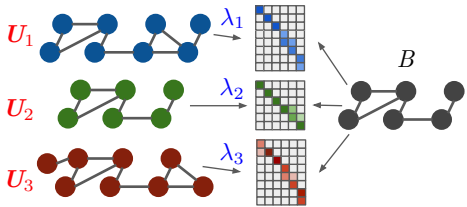




Graph representation

Extend existing representation models based on Gromov-Wasserstein distance

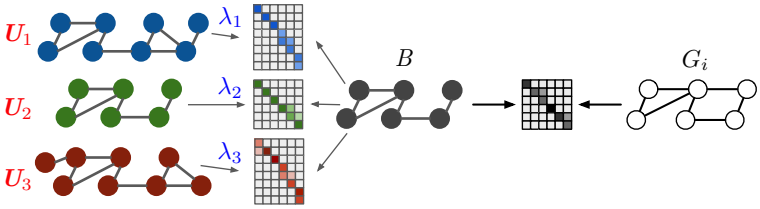
Gromov-Wasserstein factorization model



$$B_{gw}(\boldsymbol{U}_{1:K}, \boldsymbol{\lambda}) := \arg\min_B \sum_{k=1}^{K} \lambda_k d_{gw}(B, G_k(\boldsymbol{U}_k)). \tag{6}$$

- $\{G_k(\boldsymbol{U}_k)\}_{k=1}^K$: a set of graph bases.
- $\boldsymbol{\lambda} = [\lambda_k] \in \Delta^{K-1}$: the coefficients of the graph basis.

Gromov-Wasserstein factorization model



$$B_{gw}(\boldsymbol{U}_{1:K}, \boldsymbol{\lambda}) := \arg\min_B \sum_{k=1}^{K} \lambda_k d_{gw}(B, G_k(\boldsymbol{U}_k)). \tag{6}$$

- $\{G_k(\boldsymbol{U}_k)\}_{k=1}^K$: a set of graph bases.
- $\boldsymbol{\lambda} = [\lambda_k] \in \Delta^{K-1}$: the coefficients of the graph basis.
- Estimate each graph by a GW barycenter graph *[Xu, AAAI 2020]*:

$$\min_{\mathbf{1} \geq \boldsymbol{U}_{1:K} \geq \mathbf{0},\ \boldsymbol{\lambda}_{1:I} \in \Delta^{K-1}} \sum_{i=1}^{I} d_{gw}(B_{gw}(\boldsymbol{U}_{1:K}, \overbrace{\boldsymbol{\lambda}_i}^{\text{Rep. of } G_i}), G_i). \tag{7}$$

Gromov-Wasserstein factorization model

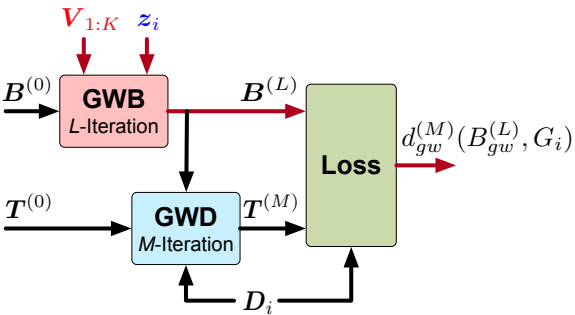
Reparameterize the problem to an unconstrained optimization problem:

$$\min_{\boldsymbol{V}_{1:K},\ \boldsymbol{z}_{1:I}} \sum_{i=1}^{I} d_{gw}(B_{gw}(\underbrace{\sigma(\boldsymbol{V}_{1:K})}_{\boldsymbol{U}_{1:K}}, \underbrace{\mathsf{softmax}(\boldsymbol{z}_i)}_{\boldsymbol{\lambda}_i}),\ G_i). \tag{8}$$

Gromov-Wasserstein factorization model

Reparameterize the problem to an unconstrained optimization problem:

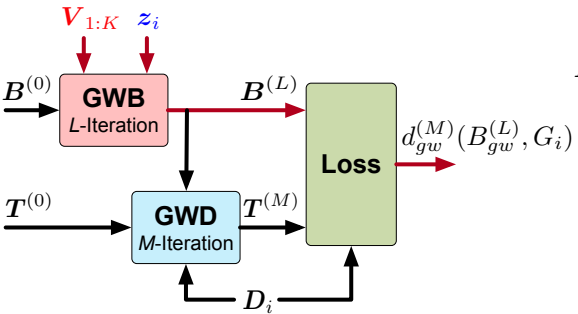
$$\min_{\boldsymbol{V}_{1:K},\ \boldsymbol{z}_{1:I}} \sum_{i=1}^{I} d_{gw}(B_{gw}(\underbrace{\sigma(\boldsymbol{V}_{1:K})}_{\boldsymbol{U}_{1:K}}, \underbrace{\mathsf{softmax}(\boldsymbol{z}_i)}_{\boldsymbol{\lambda}_i}),\ G_i). \tag{8}$$

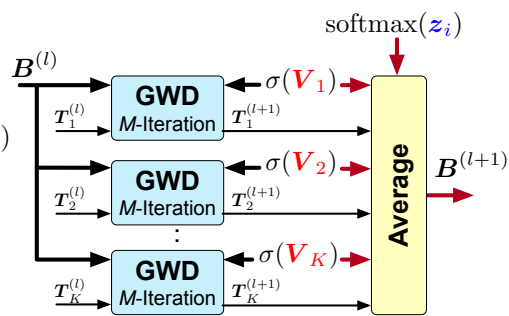


Gromov-Wasserstein factorization model

Reparameterize the problem to an unconstrained optimization problem:

$$\min_{\boldsymbol{V}_{1:K},\ \boldsymbol{z}_{1:I}} \sum_{i=1}^{I} d_{gw}(B_{gw}(\underbrace{\sigma(\boldsymbol{V}_{1:K})}_{\boldsymbol{U}_{1:K}}, \underbrace{\mathrm{softmax}(\boldsymbol{z}_i)}_{\boldsymbol{\lambda}_i}),\ G_i). \tag{8}$$





