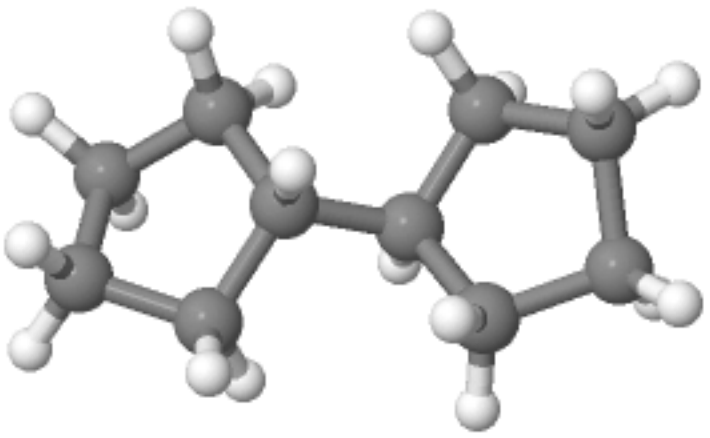
Distance-Restricted Folklore Weisfeiler-Leman GNNs with Provable Cycle Counting Power

Junru Zhou · Jiarui Feng · Xiyuan Wang · Muhan Zhang

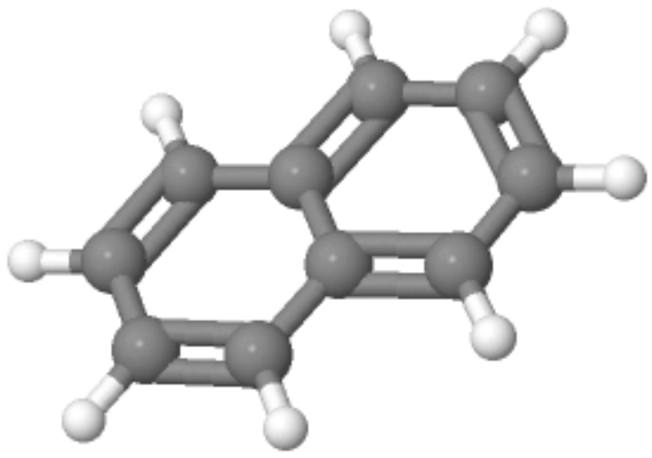
Introduction: Cycle counting

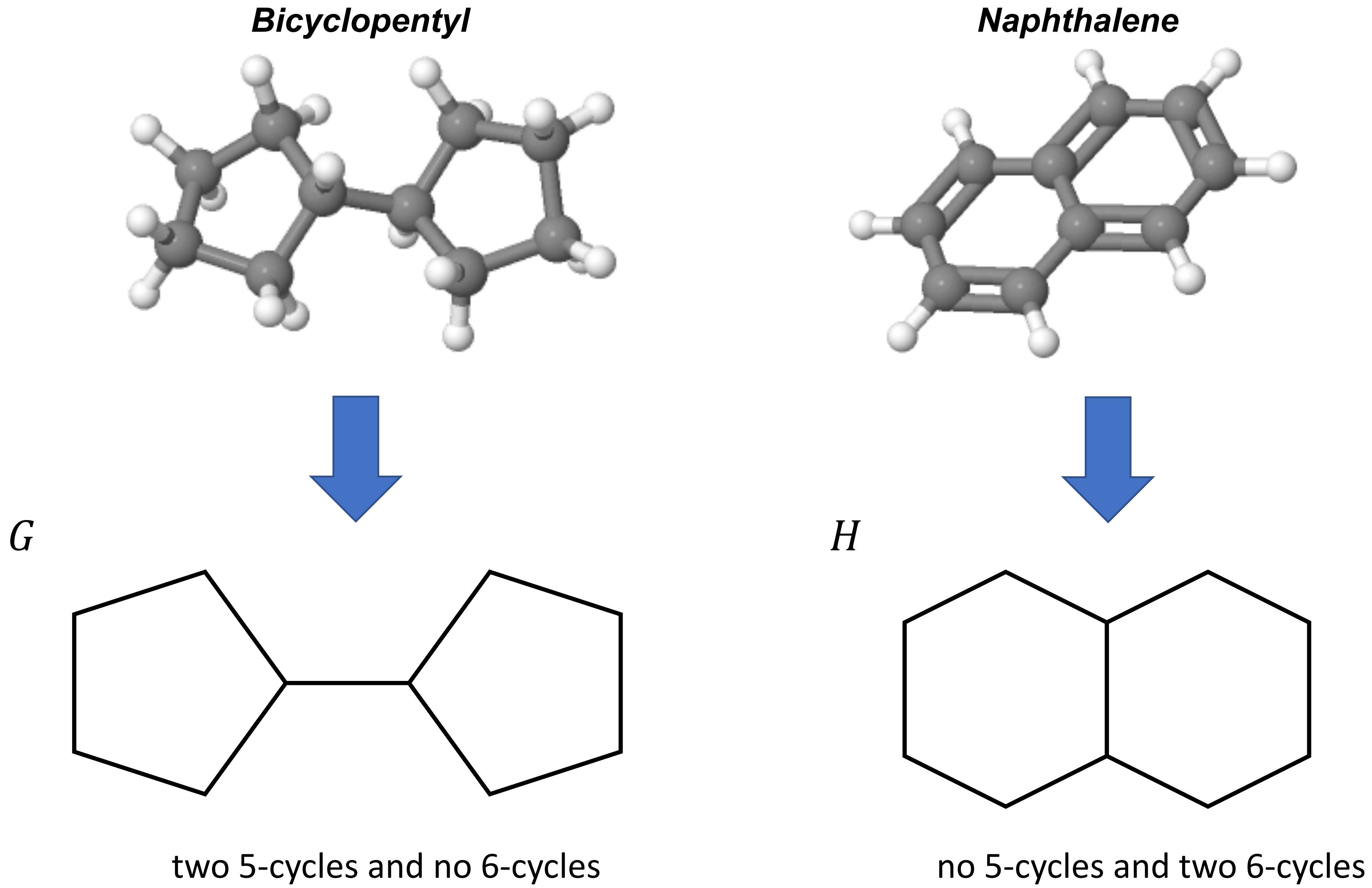
- Cycles are important local structures in graphs.
  (*especially in the context of **chemistry**!*)

