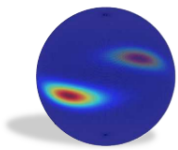
Geometric Learning on Graph Structured Data

Asiri Wijesinghe

asiri.wijesinghe@anu.edu.au
School of Computing
College of Engineering and Computer Science
Australian National University, Canberra, Australia

Geometric Graph Learning

Surfaces



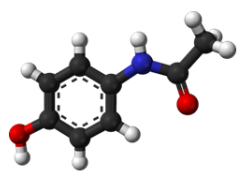
Distributions

Graphs / Networks

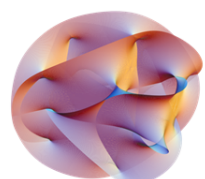
Functions on Manifolds

Hyperbolic spaces

Hyper-surfaces



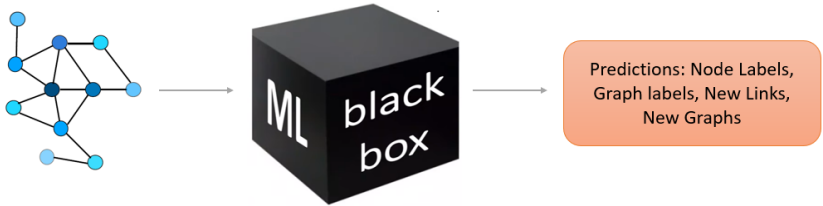
Molecules



General manifolds

Research Questions

- How can we design more powerful machine learning models to take the advantage of rich structure of graphs for better prediction?



  - Graph representation learning
  - Graph similarity learning

Key Challenges

- How to improve the representation power of machine learning models?
- How to effectively design a model to preserve intrinsic properties of graphs during the learning?
- How to ensure theoretical properties to guarantee the convergence and numerical stability for machine learning techniques on graphs?

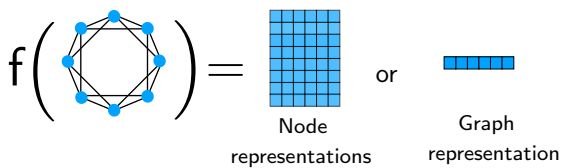
Thesis Contributions

* **Part 1:** Graph Representation Learning
  - **Spatial GNNs** - A. Wijesinghe and Q. Wang. A New Perspective on "How Graph Neural Networks Go Beyond Weisfeiler-Lehman?". (ICLR 2022)
  - **Spectral GNNs** - A. Wijesinghe and Q. Wang. DFNets: Spectral CNNs for Graphs with Feedback-Looped Filters. (NeurIPS 2019)
  - **Diffusion GNNs** - A. Wijesinghe and Q. Wang. Dynamic PageRank for Graph Neural Networks. (Under Review)
* **Part 2:** Graph Similarity Learning
  - **Optimal Transport Graph Kernel** - A. Wijesinghe, Q. Wang and S. Gould. A Regularized Wasserstein Framework for Graph Kernels. (ICDM 2021)

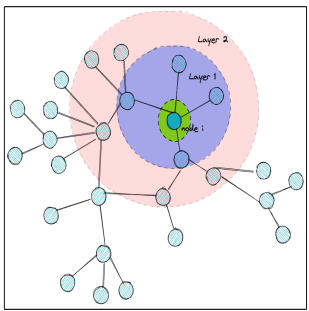
Graph Representation Learning

Graph Neural Networks:

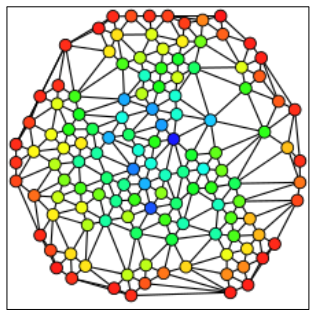
How to represent nodes/graphs in a vector space?



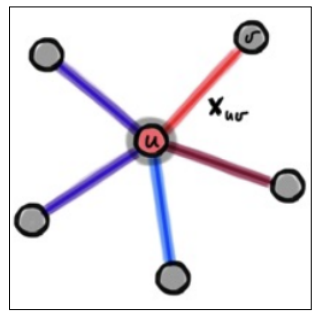
Graph Representation Learning



Spatial GNNs

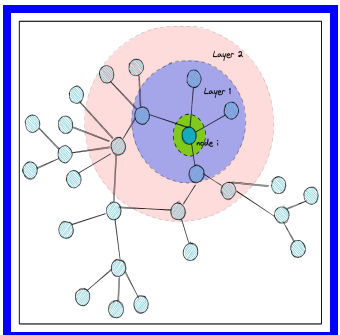


Spectral GNNs

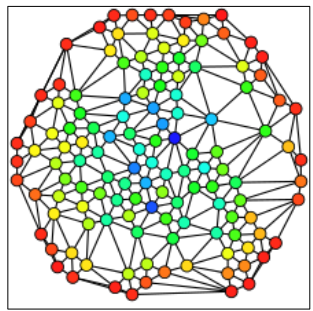


Diffusion GNNs

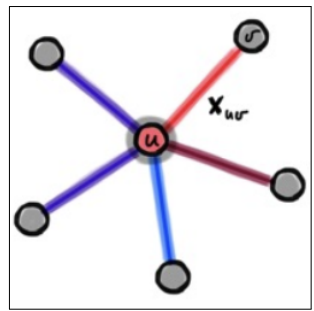
Graph Representation Learning



Spatial GNNs [1]



Spectral GNNs

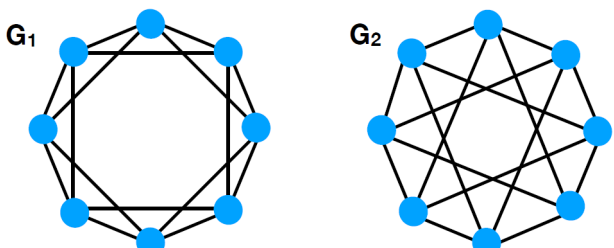


Diffusion GNNs

[1] A. Wijesinghe and Q. Wang. **A New Perspective on "How Graph Neural Networks Go Beyond Weisfeiler-Lehman?"**. (ICLR 2022)

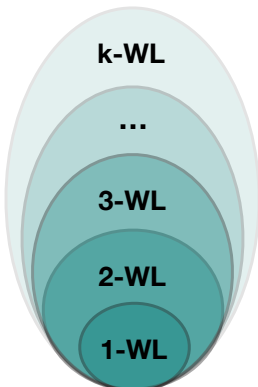
Research Question

- How to design expressive yet simple GNNs that can go beyond the WL test with a theoretically provable guarantee?



