Gromov-Wasserstein Learning for Graph Matching, Partitioning, and Embedding

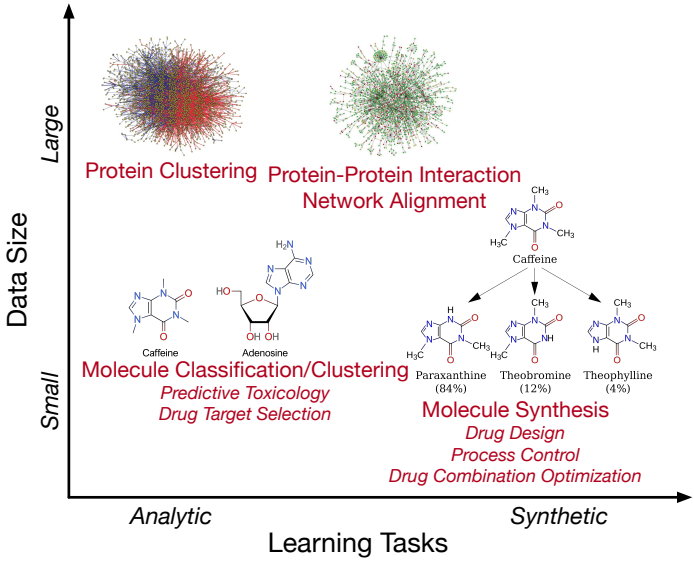
Hongteng Xu

Gaoling School of Artificial Intelligence, Renmin University of China

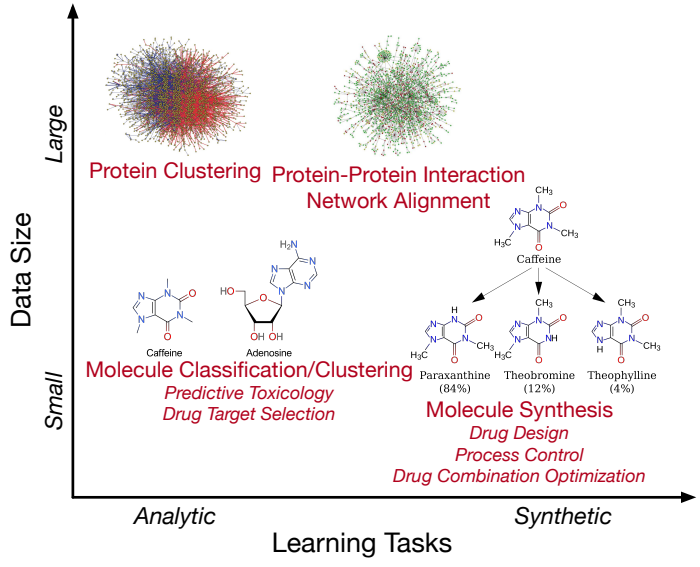
中國人民大學
RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院
Gaoling School of Artificial Intelligence

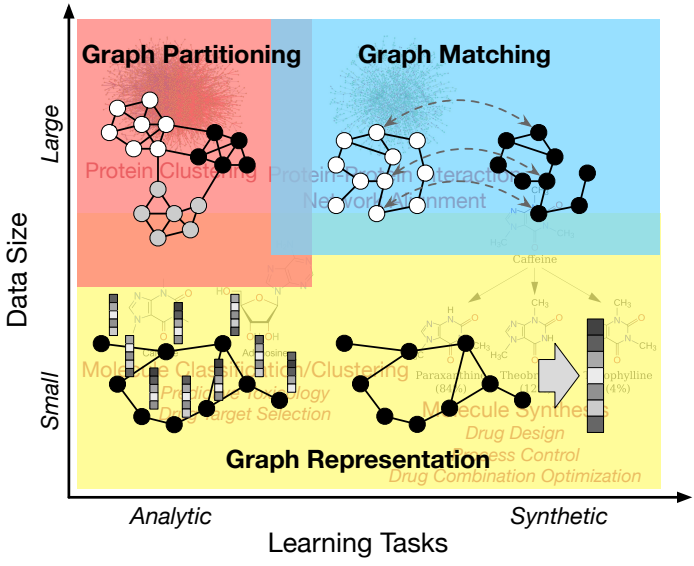
The space of graph-related problems



The space of graph-related problems



The space of graph-related problems

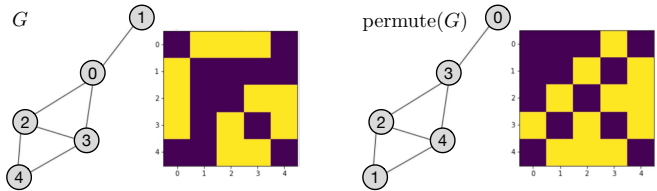


The challenges of the problems

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

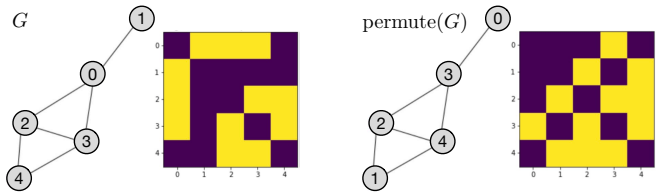
The challenges of the problems

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



The challenges of the problems

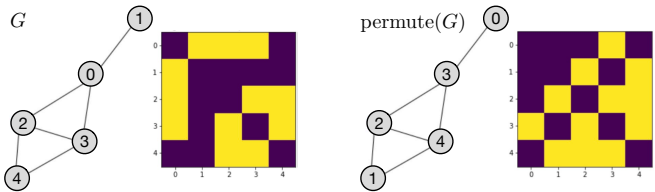
- NP-completeness
  - Approximation algorithms with high stability and scalability
- Permutation invariance