Bicyclopentyl



Naphthalene




Bicyclopentyl
Naphthalene
G
H
two 5-cycles and no 6-cycles
no 5-cycles and two 6-cycles

Introduction: Message passing GNNs

- Message passing GNNs (MPNNs) update node representation h_u by

$$h_u^{(t)} = f^{(t)}\left(h_u^{(t-1)}, \bigoplus_{v \in \mathcal{N}(u)} m^{(t)}\left(h_u^{(t-1)}, h_v^{(t-1)}, e_{uv}\right)\right)$$

$f^{(t)}, m^{(t)}$: learnable functions

$\bigoplus$: permutation-invariant aggregation function (e.g. sum, mean)

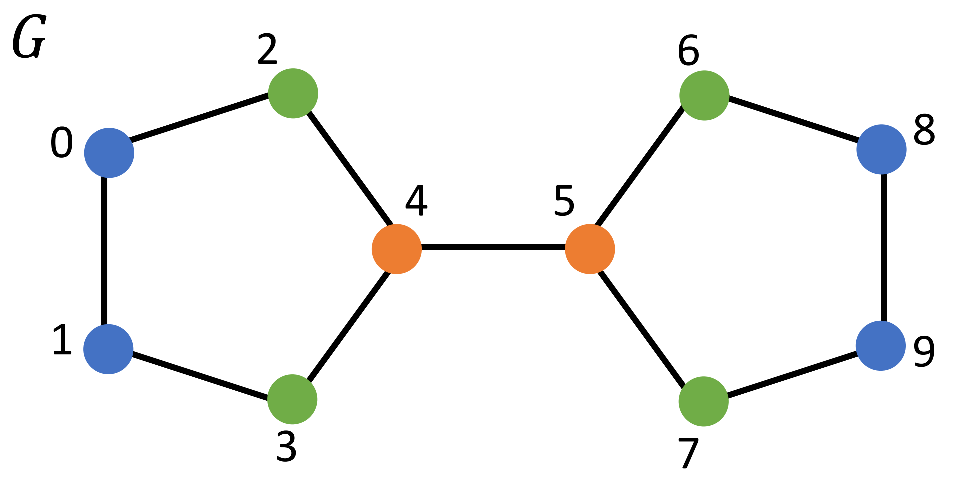
Use a readout layer to encode graph G: $h_G = R(\{\!\{h_u : u \in \mathcal{V}_G\}\!\})$,

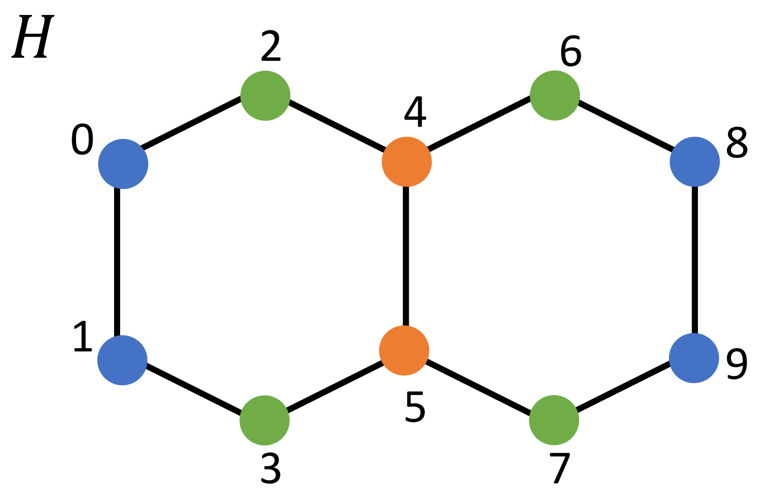
- MPNNs are not more powerful than WL(1) test. [Xu et al., 2018]

$$W^{(t)}(v) = \mathrm{HASH}^{(t)}\left(W^{(t-1)}(v), \mathrm{POOL}^{(t)}\left(\{\!\{W^{(t-1)}(u) : u \in \mathcal{N}(v)\}\!\}\right)\right)$$

