A Theoretical Comparison of Graph Neural Network Extensions

Pál András Papp

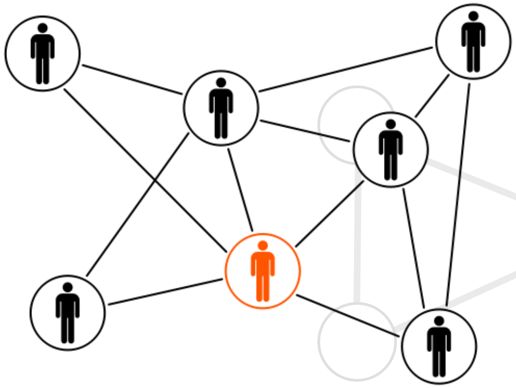
Huawei Tech., Switzerland
(this work at ETH Zürich)

Roger Wattenhofer

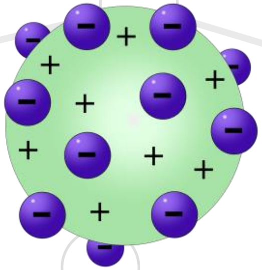
ETH Zürich

Graph Neural Networks

social networks

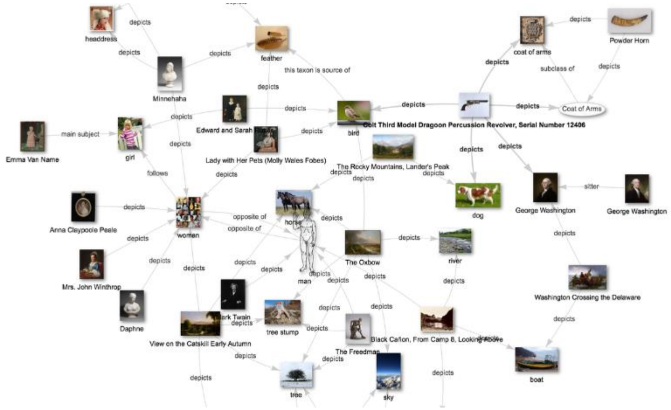


chemo-informatics

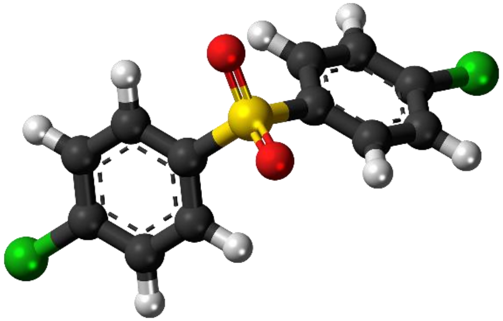


question answering systems

knowledge graphs

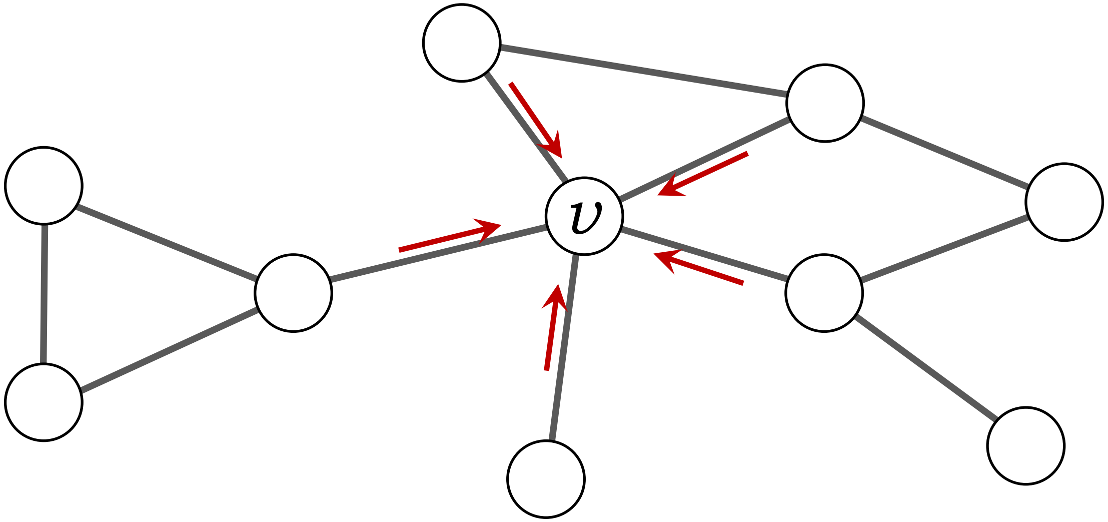


molecule recognition



recommender systems

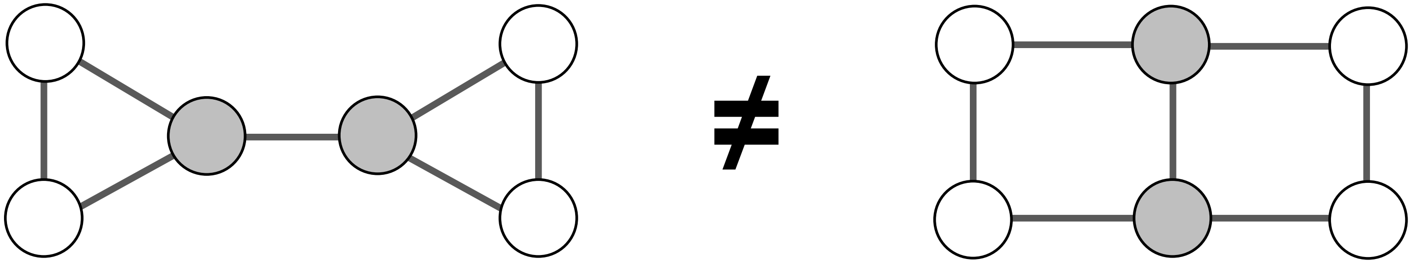
Graph Neural Networks



$$a_v = \textsc{Aggregate}\left(\{\{ h_u \mid u \in N(v) \}\} \right)$$

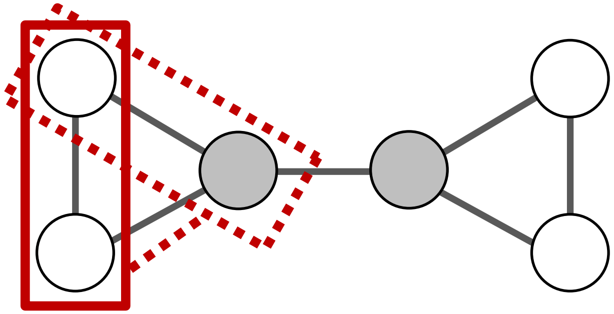
$$h_v^{(t+1)} = \textsc{Update}\left(h_v, a_v \right)$$

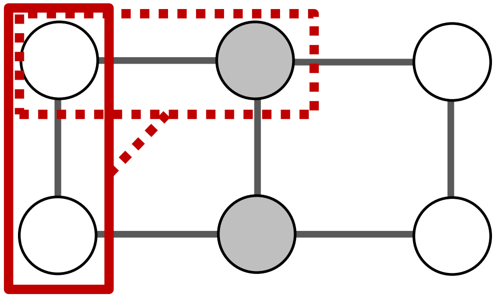
Graph Neural Networks



Cannot distinguish some very simple graphs!