Current State

- **Known results:**



- $>$ 1-WL

  e.g., k-dimensional GNN (k-GNN)
  [Morris et al. 2019]

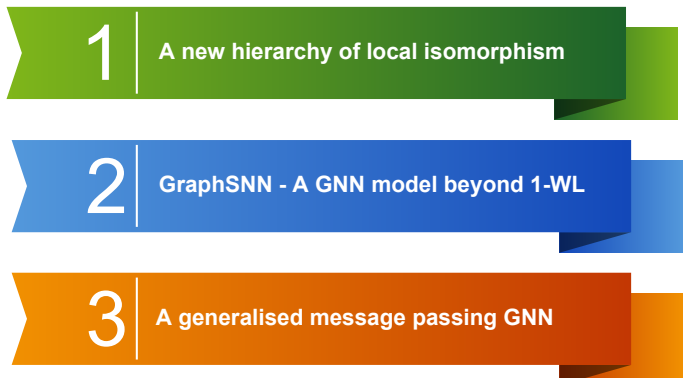
- $\equiv$ 1-WL

  e.g., Graph Isomorphism Network (GIN)
  [Xu et al. 2019]

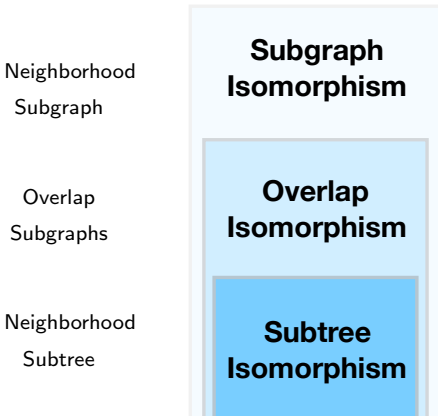
- $<$ 1-WL

  e.g., Graph Convolutional Network (GCN)
  [Kipf and Welling 2017]

Contributions



A New Hierarchy of Local Isomorphism



Theorem

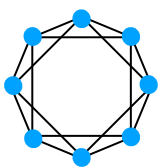
If $S_i \simeq_{subgraph} S_j$, then $S_i \simeq_{overlap} S_j$, but not vice versa.

Theorem

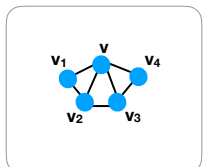
If $S_i \simeq_{overlap} S_j$, then $S_i \simeq_{subtree} S_j$, but not vice versa.

GraphSNN - A GNN Model Beyond 1-WL

- For each vertex v, the **neighborhood subgraph** S_v is the subgraph induced by $\mathcal{N}(v) \cup \{v\}$.

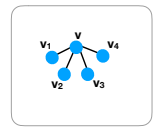


Graph

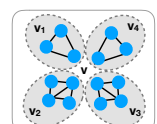


Neighborhood subgraph

$\Longrightarrow$



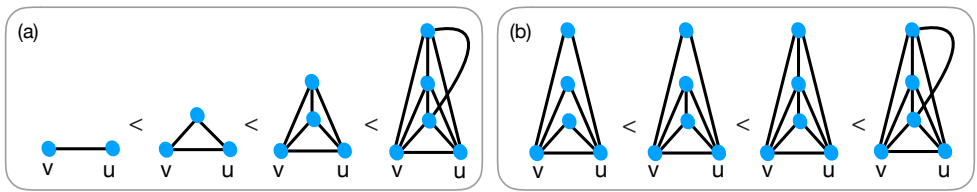
Neighborhood subtree (GIN)



Overlap subgraphs (our work)

- For two adjacent vertices v and u, the **overlap subgraph** is $S_{vu} = S_v \cap S_u$.

GraphSNN - A GNN Model Beyond 1-WL



- **An instance:**

$$A_{vu} = \frac{|E_{vu}|}{|V_{vu}| \cdot |V_{vu} - 1|} |V_{vu}|^{\lambda},\ \lambda > 0$$

GraphSNN - A GNN Model Beyond 1-WL

- A single layer:

$$h_v^{(t)} = \textsc{Mlp}\Big(\gamma^{(t)}\Big(\sum_{u\in\mathcal{N}(v)} \tilde{A}_{vu} + 1\Big)h_v^{(t-1)} + \sum_{u\in\mathcal{N}(v)} \Big(\tilde{A}_{vu} + 1\Big)h_u^{(t-1)}\Big)$$

- Multiple layers (same as GIN)[1]:

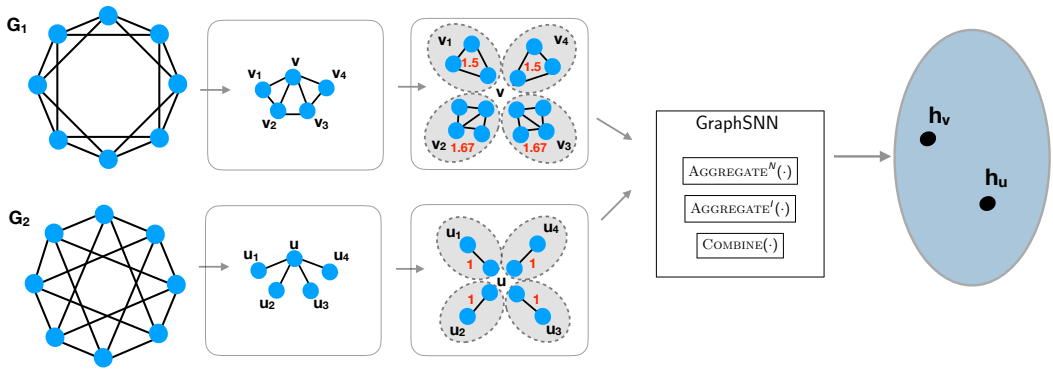
$$h_G = \textsc{Concat}(\textsc{Readout}(\{\{h_v^{(t)} | v \in V\}\}) | t = 1, \ldots, k)$$

Theorem

GraphSNN is strictly more expressive than 1-WL.