From transductive to inductive

The GWF is a transductive model:

$$\boldsymbol{z}_{\mathsf{new}} = \arg\min_{\boldsymbol{z}} d_{gw}(B_{gw}(\sigma(\boldsymbol{V}_{1:K}), \mathsf{softmax}(\boldsymbol{z})), G_{\mathsf{new}}). \tag{9}$$

From transductive to inductive

The GWF is a transductive model:

$$\boldsymbol{z}_{\mathsf{new}} = \arg\min_{\boldsymbol{z}} d_{gw}(B_{gw}(\sigma(\boldsymbol{V}_{1:K}), \mathsf{softmax}(\boldsymbol{z})), G_{\mathsf{new}}). \tag{9}$$

Make an **inductive GWF** model *[Xu, et al. ICML 2020 submission]*:

$$\min_{\boldsymbol{V}_{1:K},\, \theta} \sum_{i=1}^{I} d_{gw}(B_{gw}(\sigma(\boldsymbol{V}_{1:K}), \mathsf{softmax}(\underbrace{\mathsf{GCN}_\theta(G_i)}_{\boldsymbol{z}_i})), G_i). \tag{10}$$

From transductive to inductive

The GWF is a transductive model:

$$\boldsymbol{z}_{\text{new}} = \arg\min_{\boldsymbol{z}} d_{gw}(B_{gw}(\sigma(\boldsymbol{V}_{1:K}), \text{softmax}(\boldsymbol{z})), G_{\text{new}}). \tag{9}$$

Make an **inductive GWF** model *[Xu, et al. ICML 2020 submission]*:

$$\min_{\boldsymbol{V}_{1:K},\ \theta} \sum_{i=1}^{I} d_{gw}(B_{gw}(\sigma(\boldsymbol{V}_{1:K}), \text{softmax}(\underbrace{\text{GCN}_\theta(G_i)}_{\boldsymbol{z}_i})), G_i). \tag{10}$$

$$\boldsymbol{z}_{\text{new}} = \text{GCN}_\theta(G_{\text{new}}). \tag{11}$$

Experiments on molecule clustering

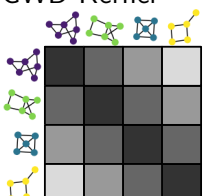
Experiments on molecule clustering

- AIDS: 2,000 compounds active/inactive to anti-HIV
- PROTEIN: 1,113 enzymatic/non-enzymatic proteins

Experiments on molecule clustering

- AIDS: 2,000 compounds active/inactive to anti-HIV
- PROTEIN: 1,113 enzymatic/non-enzymatic proteins

GWD Kernel



GWD+Kmeans

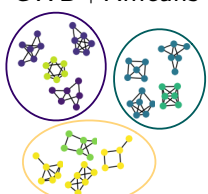


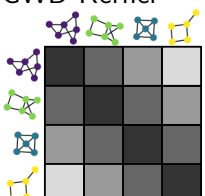
Table: Comparisons on clustering accuracy (%)

Method	AIDS	PROTEIN
GWD Kernel + SC	91.0±0.7	66.4±0.8
GWD + Kmeans	95.2±0.9	64.7±1.1

Experiments on molecule clustering

- AIDS: 2,000 compounds active/inactive to anti-HIV
- PROTEIN: 1,113 enzymatic/non-enzymatic proteins

GWD Kernel



GWD+Kmeans

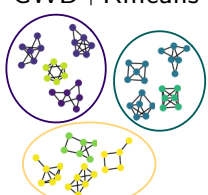


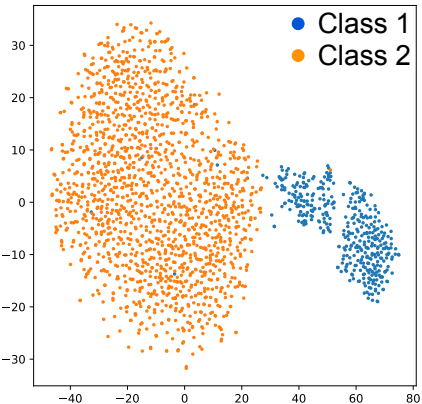
Table: Comparisons on clustering accuracy (%)

Method	AIDS	PROTEIN
GWD Kernel + SC	91.0±0.7	66.4±0.8
GWD + Kmeans	95.2±0.9	64.7±1.1
GWF + Kmeans	**99.5±0.4**	**70.7±0.7**
I-GWF + Kmeans	**99.2±0.5**	**73.3±0.6**

