A Regularized Wasserstein Framework for Graph Kernels

21st IEEE International Conference on Data Mining (ICDM 2021)

Authors: Asiri Wijesinghe, Qing Wang, Stephen Gould

Reporter: Fengjiao Gong

4th

Date: 2021-12-09

Outline

Denotions

Graph Similarity Matrix

Regularized Wasserstein Framework

Regularized Wasserstein Kernel

Experiments

Denotions

Given **undirected** Graph $G = (V, E)$:

V → vertices set

E → edges set

normalized graph Laplacian:

$$L = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

discrete probability distribution:

$$\Sigma_n = \{\mu \in R_+^n : \sum_i^n \mu_i = 1\}$$

where μ_i is the weight of each vertex $v_i \in V$.

Denotions

Given **one undirected Graph** $G = (V, E)$:

$$\Sigma_n = \{\mu \in R_+^n : \sum_i^n \mu_i = 1\}$$

where μ_i is the weight of each vertex $v_i \in V$.

How to get μ_i?

(a) according to some prior information

(b) uniform distribution $\mu_i = \frac{1}{n}, \mu = \frac{1}{n} 1_n$

Denotions

Given **two Graphs**:

$$G1 : \ \Sigma_{n1} = \{\mu \in R_+^{n_1} : \sum_i^{n_1} \mu_i = 1\}$$

$$G_2 : \ \Sigma_{n2} = \{\nu \in R_+^{n_2} : \sum_i^{n_2} \nu_i = 1\}$$

joint distribution:

$$\pi(\mu, \nu) = \{\gamma \in R_+^{n_1 \times n_2} : \gamma 1_{n_2} = \mu, \gamma^T 1_{n_1} = v\}$$

formalize **a regularized optimal transport problem**:

$$\hat{\gamma} = arg\ mim < \gamma, C >_F + \lambda \Theta(\gamma),\ \lambda \in [0,1]$$

Denotions

Given **a Graph set** $\mathscr{G}$:

$$\mathscr{G} = \{G_1, \cdots, G_n\}$$

Define **graph kernel**:

$$\mathscr{G} \times \mathscr{G} \rightarrow R$$

where **kernel value** is defined upon:

optimal transport distance

Denotions

Given one undirected graph $G = (V, E)$

Consider **embedding functions**:

Feature $\xi_f : V \rightarrow R^m,\ d_f$

Structure $\xi_s : V \rightarrow R^k,\ d_s$

Denote graph discrete distribution:

$$p = \sum_{i=1}^{n} \mu_i \delta(\xi_f(v_i), \xi_s(v_i))$$

Graph Similarity Matrix

Graph Similarity Matrix

measure feature and structure of the graph:

(1) Feature Similarity Matrix

(2) Structure Similarity Matrix

(i) neighborhood similarity matrix

(i) pairwise similarity matrix

Feature Similarity Matrix

Graph signals = features:

$$mapping: \ V \rightarrow R^m$$

graph signal matrix:

$$X \in R^{n \times m}$$

where

$$x_i \in R^m$$

$$n = |V|$$

Feature Similarity Matrix

Local variation matrix:

$$\Delta(X) = \left| X - \frac{L^j X}{\lambda_{max}(L)} \right|$$

to quantify graph signals changing from vertex to its neighbors

where,

$L^j X$ → aggregated graph signals of all vertices in G within **j-hop neighborhood**

$\lambda_{max}(L)$ → normalize $L^j X$ to ensure the numerical stability

Feature Similarity Matrix

Feature Embedding Vector

For **one Graph**:

$$a_i = \xi_f(v_i) = x_i \otimes \Delta(x_i) \in R^{2m}$$

where $\otimes$ refers to **concatenation**.

Feature Similarity Matrix

For one Graph:

$$a_i = \xi_f(v_i) = x_i \otimes \Delta(x_i) \in R^{2m}$$

where $\otimes$ refers to concatenation.

For **two Graphs**:

$$C^V(i,j) = (d_f(a_i, a_j))_{i,j} \in R^{n_1 \times n_2}$$

Feature Similarity Matrix

Graph Similarity Matrix

measure feature and structure of the graph:

(1) Feature Similarity Matrix

(2) Structure Similarity Matrix

(i) neighborhood similarity matrix

(i) pairwise similarity matrix

Neighborhood Similarity Matrix

For **one graph**, node embedding:

$$e_i = \xi_s(v_i) \in R^k$$

learned by heat kernel random walks with graph attention.

Probability transition matrix:

$$M = e^{-tL}$$

where

t is the length of random walks

[reference]S. Abu-El-Haija, B. Perozzi, R. Al-Rfou, and A. A. Alemi. Watch your step: Learning node embeddings via graph attention. In NeurIPS, 2018.

Neighborhood Similarity Matrix

For one graph, node embedding:

$$e_i = \xi_s(v_i) \in R^k$$

learned by heat kernel random walks with graph attention.

For **two Graphs**:

$$C^N(i,j) = (d_s(e_i, e_j))_{i,j} \in R^{n_1 \times n_2}$$

neighborhood similarity matrix

Graph Similarity Matrix

measure feature and structure of the graph:

(1) Feature Similarity Matrix

(2) Structure Similarity Matrix

(i) neighborhood similarity matrix

(i) pairwise similarity matrix

Pairwise Similarity Matrix

For **one graph**:

$$C^P(i,j) = (d_s(e_i, e_j))_{i,j} \in R^{n \times n}$$

Pairwise Similarity matrix

Pairwise Similarity Matrix

For one graph:

$$C^P(i,j) = (d_s(e_i, e_j))_{i,j} \in R^{n\times n}$$

Pairwise Similarity matrix

For **two graphs**:

$$L_2(C_1^P(i,j), C_2^P(k,l)) = \frac{1}{2}\left| C_1^P(i,j) - C_2^P(k,l) \right|^2$$

pairwise similarity between G_1 and G_2

Regularized Wasserstein Framework

Regularized Wasserstein Framework

a novel OT framework

preserve local and global **graph structure**:

(1) local barycentric Wasserstein distance

(2) global connectivity Wasserstein distance

preserve both features and structures:

(3)regularized Wasserstein discrepancy

Local Barycentric Wasserstein Distance

Wasserstein distance defined on **neighborhood similarity matrix**:

$$LW(\mu,\nu) = \min_{\gamma \in \pi(\mu,\nu)} < \gamma, C^N >_F + \Theta_w(\gamma)$$

Design $\Theta_w(\gamma)$

Local Barycentric Wasserstein Distance

Define **transport map**:

$$T : \mu \rightarrow \nu$$

by mapping:

$$e_i^\mu \rightarrow \hat{e}_i^\mu$$

where $\hat{e}_i^\mu$ is weighted average of node embeddings of vertices in ν

$$\hat{e}_i^\mu = T(e_i^\mu) = \frac{\sum_{j=1}^{n_2} \gamma(i,j) e_j^\nu}{\sum_{j=1}^{n_2} \gamma(i,j)}$$

Local Barycentric Wasserstein Distance

Define **node embedding matrix**:

$$E_\mu = \begin{bmatrix} e_1^\mu \\ \vdots \\ e_{n_1}^\mu \end{bmatrix} \in R^{n_1 \times k}, \quad E_\nu = \begin{bmatrix} e_1^\nu \\ \vdots \\ e_{n_1}^\nu \end{bmatrix} \in R^{n_2 \times k}$$

Then

$$\hat{E}_\mu = T(E_\mu) = (diag(\gamma 1_{n_2}))^{-1} \gamma E_\nu$$

Local Barycentric Wasserstein Distance

Define **node embedding matrix**:

$$E_\mu = \begin{bmatrix} e_1^\mu \\ \vdots \\ e_{n_1}^\mu \end{bmatrix} \in R^{n_1 \times k}, \quad E_\nu = \begin{bmatrix} e_1^\nu \\ \vdots \\ e_{n_1}^\nu \end{bmatrix} \in R^{n_2 \times k}$$

Then

$$\hat{E}_\mu = T(E_\mu) = (diag(\gamma 1_{n_2}))^{-1} \gamma E_\nu$$

If μ and ν are unifom distributions,

$$\hat{E}_\mu = n_1 \gamma E_\nu$$

Local Barycentric Wasserstein Distance

Define **source regularization**

$$\Omega_\mu(\gamma) = \frac{1}{n_1^2}\sum_{i,j} a_{i,j} \left\| \hat{e}_i^\mu - \hat{e}_j^\mu \right\|_2^2 = \frac{1}{n_1^2}\,\mathrm{tr}\left(\hat{\mathbf{E}}_\mu^T \mathbf{L}_\mu \hat{\mathbf{E}}_\mu\right)$$

a spatially **localized barycentric term**.