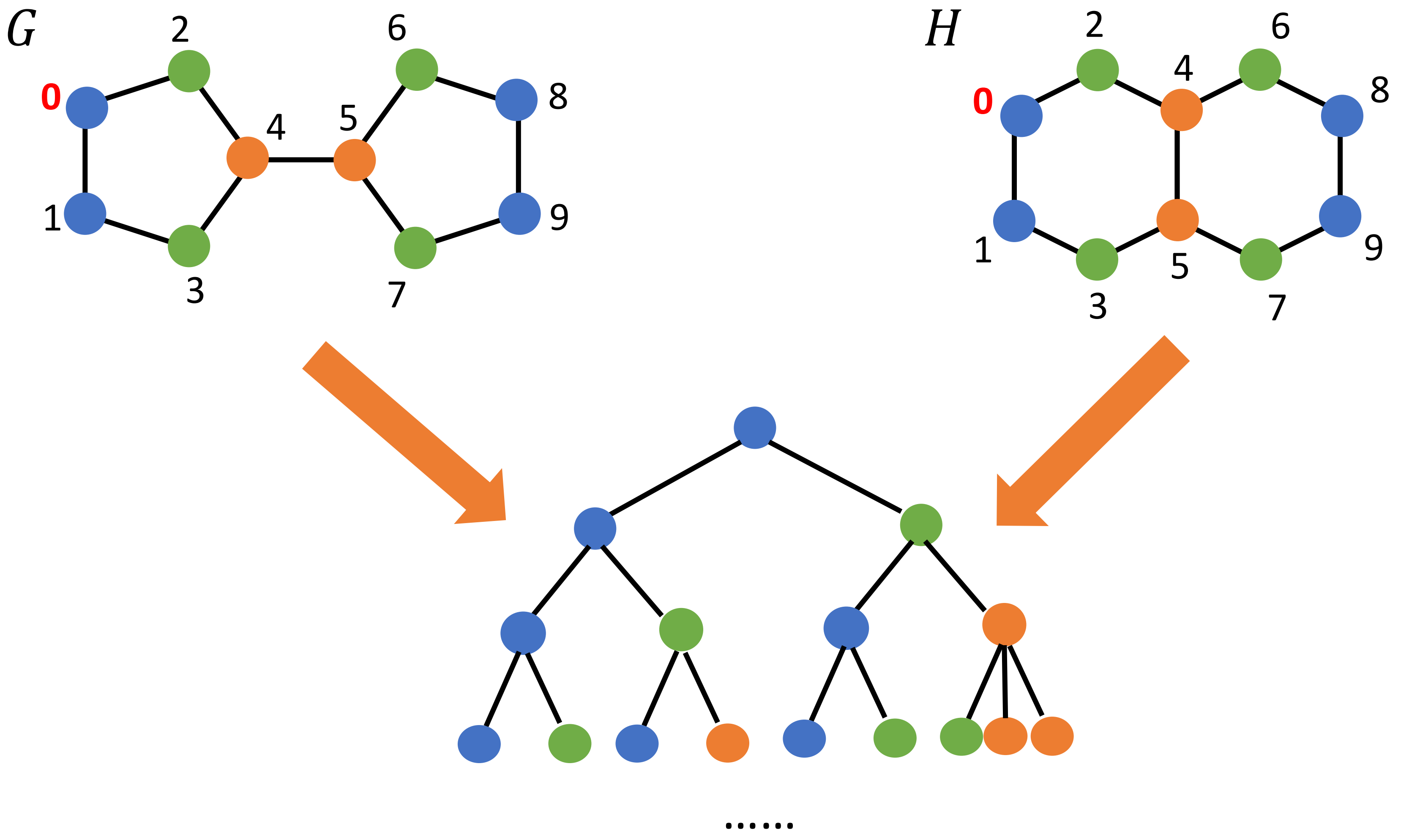
The difficulty of MPNNs to count cycles

- MPNNs fail to distinguish between G and H.