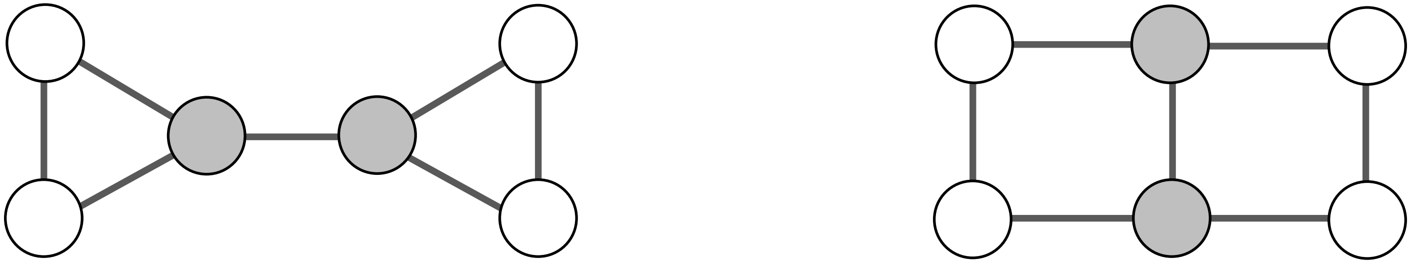
k-WL: Higher-Order Methods





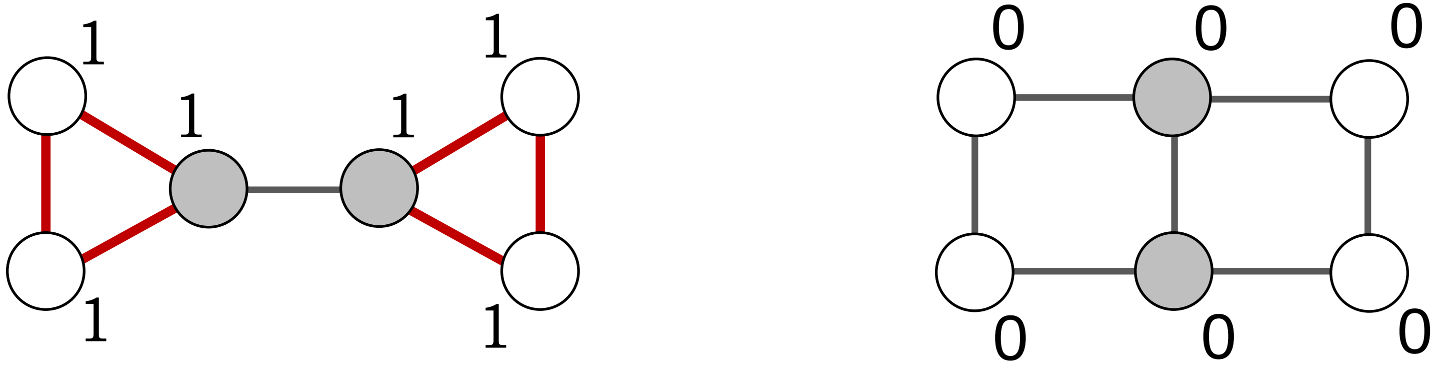
GNN on *k*-tuples of original nodes

$\boldsymbol{S_k}$: substructures up to size *k*



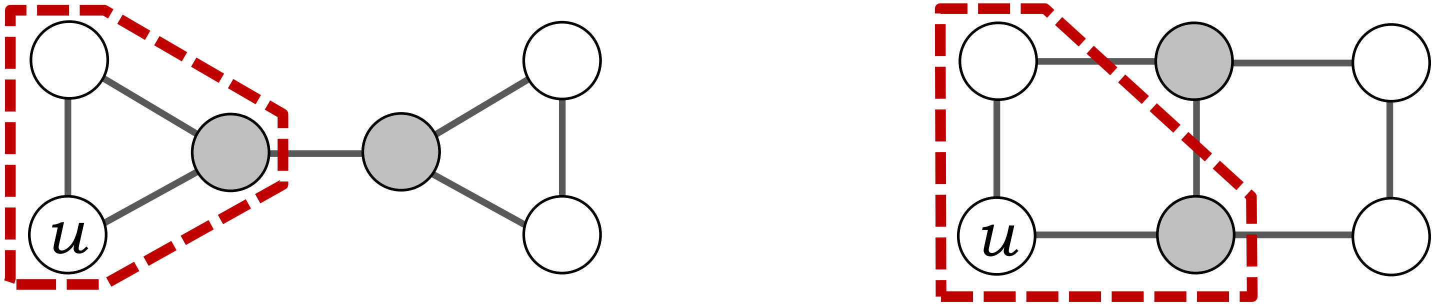
- Preprocess substructures up to size *k*
- Add as extra node features
- Run standard GNN

S_k: substructures up to size *k*



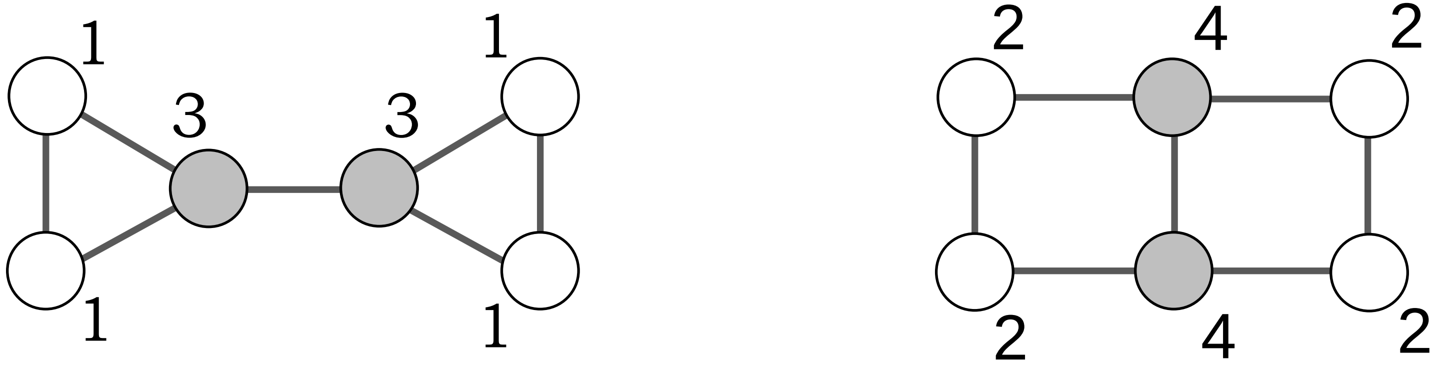
- Preprocess substructures up to size *k*
- Add as extra node features
- Run standard GNN