Local Barycentric Wasserstein Distance

Define **source regularization**

$$\Omega_\mu(\gamma) = \frac{1}{n_1^2} \sum_{i,j} a_{i,j} \left\| \hat{e}_i^\mu - \hat{e}_j^\mu \right\|_2^2 = \frac{1}{n_1^2} \operatorname{tr}\left(\hat{\mathbf{E}}_\mu^T \mathbf{L}_\mu \hat{\mathbf{E}}_\mu \right)$$

a spatially **localized barycentric term**.

If μ and ν are unifom distributions,

$$\Omega_\mu(\gamma) = \operatorname{tr}\left(\mathbf{E}_\nu^T \gamma^T \mathbf{L}_\mu \gamma \mathbf{E}_\nu \right)$$

Local Barycentric Wasserstein Distance

Similarly **target regularization**

$$\Omega_\nu(\gamma) = \frac{1}{n_2^2} \sum_{i,j} a_{i,j} \left\| \hat{e}_i^\nu - \hat{e}_j^\nu \right\|_2^2 = \frac{1}{n_2^2} \operatorname{tr}\left(\hat{\mathbf{E}}_\nu^T \mathbf{L}_\nu \hat{\mathbf{E}}_\nu \right)$$

a spatially **localized barycentric term**.

Local Barycentric Wasserstein Distance

Regularization term:

$$\Theta_w(\gamma) = \lambda_\mu \Omega_\mu(\gamma) + \lambda_\nu \Omega_\nu(\gamma) + \frac{\rho}{2} \|\gamma\|_F^2$$

where

$$0 \leq \lambda_\mu, \lambda_\nu \leq 1$$

$\|\gamma\|_F^2$ to smooth transport mass conservation

$\rho \in (0,1]$ is degree of smoothness

Local Barycentric Wasserstein Distance

Regularization term:

$$\Theta_w(\gamma) = \lambda_\mu \Omega_\mu(\gamma) + \lambda_\nu \Omega_\nu(\gamma) + \frac{\rho}{2}\|\gamma\|_F^2$$

here to avoid strict mass conservation (i.e, a bijective mapping between μ and ν)

[reference] A function is said to be **bijective** or bijection, if a function f: A → B satisfies both the injective (one-to-one function) and surjective function (onto function) properties.It means that every element “b” in the codomain B, there is exactly one element “a” in the domain A. such that f(a) = b. If the function satisfies this condition, then it is known as **one-to-one correspondence**. (https://byjus.com/maths/bijective-function/)

Wasserstein Distance

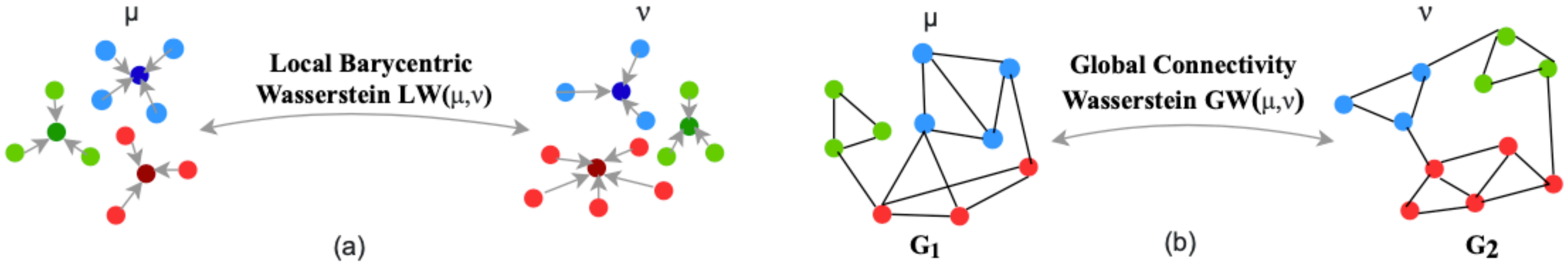

μ
ν
Local Barycentric Wasserstein LW(μ,ν)
(a)
μ
ν
Global Connectivity Wasserstein GW(μ,ν)
G1
(b)
G2

Fig. 2: (a) shows the local barycentric Wasserstein distance that transports each vertex in μ to a spatially localized barycenter of its corresponding neighbors in ν and vice versa; (b) shows the global connectivity Wasserstein distance that captures the pairwise similarity between vertices under the preservation of degree distributions.

Local Barycentric Wasserstein Distance

Convergence

Lemma 1. $LW(\mu, \nu)$ is strongly convex and smooth w.r.t. γ.

Regularized Wasserstein Framework

a novel OT framework

preserve local and global **graph structure**:

(1) local barycentric Wasserstein distance

(2) global connectivity Wasserstein distance

preserve both features and structures:

(3)regularized Wasserstein discrepancy

Global Connectivity Wasserstein Distance

Design **degree entropy regularization term $\Theta_g(\gamma)$** on pairwise similarity matrix:

$$GW(\mu,\nu) = \min_{\gamma \in \pi(\mu,\nu)} \left\langle \gamma, L_2\left(\mathbf{C}^P_\mu, \mathbf{C}^P_\nu\right) \otimes \gamma \right\rangle_F - \lambda_g \Theta_g(\gamma)$$

where

$$\left\langle \gamma, L_2\left(\mathbf{C}^P_\mu, \mathbf{C}^P_\nu\right) \otimes \gamma \right\rangle_F = \sum_{i,j,k,l} L_2\left(\mathbf{C}^P_\mu(i,j), \mathbf{C}^P_\nu(k,l)\right) \gamma(i,k)\gamma(j,l)$$

$$\lambda_g \in (0,1]$$

Global Connectivity Wasserstein Distance

Design **degree entropy regularization term** $\Theta_g(\gamma)$ on pairwise similarity matrix:

$$GW(\mu,\nu) = \min_{\gamma \in \pi(\mu,\nu)} \left\langle \gamma, L_2\left(\mathbf{C}_{\mu}^{P}, \mathbf{C}_{\nu}^{P}\right) \otimes \gamma \right\rangle_F - \lambda_g \Theta_g(\gamma)$$

when

$\mathbf{C}_{\mu}^{P}, \mathbf{C}_{\nu}^{P} \rightarrow$ **shortest path distance matrices**

adjacency matrices

graph Laplacians

to preserve connectivity structure of graphs.

Global Connectivity Wasserstein Distance

Specifically,

$$\Theta_g(\gamma) = KL\left(\gamma \| \gamma'\right) = \sum_{i,j} \gamma(i,j) \log\left(\frac{\gamma(i,j)}{\gamma'(i,j)}\right)$$

where, $D_\mu \in R^{n_1}$, $D_\nu \in R^{n_2}$ are node degree vectors, γ' is **prior node degree distribution**

$$\gamma'(i,j) = \frac{\tilde{\gamma}(i,j)}{\left\| \sum_j \tilde{\gamma}(i,j) \right\|_1}$$

$$\tilde{\gamma}(i,j) = 1 - \frac{\left| D_\mu^i - D_\nu^j \right|}{\max\left\{ D_\mu^i, D_\nu^j \right\}}$$

Global Connectivity Wasserstein Distance

Convergence

Lemma 2. $KL\left(\gamma || \gamma'\right)$ is strongly convex w.r.t. γ.