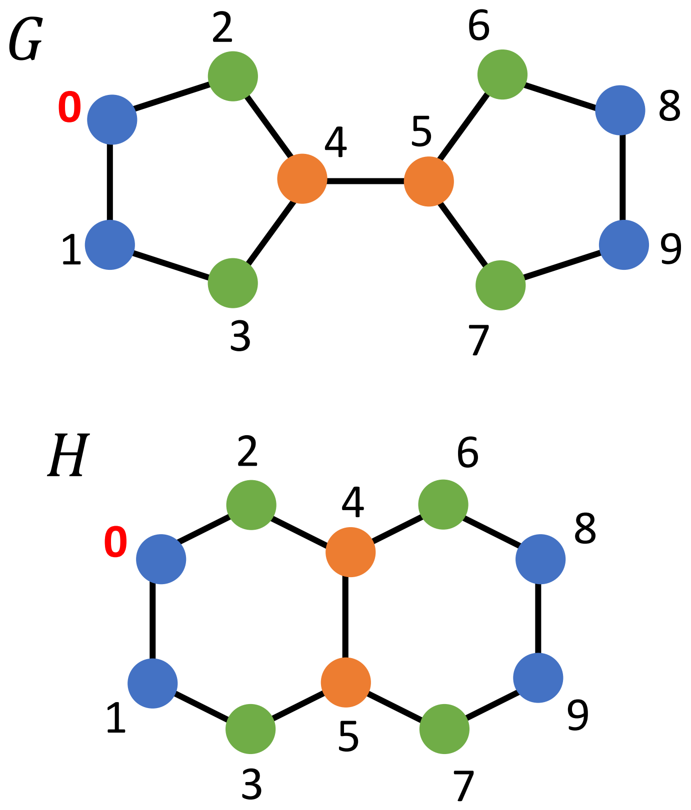


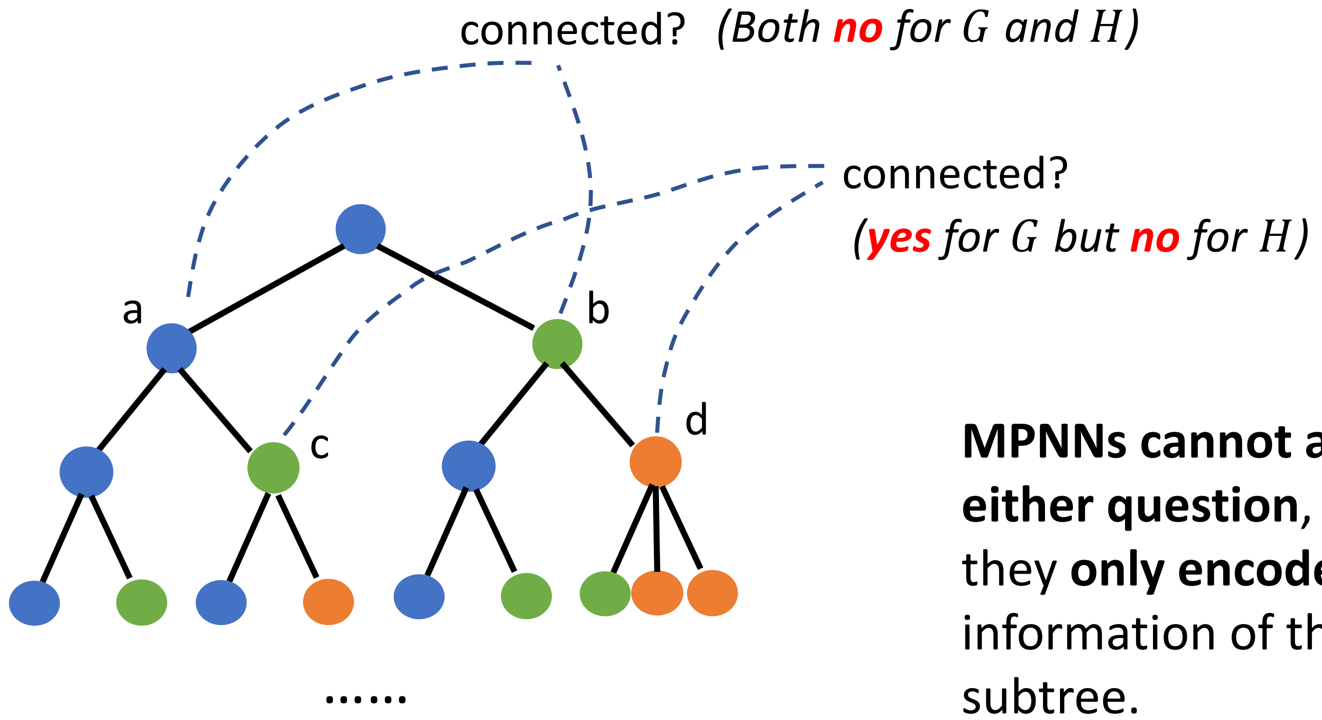


Why MPNNs cannot count cycles?



Why MPNNs cannot count cycles?





MPNNs cannot answer either question, since they **only encode** information of the local subtree.

Introduction: FWL(2) test

- FWL(2) test assigns a color $W(u,v)$ for every **2-tuple** $(u,v) \in \mathcal{V}_G^2$,

$O(n^2)$ space

Initialize: Give a unique color for three different cases, (i) $u = v$, (ii) u, v connected, and (iii) u, v not connected.

Update:

$O(n^3)$ time

$$W^{(t)}(u,v) = \mathrm{HASH}^{(t)}\left(\begin{array}{c} W^{(t-1)}(u,v), \\ \mathrm{POOL}^{(t)}\left(\left[\!\left[\left(W^{(t-1)}(u,w), W^{(t-1)}(w,v)\right) : w \in \mathcal{V}_G\right]\!\right]\right) \end{array}\right)$$

Readout: $W^{(\infty)}(G) = \mathrm{READOUT}\left(\left[\!\left[W^{(\infty)}(u,v) : (u,v) \in \mathcal{V}_G^2\right]\!\right]\right)$

Can FWL(2) test count cycles?

- **FWL(2) test can count 3-cycles.**

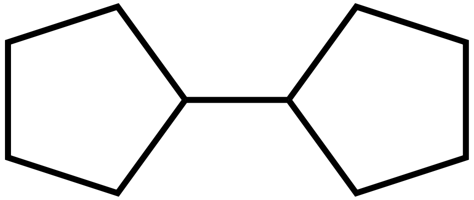
- Actually, FWL(2) can simulate the following procedure

Procedure: Count 3-cycles that passes nodes u and v.

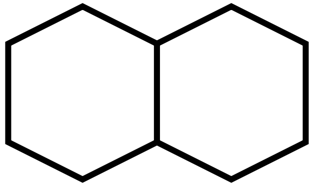
if not (u and v are connected):
    return 0
else:
    return (# of w such that both u,w and w,v are connected)

Definitions of cycle counting

- Graph-level counts:

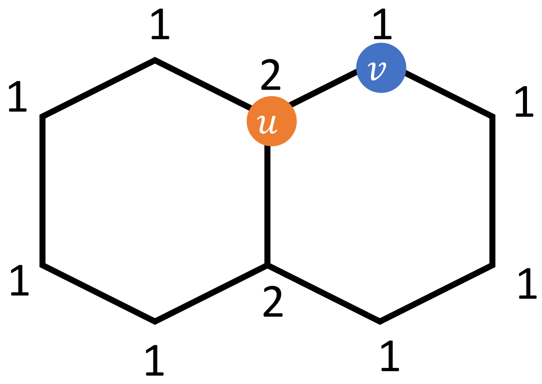


#(5-cycle) = 2, #(6-cycle) = 0



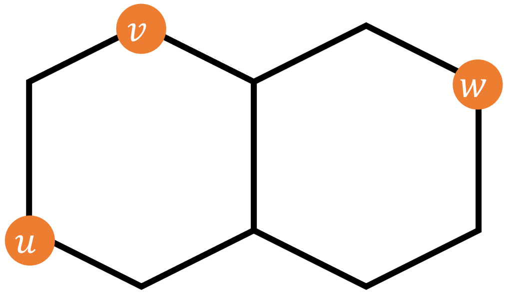
#(5-cycle) = 0, #(6-cycle) = 2

- Node-level counts:



#(6-cycle passing u) = 2
#(6-cycle passing v) = 1

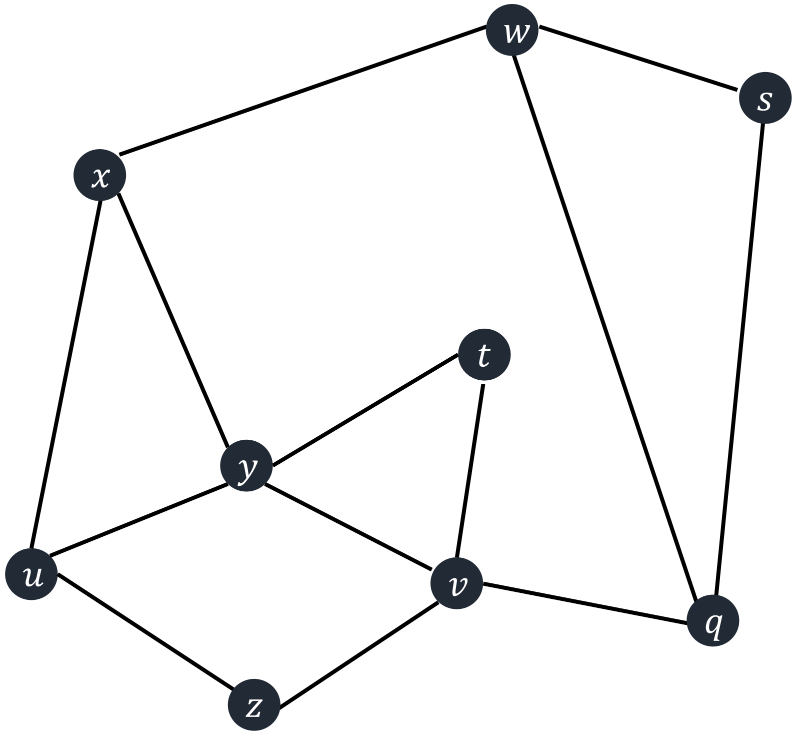
- Pair-level counts:



#(6-cycle passing u, v) = 1
#(6-cycle passing u, w) = 0

Can FWL(2) test count cycles?

- FWL(2) counts cycles by counting **closed walks**!
- Take the example of counting 6-cycles.



Can be done by FWL(2)!

#(2-walk $u \rightarrow w$) = 1
#(2-walk $w \rightarrow v$) = 1
#(2-walk $u \rightarrow v$) = 2

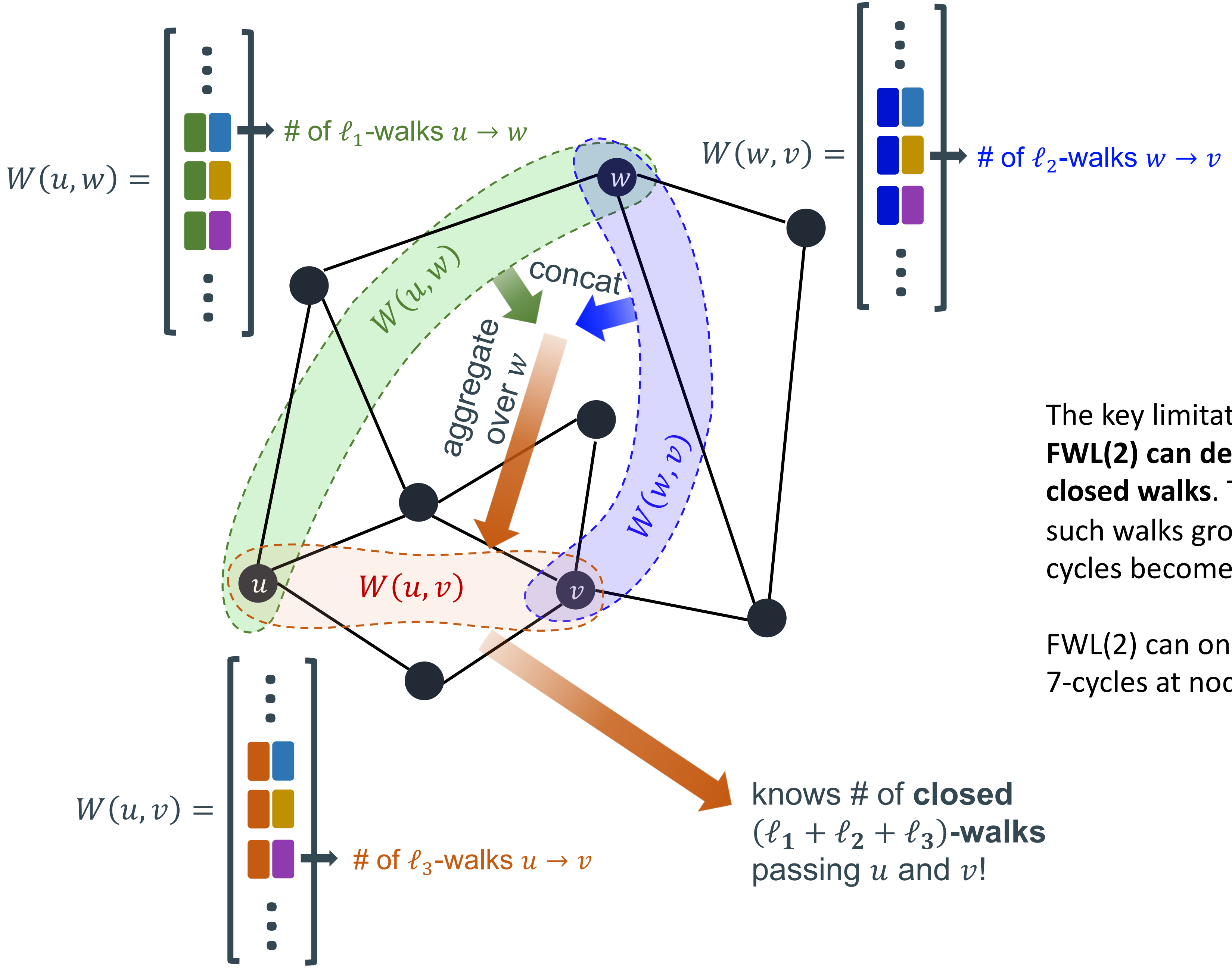
#(6-closed walk $u \rightarrow w \rightarrow v \rightarrow u$) = 1*1*2 = 2

sum over all possible w

#(6-closed walk $u \rightarrow \cdots \rightarrow v \rightarrow u$) =
2 + 2 + 12 + 16 + 2 + 0 + 2 + 0 + 0 = 36

remove non-cycles

#(6-cycle passing u, v) = 3



The key limitation is **whether FWL(2) can detect non-cycle closed walks**. The number of such walks grows quickly as cycles become longer.

