$\mathscr{N}$-WL: A New Hierarchy of Expressivity for Graph Neural Networks

Qing Wang, Dillon Chen, Asiri Wijesinghe, Shouheng Li, and Muhammad Farhan

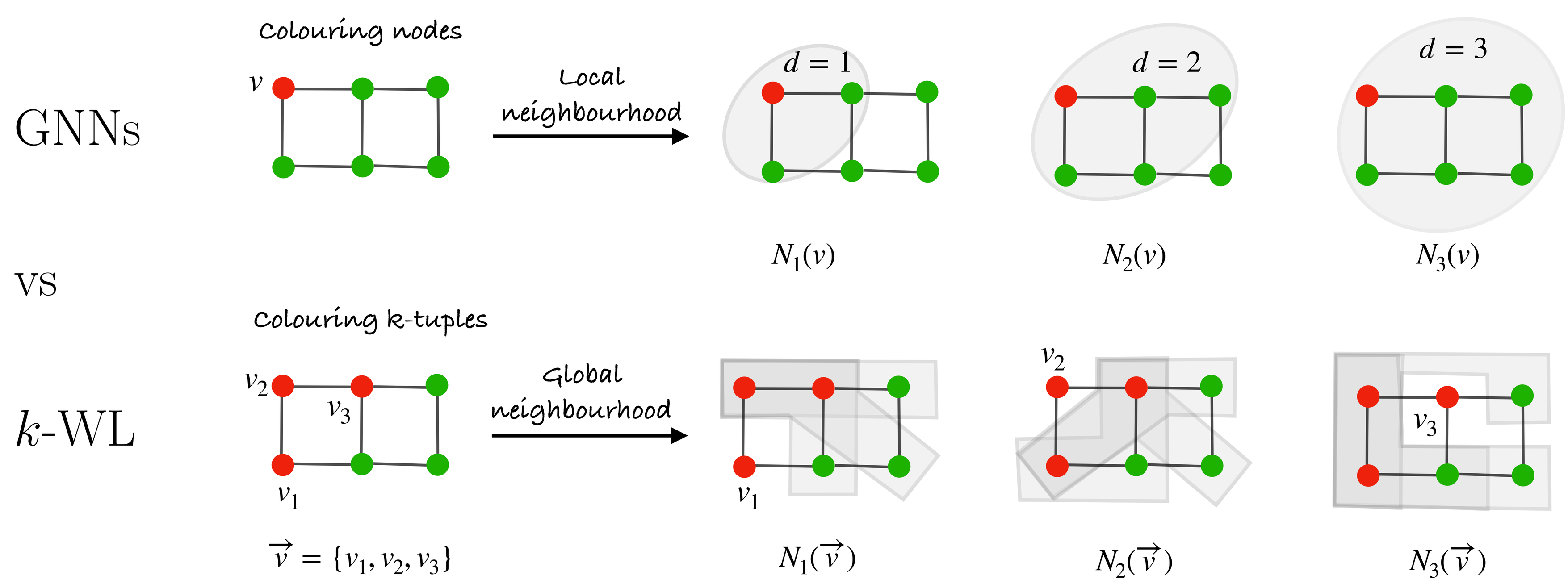
Graph Research Lab, School of Computing, Australian National University

ICLR

Australian National University

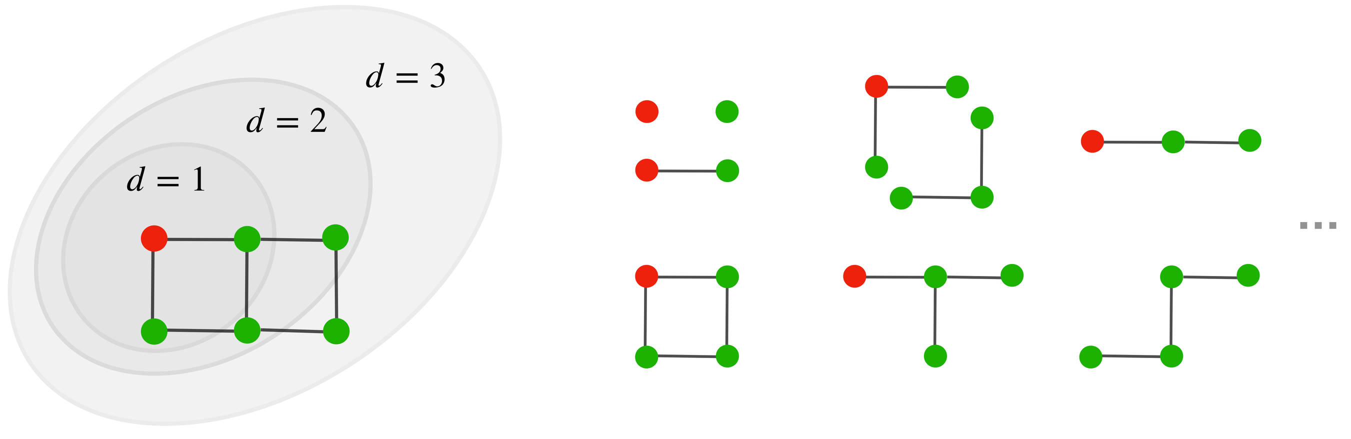
Introduction

Is k-WL hierarchy a good yardstick for measuring expressivity of GNNs?