Although $GW(\mu, \nu)$ remains non-convex, strongly convex of $KL\left(\gamma || \gamma'\right)$ enables better optimization convergence.

Wasserstein Distance

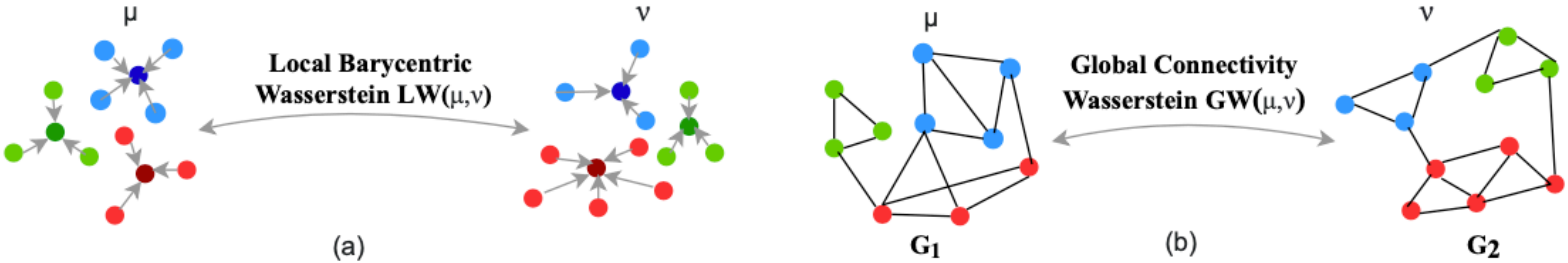


Fig. 2: (a) shows the local barycentric Wasserstein distance that transports each vertex in μ to a spatially localized barycenter of its corresponding neighbors in ν and vice versa; (b) shows the global connectivity Wasserstein distance that captures the pairwise similarity between vertices under the preservation of degree distributions.

Regularized Wasserstein Framework

a novel OT framework

preserve local and global graph structure:

(1) local barycentric Wasserstein distance

(2) global connectivity Wasserstein distance

preserve both **features** and **structures**:

(3)regularized Wasserstein discrepancy

Regularized Wasserstein discrepancy

Similarity Matrices:

$$C^V(i,j) = (d_f(a_i, a_j))_{i,j} \in R^{n_1 \times n_2}$$ (**feature**)

$$C^N(i,j) = (d_s(e_i, e_j))_{i,j} \in R^{n_1 \times n_2}$$

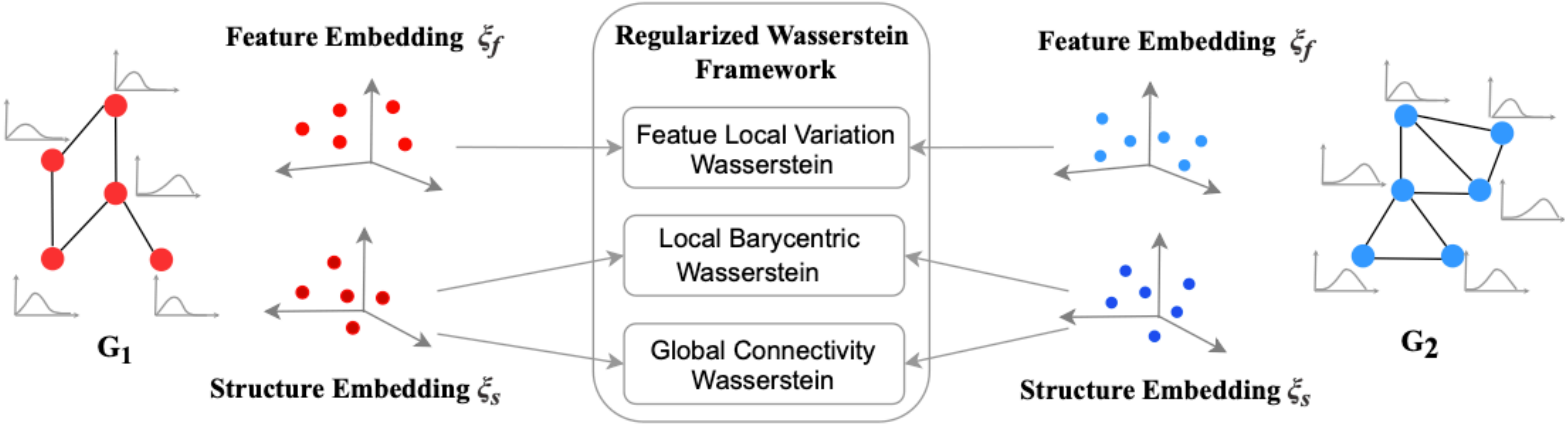
$$C^P(i,j) = (d_s(e_i, e_j))_{i,j} \in R^{n \times n}$$

Wasserstein distances:

$$LW(\mu,\nu) = \min_{\gamma \in \pi(\mu,\nu)} < \gamma, C^N >_F + \Theta_w(\gamma)$$ (**vertices**)

$$GW(\mu,\nu) = \min_{\gamma \in \pi(\mu,\nu)} \left\langle \gamma, L_2\left(\mathbf{C}^P_\mu, \mathbf{C}^P_\nu\right) \otimes \gamma \right\rangle_F - \lambda_g \Theta_g(\gamma)$$ (**edges**)

Regularized Wasserstein discrepancy



Regularized Wasserstein Framework

Featue Local Variation Wasserstein

Local Barycentric Wasserstein

Global Connectivity Wasserstein

Feature Embedding ξ_f
Feature Embedding ξ_f
G_1
G_2
Structure Embedding ξ_s
Structure Embedding ξ_s

Fig. 1: An overview of the proposed framework for regularized Wasserstein kernels (RWKs), which unifies feature local variation, local barycentric and global connectivity Wasserstein distances based on feature and structure embeddings.

Regularized Wasserstein discrepancy

RW discrepancy:

$$RW(\mu, \nu) = \min_{\gamma \in \pi(\mu, \nu)} \left\langle \gamma, \mathbf{C}^V \right\rangle_F + \beta_1 LW(\mu, \nu) + \beta_2 GW(\mu, \nu)$$

Regularized Wasserstein discrepancy

RW discrepancy:

$$RW(\mu,\nu) = \min_{\gamma\in\pi(\mu,\nu)} \left\langle \gamma, \mathbf{C}^V \right\rangle_F + \beta_1 LW(\mu,\nu) + \beta_2 GW(\mu,\nu)$$

transform into:

$$\min_{\gamma\in\pi(\mu,\nu)} H(\gamma) = \min_{\gamma\in\pi(\mu,\nu)} f(\gamma) + g(\gamma) - h(\gamma) \qquad \textbf{(RW)}$$

where

$$f(\gamma) = \left\langle \gamma, \mathbf{C}^V \right\rangle_F + \beta_1 LW(\mu,\nu)$$

$$g(\gamma) = \left\langle \gamma, \beta_2 \left(L_2 \left(\mathbf{C}^P_\mu, \mathbf{C}^P_\nu \right) \otimes \gamma \right) \right\rangle_F$$

$$h(\gamma) = \beta_2 \left(\lambda_g \Theta_g(\gamma) \right)$$

Regularized Wasserstein discrepancy

RW discrepancy:

$$\min_{\gamma \in \pi(\mu,\nu)} H(\gamma) = \min_{\gamma \in \pi(\mu,\nu)} f(\gamma) + g(\gamma) - h(\gamma)$$

here

NP-hard

Sinkhorn Conditional Gradient(SCG)

linearize

Sinkhorn Conditional Gradient(SCG)

RW discrepancy:

$$\min_{\gamma \in \pi(\mu,\nu)} H(\gamma) = \min_{\gamma \in \pi(\mu,\nu)} f(\gamma) + g(\gamma) - h(\gamma)$$