N_k: preprocessing up to *k* hops



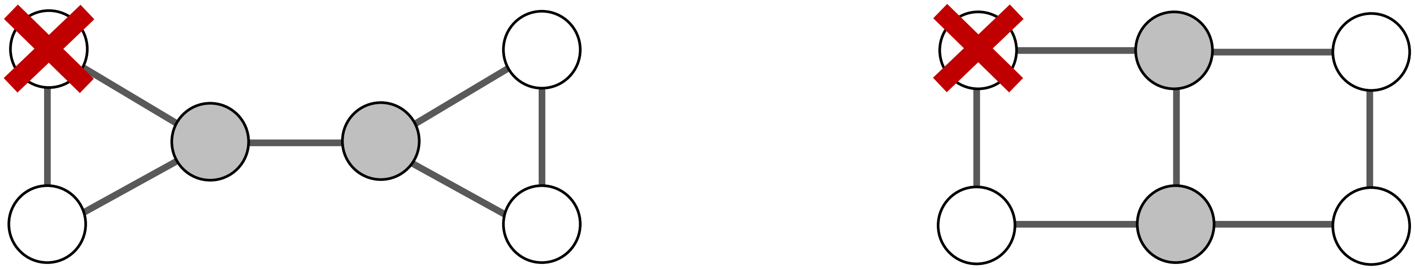
- Preprocess *k*-hop neighborhood

$\boldsymbol{N_k}$: preprocessing up to *k* hops



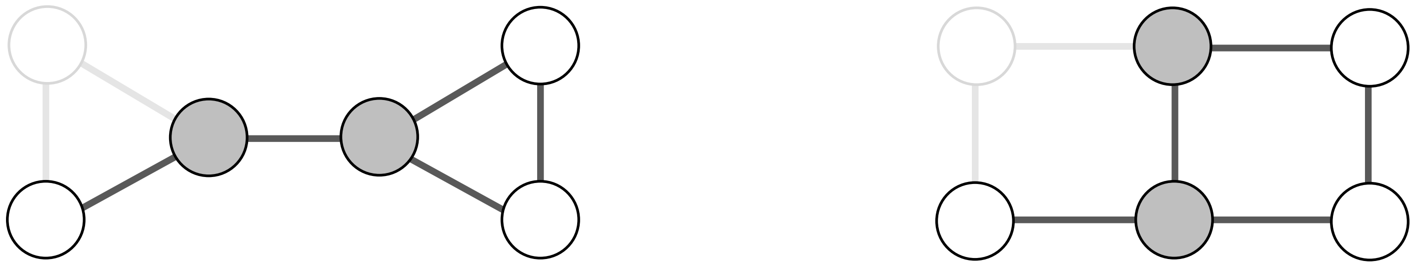
- Preprocess *k*-hop neighborhood
- Add extra features
- Run standard GNN

Removing some of the nodes



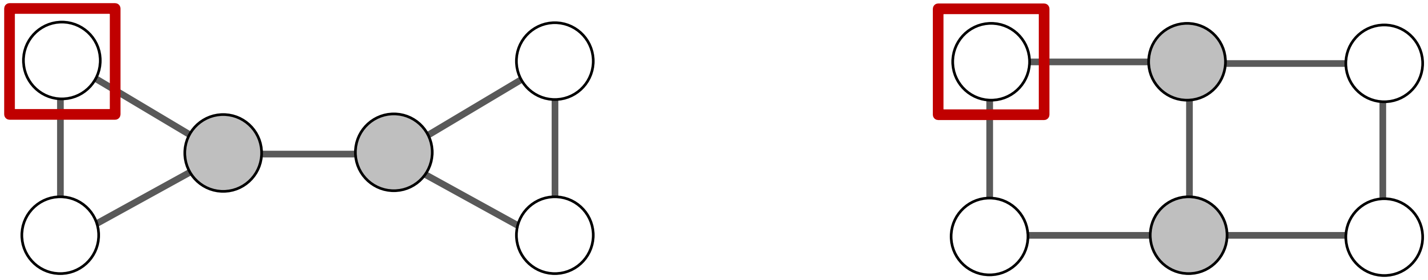
- Multiple runs of a standard GNN
- Removing some nodes in each run
- Aggregate runs in the end