  - 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

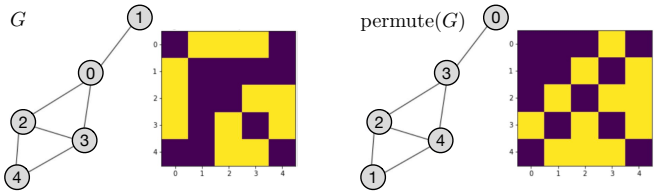
- NP-completeness
  - Approximation algorithms with high stability and scalability
- Permutation invariance



  - 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

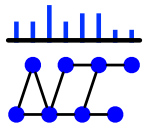
- NP-completeness
  - Approximation algorithms with high stability and scalability
- Permutation invariance



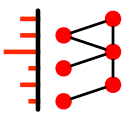
  - A permutation-invariant metric d: $d(G_X, G_Y) = d(G_X, \text{permute}(G_Y))$
  - A permutation-invariant representation model f: $f(G) = f(\text{permute}(G))$
- (Often) No labels
  - Unsupervised or semi-supervised learning

Gromov-Wasserstein Learning (GWL) provides a potential solution.
Applications: PPI network alignment, molecule clustering and classification.

Gromov-Wasserstein distance (GWD) between two graphs



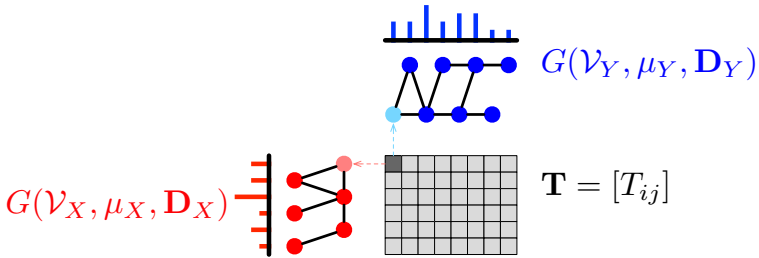
$G(\mathcal{V}_Y, \mu_Y, \mathbf{D}_Y)$



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

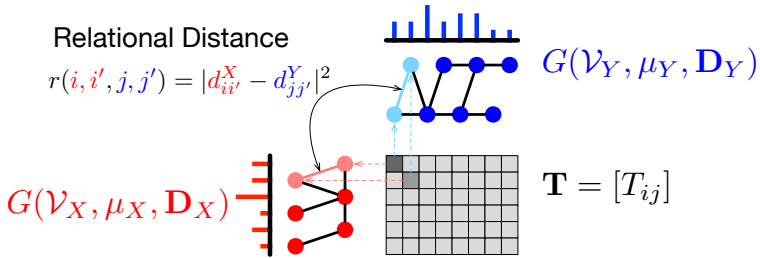
- $\mathcal{V}$: the node set
- $\boldsymbol{\mu}$: a predefined distribution of nodes
- $\boldsymbol{D} = [d_{ii'}]$: the adjacency / distance / kernel matrix

Gromov-Wasserstein distance (GWD) between two graphs



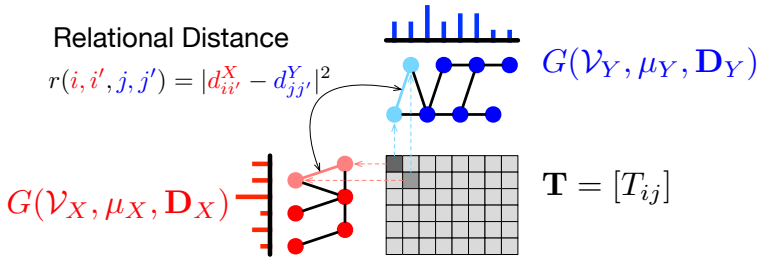
- $\boldsymbol{T} = [T_{ij}]$: a joint distribution of nodes
- $(i \in \mathcal{V}_X, j \in \mathcal{V}_j) \sim \boldsymbol{T}$.
- $\boldsymbol{T} \in \Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y) = \{\boldsymbol{T} \geq \mathbf{0} \mid \boldsymbol{T}\mathbf{1} = \boldsymbol{\mu}_X,\ \boldsymbol{T}^\top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

Gromov-Wasserstein distance (GWD) between two graphs



- $\underbrace{\boldsymbol{T} \otimes \boldsymbol{T}}_{\text{Kronecker product}}$: 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.