Here

t: maximum number of iterations for SCG

b: maximal number of Sinkhorn iterations

$\lambda \in [0,\infty]$

```
initialize $i = 0,\ \gamma^0 \leftarrow \mu\nu^T,$ and $c^0 \leftarrow H\left(\gamma^0\right)$
while $i \le t$ do
    $i \leftarrow i + 1$
    $\nabla H(\gamma) \leftarrow$ Gradient of $H(\gamma)$ w.r.t $\gamma^{(i-1)}$
    $\hat{\gamma}^{(i-1)} \leftarrow$ Sinkhorn-knopp $(\mu, \nu, \nabla H(\gamma), \lambda, b)$
    $\Delta\gamma \leftarrow \hat{\gamma}^{(i-1)} - \gamma^{(i-1)}$
    $\alpha^{(i)}, c^{(i)} \leftarrow$ Line-search $\left(\gamma^{(i-1)}, \Delta\gamma, \nabla H(\gamma), c^{(i-1)}\right)$
    $\gamma^{(i)} \leftarrow \gamma^{(i-1)} + \alpha^{(i)}\Delta\gamma$
    if $\delta^{(i-1)} \leftarrow \langle \Delta\gamma, -\nabla H(\gamma)\rangle_F \le \epsilon$ then
        stop
    end-if
end-while
```

Sinkhorn Conditional Gradient(SCG)

RW discrepancy:

$$\min_{\gamma \in \pi(\mu,\nu)} H(\gamma) = \min_{\gamma \in \pi(\mu,\nu)} f(\gamma) + g(\gamma) - h(\gamma)$$

SCG has nice convergence properties!

```
initialize $i = 0,\ \gamma^0 \leftarrow \mu\nu^T,$ and $c^0 \leftarrow H\left(\gamma^0\right)$
while $i \leq t$ do
    $i \leftarrow i + 1$
    $\nabla H(\gamma) \leftarrow$ Gradient of $H(\gamma)$ w.r.t $\gamma^{(i-1)}$
    $\hat{\gamma}^{(i-1)} \leftarrow$ Sinkhorn-knopp $(\mu, \nu, \nabla H(\gamma), \lambda, b)$
    $\Delta\gamma \leftarrow \hat{\gamma}^{(i-1)} - \gamma^{(i-1)}$
    $\alpha^{(i)}, c^{(i)} \leftarrow$ Line-search $\left(\gamma^{(i-1)}, \Delta\gamma, \nabla H(\gamma), c^{(i-1)}\right)$
    $\gamma^{(i)} \leftarrow \gamma^{(i-1)} + \alpha^{(i)}\Delta\gamma$
    if $\delta^{(i-1)} \leftarrow \langle \Delta\gamma, -\nabla H(\gamma) \rangle_F \leq \epsilon$ then
        stop
    end-if
end-while
```

Sinkhorn Conditional Gradient(SCG)

RW discrepancy:

$$\min_{\gamma \in \pi(\mu,\nu)} H(\gamma) = \min_{\gamma \in \pi(\mu,\nu)} f(\gamma) + g(\gamma) - h(\gamma)$$

SCG has nice convergence properties!

Define **sub-optimality gap**:

$$\delta_i = \max_{\hat{\gamma} \in \pi(\mu,\nu)} \langle (\gamma - \hat{\gamma}), \nabla H(\gamma) \rangle_F$$

```
initialize $i = 0$, $\gamma^0 \leftarrow \mu\nu^T$, and $c^0 \leftarrow H\left(\gamma^0\right)$
while $i \leq t$ do
    $i \leftarrow i + 1$
    $\nabla H(\gamma) \leftarrow$ Gradient of $H(\gamma)$ w.r.t $\gamma^{(i-1)}$
    $\hat{\gamma}^{(i-1)} \leftarrow$ Sinkhorn-knopp $(\mu, \nu, \nabla H(\gamma), \lambda, b)$
    $\Delta\gamma \leftarrow \hat{\gamma}^{(i-1)} - \gamma^{(i-1)}$
    $\alpha^{(i)}, c^{(i)} \leftarrow$ Line-search $\left(\gamma^{(i-1)}, \Delta\gamma, \nabla H(\gamma), c^{(i-1)}\right)$
    $\gamma^{(i)} \leftarrow \gamma^{(i-1)} + \alpha^{(i)}\Delta\gamma$
    if $\delta^{(i-1)} \leftarrow \langle \Delta\gamma, -\nabla H(\gamma) \rangle_F \leq \epsilon$ then
        stop
    end-if
end-while
```

Sinkhorn Conditional Gradient(SCG)

Convergence

Theorem 1 (Convergence). SCG has the minimal suboptimality gap δ_i that satisfies the following condition:

$$\min_{0 \le i \le k} \delta_i \le \frac{\max\left\{2h_0, (L-\sigma)\cdot \operatorname{diam}_{\|\cdot\|}(\pi(\mu,\nu))^2\right\}}{\sqrt{k+1}}$$

where

$$\sigma = 1$$

$h_0 = H\left(\gamma^0\right) - \min\limits_{\gamma \in \pi(\mu,\nu)} H(\gamma)$ is the initial suboptimality gap

L is a Lipschitz constant of $\nabla(f+g)(\gamma)$

diam $_{\|\cdot\|}(\pi(\mu,\nu))^2$ denotes the $\|\cdot\|_F$-diameter of the $\pi(\mu,\nu)$

Sinkhorn Conditional Gradient(SCG)

Convergence

Corollary 1. For SCG, the minimal suboptimality gap is $O\left(\frac{1}{\sqrt{k}}\right)$ after the number k of iterations. It takes at most $O\left(\frac{1}{\epsilon^2}\right)$ iterations to find **an approximate stationary point** with a suboptimality gap smaller than $O\left(\frac{1}{\epsilon^2}\right)$.

Regularized Wasserstein Kernel (RWK)

Regularized Wasserstein Kernels

RWK

Define Kernel matrix:

$$K_{\mu\nu} = e^{-\eta RW(\mu,\nu)} \in R^{|\mathcal{G}| \times |\mathcal{G}|}, \eta > 0$$

indefinite

the noisy observation of a true positive semi-definite kernel

[reference]R. Luss and A. d'Aspremont. Support vector machine classification with indefinite kernels. In NeurIPS, 2008.

*A matrix m is **indefinite** if its Hermitian part is neither a positive nor a negative semidefinite matrix.*

Regularized Wasserstein Kernels

RWK

Define Kernel matrix:

$$K_{\mu\nu} = e^{-\eta RW(\mu,\nu)} \in R^{|\mathcal{G}|\times|\mathcal{G}|}, \eta > 0$$

graph classification

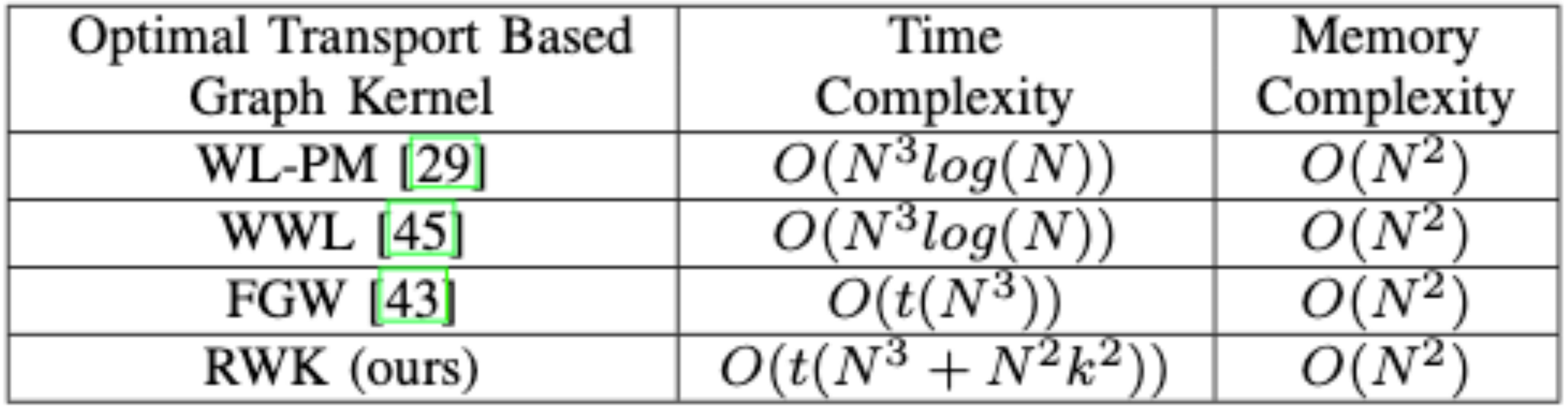
a robust classification problem under a perturbation of a true positive semidefinite kernel

[reference]R. Luss and A. d'Aspremont. Support vector machine classification with indefinite kernels. In NeurIPS, 2008.