[1] code: https://github.com/wokas36/GraphSNN

GraphSNN - A GNN Model Beyond 1-WL



A Generalised Message Passing GNN

- A layer of a **Message-Passing GNN** is defined as:
  1. Aggregate "messages" from neighbors $\mathcal{N}(v)$

     ~~$h^{(t)} = \textsc{Aggregate}\Big(\{\!\{h_u^{(t)} | u \in \mathcal{N}(v)\}\!\}\Big)$~~

     $\hookrightarrow\ m_a^{(t)} = \textsc{Aggregate}^N\Big(\{\!\{(\tilde{A}_{vu}, h_u^{(t)}) | u \in \mathcal{N}(v)\}\!\}\Big)$

     $\hookrightarrow\ m_v^{(t)} = \textsc{Aggregate}^I\Big(\{\!\{\tilde{A}_{vu} | u \in \mathcal{N}(v)\}\!\}\Big) h_v^{(t)}$

  2. Combine with its own "message" $h_v^{(t)}$

     ~~$h_v^{(t+1)} = \textsc{Combine}\Big(h_v^{(t)}, h^{(t)}\Big)$~~

     $\hookrightarrow\ h_v^{(t+1)} = \textsc{Combine}\Big(m_v^{(t)}, m_a^{(t)}\Big)$

Numerical Experiments

- Classification on Open Graph Benchmark (OGB) datasets, including four molecular graph datasets and one protein-protein association network.

Method	ogbg-molhiv	ogbg-moltox21	ogbg-moltoxcast	ogbg-ppa	ogbg-molpcba
GIN	75.58±1.40	74.91±0.51	63.41±0.74	68.92±1.00	22.66±0.28
GIN+VN	75.20±1.30	76.21±0.82	66.18±0.68	70.37±1.07	27.03±0.23
GSN	77.99±1.00	-	-	-	-
PNA	79.05±1.30	-	-	-	28.38±0.35
ID-GNN	78.30±2.00	-	-	-	-
Deep LRP	77.19±1.40	-	-	-	-
GraphSNN	78.51±1.70	75.45±1.10	65.40±0.71	70.66±1.65	24.96±1.50
GraphSNN+VN	**79.72±1.83**	**76.78±1.27**	**67.68±0.92**	**72.02±1.48**	**28.50±1.68**

Table: Classification accuracy on large graph classification.

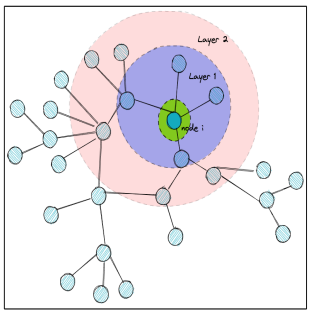
Numerical Experiments

- Classification w.r.t GraphSNN$_M$ models by replacing GCN, GAT, GIN, and GraphSAGE aggregation schemes by our aggregation scheme.

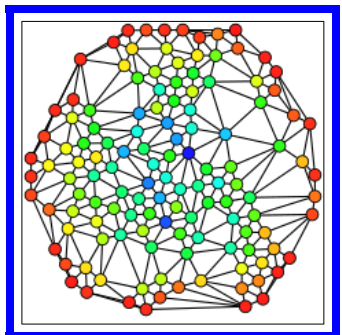
Method	Cora	Citeseer	Pubmed	NELL	ogbn-arxiv
GCN	81.5 ± 0.4	70.3 ± 0.5	79.0 ± 0.5	66.0 ± 1.7	71.74 ± 0.29
GraphSNN$_{GCN}$	**83.1 ± 1.8**	**72.3 ± 1.5**	**79.8 ± 1.2**	**68.3 ± 1.6**	**72.20 ± 0.90**
GAT	83.0 ± 0.6	72.6 ± 0.6	78.5 ± 0.3	-	-
GraphSNN$_{GAT}$	**83.8 ± 1.2**	**73.5 ± 1.6**	**79.6 ± 1.4**	-	-
GIN	77.6 ± 1.1	66.1 ± 1.5	77.0 ± 1.2	61.5 ± 2.3	-
GraphSNN$_{GIN}$	**79.2 ± 1.7**	**68.3 ± 1.5**	**78.8 ± 1.3**	**63.8 ± 2.7**	-
GraphSAGE	79.2 ± 3.7	71.6 ± 1.9	77.4 ± 2.2	63.7 ± 5.2	71.49 ± 0.27
GraphSNN$_{GraphSAGE}$	**80.5 ± 2.5**	**72.7 ± 3.2**	**79.0 ± 3.5**	**66.3 ± 5.6**	**71.80 ± 0.70**

Table: Classification accuracy on semi-supervised node classification.

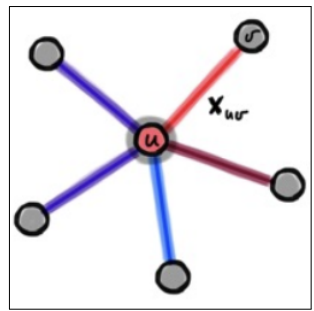
Graph Representation Learning



Spatial GNNs



Spectral GNNs [2]

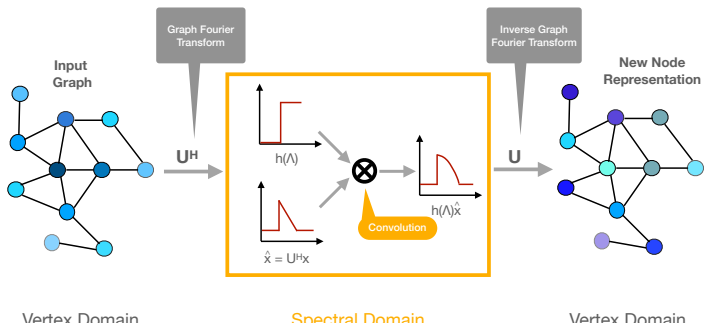


Diffusion GNNs

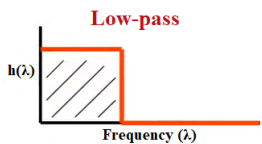
[2] A. Wijesinghe and Q. Wang. **DFNets: Spectral CNNs for Graphs with Feedback-Looped Filters**. (NeurIPS 2019)

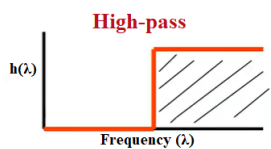
Research Question

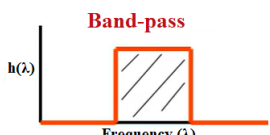
- How to design spectral graph filters and tackle the problem of designing an effective, yet efficient GNNs with spectral graph filters?