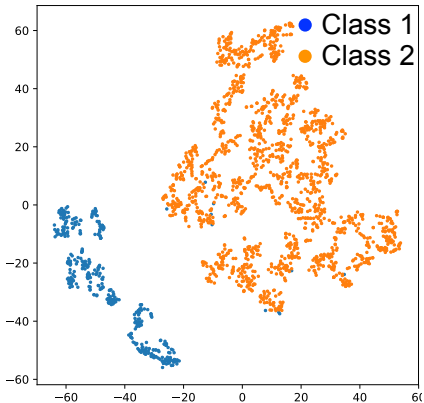
Visualization of coefficient vectors

AIDS

GWF



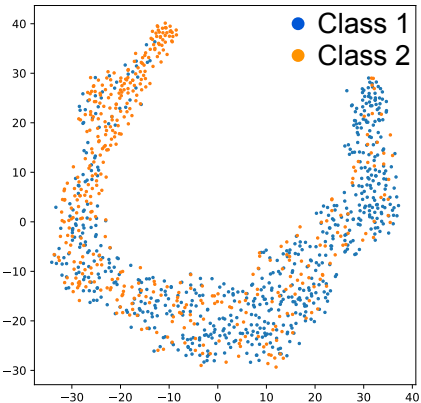
I-GWF



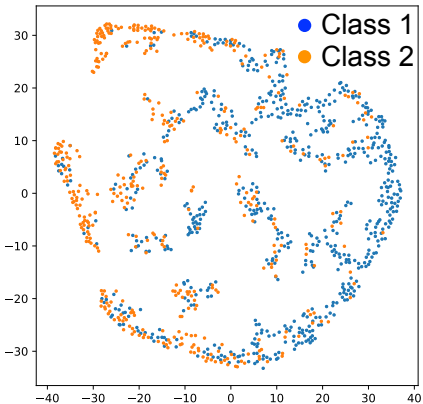
Visualization of coefficient vectors

PROTEIN

GWF



I-GWF



Semi-supervised learning of the I-GWF model

$$\min_{\boldsymbol{V}_{1:K},\theta,\phi} \underbrace{\sum_{i=1}^{I} d_{gw}(B_{gw}(\sigma(\boldsymbol{V}_{1:K}), \mathsf{softmax}(\mathsf{GCN}_\theta(G_i))), G_i) +}_{\text{Reconstruction loss for both unlabeled and labeled graphs}}$$

$$\underbrace{\sum_{i\in\mathcal{G}_{\text{labeled}}} \mathsf{loss}(\overbrace{\phi(\mathsf{GCN}_\theta(G_i))}^{\text{classifier}}, l_i)\,.}_{\text{MLE for labeled graphs}} \tag{12}$$

Experiments on molecule classification

Experiments on molecule classification

- MUTAG: 188 compounds with high/low mutagenicity
- PTC-MR: 344 drugs with/without rodent carcinogenicity

Experiments on molecule classification

- MUTAG: 188 compounds with high/low mutagenicity
- PTC-MR: 344 drugs with/without rodent carcinogenicity

Table: Comparison on classification accuracy (%).

Category	Method	MUTAG	PTC-MR
Graph kernel	Random Walk	83.71	57.85
	Shortest Path	85.22	58.24
	Graphlet	81.66	57.26
	Weisfeiler-Lehman	80.72	57.97
	Deep Graph	87.44	60.08
	Multi-Scale Laplacian	87.94	63.26

Experiments on molecule classification

- MUTAG: 188 compounds with high/low mutagenicity
- PTC-MR: 344 drugs with/without rodent carcinogenicity

Table: Comparison on classification accuracy (%).

Category	Method	MUTAG	PTC-MR
Graph kernel	Random Walk	83.71	57.85
	Shortest Path	85.22	58.24
	Graphlet	81.66	57.26
	Weisfeiler-Lehman	80.72	57.97
	Deep Graph	87.44	60.08
	Multi-Scale Laplacian	87.94	63.26
Graph embedding dimension=512	node2vec	72.63	58.58
	sub2vec	61.05	59.99
	graph2vec	83.15	60.17
	InfoGraph [ICLR 2020]	89.01	61.65

Experiments on molecule classification

- MUTAG: 188 compounds with high/low mutagenicity
- PTC-MR: 344 drugs with/without rodent carcinogenicity

Table: Comparison on classification accuracy (%).

Category	Method	MUTAG	PTC-MR
Graph kernel	Random Walk	83.71	57.85
	Shortest Path	85.22	58.24
	Graphlet	81.66	57.26
	Weisfeiler-Lehman	80.72	57.97
	Deep Graph	87.44	60.08
	Multi-Scale Laplacian	87.94	63.26
Graph embedding dimension=512	node2vec	72.63	58.58
	sub2vec	61.05	59.99
	graph2vec	83.15	60.17
	InfoGraph [ICLR 2020]	89.01	61.65
GWL	I-GWF	**90.11**	**63.58**

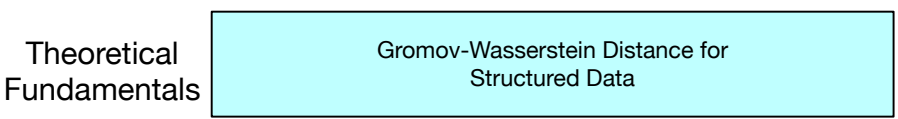
Experiments on molecule classification

- MUTAG: 188 compounds with high/low mutagenicity
- PTC-MR: 344 drugs with/without rodent carcinogenicity

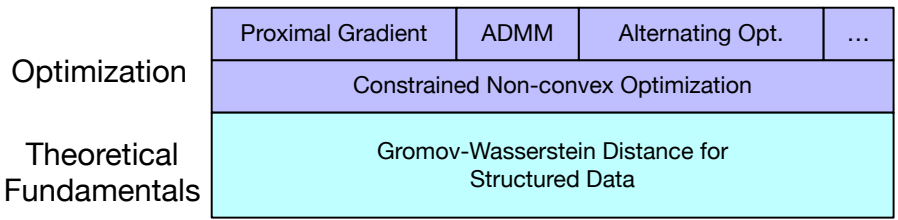
Table: Comparison on classification accuracy (%).