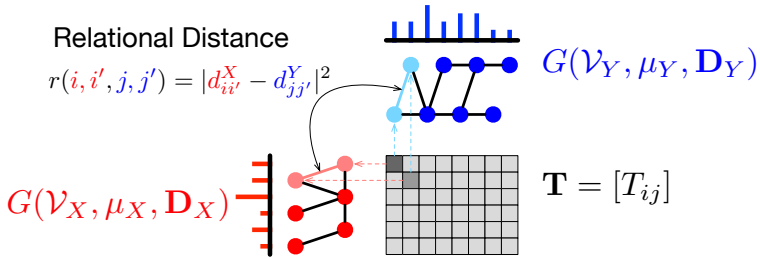
Gromov-Wasserstein distance (GWD) between two graphs



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) between two graphs

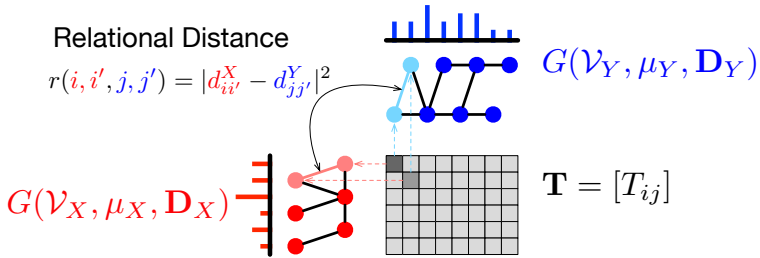


The GWD is **the minimum expectation of the relational distance:**

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

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

Gromov-Wasserstein distance (GWD) between two graphs

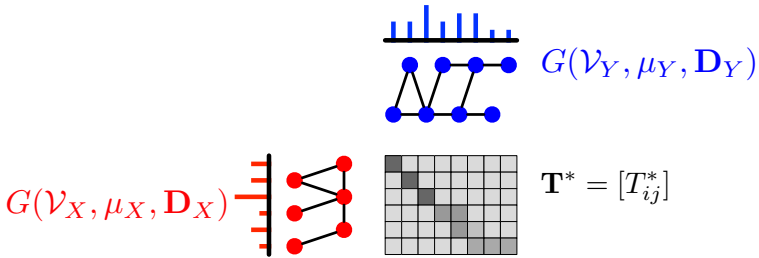


The GWD is **the minimum expectation of the relational distance:**

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

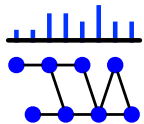
- $\boldsymbol{D}_{XY} = (\boldsymbol{D}_X \odot \boldsymbol{D}_X)\boldsymbol{\mu}_X \mathbf{1}_{|\mathcal{V}_Y|}^\top + \mathbf{1}_{|\mathcal{V}_X|} \boldsymbol{\mu}_Y^\top (\boldsymbol{D}_Y \odot \boldsymbol{D}_Y)^\top$.
- Given comparable node attributes, $\boldsymbol{D}_{XY} \leftarrow \boldsymbol{D}_{XY}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

Advantages of GWD



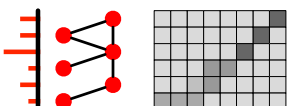
- The optimal joint distribution $\boldsymbol{T}^*$ (or called "**optimal transport**" matrix) indicates the correspondence between the two graphs.

Advantages of GWD



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

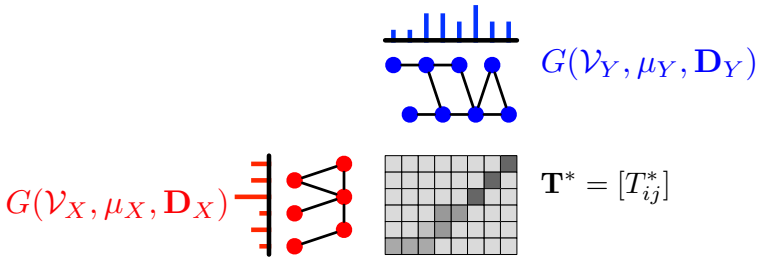
$G(\mathcal{V}_X, \mu_X, \mathbf{D}_X)$



$\mathbf{T}^* = [T^*_{ij}]$

- 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

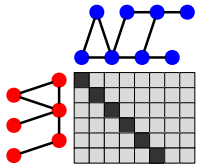


- The optimal joint distribution $\boldsymbol{T}^*$ (or called "**optimal transport**" matrix) indicates the correspondence between the two graphs.
- A **permutation-invariant** (pseudo) metric
  - $d_{gw}(G_X, G_Y) = d_{gw}(G_X, \text{permute}(G_Y))$
- Applicable to the graphs with different sizes,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

Matching via learning optimal transport

Quadratic assignment problem (QAP):

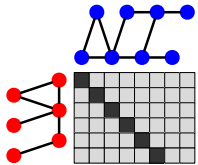
$$\max_{\boldsymbol{P}\in\mathcal{P}} \langle \boldsymbol{D}_X \boldsymbol{P} \boldsymbol{D}_Y^\top, \boldsymbol{P} \rangle,$$
$$\mathcal{P} = \{\boldsymbol{P} \in \{0,1\}^{|\mathcal{V}_X|\times|\mathcal{V}_Y|} \mid \boldsymbol{P}\mathbf{1} = \mathbf{1}, \boldsymbol{P}^\top\mathbf{1} \leq \mathbf{1}\}.$$



Matching via learning optimal transport

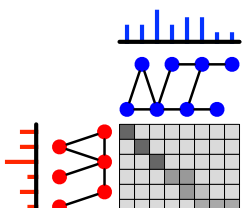
Quadratic assignment problem (QAP):

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



Gromov-Wasserstein distance (GWD):

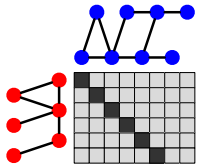
$$\min_{\boldsymbol{T}\in\Pi(\boldsymbol{\mu}_X,\boldsymbol{\mu}_Y)} \langle \boldsymbol{D}_{XY} - 2\boldsymbol{D}_X \boldsymbol{T} \boldsymbol{D}_Y^\top, \boldsymbol{T} \rangle,$$
$$\Pi(\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y) = \{\boldsymbol{T} \geq \mathbf{0} \mid \boldsymbol{T}\mathbf{1} = \boldsymbol{\mu}_X,\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