FWL(2) can only count up to 7-cycles at node level.

Why FWL(2) can count cycles?

- Basically, two features are important for the cycle counting power of FWL(2):
- 1. use 2-tuple (instead of nodes) as the basis of message passing
- 2. use the *walk-like* update rule

- It is possible to design **more efficient algorithms** than FWL(2) but keep (almost) all of its counting power, as long as these two features are kept.

The "local" nature of cycle counting

- Consider the following algorithm to count 3-cycles.

```
Procedure: Count 3-cycles that passes nodes u and v.

if not (u and v are connected):
        return 0
else:
        return (# of w such that both u,w and w,v are connected)
```

- We notice that even if all tuples (u, v) with $d(u, v) > 1$ are ignored in FWL(2) update, we can still calculate the count.

The “local” nature of cycle counting

- Namely, if we modify the update rule of FWL(2) to

if $d(u,v)=1$**, then**

$$W^{(t)}(u,v) = \mathrm{HASH}^{(t)}\begin{pmatrix} W^{(t-1)}(u,v), \\ \mathrm{POOL}^{(t)}\left(\left\llbracket \left(W^{(t-1)}(u,w), W^{(t-1)}(w,v)\right) : w \in \mathcal{N}_1(u) \cap \mathcal{N}_1(v) \right\rrbracket\right) \end{pmatrix}$$

else

$$W^{(t)}(u,v) = 0$$

- Then the ability to count 3-cycle is retained.

The "local" nature of cycle counting

- Similarly, if we modify the update rule of FWL(2) to

if $d(u,v) \leq 2$**, then**

$$W^{(t)}(u,v) = \mathrm{HASH}^{(t)}\left(\begin{matrix} W^{(t-1)}(u,v), \\ \mathrm{POOL}^{(t)}\left(\left[\!\left[\left(W^{(t-1)}(u,w), W^{(t-1)}(w,v)\right) : d(u,w), d(w,v) \leq 2\right]\!\right]\right)\end{matrix}\right)$$

else

$$W^{(t)}(u,v) = 0$$

- Then the abilities to node-level count 3, 4, 5, 6-cycle are all retained.

d-Distance Restricted FWL(2) tests

- We propose **d-Distance Restricted FWL(2) tests**, or **d-DRFWL(2) tests** as following. Different from FWL(2), d-DRFWL(2) test assigns a color **only** to all $(u, v) \in \mathcal{V}_G^2$ **that satisfies** $0 \leq d(u, v) \leq d$.

Initialize: Give a unique color for $(d + 1)$ different cases, (i) $d(u, v) = 0$, or $u = v$, (ii) $d(u, v) = 1$, (iii) $d(u, v) = 2$, …

Note: We remark that this step can be unnecessary. One can still adopt the FWL(2) initialization (only considering three cases, coinciding, connected or disconnected), and use the update rule to generate distance encoding.

d-Distance Restricted FWL(2) tests

Update:

$\boxed{\text{For each } k = 0, 1, \ldots, d,}$

$$W^{(t)}(u,v) = \mathrm{HASH}_k^{(t)}\left(W^{(t-1)}(u,v), \left(M_{ij}^{k(t)}(u,v)\right)_{0\leqslant i,j\leqslant d}\right), \quad \text{if } d(u,v) = k, \tag{6}$$

where $\mathrm{HASH}_k^{(t)}$ is an injective hashing function for distance k and iteration t, and $M_{ij}^{k(t)}(u,v)$ is defined as

$$M_{ij}^{k(t)}(u,v) = \mathrm{POOL}_{ij}^{k(t)}\left(\{\!\{\left(W^{(t-1)}(w,v), W^{(t-1)}(u,w)\right) : \boxed{w \in \mathcal{N}_i(u) \cap \mathcal{N}_j(v)}\}\!\}\right). \tag{7}$$

The symbol $\left(M_{ij}^{k(t)}(u,v)\right)_{0\leqslant i,j\leqslant d}$ stands for $\left(M_{00}^{k(t)}(u,v), M_{01}^{k(t)}(u,v), \ldots, M_{0d}^{k(t)}(u,v), \ldots, M_{dd}^{k(t)}(u,v)\right)$. Each of the $\mathrm{POOL}_{ij}^{k(t)}$ with $0 \leqslant i,j,k \leqslant d$ is an injective multiset hashing function.

d-Distance Restricted FWL(2) tests

Readout:

$$W(G) = \mathrm{READOUT}\left(\{\!\!\{W^{(\infty)}(u,v) : (u,v) \in \mathcal{V}_G^2 \textbf{ and } \boxed{0 \leqslant d(u,v) \leqslant d}\}\!\!\}\right).$$

d-DRFWL(2) GNNs

- d-DRFWL(2) GNNs are neural versions of d-DRFWL(2) tests.

Initialize: generate *initial labeling* $h_{uv}^{(0)}, 0 \leqslant d(u,v) \leqslant d$.