Category	Method	MUTAG	PTC-MR
Graph kernel	Random Walk	83.71	57.85
	Shortest Path	85.22	58.24
	Graphlet	81.66	57.26
	Weisfeiler-Lehman	80.72	57.97
	Deep Graph	87.44	60.08
	Multi-Scale Laplacian	87.94	63.26
Graph embedding dimension=512	node2vec	72.63	58.58
	sub2vec	61.05	59.99
	graph2vec	83.15	60.17
	InfoGraph [ICLR 2020]	89.01	61.65
GWL **dimension=15**	I-GWF	**90.11**	**63.58**
	I-GWF + SSL	**91.36**	**65.21**

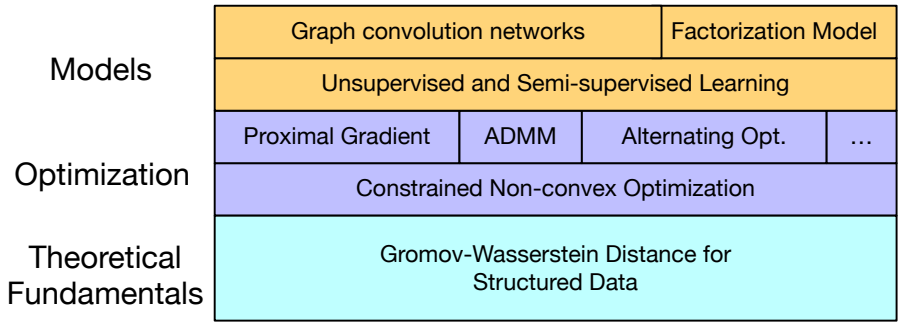
Summary of the GWL framework



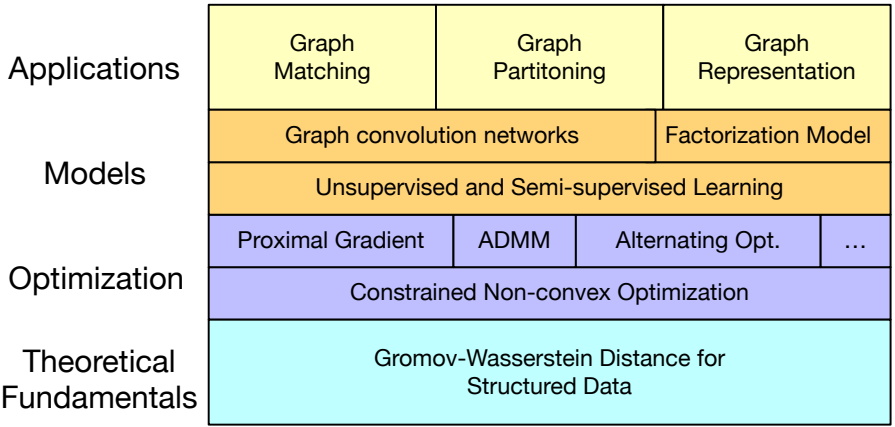
Summary of the GWL framework



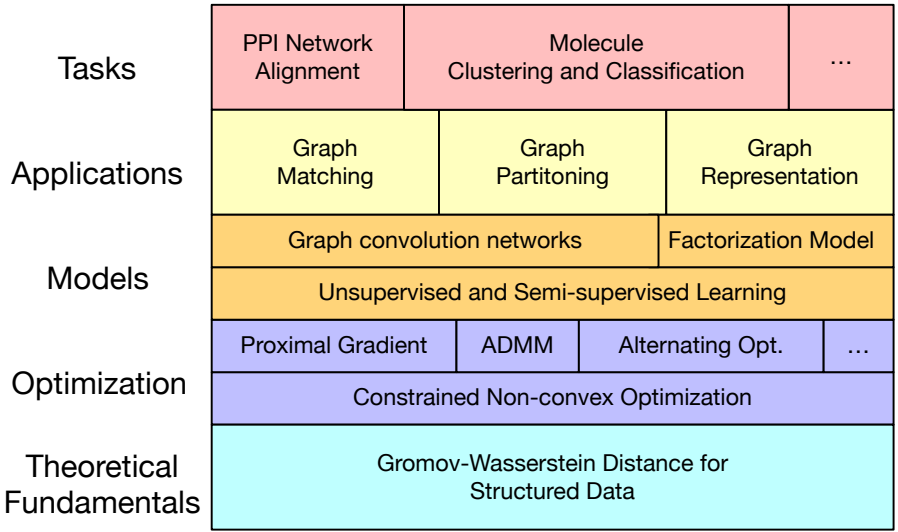
Summary of the GWL framework



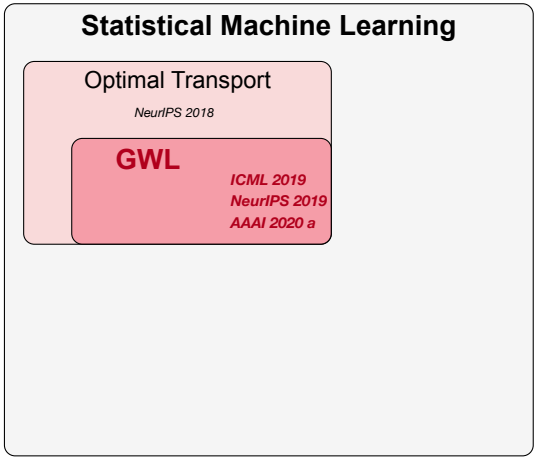
Summary of the GWL framework



Summary of the GWL framework

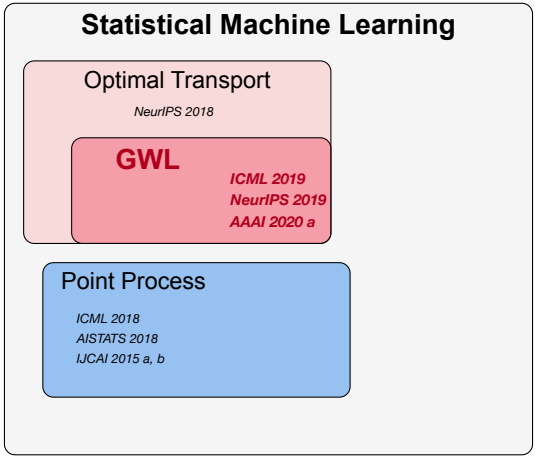


Existing and ongoing work



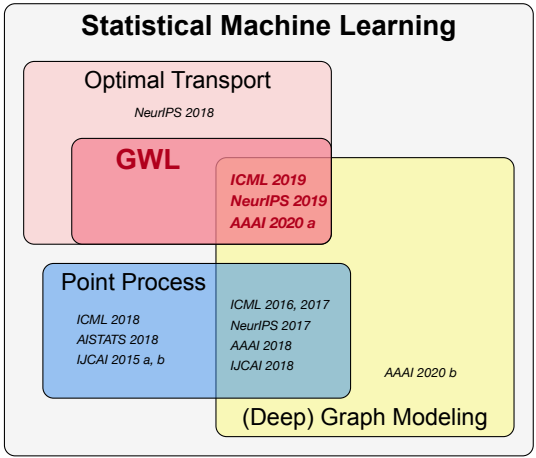
1. **[ICML 2019] Hongteng Xu**, D. Luo, H. Zha, and L. Carin, *Gromov-Wasserstein Learning for Graph Matching and Node Embedding.*
2. **[NeurIPS 2019] Hongteng Xu**, D. Luo, L. Carin, *Scalable Gromov-Wasserstein Learning for Graph Partitioning and Matching.*
3. **[AAAI 2020 a] Hongteng Xu**, *Gromov-Wasserstein Factorization Models for Graph Clustering.*
4. **[NeurIPS 2018] Hongteng Xu**, W. Wang, W. Liu, and L. Carin, *Distilled wasserstein learning for word embedding and topic modeling.*