Matching via learning optimal transport

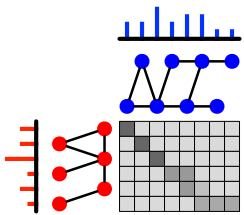
Quadratic assignment problem (QAP):

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



Gromov-Wasserstein distance (GWD):

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

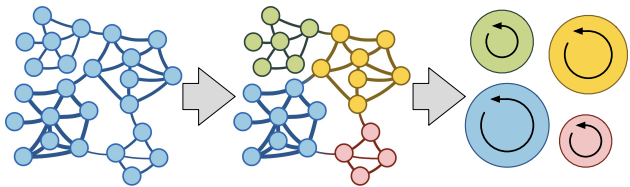


- Conditional prob. $\boldsymbol{P}^* = \frac{\boldsymbol{T}^*}{\boldsymbol{\mu}_X\mathbf{1}^\top} = [P^*(j|i)]$.
- For each node $i \in \mathcal{V}_X$, $j^* = \arg\max_j P^*(j|i)$.

Partitioning is also matching

Modularity maximization principle

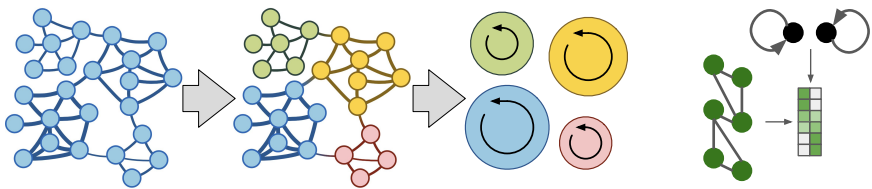
- Dense internal edges + sparse external edges.



Partitioning is also matching

Modularity maximization principle

- Dense internal edges + sparse external edges.



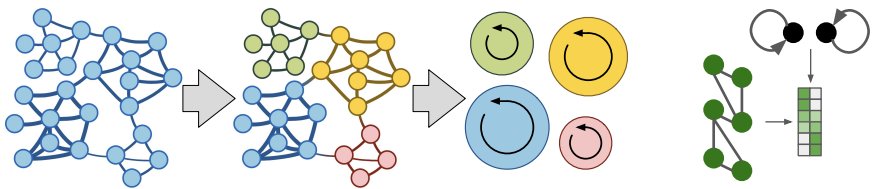
A GWD-based solution *[Xu, et al., NeurIPS 2019]*:

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

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 Optimization Methods

- **Entropic Regularization** *[Peyré, et al., ICML 2016]*
- **Proximal Point Algorithm** *[Xu, et al., ICML 2019]*
- **Bregman ADMM** *[Xu, AAAI 2020]*

Proposed Optimization Methods

- **Entropic Regularization** *[Peyré, et al., ICML 2016]*
- **Proximal Point Algorithm** *[Xu, et al., ICML 2019]*
- **Bregman ADMM** *[Xu, AAAI 2020]*

Convergence

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

Proposed Optimization Methods

- **Entropic Regularization** *[Peyré, et al., ICML 2016]*
- **Proximal Point Algorithm** *[Xu, et al., ICML 2019]*
- **Bregman ADMM** *[Xu, AAAI 2020]*

Convergence

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

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: $\mathcal{O}(V^3)$.

Proposed Optimization Methods

- **Entropic Regularization** *[Peyré, et al., ICML 2016]*
- **Proximal Point Algorithm** *[Xu, et al., ICML 2019]*
- **Bregman ADMM** *[Xu, AAAI 2020]*

Convergence

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

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: $\mathcal{O}(VE)$.

Proposed Optimization Methods

- **Entropic Regularization** *[Peyré, et al., ICML 2016]*
- **Proximal Point Algorithm** *[Xu, et al., ICML 2019]*
- **Bregman ADMM** *[Xu, AAAI 2020]*

Convergence

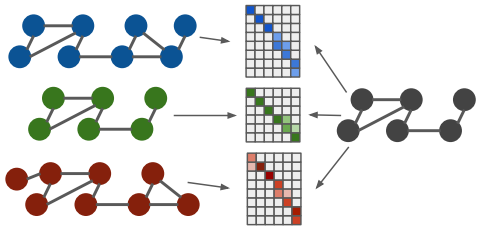
- $\lim_{m\to\infty} \boldsymbol{T}^{(m)}$ is a stationary point.
- Linear convergence.

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): $\mathcal{O}(N(E+V))$.

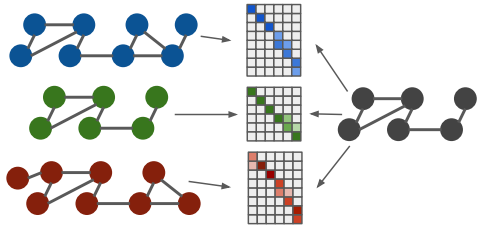
Large-scale matching based on GW barycenters



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

$$\underbrace{B_{gw}(\bar{\mathcal{V}}, \bar{\boldsymbol{\mu}}, \boldsymbol{B}^*)}_{\text{Barycenter graph}}, \underbrace{\{\boldsymbol{T}_k^*\}_{k=1}^K}_{\text{OT matrices}} := \arg\min_B \sum_{k=1}^K \lambda_k d_{gw}(B, G_k), \tag{2}$$