Removing some of the nodes

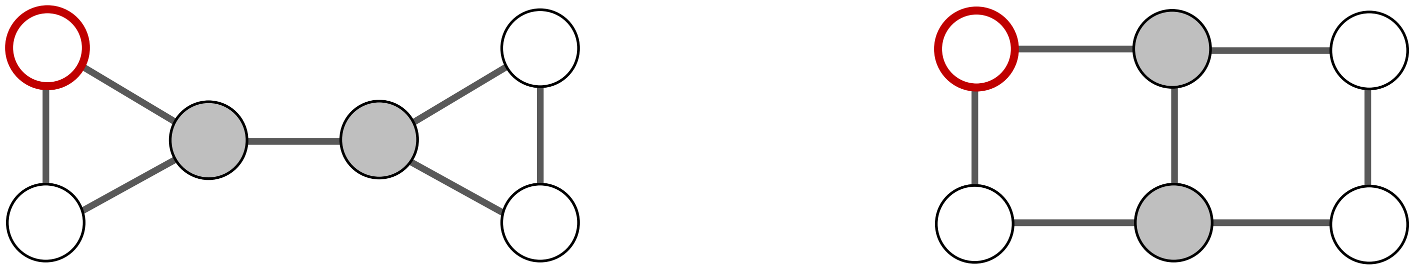


- Multiple runs of a standard GNN
- Removing some nodes in each run
- Aggregate runs in the end

$\boldsymbol{M}_k$: marking up to *k* nodes



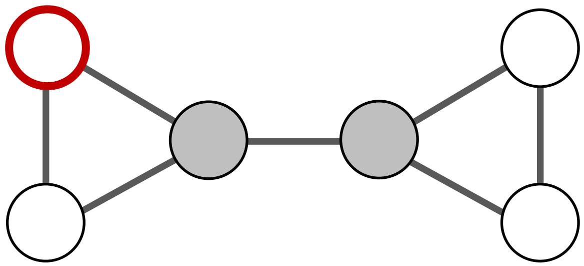
M_k: marking up to k nodes

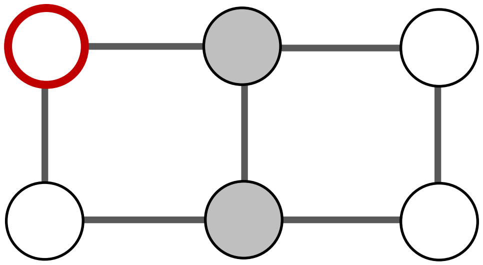


- **Mark** some nodes in each run
- Handle the marked nodes differently

$$a_{\upsilon} = \mathrm{A{\small GGR}}_{\mathit{marked}}\,(\,\dots\,) + \mathrm{A{\small GGR}}_{\mathit{unmarked}}\,(\,\dots\,)$$

***M*_*k*_: marking up to *k* nodes**





- **Mark** some nodes in each run
- Handle the marked nodes differently

$$a_\upsilon = \text{A}\textsc{ggr}_{\textit{marked}} \,(\,\ldots\,) + \text{A}\textsc{ggr}_{\textit{unmarked}} \,(\,\ldots\,)$$

↓

Strictly more expressive than node removals!