Existing and ongoing work



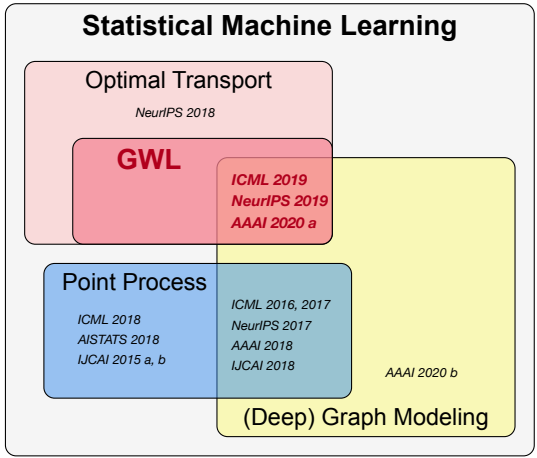
5. **[ICML 2018] Hongteng Xu**, H. Zha, and L. Carin, *Learning registered point processes from idiosyncratic observations.*
6. **[AISTATS 2018] Hongteng Xu**, D. Luo, X. Chen, and L. Carin, *Benefits from superposed hawkes processes.*
7. **[IJCAI 2015 a] Hongteng Xu**, D. Luo, et al., *Multi-Task Multi-Dimensional Hawkes Processes for Modeling Event Sequences.*
8. **[IJCAI 2015 b] Hongteng Xu**, Y. Zhen, and H. Zha, *Trailer Generation via a Point Process-Based Visual Attractiveness Model.*

Existing and ongoing work



9. **[ICML 2016] Hongteng Xu**, M. Farajtabar, and H. Zha, *Learning Granger Causality for Hawkes Processes*.
10. **[ICML 2017] Hongteng Xu**, D. Luo, and H. Zha, *Learning Hawkes processes from short doubly-censored event sequences*.
11. **[NeurIPS 2017] Hongteng Xu** and H. Zha, *A Dirichlet mixture model of Hawkes processes for event sequence clustering*.
12. **[AAAI 2020 b]** W. Wang, **Hongteng Xu**, et al., *Graph-Driven Generative Models for Heterogeneous Multi-Task Learning*.

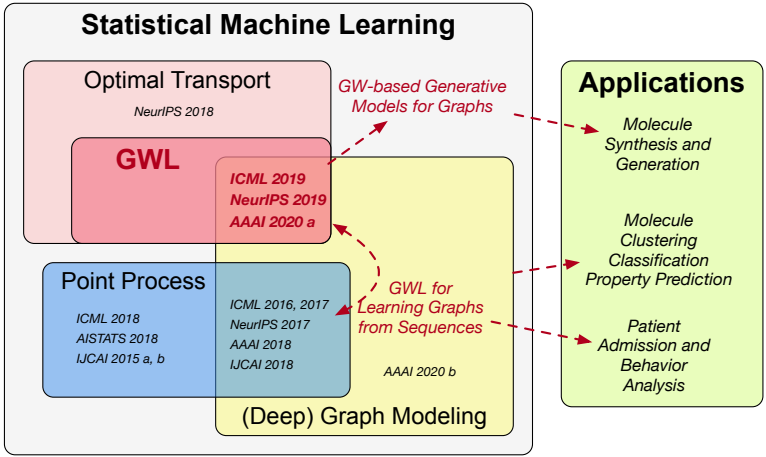
Existing and ongoing work



Miscellaneous (Manifold learning, vision, recommendation, ...)