Regularized Wasserstein Kernels

Computation Complexity

Summary:



Optimal Transport Based Graph Kernel	Time Complexity	Memory Complexity
WL-PM [29]	$O(N^3 log(N))$	$O(N^2)$
WWL [45]	$O(N^3 log(N))$	$O(N^2)$
FGW [43]	$O(t(N^3))$	$O(N^2)$
RWK (ours)	$O(t(N^3 + N^2 k^2))$	$O(N^2)$

TABLE I: A summary of time and memory complexities.

where

$t \ll N$ is the maximal number of SCG iteration

k is the dimension of node embedding

Experiments

Experiments

Datasets

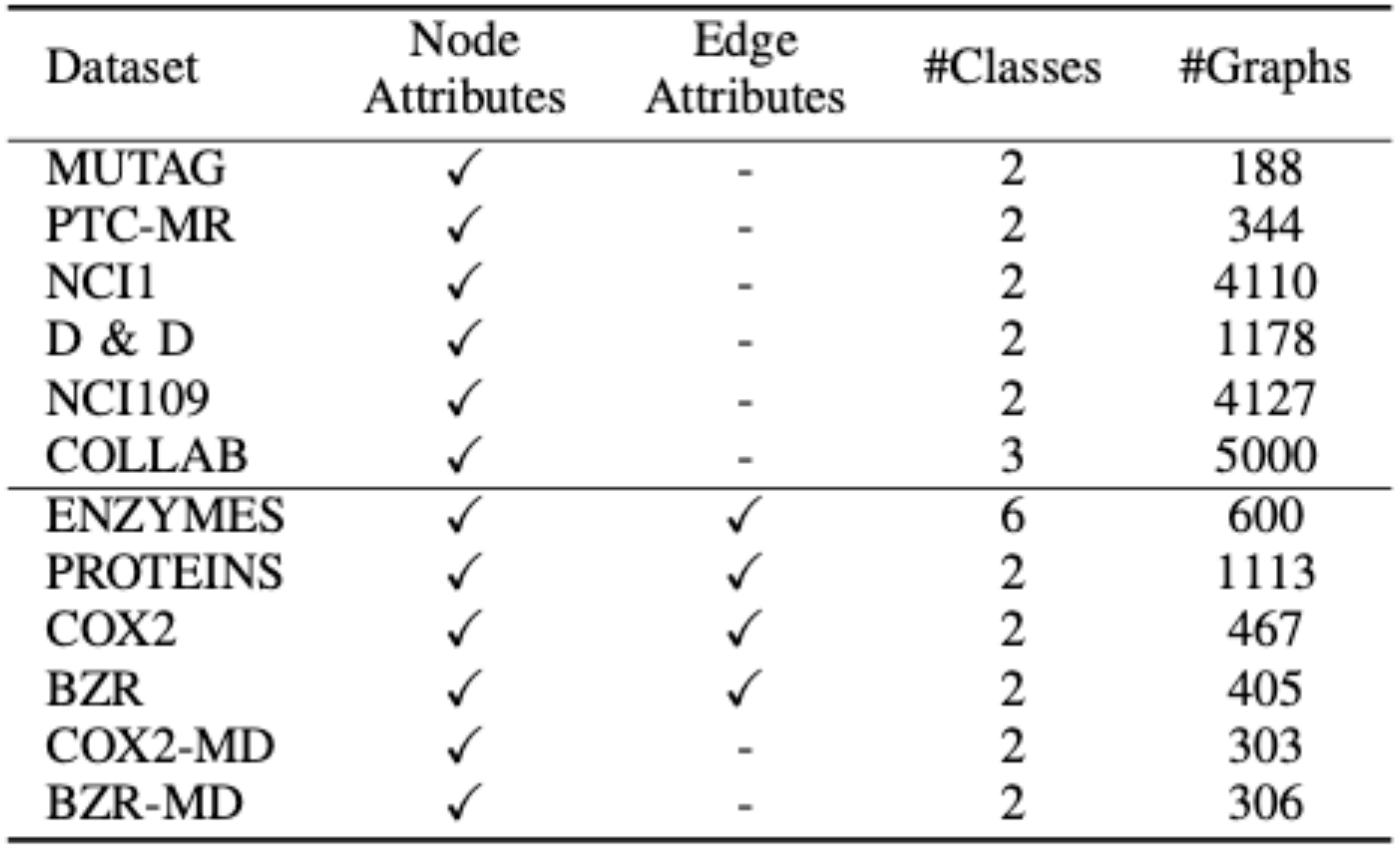
12 benchmark datasets in two categories:

(1) Graphs with **discrete attributes**:

MUTAG, PTC-MR, NCI1, NCI109 and D&D are bioinformatics datasets, and COLLAB is a social network.

(2) Graphs with **continuous attributes**:

COX2, COX2-MD, BZR, BZR-MD, PROTEINS and ENZYMES are bioinformatics datasets.

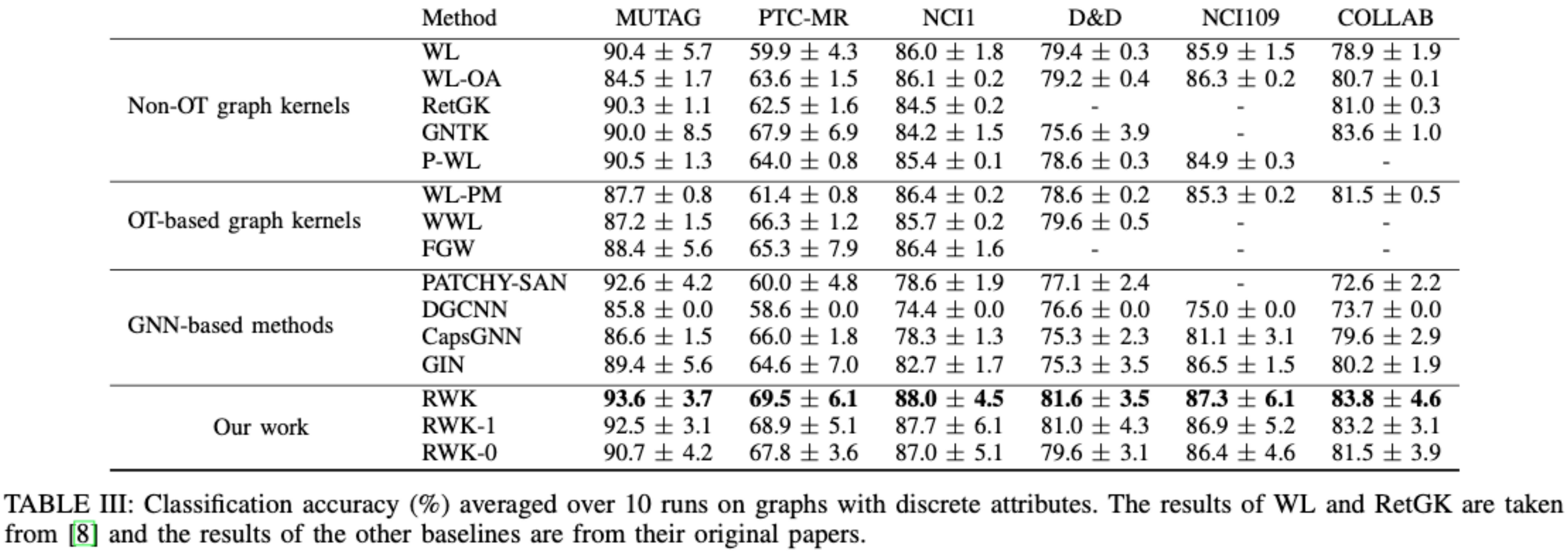


Dataset	Node Attributes	Edge Attributes	#Classes	#Graphs
MUTAG	✓	-	2	188
PTC-MR	✓	-	2	344
NCI1	✓	-	2	4110
D & D	✓	-	2	1178
NCI109	✓	-	2	4127
COLLAB	✓	-	3	5000
ENZYMES	✓	✓	6	600
PROTEINS	✓	✓	2	1113
COX2	✓	✓	2	467
BZR	✓	✓	2	405
COX2-MD	✓	-	2	303
BZR-MD	✓	-	2	306

TABLE II: Dataset statistics.