GNN extensions

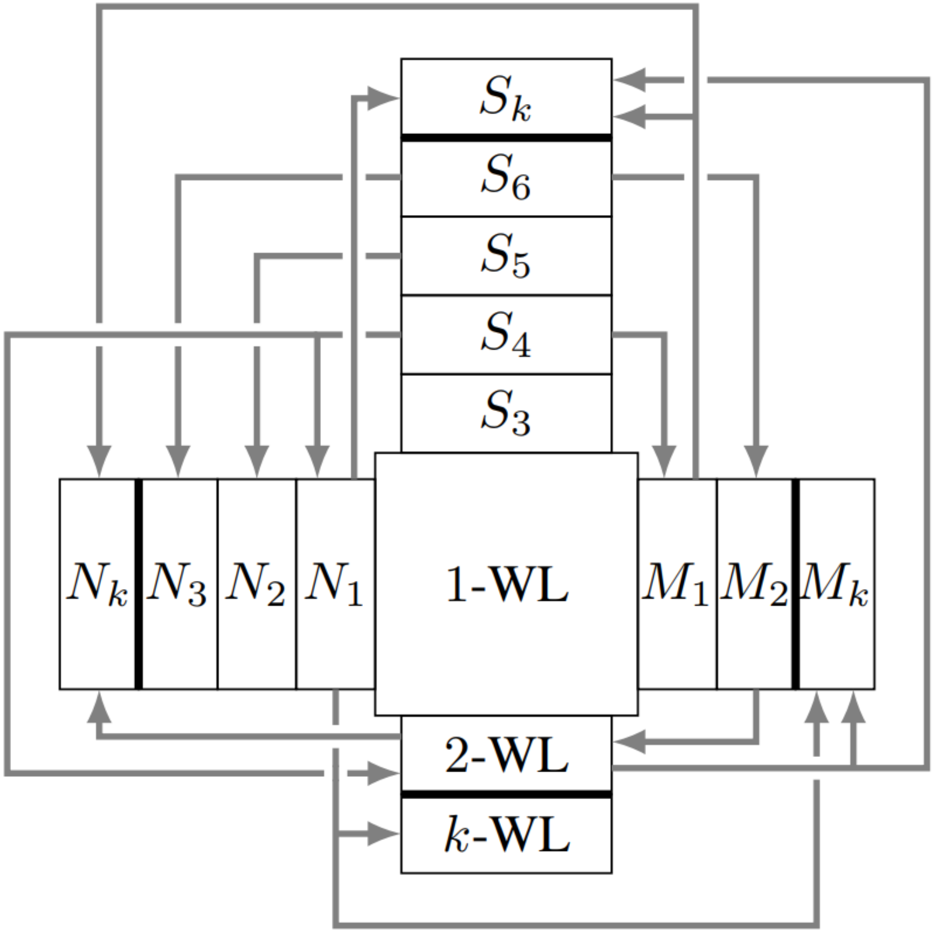
→ k-WL: k^{th} order methods

→ S_k: substructures up to size k

→ N_k: k-hop neighborhood

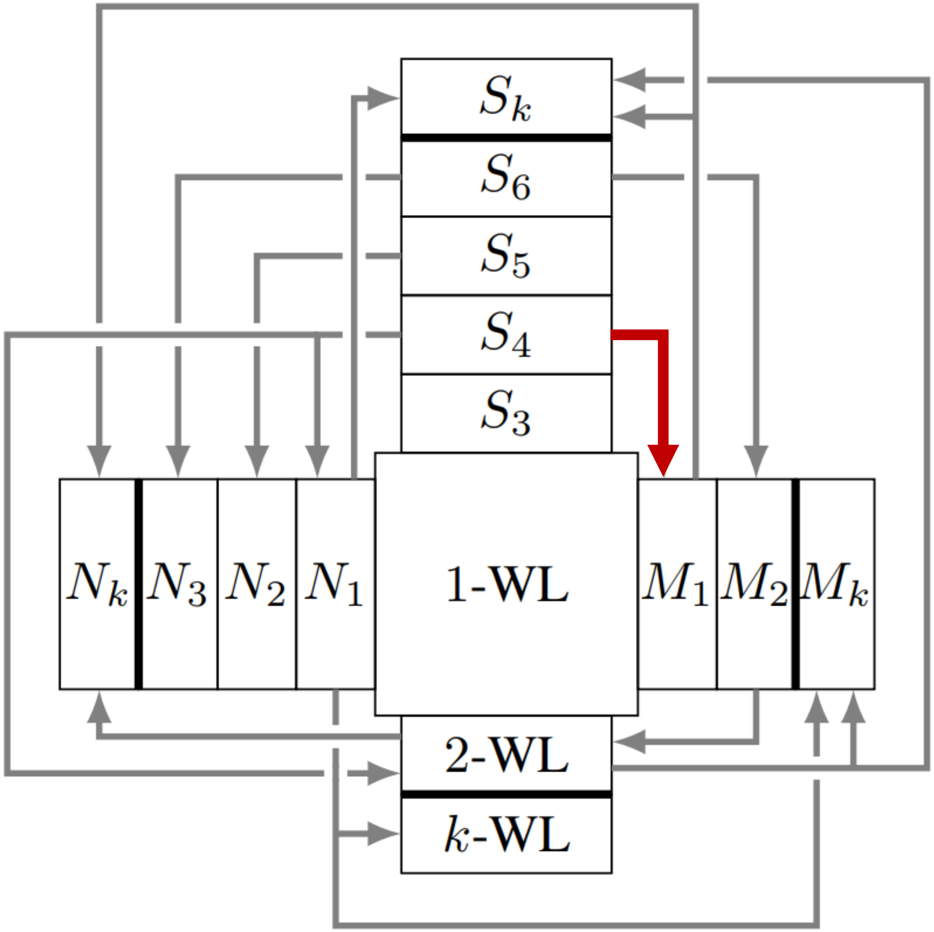