Neighbourhood WL Hierarchy

Neighbourhood WL ($\mathscr{N}$-WL) hierarchy colours nodes via t-order induced subgraphs within d-hop neighbourhoods:

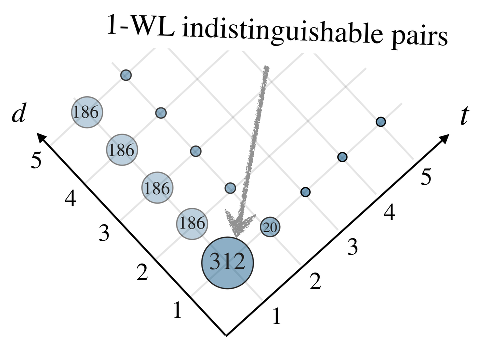


d-hop neighbourhoods *t-order induced subgraphs*

A Simple Experiment

A graph isomorphism test on 312 pairs of simple graphs of 8 vertices:

- None-or-all:
  None by 1-WL but all by 3-WL
- Progressive:
  Varying with d and t by $\mathscr{N}$-WL



Main Results

- Increasing the order of induced subgraphs, the expressive power increases:

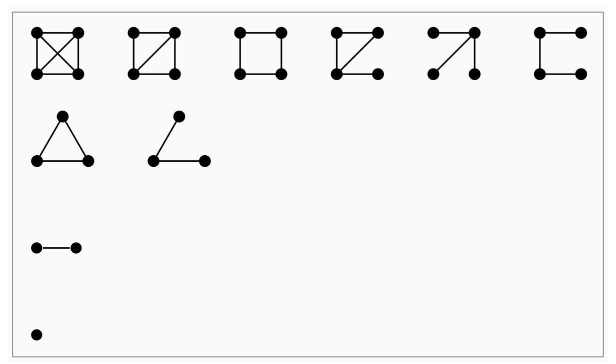
Theorem (Weak Hierarchy) $\mathscr{N}^-(t,d)\text{-WL} \subsetneq \mathscr{N}^-(t+1,d)\text{-WL}$

- Increasing the hops of neighbourhoods, the expressive power may decrease:

Theorem (Strong Hierarchy) $\mathscr{N}(t,d)\text{-WL} \subsetneq \mathscr{N}(t+1,d)\text{-WL}$; $\mathscr{N}(t,d)\text{-WL} \not\subsetneq \mathscr{N}(t,d+1)\text{-WL}$

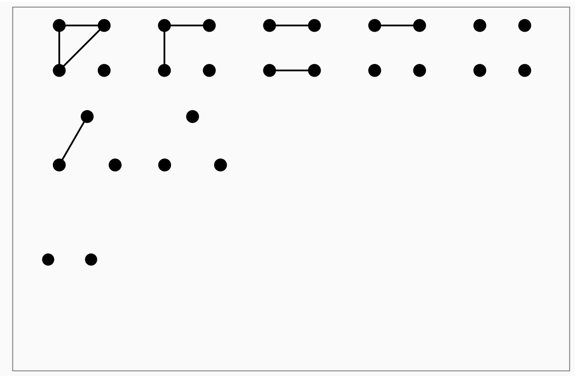
- Induced connected subgraphs remain the same expressive power:

Theorem (Equivalence) $\mathscr{N}^c(t,d)\text{-WL} \equiv \mathscr{N}(t,d)\text{-WL}$