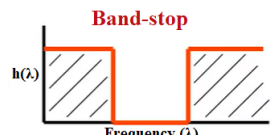


Current State







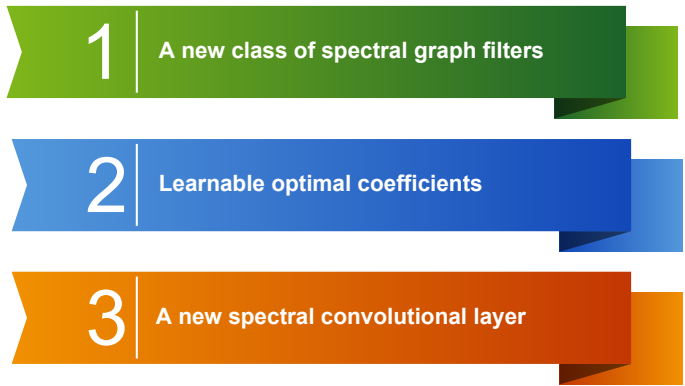


- Rational polynomial filters

  e.g., Cayley
  [Ron et al. 2017]
- Polynomial filters

  e.g., Lanczos
  [Renjie et al. 2019]
- Basis-dependent filters

Contributions



A New Class of Spectral Graph Filters

- Feedback-looped filters belong to a class of ARMA filters.

$$h_{\psi,\phi}(L)x = \Big(I + \sum_{j=1}^{p} \psi_j L^j\Big)^{-1} \Big(\sum_{j=0}^{q} \phi_j L^j\Big) x \tag{1}$$

- Feedback-looped filters use the following approximation.

$$\bar{x}^{(0)} = x \text{ and } \bar{x}^{(t)} = -\sum_{j=1}^{p} \psi_j \tilde{L}^j \bar{x}^{(t-1)} + \sum_{j=0}^{q} \phi_j \tilde{L}^j x \tag{2}$$

Learnable Optimal Coefficients

- The frequency response of feedback-looped filters is defined as:

$$h(\lambda_i) = \frac{\sum_{j=0}^{q} \phi_j \lambda_i^j}{1 + \sum_{j=1}^{p} \psi_j \lambda_i^j}. \tag{3}$$

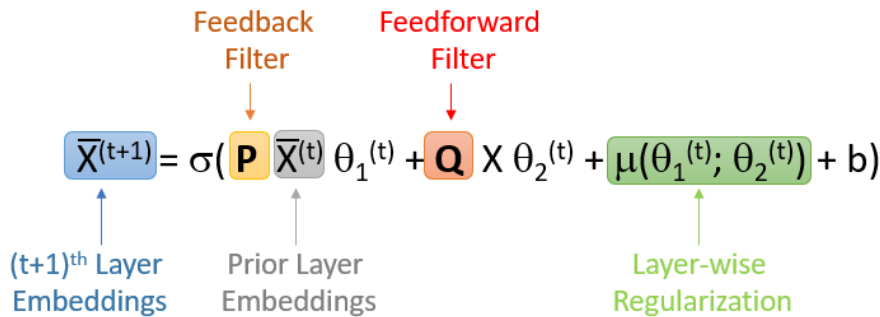
- The stable coefficients ψ and ϕ can be learned by a convex constrained least-squares optimization problem:

$$\textbf{minimize}_{\psi,\phi} \; ||\hat{h} + diag(\hat{h})\alpha\psi - \beta\phi||_2 \tag{4}$$

$$\textbf{subject to} \; ||\alpha\psi||_\infty \leq \gamma \text{ and } \gamma<1$$

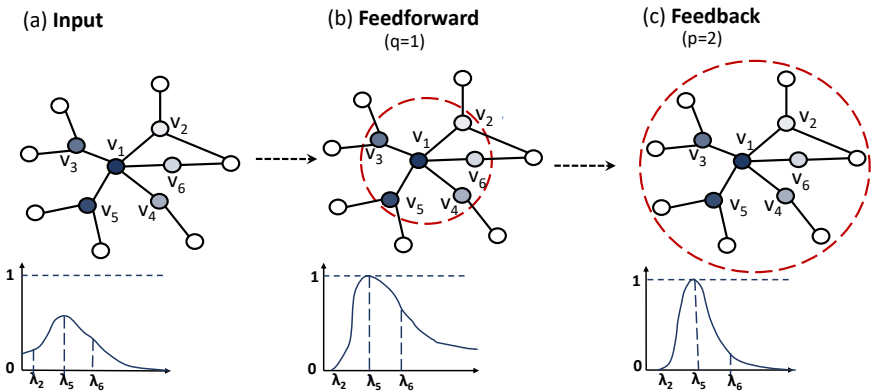
A New Spectral Convolutional Layer

- Propagation rule of a spectral convolutional layer is defined as: [2]



