The Expressive Power of GNNs

Pankaj Kumar
24126002

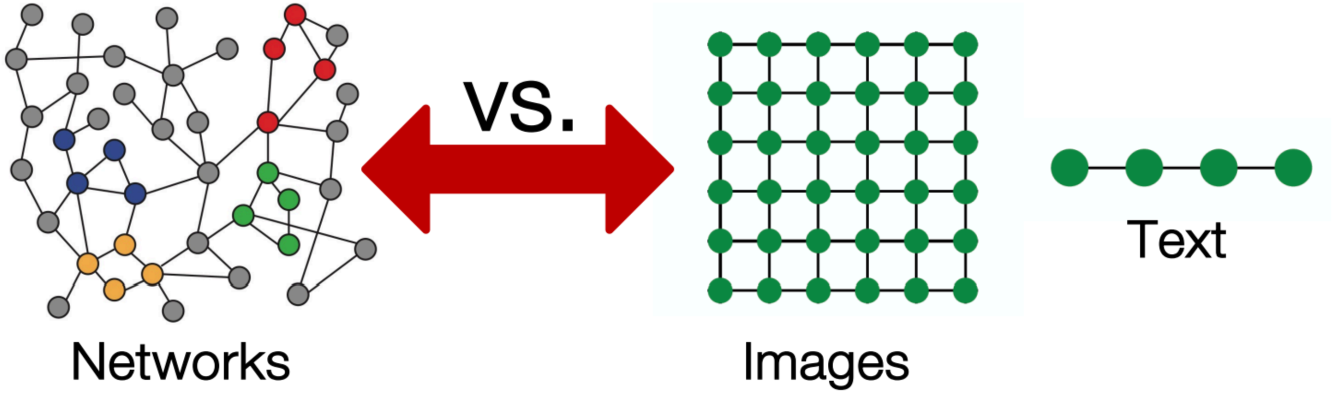
School of Computer Science
NISER, HBNI
Feburary 04, 2025

Outline

- Introduction
- GNNs
- Representation Learning
- Expressivity
- WL tests
- Variant of GNNs
- Characteristics of expressivity
- Limitations
- References

Introduction

- Graph data contains both feature and structural information
- No fixed node ordering or reference point

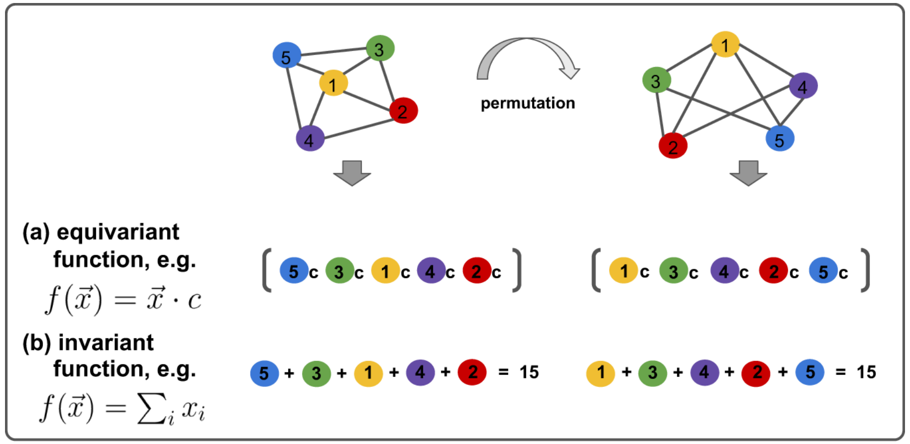


What are GNNs

- Characterized by the following equations $\leftarrow$ Message Passing Neural Networks (MPNNs) :

$$\boxed{\begin{aligned} h_v^{(t+1)} &= \mathrm{UPDATE}\big(h_v^{(t)}, \mathrm{AGGREGATE}(\{h_u^{(t)} : u \in \mathcal{N}(v)\})\big) \\ h_v^{(t+1)} &= \mathrm{READOUT}(\{h_v^{(t+1)} : v \in \mathcal{V}\}) \end{aligned}}$$

- $\mathrm{AGGREGATE}$ is learnable and permutation-invariant
- $\mathrm{READOUT}$ is learnable and permutation equivariant