Experiments

Graph Classification



	Method	MUTAG	PTC-MR	NCI1	D&D	NCI109	COLLAB
Non-OT graph kernels	WL	90.4 ± 5.7	59.9 ± 4.3	86.0 ± 1.8	79.4 ± 0.3	85.9 ± 1.5	78.9 ± 1.9
	WL-OA	84.5 ± 1.7	63.6 ± 1.5	86.1 ± 0.2	79.2 ± 0.4	86.3 ± 0.2	80.7 ± 0.1
	RetGK	90.3 ± 1.1	62.5 ± 1.6	84.5 ± 0.2	-	-	81.0 ± 0.3
	GNTK	90.0 ± 8.5	67.9 ± 6.9	84.2 ± 1.5	75.6 ± 3.9	-	83.6 ± 1.0
	P-WL	90.5 ± 1.3	64.0 ± 0.8	85.4 ± 0.1	78.6 ± 0.3	84.9 ± 0.3	-
OT-based graph kernels	WL-PM	87.7 ± 0.8	61.4 ± 0.8	86.4 ± 0.2	78.6 ± 0.2	85.3 ± 0.2	81.5 ± 0.5
	WWL	87.2 ± 1.5	66.3 ± 1.2	85.7 ± 0.2	79.6 ± 0.5	-	-
	FGW	88.4 ± 5.6	65.3 ± 7.9	86.4 ± 1.6	-	-	-
GNN-based methods	PATCHY-SAN	92.6 ± 4.2	60.0 ± 4.8	78.6 ± 1.9	77.1 ± 2.4	-	72.6 ± 2.2
	DGCNN	85.8 ± 0.0	58.6 ± 0.0	74.4 ± 0.0	76.6 ± 0.0	75.0 ± 0.0	73.7 ± 0.0
	CapsGNN	86.6 ± 1.5	66.0 ± 1.8	78.3 ± 1.3	75.3 ± 2.3	81.1 ± 3.1	79.6 ± 2.9
	GIN	89.4 ± 5.6	64.6 ± 7.0	82.7 ± 1.7	75.3 ± 3.5	86.5 ± 1.5	80.2 ± 1.9
Our work	RWK	**93.6 ± 3.7**	**69.5 ± 6.1**	**88.0 ± 4.5**	**81.6 ± 3.5**	**87.3 ± 6.1**	**83.8 ± 4.6**
	RWK-1	92.5 ± 3.1	68.9 ± 5.1	87.7 ± 6.1	81.0 ± 4.3	86.9 ± 5.2	83.2 ± 3.1
	RWK-0	90.7 ± 4.2	67.8 ± 3.6	87.0 ± 5.1	79.6 ± 3.1	86.4 ± 4.6	81.5 ± 3.9

TABLE III: Classification accuracy (%) averaged over 10 runs on graphs with discrete attributes. The results of WL and RetGK are taken from [8] and the results of the other baselines are from their original papers.

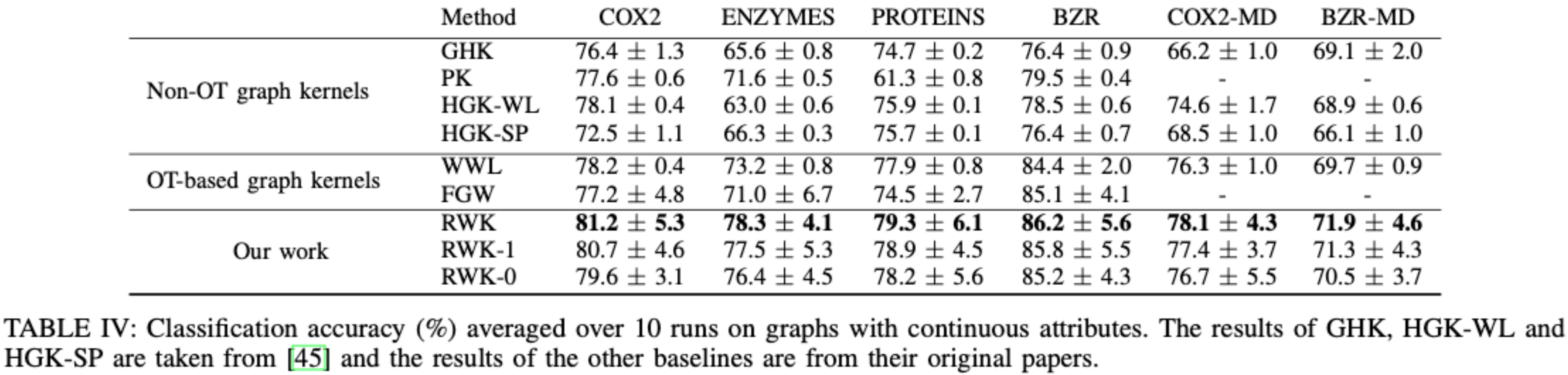
RWK: with using 2-hop feature local variations(default)

RWK-1: with using 1-hop feature local variations

RWK-0: without using any feature local variations

Experiments

Graph Classification



	Method	COX2	ENZYMES	PROTEINS	BZR	COX2-MD	BZR-MD
Non-OT graph kernels	GHK	76.4 ± 1.3	65.6 ± 0.8	74.7 ± 0.2	76.4 ± 0.9	66.2 ± 1.0	69.1 ± 2.0
	PK	77.6 ± 0.6	71.6 ± 0.5	61.3 ± 0.8	79.5 ± 0.4	-	-
	HGK-WL	78.1 ± 0.4	63.0 ± 0.6	75.9 ± 0.1	78.5 ± 0.6	74.6 ± 1.7	68.9 ± 0.6
	HGK-SP	72.5 ± 1.1	66.3 ± 0.3	75.7 ± 0.1	76.4 ± 0.7	68.5 ± 1.0	66.1 ± 1.0
OT-based graph kernels	WWL	78.2 ± 0.4	73.2 ± 0.8	77.9 ± 0.8	84.4 ± 2.0	76.3 ± 1.0	69.7 ± 0.9
	FGW	77.2 ± 4.8	71.0 ± 6.7	74.5 ± 2.7	85.1 ± 4.1	-	-
Our work	RWK	**81.2 ± 5.3**	**78.3 ± 4.1**	**79.3 ± 6.1**	**86.2 ± 5.6**	**78.1 ± 4.3**	**71.9 ± 4.6**
	RWK-1	80.7 ± 4.6	77.5 ± 5.3	78.9 ± 4.5	85.8 ± 5.5	77.4 ± 3.7	71.3 ± 4.3
	RWK-0	79.6 ± 3.1	76.4 ± 4.5	78.2 ± 5.6	85.2 ± 4.3	76.7 ± 5.5	70.5 ± 3.7