[2] code: https://github.com/wokas36/DFNets

A New Spectral Convolutional Layer

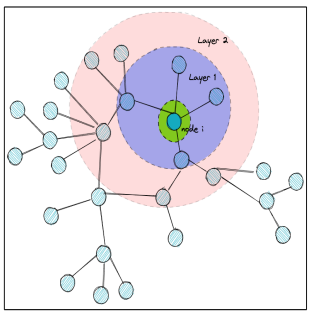


Numerical Experiments

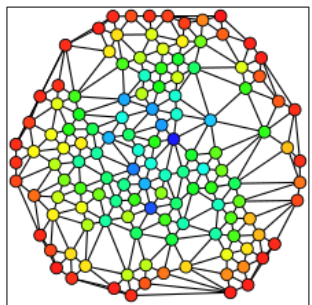
Model	Cora	Citeseer	Pubmed	NELL
SemiEmb	59.0	59.6	71.1	26.7
LP	68.0	45.3	63.0	26.5
DeepWalk	67.2	43.2	65.3	58.1
ICA	75.1	69.1	73.9	23.1
Planetoid*	64.7	75.7	77.2	61.9
Chebyshev	81.2	69.8	74.4	-
GCN	81.5	70.3	79.0	66.0
LNet	79.5	66.2	78.3	-
AdaLNet	80.4	68.7	78.1	-
CayleyNet	81.9*	-	-	-
$ARMA_1$	84.7	73.8	81.4	-
GAT	83.0	72.5	79.0	-
GCN + DenseBlock	82.7 $\pm$ 0.5	71.3 $\pm$ 0.3	81.5 $\pm$ 0.5	66.4 $\pm$ 0.3
GAT + Dense Block	83.8 $\pm$ 0.3	73.1 $\pm$ 0.3	81.8 $\pm$ 0.3	-
DFNet (ours)	**85.2** $\pm$ **0.5**	**74.2** $\pm$ **0.3**	**84.3** $\pm$ **0.4**	**68.3** $\pm$ **0.4**
DFNet-ATT (ours)	**86.0** $\pm$ **0.4**	**74.7** $\pm$ **0.4**	**85.2** $\pm$ **0.3**	**68.8** $\pm$ **0.3**
DF-ATT (ours)	83.4 $\pm$ 0.5	73.1 $\pm$ 0.4	**82.3** $\pm$ **0.3**	**67.6** $\pm$ **0.3**

Table: Classification accuracy on semi-supervised node classification.

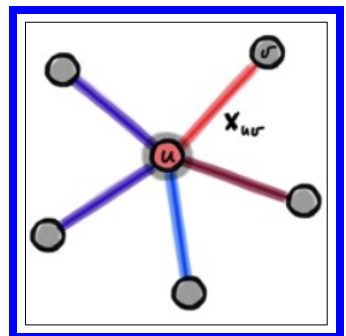
Graph Representation Learning



Spatial GNNs



Spectral GNNs

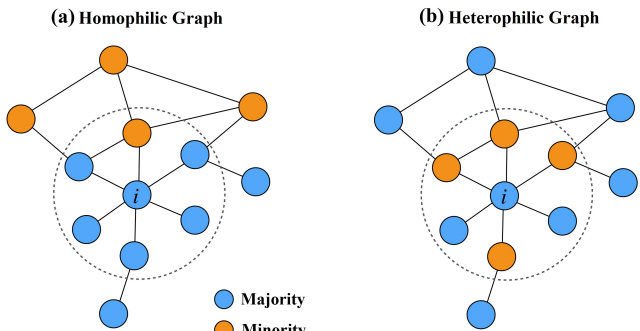


Diffusion GNNs[3]

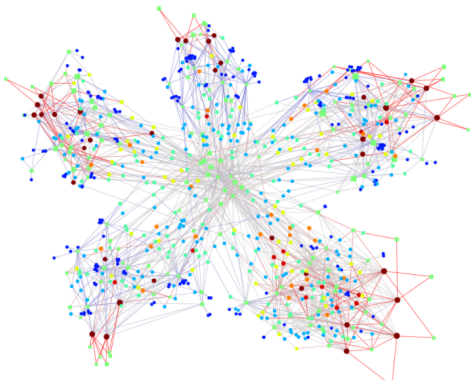
[3] A. Wijesinghe and Q. Wang. **Dynamic PageRank for Graph Neural Networks**. (Under Review)

Research Question

- How to build powerful GNNs by graph diffusion to capture rich and varying graph structures, i.e, homophily and heterophily?

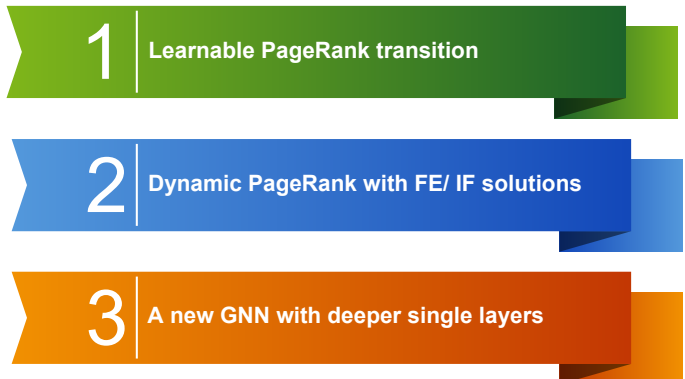


Current State



- Non-homogeneous anisotropic diffusion
  e.g., DGN
  [Dominique et al. 2021]
- Non-homogeneous isotropic diffusion
  e.g., GRAND
  [Chamberlain et al. 2021]
- Homogeneous isotropic diffusion
  e.g., GDC, APPNP, GPRGNN
  [Klicpera et al. 2019, Chien et al. 2021]

Contributions



Learnable PageRank Transition

- Limitations of standard PageRank:
  - Restricting landing probabilities to 1-hop neighbors.
  - Landing probabilities in $\mathbf{P}$ are pre-determined and fixed.

- We reformulate $\mathbf{P}$ with a learnable weighted linear combination of transition probabilities of different lengths.

$$\mathbf{P} = f_\phi(\mathbf{L}) = \sum_{i=1}^{k} \phi_i \mathbf{L}^i \tag{5}$$

Dynamic PageRank with FE Solution

- Standard PageRank can be represented as an iterative scheme;

$$y(t+1) = (1-\alpha)x + \alpha \mathbf{P} y(t) \tag{6}$$

- PageRank with time-dependent teleportation vector $x(t)$;

$$\frac{\partial y(t)}{\partial t} = (1-\alpha)x(t) - (\mathbf{I} - \alpha \mathbf{P})y(t) \tag{7}$$

- Connection with the message-passing GNNs.

$$\mathbf{Y}^{(t)} = \Big((1-\alpha)\mathbf{I} + \alpha \mathbf{P}\Big)\mathbf{Y}^{(t-1)} \tag{8}$$