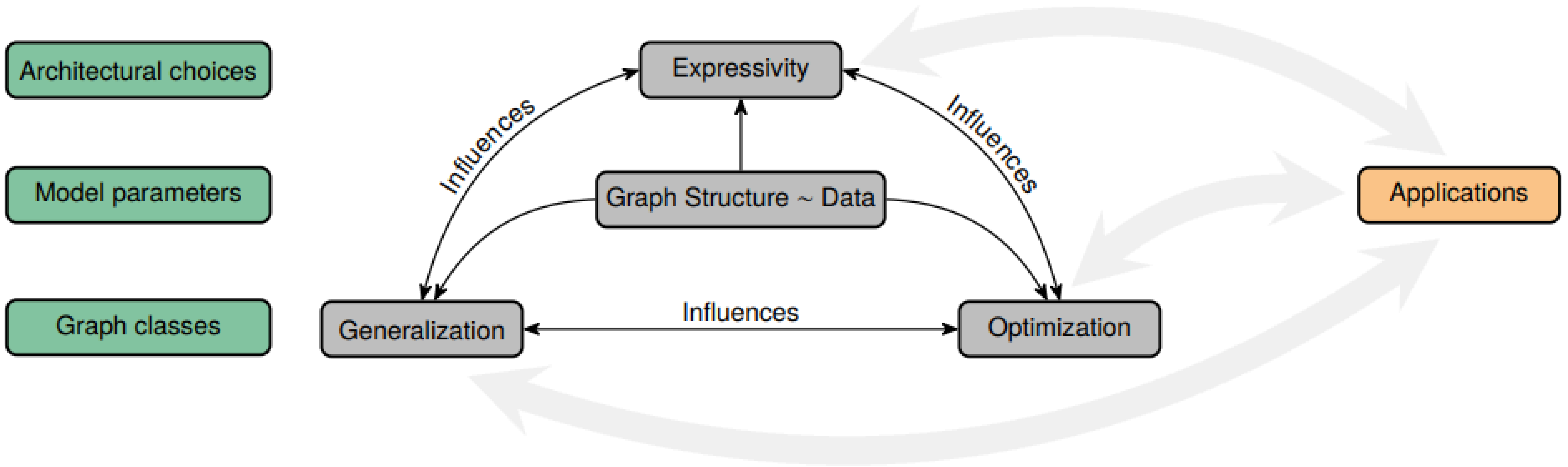


Graph Representation Learning

- Consider ground truth function $f^* : X \to Y, \quad f^*(G, S) = y$
- Given $\mathcal{T} = \{(G^{(i)}, S^{(i)}, y^{(i)})\}_{i=1}^{k}, \quad \Psi = \{(\tilde{G}^{(i)}, \tilde{S}^{(i)}, \tilde{y}^{(i)})\}_{i=1}^{k}$
- GRL problem learns f such that f is close to f^* on Ψ
  - G is a graph and $S \subset \mathcal{V}(G)$
  - Each $(G, S) \in X$ is associated with a target y in the target space Y
  - X is feature space, Y is a target space
  - $\mathcal{T}, \Psi$ are set of training and test examples