Large-scale matching based on GW barycenters



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

$$\underbrace{B_{gw}(\bar{\mathcal{V}}, \bar{\boldsymbol{\mu}}, \boldsymbol{B}^*)}_{\text{Barycenter graph}}, \underbrace{\{\boldsymbol{T}_k^*\}_{k=1}^K}_{\text{OT matrices}} := \arg\min_B \sum_{k=1}^K \lambda_k d_{gw}(B, G_k), \tag{2}$$

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

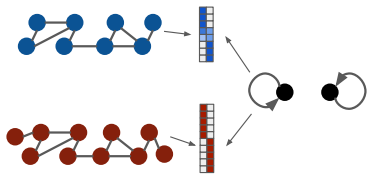
$$\boldsymbol{B}^* = \frac{1}{\bar{\boldsymbol{\mu}}\bar{\boldsymbol{\mu}}^\top} \sum_{k=1}^K \lambda_k (\boldsymbol{T}_k^*)^\top \boldsymbol{D}_k \boldsymbol{T}_k^* \tag{3}$$

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.

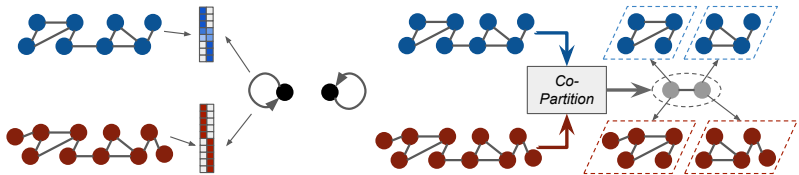


Large-scale matching based on GW barycenters

Co-partition two graphs:

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

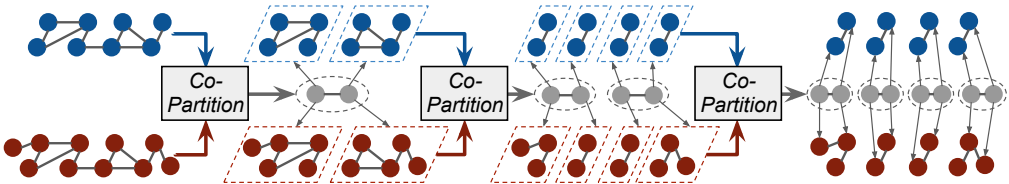
Initialize the barycenter graph by a disconnected graph.



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

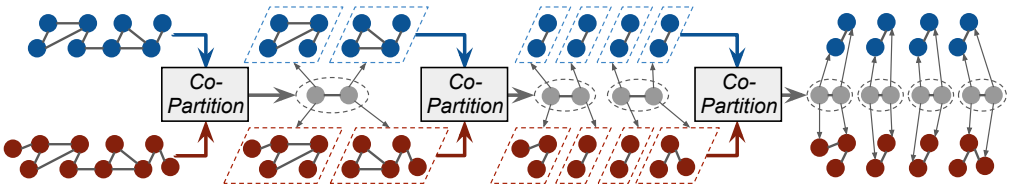
Large-scale matching based on GW barycenters

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



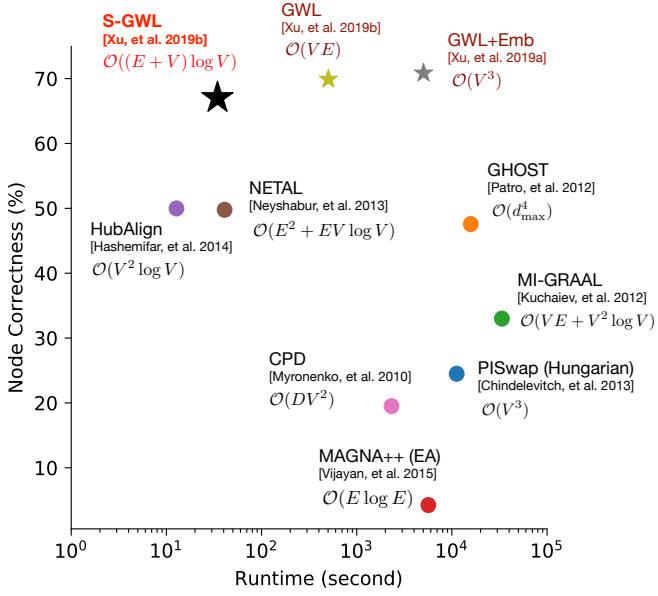
Large-scale matching based on GW barycenters

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



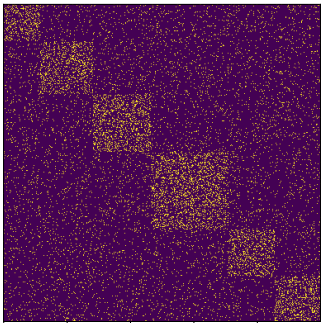
Computational complexity: $\mathcal{O}((E+V)\log V)$