Dynamic PageRank with IF Solution

- Dynamic PageRank generalizes both personalized PageRank and heat kernel;

$$y(t) = y_{ppr} + exp\Big\{ - t(\mathbf{I} - \alpha\mathbf{P})\Big\}(y(0) - y_{ppr}) \tag{9}$$

- Connection with the message-passing GNNs;

$$\mathbf{Y}^{(t)} = \mathbf{Y}_{ppr} + exp\{-t(\mathbf{I} - \alpha\mathbf{P})\}(\mathbf{Y}^{(0)} - \mathbf{Y}_{ppr}) \tag{10}$$

A New GNN with Deeper Single Layers

- A new GNN with deeper single layer is defined as;

$$\mathbf{Z}^{(l+1)} = \sigma\Big(\sum_{i=1}^{q}\Big(\mathbf{Z}_i^{(l)}\mathbf{W}_i^{(l)} + b_i^{(l)}\Big)\Big) \tag{11}$$

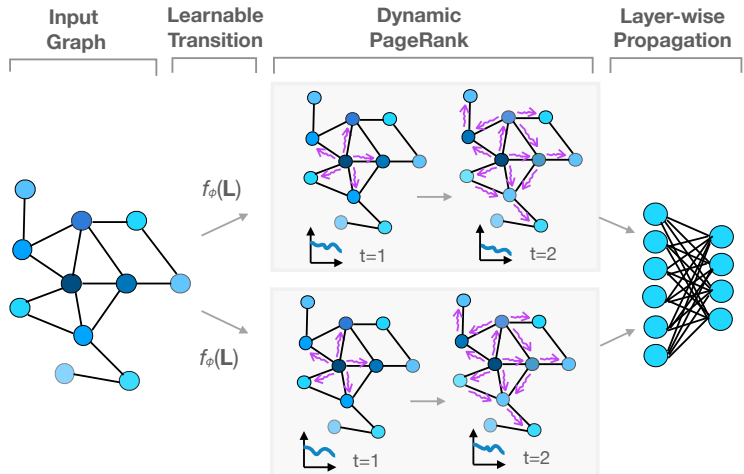
Theorem

When $t \to \infty$, dynamic PageRank diffusion schemes in $\mathcal{Z}_{EF}$ are guaranteed to converge.

Theorem

The scheme of the forward Euler solution is equivalent to the spectral convolution layer.

A New GNN with Deeper Single Layers



Numerical Experiments

Model	Cora	Pubmed	Photo	Computers	
GCN	86.87 ± 0.26	86.97 ± 0.12	90.54 ± 0.21	82.52 ± 0.32	
GAT	87.52 ± 0.24	86.64 ± 0.11	90.09 ± 0.27	81.95 ± 0.38	
APPNP	88.10 ± 0.23	89.15 ± 0.13	91.11 ± 0.26	81.99 ± 0.26	
JKNet	86.97 ± 0.27	87.38 ± 0.13	87.70 ± 0.70	77.80 ± 0.97	
Geom-GCN	85.40 ± 0.26	88.51 ± 0.08	-	-	
U-GCN	84.00	74.08	85.22	-	-
H_2GCN	86.92 ± 1.37	89.40 ± 0.34	-	-	
ASGAT-Cheb	87.50 ± 0.50	89.90 ± 0.90	-	-	
ASGAT-ARMA	87.40 ± 1.10	88.30 ± 1.00	-	-	
NLMLP	76.90 ± 1.80	88.20 ± 0.50	-	-	
NLGCN	88.10 ± 1.00	89.00 ± 0.50	-	-	
GPRGNN	88.65 ± 0.28	89.18 ± 0.15	91.93 ± 0.26	82.90 ± 0.37	
MLP+GCN	87.01 ± 1.35	89.77 ± 0.39	-	-	
PDE-GCN	88.60	89.93	-	-	
DPRN-IF (ours)	**90.18 ± 0.36**	**90.80 ± 1.96**	**93.40 ± 0.41**	**86.11 ± 0.21**	
DPRN-FE (ours)	**90.24 ± 0.42**	**89.97 ± 1.53**	**93.82 ± 0.23**	**85.76 ± 0.17**	

Table: Fully-supervised node classification on homophilic datasets.

Numerical Experiments

Model	Actor	Wisconsin	Cornell	Texas	Chameleon
GCN	30.59 ± 0.23	-	66.72 ± 1.37	75.16 ± 0.96	60.96 ± 0.78
GAT	35.98 ± 0.23	-	76.00 ± 1.01	78.87 ± 0.86	63.9 ± 0.46
APPNP	38.86 ± 0.24	-	91.80 ± 0.63	91.18 ± 0.70	51.91 ± 0.56
JKNet	33.41 ± 0.25	-	66.73 ± 1.73	75.53 ± 1.16	62.92 ± 0.49
Geom-GCN	31.81 ± 0.24	-	55.59 ± 1.59	58.56 ± 1.77	61.06 ± 0.49
U-GCN	-	69.89	69.77	71.72	54.07
H_2GCN	35.86 ± 1.03	86.67 ± 4.69	82.16 ± 4.80	84.86 ± 6.77	57.11 ± 1.58
ASGAT-Cheb	-	86.30 ± 3.70	82.70 ± 8.30	85.10 ± 5.70	66.50 ± 2.80
ASGAT-ARMA	-	84.70 ± 4.40	83.20 ± 5.50	79.50 ± 7.70	65.80 ± 2.20
NLMLP	37.90 ± 1.30	87.30 ± 4.30	84.90 ± 5.70	85.40 ± 3.80	50.70 ± 2.20
GPRGNN	39.30 ± 0.27	-	91.36 ± 0.70	92.92 ± 0.61	67.48 ± 0.40
MLP+GCN	36.24 ± 1.09	86.43 ± 4.00	84.82 ± 4.87	83.60 ± 6.04	68.04 ± 1.86
PDE-GCN	-	91.76	89.73	93.24	66.01
DPRN-IF (ours)	**41.82 ± 0.78**	**88.37 ± 2.09**	**92.78 ± 1.67**	**94.59 ± 0.85**	**68.56 ± 1.92**
DPRN-FE (ours)	**40.03 ± 0.54**	**93.25 ± 1.14**	**93.93 ± 1.04**	**94.75 ± 0.65**	65.00 ± 2.31

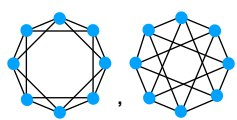
Table: Fully-supervised node classification on heterophilic datasets.