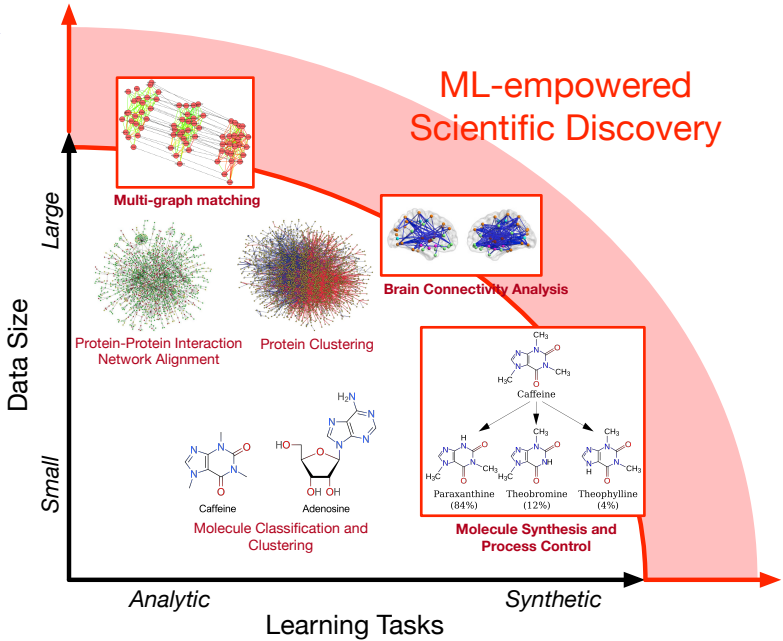
13. **[AAAI 2015] Hongteng Xu**, M. Davenport, et al., *Active Manifold Learning via Gershgorin Circle Guided Sample Selection*.
14. **[ICCV 2015] Hongteng Xu**, Y. Zhou, et al., *Unsupervised trajectory clustering via adaptive multi-kernel-based shrinkage*.
15. **[CVPR 2017] Hongteng Xu**, J. Yan, et al., *Fractal dimension invariant filtering and its CNN-based implementation*.
16. **[WSDM 2018]** X. Chen, **Hongteng Xu**, et al., *Sequential recommendation with user memory networks*.

Future work



17. **[NeurIPS workshop 2019]** D. Luo, **Hongteng Xu**, and L. Carin, *Fused Gromov-Wasserstein Alignment for Hawkes Processes.*
18. **[TKDE 2017] Hongteng Xu**, W. Wu, S. Nemati, et al., *Patient flow prediction via discriminative learning of mutually-correcting processes.*
19. **[MLHC 2018]** M. Engelhard, **Hongteng Xu**, et al., *Predicting smoking events with a time-varying semi-parametric hawkes process model.*
20. **[TechReport 2019]** D. Luo, **Hongteng Xu**, and L. Carin, *Interpretable ICD Code Embeddings with Self-and Mutual-Attention Mechanisms.*

Future work


ML-empowered
Scientific Discovery
Multi-graph matching
Brain Connectivity Analysis
Protein-Protein Interaction
Network Alignment
Protein Clustering
Caffeine
Adenosine
Molecule Classification and
Clustering
Caffeine
Paraxanthine
(84%)
Theobromine
(12%)
Theophylline
(4%)
Molecule Synthesis and
Process Control
Data Size
Large
Small
Analytic
Synthetic
Learning Tasks

Acknowledgements

- Lawrence Carin (Duke)
- Hongyuan Zha (Georgia Tech)
- Jun Liu (Infinia ML)
- Dixin Luo (Duke)
- Mark Davenport (Georgia Tech)
- Ricardo Henao (Duke)
- Le Song (Georgia Tech)
- Shamim Nemati (UCSD)
- Xia Ning (Ohio State)
- Yongfeng Zhang (Rutgers)
- Jiachang Liu (Duke)
- Matthew Engelhard (Duke)

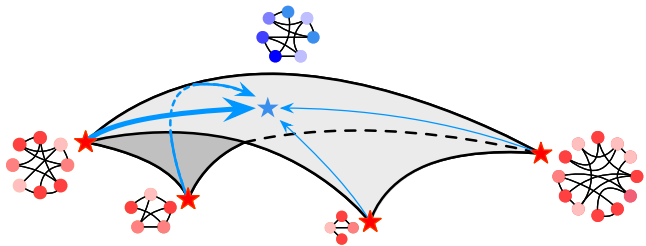
More Information

https://sites.google.com/view/hongtengxu

https://github.com/HongtengXu

hongteng.xu@duke.edu

Synthesize new graphs based on the GWF model



$$\boldsymbol{\lambda}_{\text{new}} \sim P_{\lambda},$$

$$G_{\text{new}} = \arg\min_{B} \sum_{k=1}^{K} \lambda_k^{\text{new}} d_{gw}(B, G_k).$$

Proposed algorithm: Proximal gradient algorithm (PGA)

Obj. $\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)}\langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{T}\boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle$.

Init. $m=0$, $\boldsymbol{T}^{(m)} = \boldsymbol{\mu}_X\boldsymbol{\mu}_Y^\top$.

For $m = 0 : M$

$$\begin{aligned}&\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{T}^{(m)}\boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle + \gamma \underbrace{\mathsf{KL}(\boldsymbol{T}\|\boldsymbol{T}^{(m)})}_{\text{Proximal}} \\ =&\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \underbrace{\boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{T}^{(m)}\boldsymbol{D}_Y^\top - \gamma\log\boldsymbol{T}^{(m)}}_{\text{constant}}, \boldsymbol{T}\rangle - \gamma\mathsf{H}(\boldsymbol{T}).\end{aligned} \tag{13}$$

1. Compute $\boldsymbol{\Phi} = \exp(\frac{\boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{T}^{(m)}\boldsymbol{D}_Y^\top}{\gamma}) \odot \boldsymbol{T}^{(m)}$, $\boldsymbol{a} = \boldsymbol{\mu}_X$.
2. Sinkhorn-Knopp Iterations:
   - Repeat $\boldsymbol{b} = \frac{\boldsymbol{\mu}_Y}{\boldsymbol{\Phi}^\top\boldsymbol{a}}$ and $\boldsymbol{a} = \frac{\boldsymbol{\mu}_X}{\boldsymbol{\Phi}\boldsymbol{b}}$ until convergence
3. $\boldsymbol{T}^{(m+1)} = \mathsf{diag}(\boldsymbol{a})\boldsymbol{\Phi}\mathsf{diag}(\boldsymbol{b})$.