→ M_k: randomly mark k nodes

Comparison of expressiveness



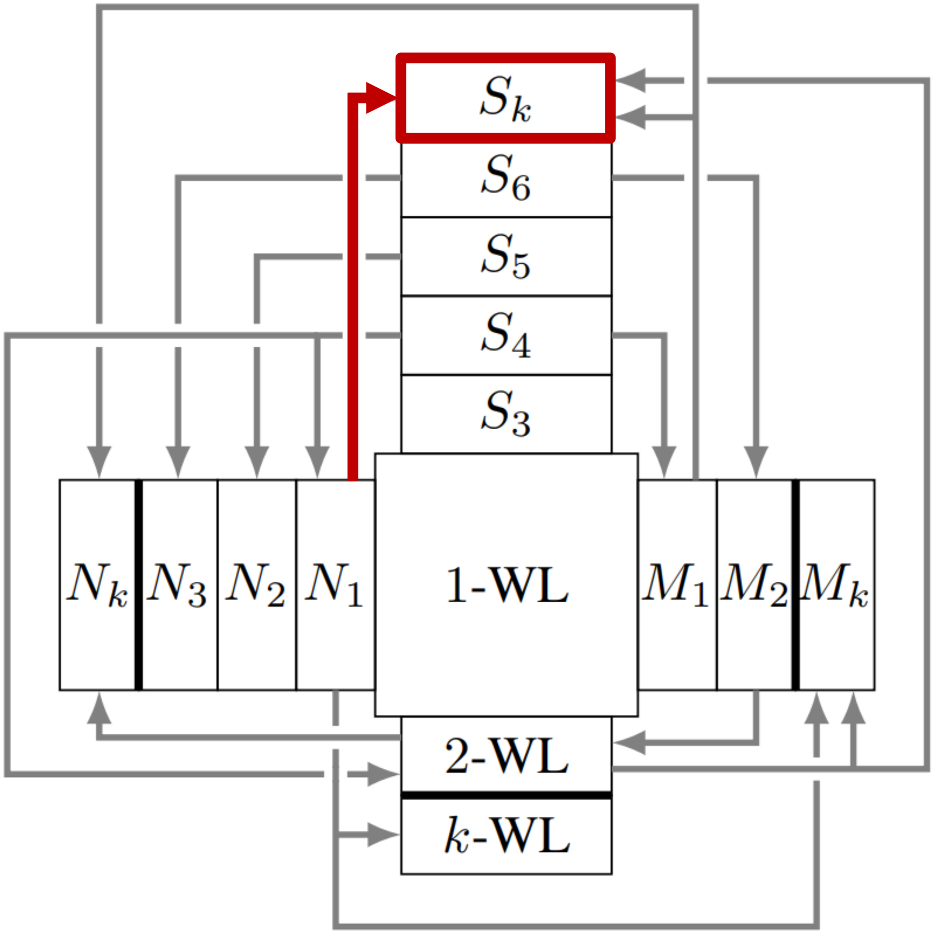
A $\longrightarrow$ ***B:*** *A can distinguish a