Update in each layer: For each $k = 0, 1, \ldots, d$,

For each $(u,v) \in \mathcal{V}_G^2$ with $d(u,v) = k$,

$$a_{uv}^{ijk(t)} = \bigoplus_{w \in \mathcal{N}_i(u) \cap \mathcal{N}_j(v)} m_{ijk}^{(t)}\left(h_{wv}^{(t-1)}, h_{uw}^{(t-1)}\right),$$

$$h_{uv}^{(t)} = f_k^{(t)}\left(h_{uv}^{(t-1)}, \left(a_{uv}^{ijk(t)}\right)_{0 \leqslant i,j \leqslant d}\right),$$

d-DRFWL(2) GNNs

- **Network structure:**

$$f = M \circ R \circ L_T \circ \sigma_{T-1} \circ \cdots \circ \sigma_1 \circ L_1.$$

$L_1, \dots, L_T$: d-DRFWL(2) GNN layers

$\sigma_1, \dots, \sigma_{T-1}$: activation functions

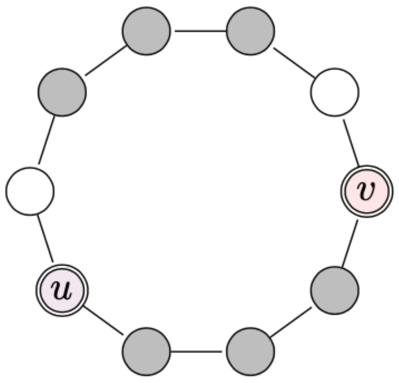
R: readout layer, giving the representation of G from the multiset

$$\{\!\!\{ h_{uv}^{(T)} : (u,v) \in \mathcal{V}_G^2 \text{ and } 0 \leqslant d(u,v) \leqslant d \}\!\!\}$$

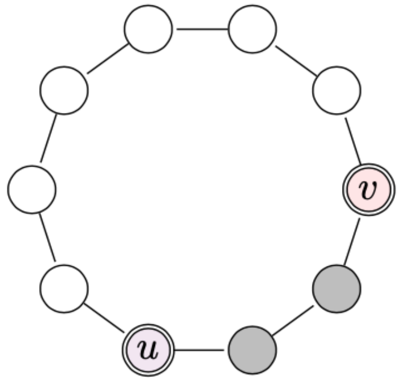
M: MLP

Discussion on d-DRFWL(2) tests

- d-DRFWL(2) test has a **finite** range of reception.
- Actually, d-DRFWL(2) tests cannot detect any $(3d+1)$-cycle in a graph, while using a larger d may make it possible.



(a) When running 4-DRFWL(2) in a 10-cycle, there are 8 (marked as colored, with 3 on the inferior arc and 3 on the superior arc) nodes contributing to the update of any distance-4 tuple (u, v)



(b) When running 3-DRFWL(2) in a 10-cycle (*or any cycle with length* $\geqslant 10$), there are only 4 (marked as colored) nodes contributing to the update of any distance-3 tuple (u, v)

Comparison with the WL hierarchy

Theorem 3.1. *In terms of the ability to distinguish between non-isomorphic graphs, the* d*-DRFWL(2) test is strictly more powerful than WL(1), for any* $\mathrm{d} \geqslant 1$.

Theorem 3.2. *In terms of the ability to distinguish between non-isomorphic graphs, FWL(2) is strictly more powerful than* d*-DRFWL(2), for any* $\mathrm{d} \geqslant 1$*. Moreover,* $(\mathrm{d}+1)$*-DRFWL(2) is strictly more powerful than* d*-DRFWL(2).*

Cycle counting power of d-DRFWL(2) GNNs

- **1-DRFWL(2) GNNs can node-level count** up to **3-cycles**, but **cannot graph-level count** more than 4-cycles.
- **2-DRFWL(2) GNNs can node-level count** up to **6-cycles**, but **cannot graph-level count** more than 7-cycles.
- **d-DRFWL(2) GNNs with $d > 2$ can node-level count** up to **7-cycles**, but **cannot graph-level count** more than 8-cycles.

- Notice that 3-DRFWL(2) GNNs already possess equal cycle counting power to FWL(2).

Complexity analysis

Method	Cycle counting power	Space	Time
2-DRFWL(2) GNN	Up to ***6-cycle*** at node level	$O(n\,\mathrm{deg}^2)$	$O(n\,\mathrm{deg}^4)$
d-DRFWL(2) GNNs $(d \geq 3)$	Up to ***7-cycle*** at node level	$O(n\,\mathrm{deg}^d)$	$O(n\,\mathrm{deg}^{2d})$
I^2-GNN	Up to ***6-cycle*** at node level, w/ subgraph height $k \geq 3$	at least $O(n\,\mathrm{deg}^4)$	at least $O(n\,\mathrm{deg}^5)$
FWL(2)-based GNNs	Up to ***7-cycle*** at node level	$O(n^2)$	$O(n^3)$

Experiments

Table 1: Normalized MAE results of node-level counting cycles and other substructures on synthetic dataset. The colored cell means an error less than 0.01.

Method	Synthetic (norm. MAE)								
	3-Cyc.	4-Cyc.	5-Cyc.	6-Cyc.	Tail. Tri.	Chor. Cyc.	4-Cliq.	4-Path	Tri.-Rect.
MPNN	0.3515	0.2742	0.2088	0.1555	0.3631	0.3114	0.1645	0.1592	0.2979
ID-GNN	0.0006	0.0022	0.0490	0.0495	0.1053	0.0454	0.0026	0.0273	0.0628
NGNN	0.0003	0.0013	0.0402	0.0439	0.1044	0.0392	0.0045	0.0244	0.0729
GNNAK+	0.0004	0.0041	0.0133	0.0238	0.0043	0.0112	0.0049	0.0075	0.1311
PPGN	0.0003	0.0009	0.0036	0.0071	0.0026	0.0015	0.1646	0.0041	0.0144
I^2-GNN	0.0003	0.0016	0.0028	0.0082	0.0011	0.0010	0.0003	0.0041	0.0013
2-DRFWL(2) GNN	0.0004	0.0015	0.0034	0.0087	0.0030	0.0026	0.0009	0.0081	0.0070

Table 2: Normalized MAE results of node-level counting k-cycles $(3 \leqslant k \leqslant 7)$ on synthetic dataset.