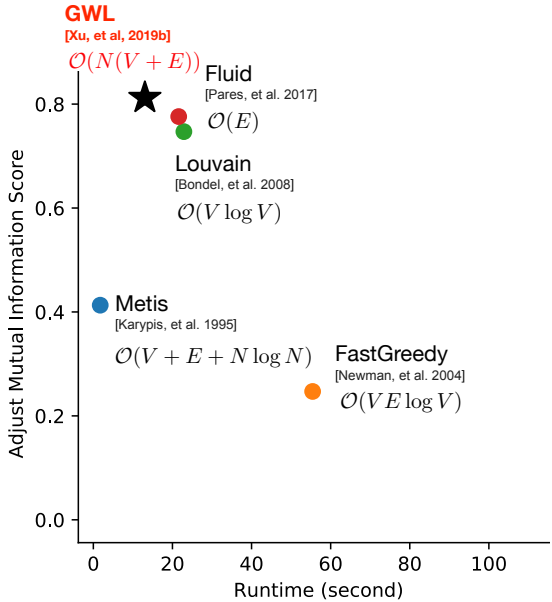
Matching synthetic graphs



Partitioning synthetic graphs

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



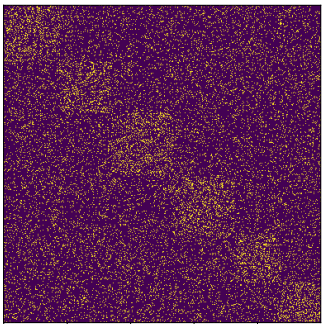


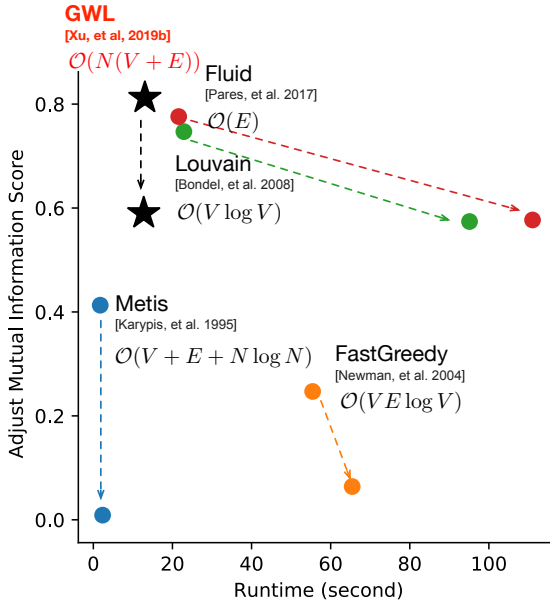
Partitioning synthetic graphs

$V = 4,000$

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

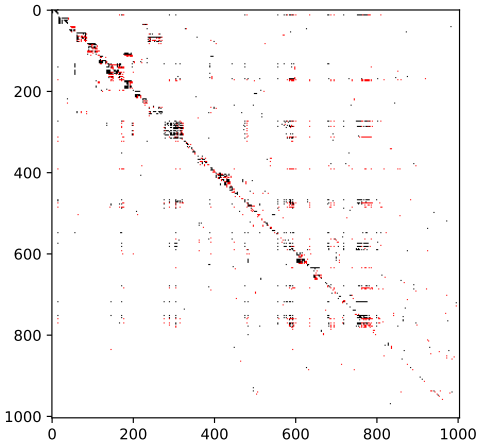
$p_{\text{across}} = 0.1$

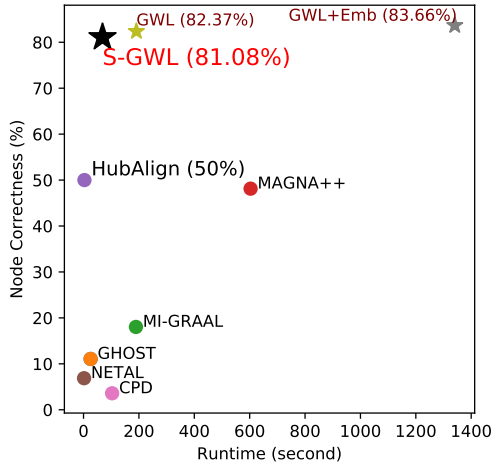




Real-world PPI network alignment

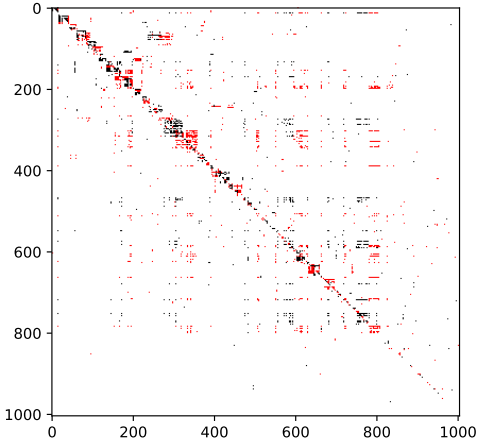
Yeast PPI $\leftrightarrow$ Yeast PPI + 5% LC edges