Graph Similarity Learning

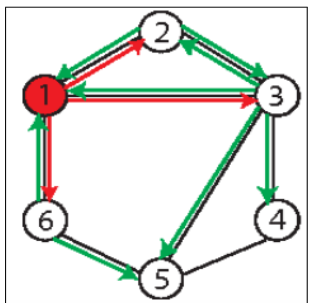
Graph Kernels:

How to compare two graphs in a vector space?

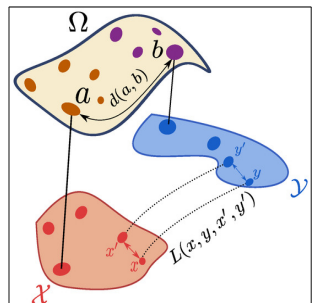
$$f\left(\quad , \quad \right) \in [0,1]$$



Graph Similarity Learning

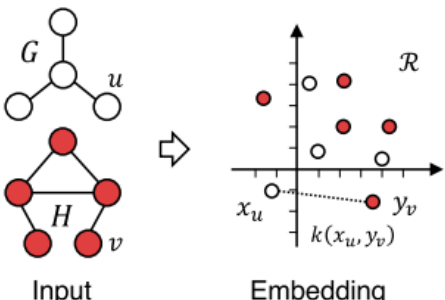


Non-OT Graph Kernels



OT Graph Kernels

Current State



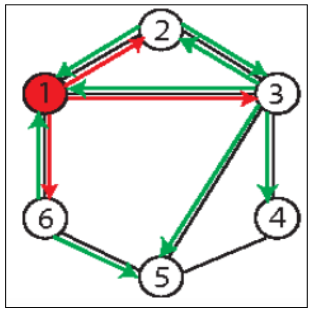
- OT graph kernels

  e.g., WWL, FGW
  [Matteo et al. 2019, Titouan et al. 2019]

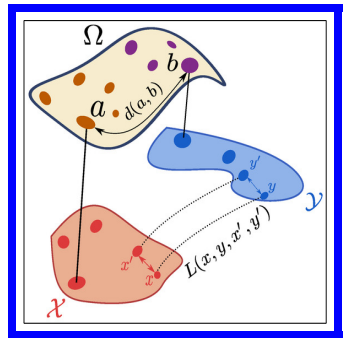
- Non-OT graph kernels

  e.g., WL subtree kernel, Graphlet kernel
  [Nino et al. 2011]

Graph Similarity Learning



Non-OT Graph Kernels



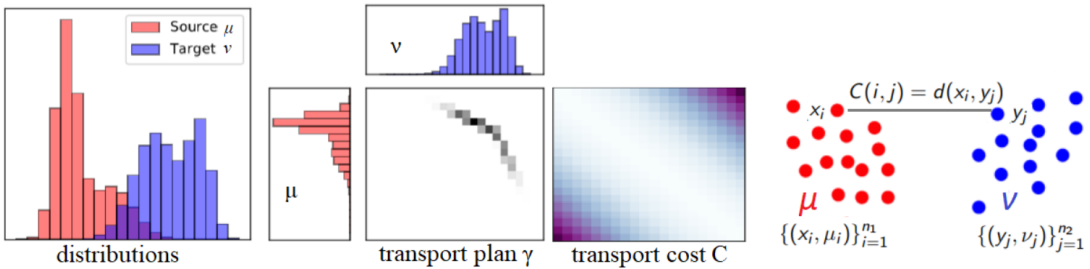
OT Graph Kernels[4]

[4] A. Wijesinghe, Q. Wang and S. Gould. **A Regularized Wasserstein Framework for Graph Kernels**. (ICDM 2021)

Research Question

- How to develop an optimal transport based kernel that can preserve intricate structures on graphs with theoretically guaranteed convergence?

Optimal Transport Theory



- Optimal transport between two discrete distributions,

$$\hat{\gamma} = \underset{\gamma \in \pi(\mu,\nu)}{\operatorname{argmin}} \langle \gamma, \mathbf{C} \rangle_F = \underset{\gamma \in \pi(\mu,\nu)}{\operatorname{argmin}} \sum_{i,j} \gamma(i,j) C(i,j) \tag{12}$$

- Set of probabilistic couplings between metric spaces,

$$\pi(\mu,\nu) = \left\{ \gamma \in \mathbb{R}_+^{n_1 \times n_2} \mid \gamma \mathbf{1}_{n_2} = \mu, \gamma^T \mathbf{1}_{n_1} = \nu \right\}$$

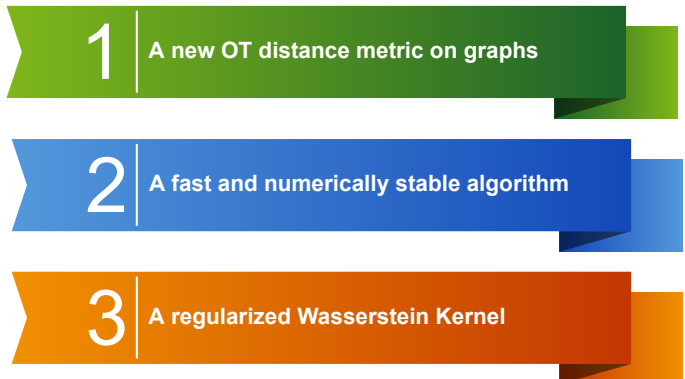
Regularized Optimal Transport

- Regularized optimal transport between two probability distributions,

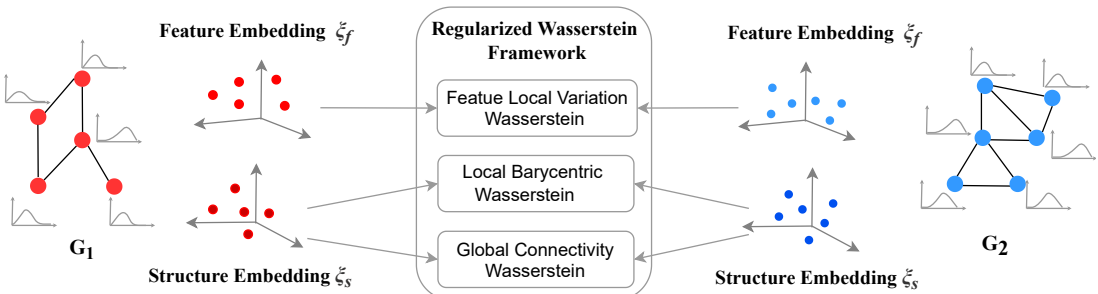
$$\hat{\gamma} = \underset{\gamma \in \pi(\mu,\nu)}{\operatorname{argmin}} \left\langle \gamma, \mathbf{C} \right\rangle_F + \lambda \Theta(\gamma)$$