Subgraph counts

implies $\longrightarrow$



Subgraph counts

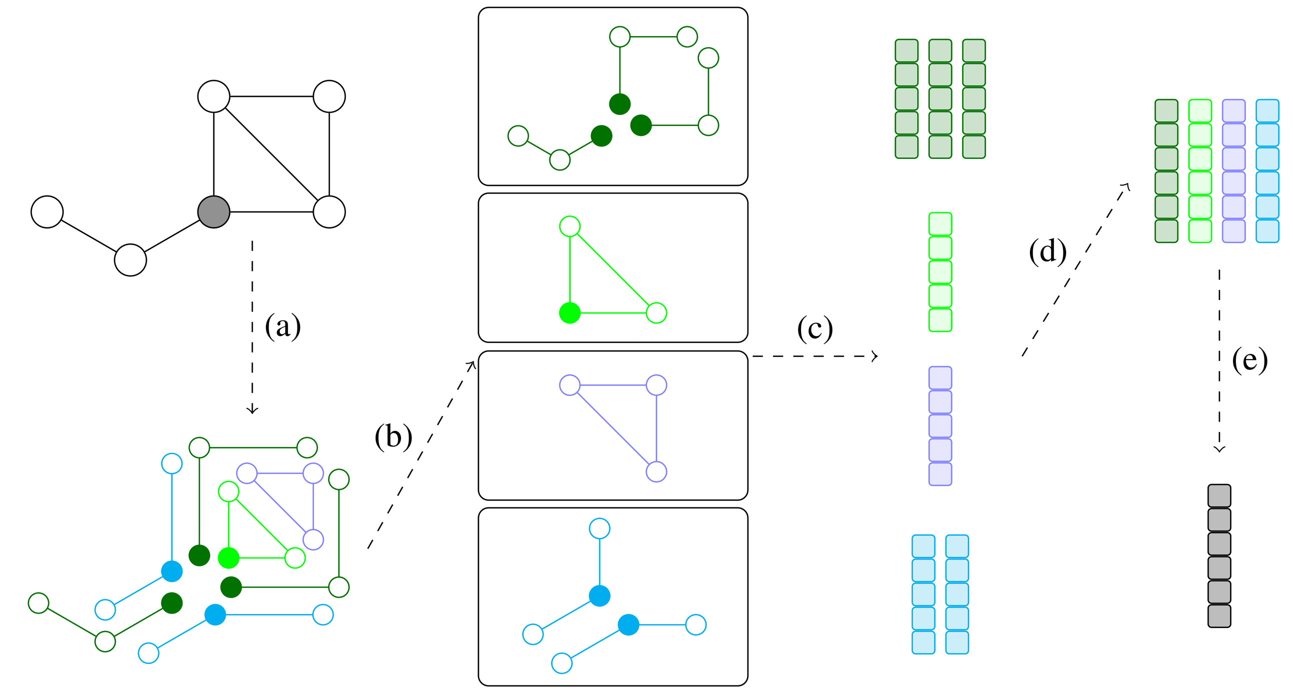
k-WL vs $\mathscr{N}$-WL

	k-WL	δ-k-LWL	(k,s)-LWL	$(k,c)(\leq)$-SETWL	$\mathscr{N}(t,d)$-WL	$\mathscr{N}^c(t,d)$-WL
#Coloured objects	n^k	n^k	subset(n^k, s)	subset$(\Sigma_{q=1}^k \binom{n}{q}, c)$	n	n
#Neighbour objects	$n \times k$	$a \times k$	$a \times k$	$n \times q$	$\binom{a^d}{t}$	subset$(\Sigma_{q=1}^t \binom{a^d}{q}, 1)$
ΔColoured objects	k-tuples	k-tuples	k-tuples	$\leq k$-sets	nodes	nodes
ΔNeighbour objects	k-tuples	k-tuples	k-tuples	$\leq k$-sets	t-sets	$\leq t$-sets
Sparsity -awareness	✗	✓	✓	✓	✗	✓

Theorem $1\text{-WL} \equiv \mathscr{N}(1,1)\text{-WL} \equiv \mathscr{N}^c(1,1)\text{-WL}$

Graph Neighbourhood Neural Network

- *Graph Neighbourhood Neural Network (G3N)* instantiates the ideas of $\mathscr{N}$-WL algorithms for graph learning.



$$h_u^{(l+1)} = \textsc{Combine}\Big(h_u^{(l)}, \textsc{Agg}^N_{(i,j)\in I_t\times J_d}\Big(\textsc{Agg}^T_{S\in\mathcal{S}_u^{(l)}(i,j)}\big(\textsc{Pool}(S)\big)\Big)\Big)$$

- Graph classification

Model	ZINC (no edge features)	ZINC (edge features)
GCN	0.4590.006	0.3210.009
PPGN	0.4070.028	-
GIN	0.3870.015	0.1630.004
PNA	0.3200.032	0.1880.004
DGN	0.2190.010	0.1680.003
DEEP LRP*	0.2230.008	-
GSN*	0.1400.006	0.1150.012
CIN*	**0.1150.003**	**0.0790.006**
G3N-(2,3)	**0.1650.018**	**0.1280.015**

Model	MolHIV (test)	MolHIV (validation)
GCN	0.76060.0097	0.82040.0141
GIN	0.75580.0140	0.82320.0090
GraphSNN	0.78510.0170	0.83590.0096
PNA	0.79050.0132	-
DGN	**0.79700.0097**	-
DEEP LRP*	0.76870.0180	0.81310.0088
GSN*	0.77990.0100	**0.86580.0084**
CIN*	**0.80940.0057**	-
G3N-(2,3)	0.79000.0134	**0.83590.0061**

- Runtime analysis