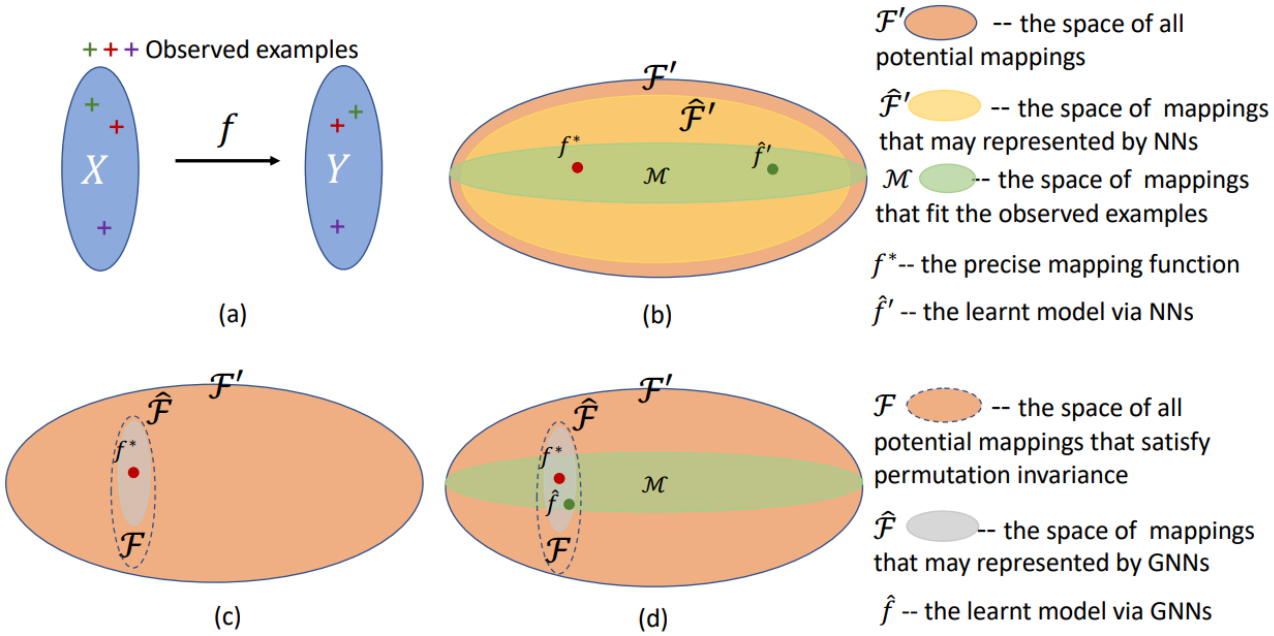
Need for Expressivity

- MPNNs have limited expressive power (can't detect cycles, get diameter ...etc)
  - fails to learn well from structural information



Expressivity

- Expressivity describes the class of functions a model can approximate.

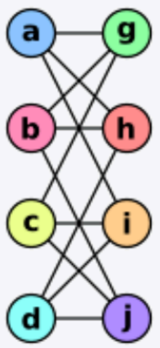
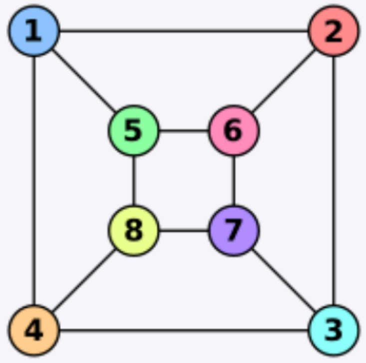


Source: Figure 5.5; Chapter 5: The Expressive Power of Graph Neural Networks

- GNN / MPNN expressivity $\uparrow$ $\iff$ f can distinguish b/w two non-isomorphic graphs

Graph Isomorphism Problem

- For two GRL examples $(G^{(1)}, S^{(1)})(G^{(2)}, S^{(2)})$, Graph Isomorphism problem asks if $\exists$ bijection $\phi : \mathcal{V}[G^{(1)}] \to \mathcal{V}[G^{(2)}]$ s.t. $A^{(1)}_{uv} = A^{(2)}_{\phi(u)\phi(v)}, X^{(1)}_u = X^{(2)}_v$

Graph G	Graph H	An isomorphism between G and H
		*f*(*a*) = 1 *f*(*b*) = 6 *f*(*c*) = 8 *f*(*d*) = 3 *f*(*g*) = 5 *f*(*h*) = 2 *f*(*i*) = 4 *f*(*j*) = 7

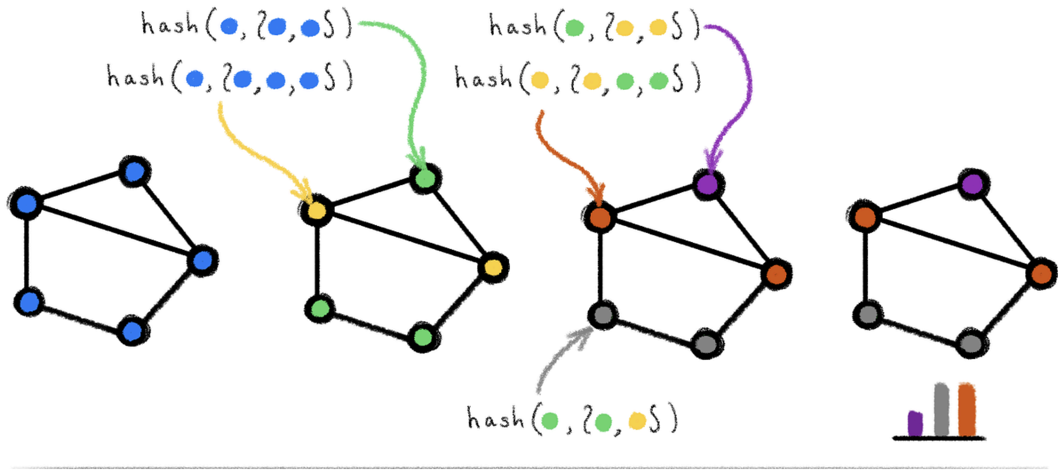
Source: Wikipedia; Graph Isomorphism

(1-)WL Test

Repeat

$$\boxed{C_v^{(i,l)} \leftarrow \mathrm{HASH}(h_v^{(i,l-1)}, \{C_u^{(i,l-1)} | u \in \mathcal{N}_v^i\}), \quad i \in \{1, 2\}}$$