pair of graphs that B cannot*

Comparison of expressiveness



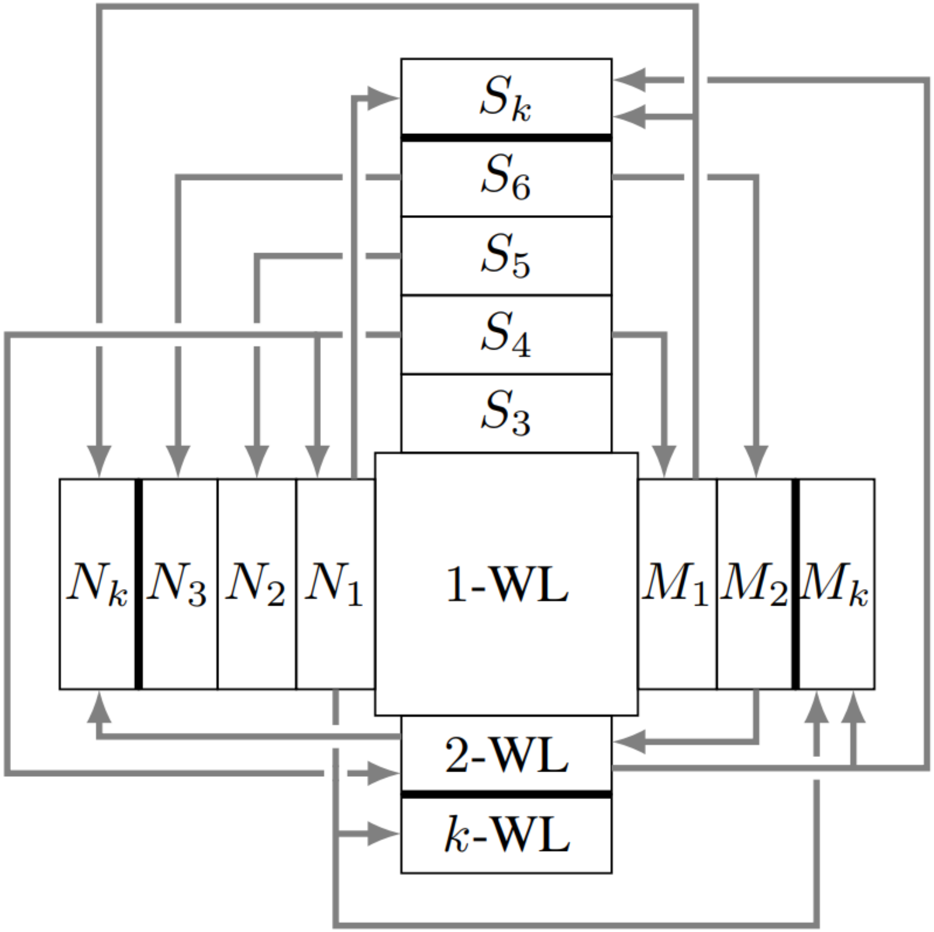
A **→** ***B:*** *A can distinguish a pair of graphs that B cannot*