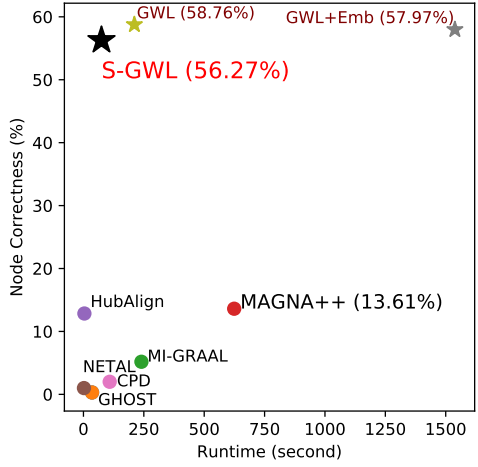




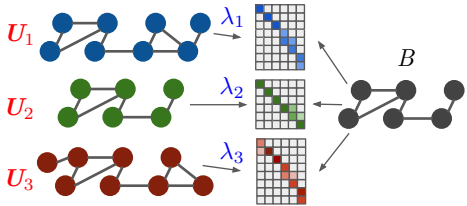
Real-world PPI network alignment

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





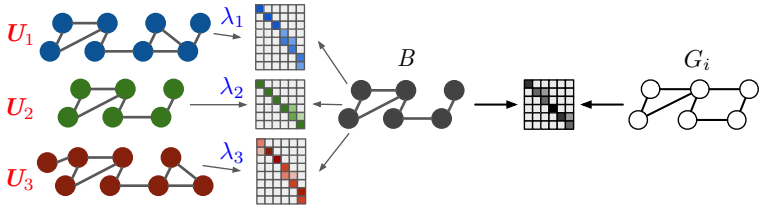
Graph representation via Gromov-Wasserstein factorization



$$B_{gw}(\boldsymbol{U}_{1:K}, \boldsymbol{\lambda}) := \arg\min_B \sum_{k=1}^{K} \lambda_k d_{gw}(B, G_k(\boldsymbol{U}_k)). \tag{4}$$

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

Graph representation via Gromov-Wasserstein factorization



$$B_{gw}(\boldsymbol{U}_{1:K}, \boldsymbol{\lambda}) := \arg\min_B \sum_{k=1}^{K} \lambda_k d_{gw}(B, G_k(\boldsymbol{U}_k)). \tag{4}$$

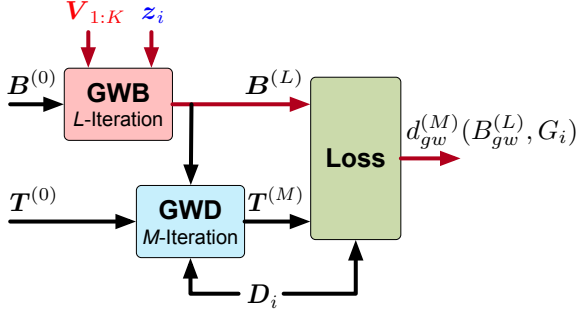
- $\{G_k(\boldsymbol{U}_k)\}_{k=1}^K$: a set of graph bases.
- $\boldsymbol{\lambda} = [\lambda_k] \in \Delta^{K-1}$: the coefficients of the graph basis.
- Estimate each graph by a GW barycenter graph *[Xu, AAAI 2020]*:

$$\min_{\mathbf{1} \geq \boldsymbol{U}_{1:K} \geq \mathbf{0},\ \boldsymbol{\lambda}_{1:I} \in \Delta^{K-1}} \sum_{i=1}^{I} d_{gw}(B_{gw}(\boldsymbol{U}_{1:K}, \overbrace{\boldsymbol{\lambda}_i}^{\text{Rep. of } G_i}), G_i). \tag{5}$$

Graph representation via Gromov-Wasserstein factorization

Reparameterize the problem to an unconstrained optimization problem:

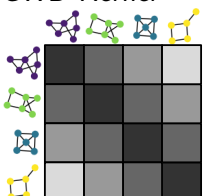
$$\min_{\boldsymbol{V}_{1:K},\ \boldsymbol{z}_{1:I}} \sum_{i=1}^{I} d_{gw}(B_{gw}(\underbrace{\sigma(\boldsymbol{V}_{1:K})}_{\boldsymbol{U}_{1:K}}, \underbrace{\mathrm{softmax}(\boldsymbol{z}_i)}_{\boldsymbol{\lambda}_i}),\ G_i). \tag{6}$$



Experiments on molecule clustering

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

GWD Kernel



GWD+Kmeans

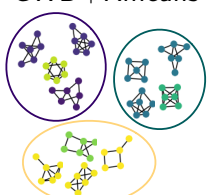
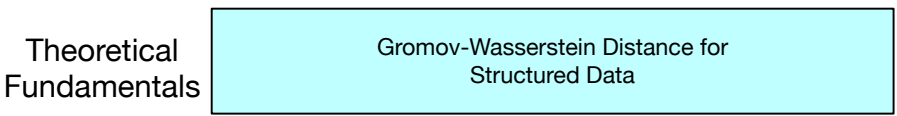


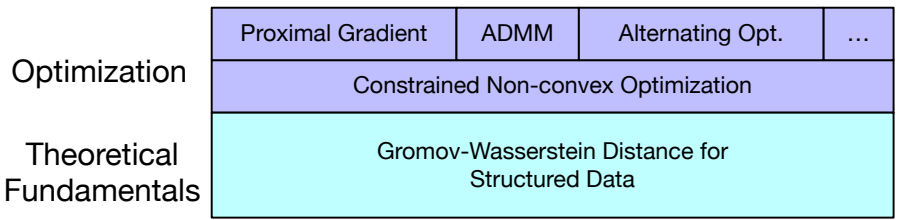
Table: Comparisons on clustering accuracy (%)