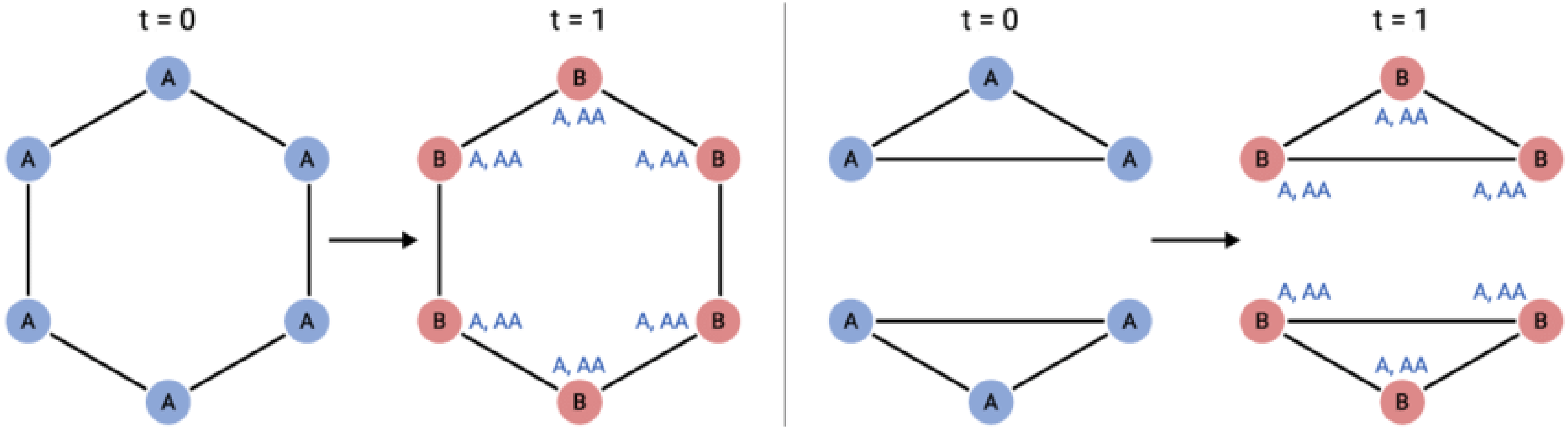
until we get a stable coloring



- Necessary condition for Graph Isomorphism problem

1-WL Test

- Different stable colorings $\implies$ Non-isomorphic
- However, same stable colorings $\not\Rightarrow$ Isomorphic



WL Test ⇔ GNN Expressivity

Theorem 3. *Let $\mathcal{A} : \mathcal{G} \rightarrow \mathbb{R}^d$ be a GNN. With a sufficient number of GNN layers, $\mathcal{A}$ maps any graphs G_1 and G_2 that the Weisfeiler-Lehman test of isomorphism decides as non-isomorphic, to different embeddings if the following conditions hold:*

a) $\mathcal{A}$ aggregates and updates node features iteratively with

$$h_v^{(k)} = \phi\left(h_v^{(k-1)}, f\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right)\right),$$

where the functions f, which operates on multisets, and ϕ are injective.

b) $\mathcal{A}$'s graph-level readout, which operates on the multiset of node features $\left\{h_v^{(k)}\right\}$, is injective.

Source: Xu et al.; How Powerful are Graph Neural Networks?