Method	Synthetic (norm. MAE)				
	3-Cyc.	4-Cyc.	5-Cyc.	6-Cyc.	7-Cyc.
2-DRFWL(2) GNN	**0.0004**	**0.0015**	**0.0034**	**0.0087**	0.0362
3-DRFWL(2) GNN	0.0006	0.0020	0.0047	0.0099	**0.0176**

Node-level cycle (& substructure) counting on synthetic datasets

Experiments

Table 3: MAE results on QM9 (smaller the better). The top two are highlighted as **First**, **Second**.

Target	1-GNN	1-2-3-GNN	DTNN	Deep LRP	PPGN	NGNN	I^2-GNN	2-DRFWL(2) GNN
μ	0.493	0.476	**0.244**	0.364	**0.231**	0.428	0.428	0.346
α	0.78	0.27	0.95	0.298	0.382	0.29	**0.230**	**0.222**
$\varepsilon_{\text{homo}}$	0.00321	0.00337	0.00388	**0.00254**	0.00276	0.00265	0.00261	**0.00226**
$\varepsilon_{\text{lumo}}$	0.00355	0.00351	0.00512	0.00277	0.00287	0.00297	**0.00267**	**0.00225**
$\Delta\varepsilon$	0.0049	0.0048	0.0112	**0.00353**	0.00406	0.0038	0.0038	**0.00324**
R^2	34.1	22.9	17.0	19.3	**16.07**	20.5	18.64	**15.04**
ZPVE	0.00124	0.00019	0.00172	0.00055	0.0064	0.0002	**0.00014**	**0.00017**
U_0	2.32	**0.0427**	2.43	0.413	0.234	0.295	0.211	**0.156**
U	2.08	**0.111**	2.43	0.413	0.234	0.361	0.206	**0.153**
H	2.23	**0.0419**	2.43	0.413	0.229	0.305	0.269	**0.145**
G	1.94	**0.0469**	2.43	0.413	0.238	0.489	0.261	**0.156**
C_v	0.27	0.0944	2.43	0.129	0.184	0.174	**0.0730**	**0.0901**

Table 8: Ten-runs MAE results on ZINC-12K (smaller the better), four-runs MAE results on ZINC-250K (smaller the better), ten-runs ROC-AUC results on ogbg-molhiv (larger the better), and four-runs AP results on ogbg-molpcba (larger the better). The * indicates the model uses virtual node on ogbg-molhiv and ogbg-molpcba.

Method	ZINC-12K (MAE)	ZINC-250K (MAE)	ogbg-molhiv (AUC)	ogbg-molpcba (AP)
GIN*	0.163 ± 0.004	0.088 ± 0.002	77.07 ± 1.49	27.03 ± 0.23
PNA	0.188 ± 0.004	–	79.05 ± 1.32	28.38 ± 0.35
DGN	0.168 ± 0.003	–	79.70 ± 0.97	28.85 ± 0.30
HIMP	0.151 ± 0.006	0.036 ± 0.002	78.80 ± 0.82	–
GSN	0.115 ± 0.012	–	80.39 ± 0.90	–
Deep LRP	–	–	77.19 ± 1.40	–
CIN-small	0.094 ± 0.004	0.044 ± 0.003	80.05 ± 1.04	–
CIN	0.079 ± 0.006	$\mathbf{0.022}\pm0.002$	$\mathbf{80.94}\pm0.57$	–
Nested GIN*	0.111 ± 0.003	0.029 ± 0.001	78.34 ± 1.86	28.32 ± 0.41
GNNAK+	0.080 ± 0.001	–	79.61 ± 1.19	$\mathbf{29.30}\pm0.44$
SUN (EGO)	0.083 ± 0.003	–	80.03 ± 0.55	–
I^2-GNN	0.083 ± 0.001	0.023 ± 0.001	78.68 ± 0.93	–
d-DRFWL(2) GNN	$\mathbf{0.077}\pm0.002$	0.025 ± 0.003	78.18 ± 2.19	25.38 ± 0.19

Performance on real-world datasets

Experiments

Efficiency & scalability

Notice: for datasets with large average degree (e.g. ogbg-ppa), our method will be slow (especially the preprocessing).

QM9: $\bar{n} = 18.0, \bar{m} = 18.7$
ogbg-molhiv: $\bar{n} = 25.5, \bar{m} = 27.5$

Table 4: Empirical efficiency of 2-DRFWL(2) GNN.

Method	QM9			ogbg-molhiv		
	Memory (GB)	Pre. (s)	Train (s/epoch)	Memory (GB)	Pre. (s)	Train (s/epoch)
MPNN	2.28	64	45.3	2.00	2.4	18.8
NGNN	13.72	2354	107.8	5.23	1003	42.7
I^2-GNN	19.69	5287	209.9	11.07	2301	84.3
2-DRFWL(2) GNN	2.31	430	141.9	4.44	201	44.3

ProteinsDB: $\bar{n} = 475.9, \bar{m} = 714.8$
HomologyTAPE: $\bar{n} = 167.3, \bar{m} = 256.7$

Table 14: Empirical efficiency on ProteinsDB and HomologyTAPE datasets. "OOM" means the method takes an amount of GPU memory more than 24 GB.

Method	ProteinsDB			HomologyTAPE		
	Memory (GB)	Pre. (s)	Train (s/epoch)	Memory (GB)	Pre. (s)	Train (s/epoch)
MPNN	2.60	235.7	0.597	1.99	243.8	5.599
NGNN	16.94	941.8	2.763	8.44	1480.7	15.249
I^2-GNN	OOM	1293.4	OOM	21.97	3173.6	38.201
PPGN	OOM	235.7	OOM	OOM	243.8	OOM
2-DRFWL(2) GNN	8.11	1843.7	3.809	3.82	2909.3	30.687

Thanks!

Paper ID: 8038

arXiv: 2309.04941 [cs.LG]

Code: https://github.com/zml72062/DR-FWL-2

Our arXiv link