- Why regularize optimal transport?
  - Smooth the distance estimation.
  - Encode prior knowledge on the data.
  - Robust and guarantee the convergence.
  - Numerical stability in optimization.
  - Fast algorithms to solve the OT problem.

Contributions



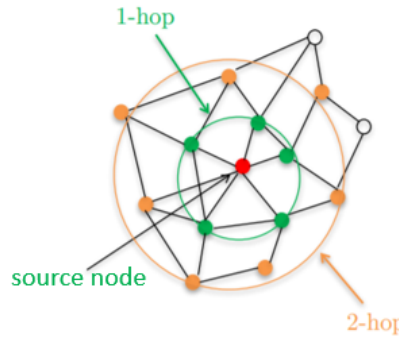
Overview


Feature Embedding ξ_f
Regularized Wasserstein Framework
Feature Embedding ξ_f
Featue Local Variation Wasserstein
Local Barycentric Wasserstein
Global Connectivity Wasserstein
G_1
Structure Embedding ξ_s
Structure Embedding ξ_s
G_2

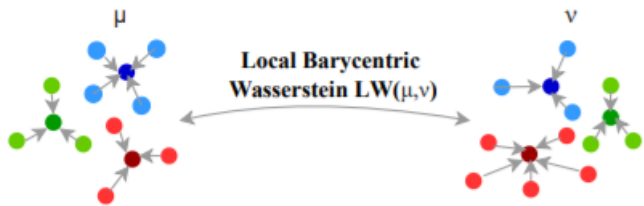
Feature Local Variation

- Feature local variation is used to quantify how graph signals change from a vertex to its neighboring vertices.

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

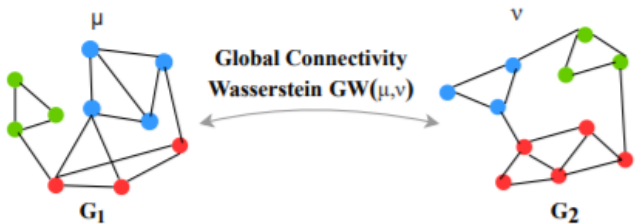


Local Barycentric Wasserstein Distance



$$LW(\mu,\nu) = \min_{\gamma \in \pi(\mu,\nu)} \left\langle \gamma, \mathbf{C}^N \right\rangle_F + \lambda_\mu tr(\mathbf{E}_\nu^T \gamma^T \mathbf{L}_\mu \gamma \mathbf{E}_\nu) + \lambda_\nu tr(\mathbf{E}_\mu^T \gamma \mathbf{L}_\nu \gamma^T \mathbf{E}_\mu) + \frac{\rho}{2} ||\gamma||_F^2$$

Global Connectivity Wasserstein Distance



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

A New OT Distance Metric on Graphs

- Regularized Wasserstein discrepancy preserves both features and structure of graphs;

$$RW(\mu,\nu) = \min_{\gamma\in\pi(\mu,\nu)} \left\langle \gamma, \mathbf{C}^V \right\rangle_F + \beta_1 LW(\mu,\nu) + \beta_2 GW(\mu,\nu) \tag{13}$$

- Transform the above optimization problem into a following form of objective [4];

$$\min_{\gamma\in\pi(\mu,\nu)} H(\gamma) = \min_{\gamma\in\pi(\mu,\nu)} f(\gamma) + g(\gamma) - h(\gamma) \tag{14}$$

[4] code: `https://github.com/wokas36/RWK`

A Fast and Numerically Stable Algorithm

Algorithm 1: Training for RW Discrepancy

initialize i=0, $\gamma^0 \leftarrow \mu\nu^T$, and $c^0 \leftarrow H(\gamma^0)$
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* (μ, ν, $\nabla H(\gamma)$, λ, b)
  $\Delta\gamma \leftarrow \hat{\gamma}^{(i-1)} - \gamma^{(i-1)}$
  $\alpha^{(i)}, c^{(i)} \leftarrow$ *Line-search* ($\gamma^{(i-1)}$, $\Delta\gamma$, $\nabla H(\gamma)$, $c^{(i-1)}$)
  $\gamma^{(i)} \leftarrow \gamma^{(i-1)} + \alpha^{(i)}\Delta\gamma$
  $\delta^{(i-1)} \leftarrow \left\langle \Delta\gamma, -\nabla H(\gamma) \right\rangle_F$
  if $\delta^{(i-1)} \leq \epsilon$ **then**
    stop
  end
end

A Regularized Wasserstein Kernel

Regularized Wasserstein Kernel

Given a set of graphs $\mathcal{G}$, RWK has a *kernel matrix* $\mathbf{K} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{G}|}$ defined as

$$\mathbf{K}_{\mu\nu} = e^{-\eta RW(\mu,\nu)},$$

where $\eta > 0$ is a parameter, μ and ν correspond to any two graphs in $\mathcal{G}$, and $RW(\mu, \nu)$ is the RW discrepancy between μ and ν.

Numerical Experiments

	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: Classification accuracy on graphs with discrete attributes.

Numerical Experiments

	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: Classification accuracy on graphs with continuous attributes.

Conclusion

In summary, we have proposed three GNN approaches and a Graph Kernel approach for graph learning.

- GraphSNN: Graph Structured Neural Network.
- DFNets: Distributed Feedback-Looped Network.
- DPRN: Dynamic PageRank Network.
- RWK: Regularized Wasserstein Kernel.

Thank You