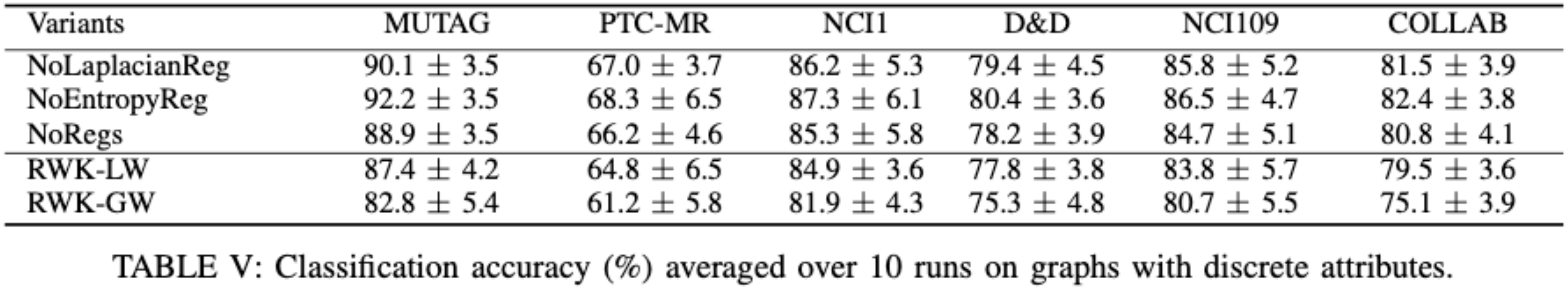
TABLE IV: Classification accuracy (%) averaged over 10 runs on graphs with continuous attributes. The results of GHK, HGK-WL and HGK-SP are taken from [45] and the results of the other baselines are from their original papers.

No baseline could achieve best performance on all datasets

RWK performs better than RWK-0, RWK-1

Experiments

Graph Classification



Variants	MUTAG	PTC-MR	NCI1	D&D	NCI109	COLLAB
NoLaplacianReg	90.1 ± 3.5	67.0 ± 3.7	86.2 ± 5.3	79.4 ± 4.5	85.8 ± 5.2	81.5 ± 3.9
NoEntropyReg	92.2 ± 3.5	68.3 ± 6.5	87.3 ± 6.1	80.4 ± 3.6	86.5 ± 4.7	82.4 ± 3.8
NoRegs	88.9 ± 3.5	66.2 ± 4.6	85.3 ± 5.8	78.2 ± 3.9	84.7 ± 5.1	80.8 ± 4.1
RWK-LW	87.4 ± 4.2	64.8 ± 6.5	84.9 ± 3.6	77.8 ± 3.8	83.8 ± 5.7	79.5 ± 3.6
RWK-GW	82.8 ± 5.4	61.2 ± 5.8	81.9 ± 4.3	75.3 ± 4.8	80.7 ± 5.5	75.1 ± 3.9

TABLE V: Classification accuracy (%) averaged over 10 runs on graphs with discrete attributes.

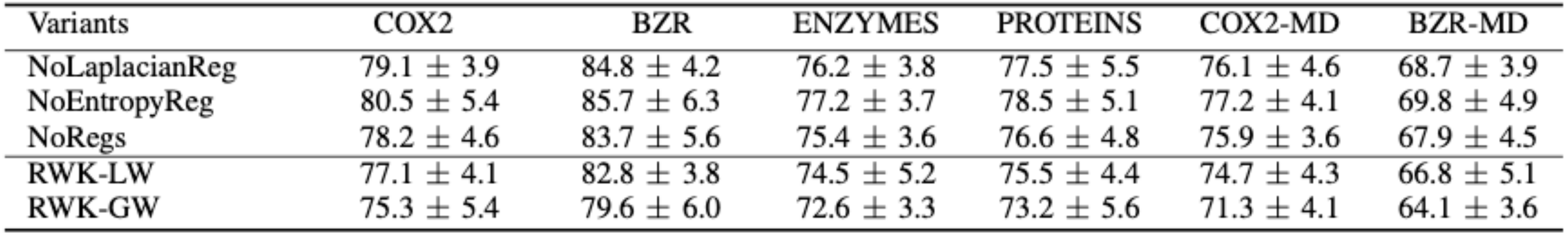
$$
\begin{aligned}
& RW(\mu,\nu) \\
&= \min_{\gamma\in\pi(\mu,\nu)} \left\langle \gamma, \mathbf{C}^V \right\rangle_F + \beta_1 LW(\mu,\nu) + \beta_2 GW(\mu,\nu) \\
&= \min_{\gamma\in\pi(\mu,\nu)} \left\langle \gamma, \mathbf{C}^V \right\rangle_F + \beta_1 \left\langle \gamma, \mathbf{C}^N \right\rangle_F + \beta_1 \Theta_w(\gamma) + \left\langle \gamma, \beta_2 \left(L_2\left(\mathbf{C}^P_\mu, \mathbf{C}^P_\nu\right) \otimes \gamma \right) \right\rangle_F - \beta_2 \left(\lambda_g \Theta_g(\gamma) \right)
\end{aligned}
$$

LW and GW are crucial to the performance

regularization terms reduce variance and boost performance

Experiments

Graph Classification



Variants	COX2	BZR	ENZYMES	PROTEINS	COX2-MD	BZR-MD
NoLaplacianReg	79.1 ± 3.9	84.8 ± 4.2	76.2 ± 3.8	77.5 ± 5.5	76.1 ± 4.6	68.7 ± 3.9
NoEntropyReg	80.5 ± 5.4	85.7 ± 6.3	77.2 ± 3.7	78.5 ± 5.1	77.2 ± 4.1	69.8 ± 4.9
NoRegs	78.2 ± 4.6	83.7 ± 5.6	75.4 ± 3.6	76.6 ± 4.8	75.9 ± 3.6	67.9 ± 4.5
RWK-LW	77.1 ± 4.1	82.8 ± 3.8	74.5 ± 5.2	75.5 ± 4.4	74.7 ± 4.3	66.8 ± 5.1
RWK-GW	75.3 ± 5.4	79.6 ± 6.0	72.6 ± 3.3	73.2 ± 5.6	71.3 ± 4.1	64.1 ± 3.6

TABLE VI: Classification accuracy (%) averaged over 10 runs on graphs with continuous attributes.

$$
\begin{aligned}
& RW(\mu,\nu) \\
= & \min_{\gamma\in\pi(\mu,\nu)} \left\langle \gamma, \mathbf{C}^V \right\rangle_F + \beta_1 LW(\mu,\nu) + \beta_2 GW(\mu,\nu) \\
= & \min_{\gamma\in\pi(\mu,\nu)} \left\langle \gamma, \mathbf{C}^V \right\rangle_F + \beta_1 \left\langle \gamma, \mathbf{C}^N \right\rangle_F + \beta_1 \Theta_w(\gamma) + \left\langle \gamma, \beta_2 \left(L_2\left(\mathbf{C}^P_\mu, \mathbf{C}^P_\nu\right) \otimes \gamma \right) \right\rangle_F - \beta_2 \left(\lambda_g \Theta_g(\gamma) \right)
\end{aligned}
$$