- $\boxed{\text{Standard MPNNs} \equiv \text{1-WL class}}$

WL Test ⇔ GNN Expressivity

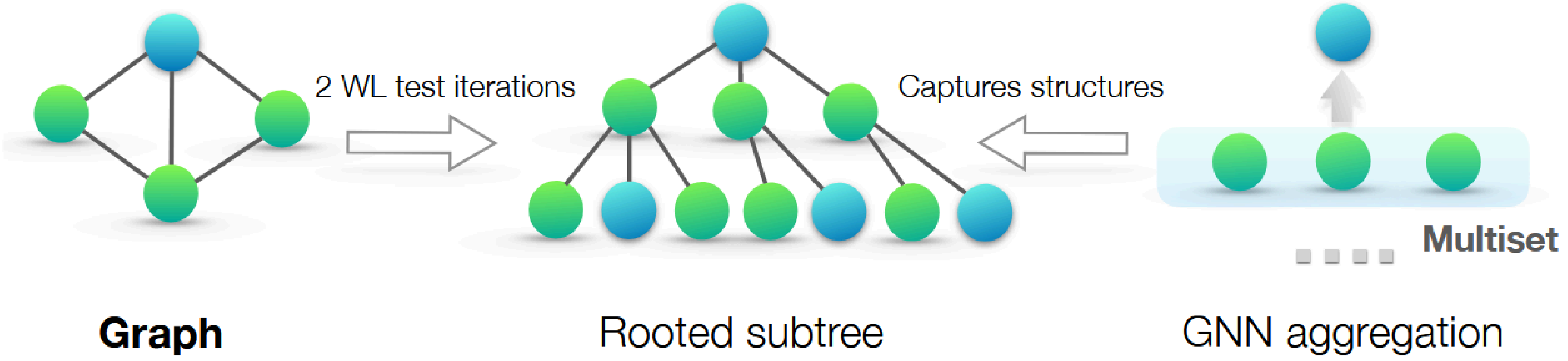


Figure 1: **An overview of our theoretical framework.** Middle panel: rooted subtree structures (at the blue node) that the WL test uses to distinguish different graphs. Right panel: if a GNN's aggregation function captures the *full multiset* of node neighbors, the GNN can capture the rooted subtrees in a recursive manner and be as powerful as the WL test.

Source: Xu et al.; How Powerful are Graph Neural Networks?

- The WL-hierarchy provides a ladder for GNN expressivity.

MPNN Variant: Graph Isomorphism Network (GIN)

$$h_v^{(0)} = x_v \quad \text{for all } v \in V.$$

Node v's initial embedding.

... is just node v's original features.

and for $k = 1, 2, \ldots$ upto K:

$$h_v^{(k)} = f^{(k)}\left(\sum_{u \in \mathcal{N}(v)} h_u^{(k-1)} + (1 + \epsilon^{(k)}) \cdot h_v^{(k-1)}\right) \quad \text{for all } v \in V.$$

Node v's embedding at step k.

Sum of v's neighbour's embeddings at step $k-1$.

Node v's embedding at step $k-1$.

Graph Isomorphism Networks (GINs)

Source: Daigavane, A. et al.; Understanding Convolutions on Graphs

Twin GNNs

- Add extra topology information
  - To increase the discrimination between nodes
- Assign color labels and identity id to nodes
  - Execute two sets of aggregation schemes simultaneously

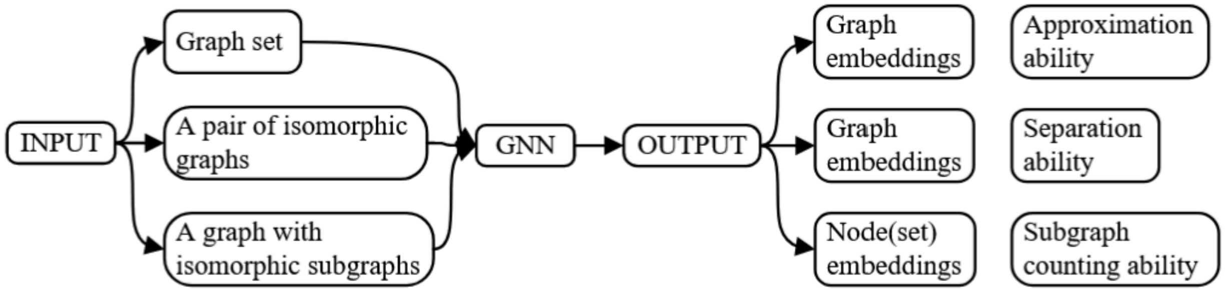
$$\boxed{id_v^{(l)} = f_{id}(id_v^{(l-1)}, \{id_u^{(l-1)} | u \in \mathcal{N}(v)\})}$$

- Allows a stronger expression than 1-WL test without significantly $\uparrow$ computational costs

Characteristics of Expressive GNNs

The expressive power of GNNs can be characterized as follows :

- Approximation ability
  - To approximate functions on graphs, considering the feature embedding and structural information
- Separation ability
  - To identify graph isomorphism, focusing solely on structural information (in unfeatured graph)
- Subgraph counting ability
  - To detect and utilize the graph subgraph structure



Limitations

- Outputs of WL tests are binary
  - does not give insights into the degree of similarity b/w two given graphs
- GNN expressivity results are fairly specific
  - No metric evaluation for some models
- No tight bounds are available
- Higher order WL tests are computationally expensive

References

1. Expressivity-Preserving GNN Simulation
2. The Expressive Power of Graph Neural Networks: A Survey
3. Future Directions in the Theory of Graph Machine Learning
4. How Powerful Are Graph Neural Networks?
5. Twin Weisfeiler-Lehman: High Expressive GNNs for Graph Classification

THANK YOU

Extra Slides

Permutation Equivariance

- The output order of the model corresponds to the input order