Comparison of expressiveness



A → ***B:*** *A can distinguish a pair of graphs that B cannot*

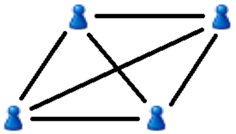
Comparison of expressiveness

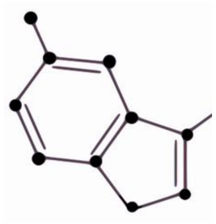


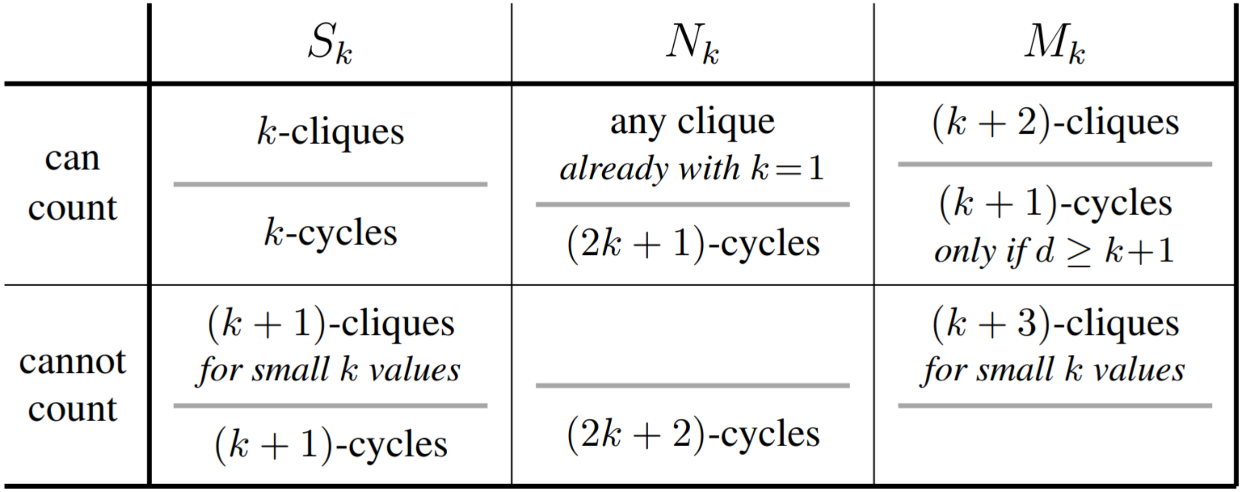
A $\longrightarrow$ ***B:*** *A can distinguish a pair of graphs that B cannot*

Counting substructures

Can GNNs count **cliques** or **cycles**?







	S_k	N_k	M_k
can count	k-cliques	any clique *already with* $k = 1$	$(k+2)$-cliques
	k-cycles	$(2k+1)$-cycles	$(k+1)$-cycles *only if* $d \geq k+1$
cannot count	$(k+1)$-cliques *for small* k *values*		$(k+3)$-cliques *for small* k *values*
	$(k+1)$-cycles	$(2k+2)$-cycles	

A Theoretical Comparison of Graph Neural Network Extensions

Pál András Papp

Huawei Tech., Switzerland
(this work at ETH Zürich)

Roger Wattenhofer

ETH Zürich