LW and GW are crucial to the performance

regularization terms reduce variance and boost performance

Experiments

Running time: OT-based graph kernels

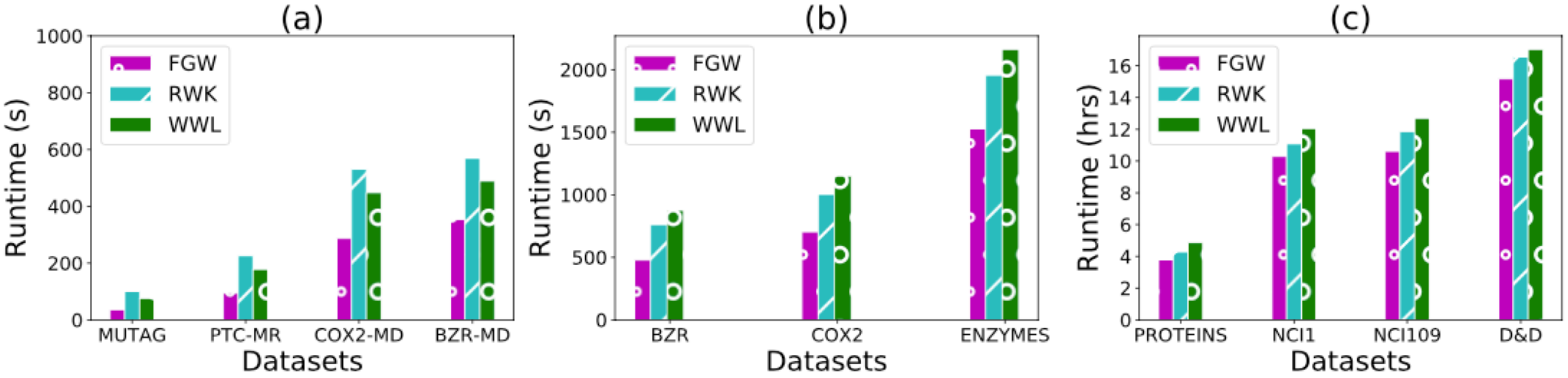


Fig. 3: Running time averaged over 10 runs on graphs with discrete and continuous attributes. There are no result for the COLLAB dataset because all methods take more than 24 hours to obtain the results

FGW is fastest

RWK is slowest in (a)

RWK is faster than WWL in (b)+(c)

Experiments

Running time: OT-based graph kernels

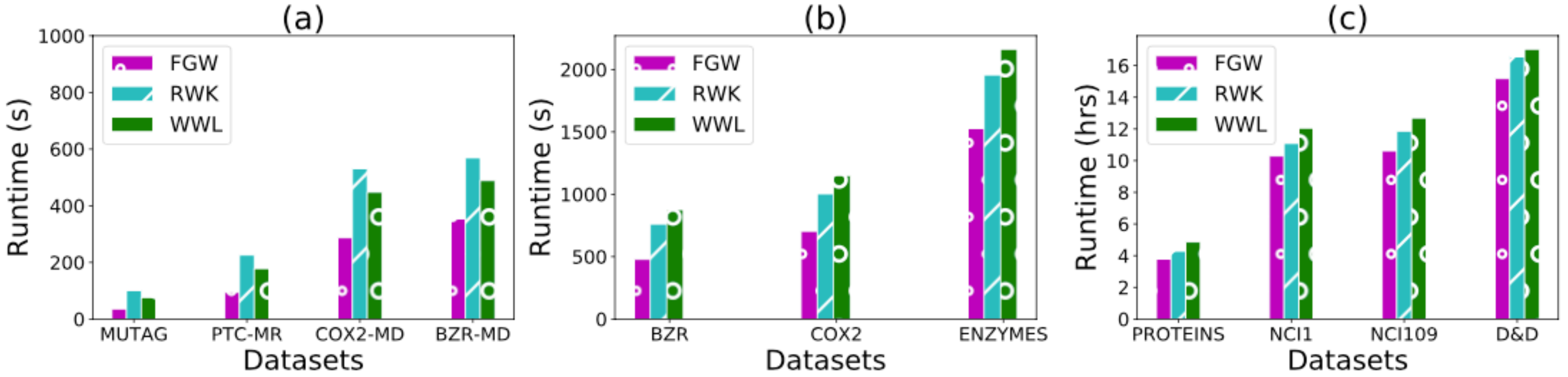
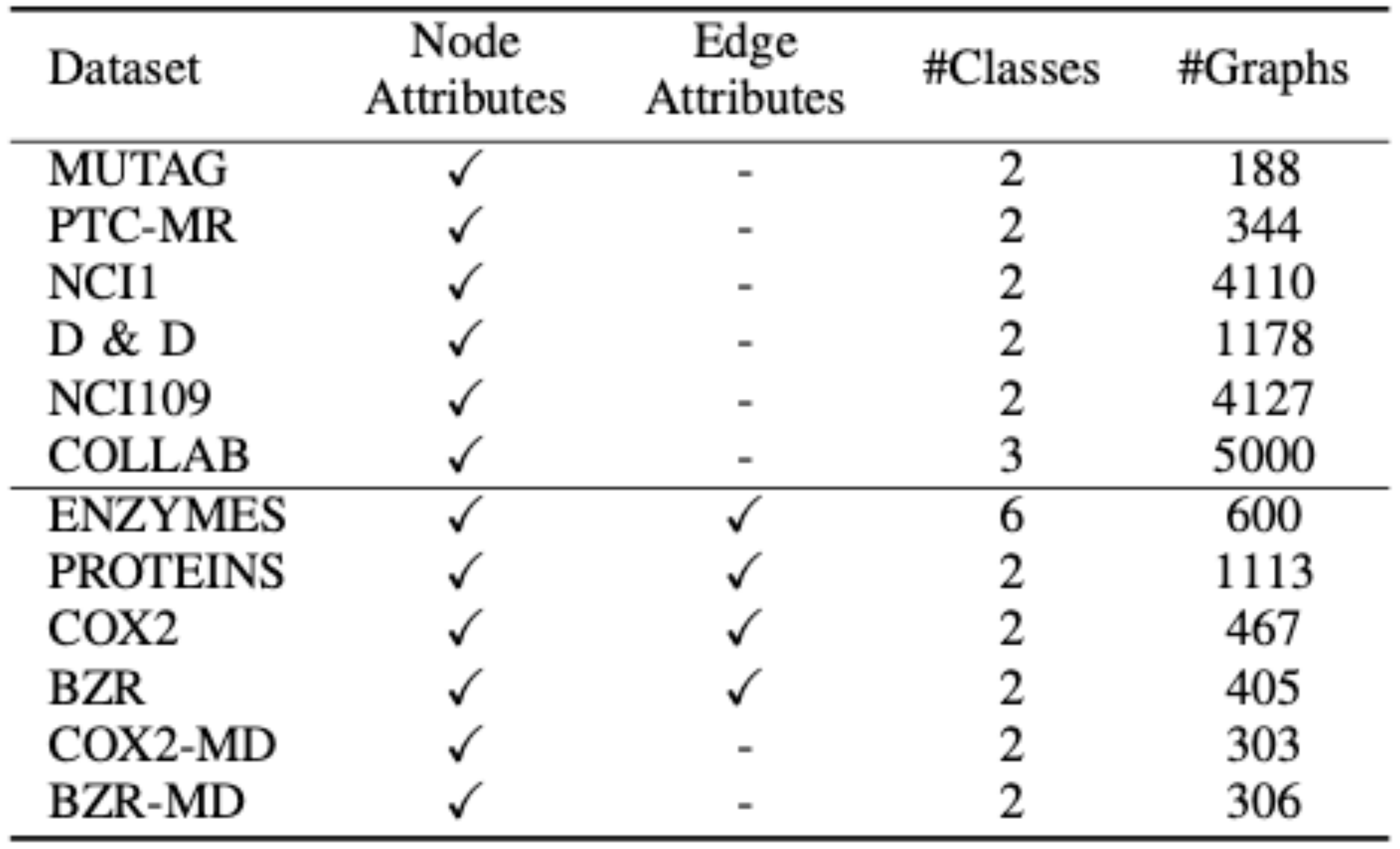


Fig. 3: Running time averaged over 10 runs on graphs with discrete and continuous attributes. There are no result for the COLLAB dataset because all methods take more than 24 hours to obtain the results

FGW is fastest

RWK is slowest on 4 smaller datasets

RWK is faster than WWL on 7 larger datasets



Dataset	Node Attributes	Edge Attributes	#Classes	#Graphs
MUTAG	✓	-	2	188
PTC-MR	✓	-	2	344
NCI1	✓	-	2	4110
D & D	✓	-	2	1178
NCI109	✓	-	2	4127
COLLAB	✓	-	3	5000
ENZYMES	✓	✓	6	600
PROTEINS	✓	✓	2	1113
COX2	✓	✓	2	467
BZR	✓	✓	2	405
COX2-MD	✓	-	2	303
BZR-MD	✓	-	2	306

TABLE II: Dataset statistics.

Thanks!