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**

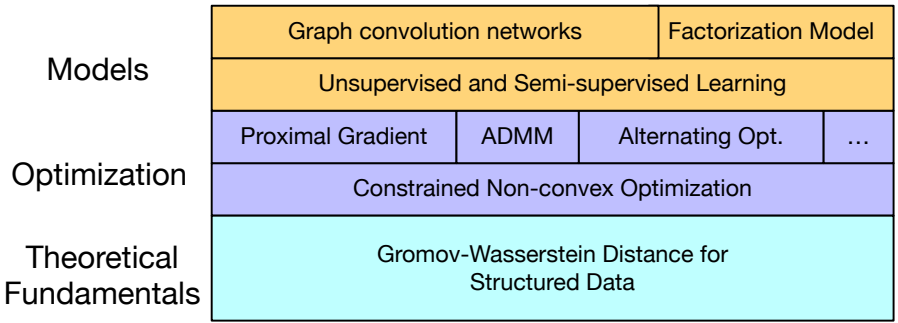
Summary of the GWL framework



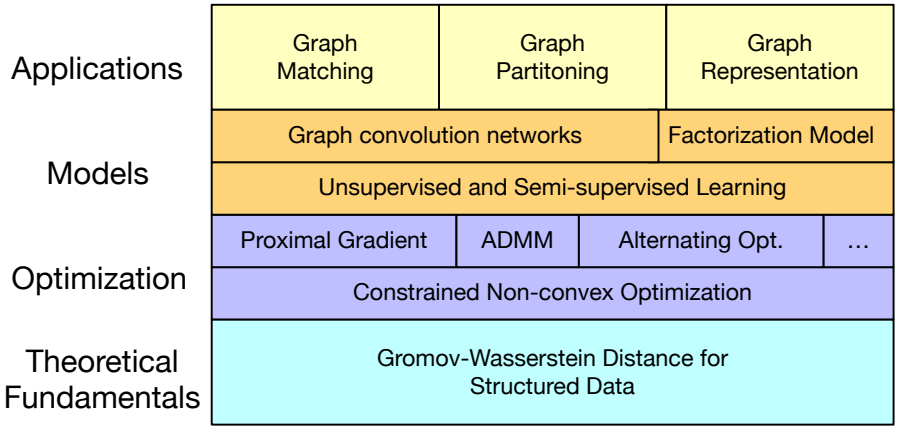
Summary of the GWL framework



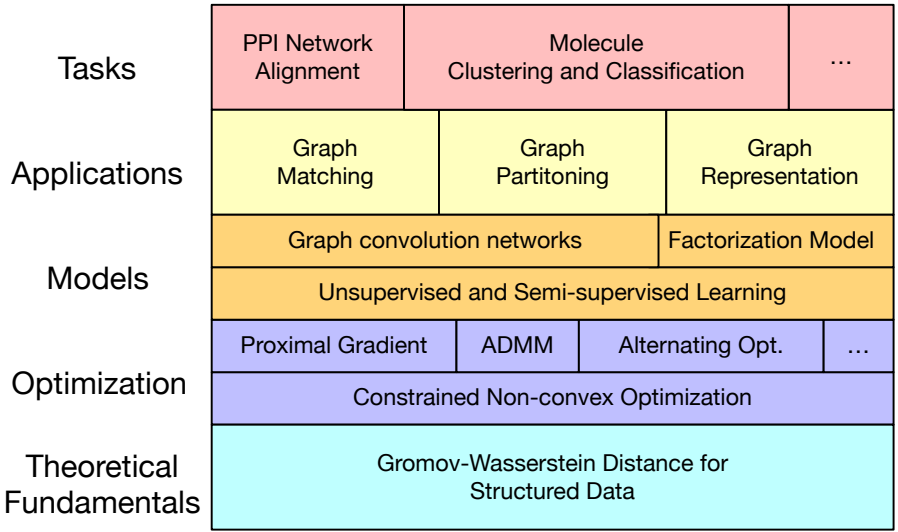
Summary of the GWL framework



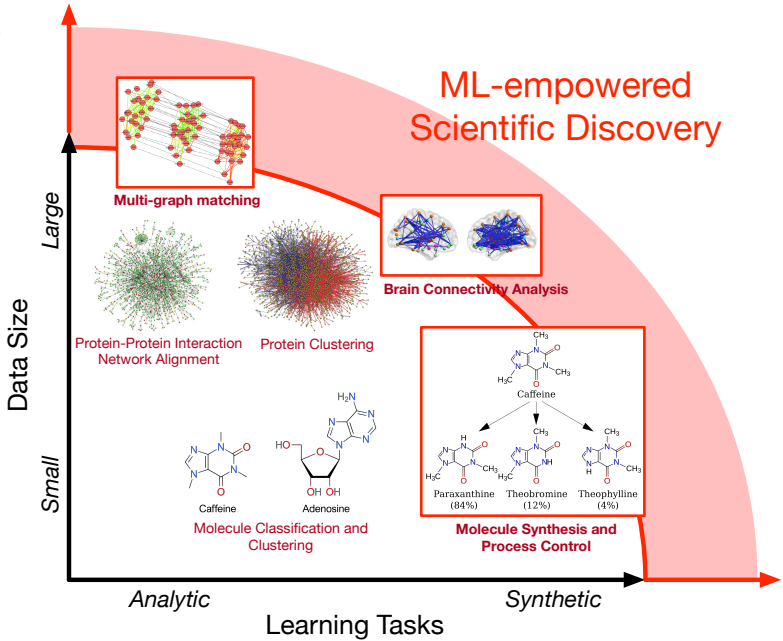
Summary of the GWL framework



Summary of the GWL framework



Future work


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

Thanks! Q&A

https://hongtengxu.github.io

https://github.com/HongtengXu

hongtengxu@ruc.edu.cn

AAAI 2022: OT-SDM Workshop https://ot-sdm.github.io + Tutorial on GWL