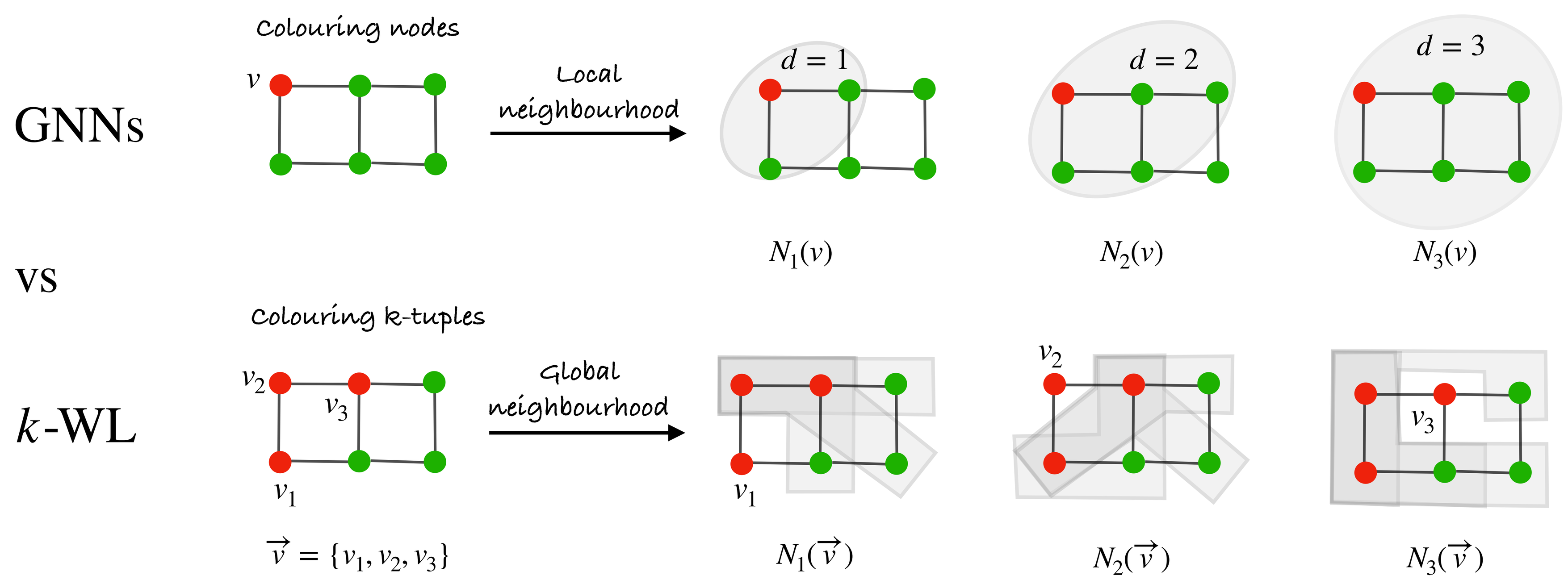
$\mathcal{N}$-WL: A New Hierarchy of Expressivity for Graph Neural Networks