Proposed algorithm: Bregman ADMM (B-ADMM)

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot),\ \boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\mu}_Y),\ \boldsymbol{T}=\boldsymbol{S}} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{S} \boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle, \tag{14}$$

Introduce a dual variable $\boldsymbol{Z}$ and initialize $\boldsymbol{Z}^{(0)} = \boldsymbol{0}$, $\boldsymbol{T}^{(0)} = \boldsymbol{S}^{(0)} = \boldsymbol{\mu}_X \boldsymbol{\mu}_Y^\top$:

Proposed algorithm: Bregman ADMM (B-ADMM)

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot),\ \boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\mu}_Y),\ \boldsymbol{T}=\boldsymbol{S}} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{S} \boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle, \tag{14}$$

Introduce a dual variable $\boldsymbol{Z}$ and initialize $\boldsymbol{Z}^{(0)} = \boldsymbol{0}$, $\boldsymbol{T}^{(0)} = \boldsymbol{S}^{(0)} = \boldsymbol{\mu}_X \boldsymbol{\mu}_Y^\top$:

$$\begin{aligned}\boldsymbol{T}^{(m+1)} =& \arg\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{S}^{(m)} \boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle + \\ & \langle \boldsymbol{Z}^{(m)}, \boldsymbol{T} - \boldsymbol{S}^{(m)}\rangle + \gamma \mathsf{KL}(\boldsymbol{T} \| \boldsymbol{S}^{(m)}) = (\frac{\boldsymbol{\mu}_X}{\boldsymbol{\Phi}_T \boldsymbol{1}_Y} \boldsymbol{1}_Y^\top) \odot \boldsymbol{\Phi}_T,\end{aligned}$$

where $\boldsymbol{\Phi}_T = \exp(-\frac{1}{\gamma}(\boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{S}^{(m)} \boldsymbol{D}_Y^\top + \boldsymbol{Z}^{(m)})) \odot \boldsymbol{S}^{(m)}$

Proposed algorithm: Bregman ADMM (B-ADMM)

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot),\ \boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\mu}_Y),\ \boldsymbol{T}=\boldsymbol{S}} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{S}\boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle, \tag{14}$$

Introduce a dual variable $\boldsymbol{Z}$ and initialize $\boldsymbol{Z}^{(0)} = \boldsymbol{0}$, $\boldsymbol{T}^{(0)} = \boldsymbol{S}^{(0)} = \boldsymbol{\mu}_X\boldsymbol{\mu}_Y^\top$:

$$\begin{aligned}\boldsymbol{T}^{(m+1)} =& \arg\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{S}^{(m)}\boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle + \\ & \langle \boldsymbol{Z}^{(m)}, \boldsymbol{T} - \boldsymbol{S}^{(m)}\rangle + \gamma \mathsf{KL}(\boldsymbol{T}\|\boldsymbol{S}^{(m)}) = (\frac{\boldsymbol{\mu}_X}{\boldsymbol{\Phi}_T\boldsymbol{1}_Y}\boldsymbol{1}_Y^\top) \odot \boldsymbol{\Phi}_T, \\ \text{where } \boldsymbol{\Phi}_T =& \exp(-\frac{1}{\gamma}(\boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{S}^{(m)}\boldsymbol{D}_Y^\top + \boldsymbol{Z}^{(m)})) \odot \boldsymbol{S}^{(m)}\end{aligned}$$

$$\begin{aligned}\boldsymbol{S}^{(m+1)} =& \arg\min_{\boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\mu}_t)} \langle -2\boldsymbol{D}_X^\top\boldsymbol{T}^{(m+1)}\boldsymbol{D}_Y, \boldsymbol{S}\rangle + \langle \boldsymbol{Z}^{(m)}, \boldsymbol{T}^{(m+1)} - \boldsymbol{S}\rangle + \\ & \gamma \mathsf{KL}(\boldsymbol{S}\|\boldsymbol{T}^{(m+1)}) = (\boldsymbol{1}_X(\frac{\boldsymbol{\mu}_Y}{\boldsymbol{\Phi}_S^\top\boldsymbol{1}_X})^\top) \odot \boldsymbol{\Phi}_S, \\ \text{where } \boldsymbol{\Phi}_S =& \exp(\frac{1}{\gamma}(2\boldsymbol{D}_X^\top\boldsymbol{T}^{(m+1)}\boldsymbol{D}_Y + \boldsymbol{Z}^{(m)})) \odot \boldsymbol{T}^{(m)}\end{aligned}$$

Proposed algorithm: Bregman ADMM (B-ADMM)

$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot),\ \boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\mu}_Y),\ \boldsymbol{T}=\boldsymbol{S}} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{S}\boldsymbol{D}_Y^\top,\ \boldsymbol{T}\rangle, \tag{14}$$

Introduce a dual variable $\boldsymbol{Z}$ and initialize $\boldsymbol{Z}^{(0)} = \boldsymbol{0}$, $\boldsymbol{T}^{(0)} = \boldsymbol{S}^{(0)} = \boldsymbol{\mu}_X\boldsymbol{\mu}_Y^\top$:

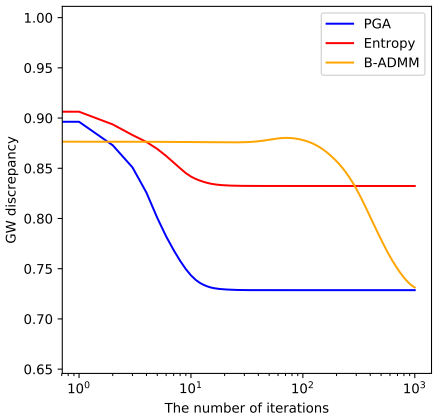
$$\begin{aligned}\boldsymbol{T}^{(m+1)} =& \arg\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\cdot)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{S}^{(m)}\boldsymbol{D}_Y^\top, \boldsymbol{T}\rangle + \\ & \langle \boldsymbol{Z}^{(m)}, \boldsymbol{T} - \boldsymbol{S}^{(m)}\rangle + \gamma \mathsf{KL}(\boldsymbol{T}\|\boldsymbol{S}^{(m)}) = (\frac{\boldsymbol{\mu}_X}{\boldsymbol{\Phi}_T\boldsymbol{1}_Y}\boldsymbol{1}_Y^\top) \odot \boldsymbol{\Phi}_T, \\ \text{where } \boldsymbol{\Phi}_T =& \exp(-\frac{1}{\gamma}(\boldsymbol{D}_{XY} - 2\boldsymbol{D}_X\boldsymbol{S}^{(m)}\boldsymbol{D}_Y^\top + \boldsymbol{Z}^{(m)})) \odot \boldsymbol{S}^{(m)}\end{aligned}$$

$$\begin{aligned}\boldsymbol{S}^{(m+1)} =& \arg\min_{\boldsymbol{S}\in\Pi(\cdot,\boldsymbol{\mu}_t)} \langle -2\boldsymbol{D}_X^\top\boldsymbol{T}^{(m+1)}\boldsymbol{D}_Y, \boldsymbol{S}\rangle + \langle \boldsymbol{Z}^{(m)}, \boldsymbol{T}^{(m+1)} - \boldsymbol{S}\rangle + \\ & \gamma \mathsf{KL}(\boldsymbol{S}\|\boldsymbol{T}^{(m+1)}) = (\boldsymbol{1}_X(\frac{\boldsymbol{\mu}_Y}{\boldsymbol{\Phi}_S^\top\boldsymbol{1}_X})^\top) \odot \boldsymbol{\Phi}_S, \\ \text{where } \boldsymbol{\Phi}_S =& \exp(\frac{1}{\gamma}(2\boldsymbol{D}_X^\top\boldsymbol{T}^{(m+1)}\boldsymbol{D}_Y + \boldsymbol{Z}^{(m)})) \odot \boldsymbol{T}^{(m)}\end{aligned}$$

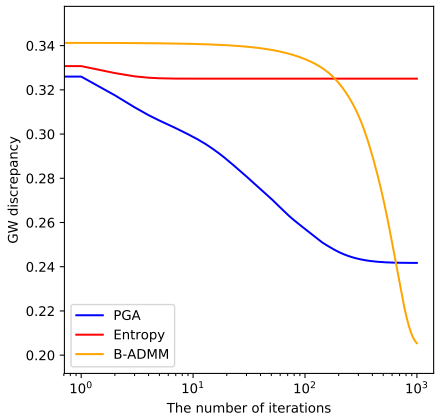
$$\boldsymbol{Z}^{(m+1)} = \boldsymbol{Z}^{(m)} + \gamma(\boldsymbol{T}^{(m+1)} - \boldsymbol{S}^{(m+1)}).$$

Empirical convergence

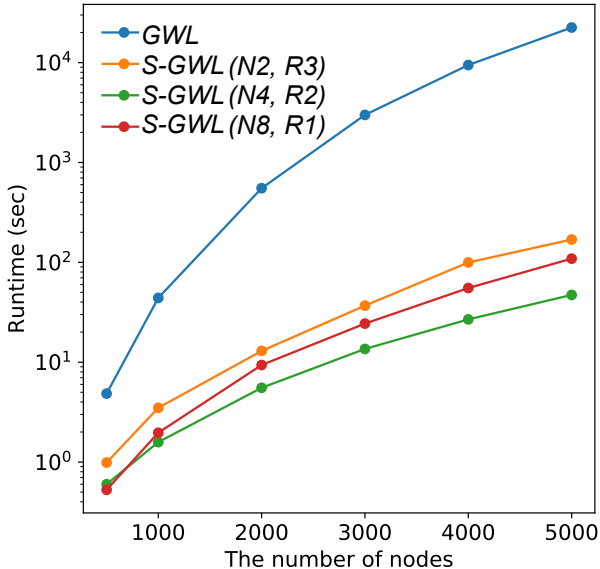
Undirected graph



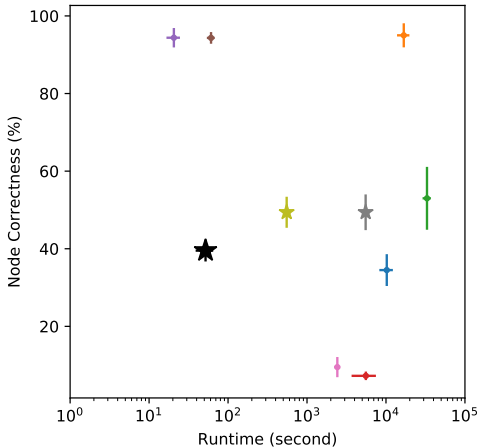
Directed graph



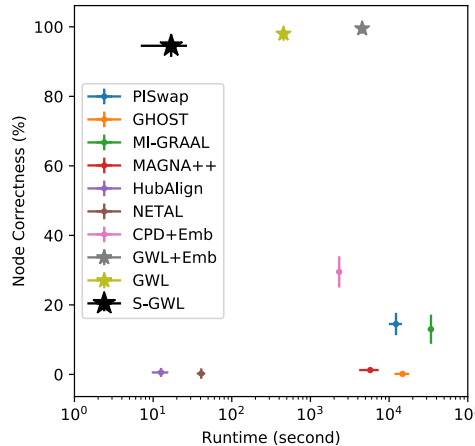
Improvements on scalability



More experimental results on graph matching



(a) BA graphs



(b) GRP graphs