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

Graph Research Lab, School of Computing, Australian National University

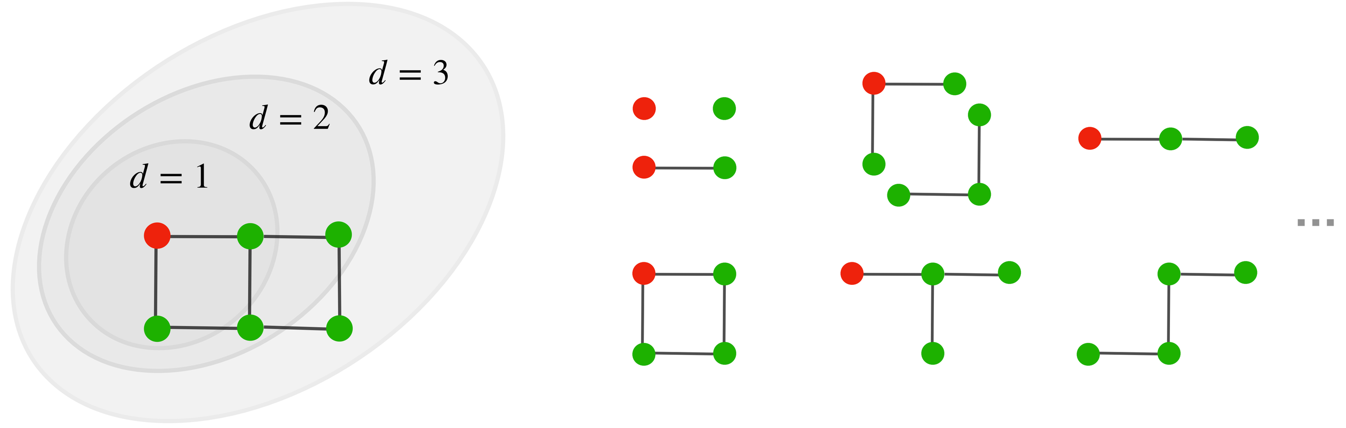
Introduction

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



Neighbourhood WL Hierarchy

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

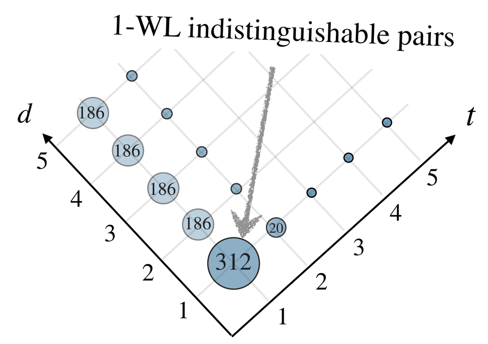


d-hop neighbourhoods *t-order induced subgraphs*

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 $\mathcal{N}$-WL



Main Results

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

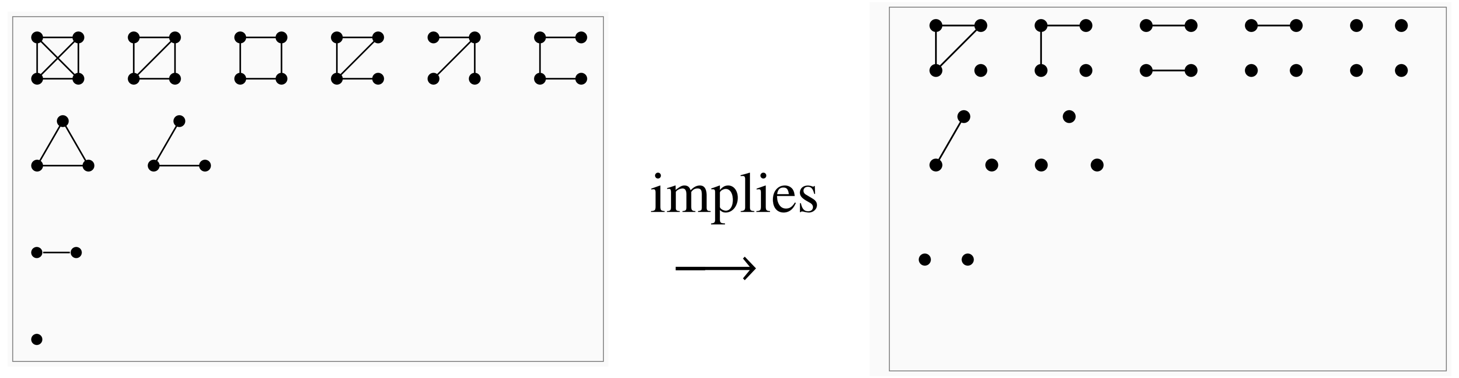
Theorem (Weak Hierarchy) $\mathcal{N}^-(t,d)\text{-WL} \subsetneq \mathcal{N}^-(t{+}1,d)\text{-WL}$

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

Theorem (Strong Hierarchy) $\mathcal{N}(t,d)\text{-WL} \subsetneq \mathcal{N}(t{+}1,d)\text{-WL}$
$\mathcal{N}(t,d)\text{-WL} \subsetneq \mathcal{N}(t,d{+}1)\text{-WL}$

- Induced connected subgraphs remain the same expressive power:

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



Subgraph counts → Subgraph counts

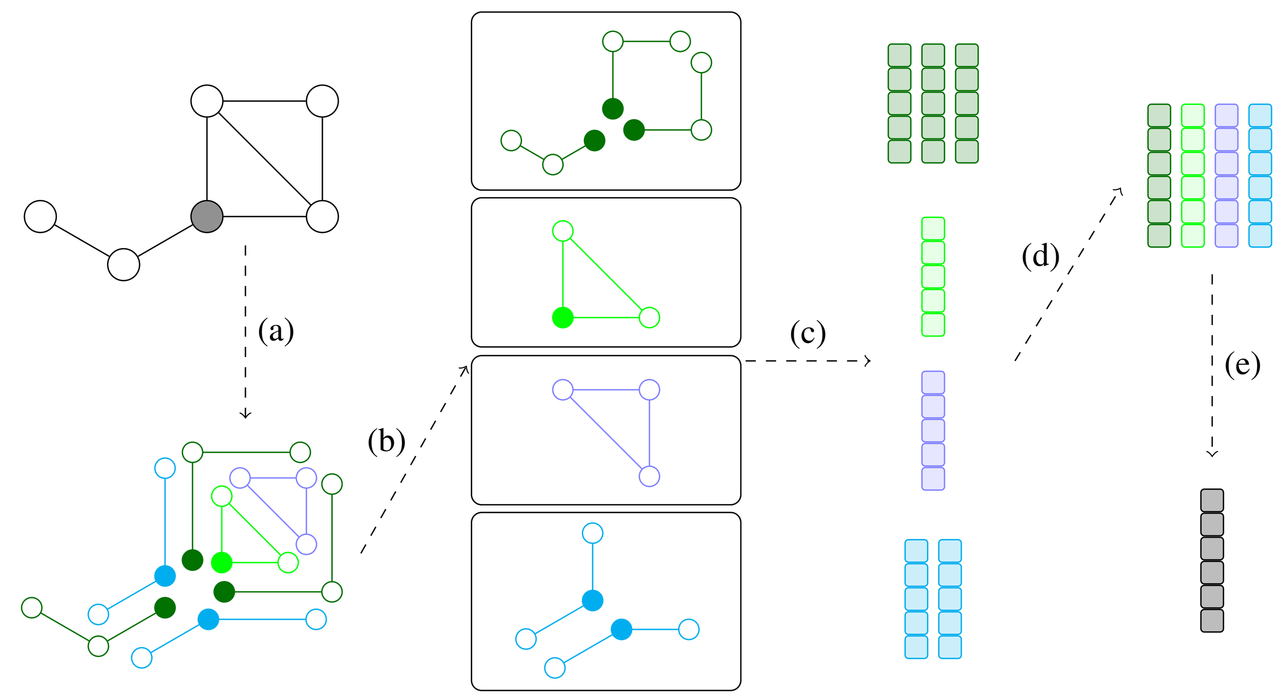
k-WL vs $\mathcal{N}$-WL

	k-WL	δ-k-LWL	(k,s)-LWL	$(k,c)(\leq)$-SETWL	$\mathcal{N}(t,d)$-WL	$\mathcal{N}^c(t,d)$-WL
#Coloured objects	n^k	n^k	subset(n^k, s)	subset$(\sum_{q=1}^{k}\binom{n}{q}, c)$	n	n
#Neighbour objects	$n\times k$	$a\times k$	$a\times k$	$n\times q$	$\binom{a^d}{t}$	subset$(\sum_{q=1}^{t}\binom{a^d}{q}, 1)$
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 $1\text{-WL} \equiv \mathcal{N}(1,1)\text{-WL} \equiv \mathcal{N}^c(1,1)\text{-WL}$

Graph Neighbourhood Neural Network

- *Graph Neighbourhood Neural Network (G3N)* instantiates the ideas of $\mathcal{N}$-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	$0.459_{\pm0.006}$	$0.321_{\pm0.009}$
PPGN	$0.407_{\pm0.028}$	-
GIN	$0.387_{\pm0.015}$	$0.163_{\pm0.004}$
PNA	$0.320_{\pm0.032}$	$0.188_{\pm0.004}$
DGN	$0.219_{\pm0.010}$	$0.168_{\pm0.003}$
DEEP LRP*	$0.223_{\pm0.008}$	-
GSN*	$0.140_{\pm0.006}$	$0.115_{\pm0.012}$
CIN*	$\mathbf{0.115_{\pm0.003}}$	$\mathbf{0.079_{\pm0.006}}$
G3N-(2,3)	$\mathbf{0.165_{\pm0.018}}$	$\mathbf{0.128_{\pm0.015}}$

Model	MolHIV (test)	MolHIV (validation)
GCN	$0.7606_{\pm0.0097}$	$0.8204_{\pm0.0141}$
GIN	$0.7558_{\pm0.0140}$	$0.8232_{\pm0.0090}$
GraphSNN	$0.7851_{\pm0.0170}$	$0.8359_{\pm0.0096}$
PNA	$0.7905_{\pm0.0132}$	-
DGN	$\mathbf{0.7970_{\pm0.0097}}$	-
DEEP LRP*	$0.7687_{\pm0.0180}$	$0.8131_{\pm0.0088}$
GSN*	$0.7799_{\pm0.0100}$	$\mathbf{0.8658_{\pm0.0084}}$
CIN*	$\mathbf{0.8094_{\pm0.0057}}$	-
G3N-(2,3)	$0.7900_{\pm0.0134}$	$\mathbf{0.8359_{\pm0.0061}}$

- Runtime analysis