

# Understanding and Improving the Expressivity of Subgraph GNNs

Bohang Zhang

Peking University

June 23, 2023

# Index

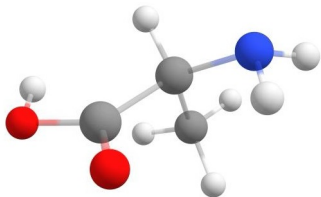
- 1 Introduction
- 2 Subgraph GNNs
- 3 A Complete Expressiveness Hierarchy for Subgraph GNNs
- 4 Localized (Folklore) Weisfeiler-Lehman Test
- 5 Strict Expressicity Separation Results
- 6 Experiments & Conclusion

# Index

- 1 Introduction
- 2 Subgraph GNNs
- 3 A Complete Expressiveness Hierarchy for Subgraph GNNs
- 4 Localized (Folklore) Weisfeiler-Lehman Test
- 5 Strict Expressicity Separation Results
- 6 Experiments & Conclusion

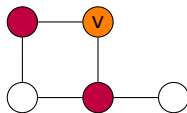
# Introduction

- Graph neural networks (GNNs) have become the dominant approach for learning on graph-structured data.



# Introduction

- The simplest GNNs are Message-passing neural networks (MPNNs): [Gilmer et al., 2017, Kipf and Welling, 2017, Hamilton et al., 2017, Veličković et al., 2018]:
  - ▶ Maintain a node feature  $h(v)$  for each node  $v$ ;
  - ▶ Update:
 
$$h^{(l)}(v) = \text{UPDATE}^{(l)} \left( h^{(l-1)}(v), \text{AGGR}^{(l)} \left( \{h^{(l-1)}(u) : u \in \mathcal{N}_G(v)\} \right) \right)$$
  - ▶ Graph representation is obtained by pooling all node representations.



MPNN Update

# Introduction

- MPNNs:

- ▶ Maintain a node feature  $h(v)$  for each node  $v$ ;

- ▶ Update:

$$h^{(l)}(v) = \text{UPDATE}^{(l)} \left( h^{(l-1)}(v), \text{AGGR}^{(l)} \left( \{ \{ h^{(l-1)}(u) : u \in \mathcal{N}_G(v) \} \} \right) \right)$$

- ▶ Graph representation is obtained by pooling all node representations.

- Examples:

- ▶ GCN [Kipf and Welling, 2017]:

$$h_v^{(l)} = \text{ReLU} \left( \mathbf{W} \left( \frac{1}{|\mathcal{N}_G(v)| + 1} \sum_{u \in \mathcal{N}_G(v) \cup v} h_u^{(l-1)} \right) + \mathbf{b} \right)$$

- ▶ GIN [Xu et al., 2019]:

$$h_v^{(l)} = \text{MLP} \left( (1 + \epsilon) h_v^{(l-1)} + \sum_{u \in \mathcal{N}_G(v)} h_u^{(l-1)} \right)$$

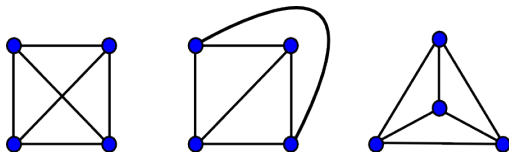
# Limitations of MPNNs

- Cannot extract pair-wise relationship between nodes
  - ▶ Not applicable to link prediction tasks
- Limited expressive power in representing graph functions
  - ▶ MPNNs has inherent drawbacks in distinguishing topologically different graphs.

$$f\left(\text{Graph}\right) = y$$


# Graph isomorphism

- Graph isomorphism problem: Given two graphs  $G = (\mathcal{V}_G, \mathcal{E}_G)$  and  $H = (\mathcal{V}_H, \mathcal{E}_H)$ , determine if there is a bijective mapping  $f: \mathcal{V}_G \rightarrow \mathcal{V}_H$ , such that  $\{u, v\} \in \mathcal{E}_G$  iff  $\{f(u), f(v)\} \in \mathcal{E}_H$ .



- Seminal work: Morris et al. [2019], Xu et al. [2019] first linked MPNN expressivity to an important algorithm called Weisfeiler-Lehman test [Weisfeiler and Leman, 1968].



# The Classic Weisfeiler-Lehman Test

- Given a graph  $G = (\mathcal{V}, \mathcal{E})$ , 1-WL computes a color mapping  $\chi_G : \mathcal{V}_G \rightarrow \mathcal{C}$  by iteratively refining each node color using its neighboring node colors.

---

## Algorithm 1: The 1-dimensional Weisfeiler-Lehman Algorithm

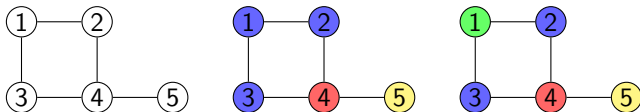
---

```

1 Initialize:  $\chi_G^0(v) := c$  for all  $v \in \mathcal{V}$  ( $c \in \mathcal{C}$  is a fixed color)
2 for  $t \leftarrow 1$  to  $T$  do
3   for each  $v \in \mathcal{V}$  do
4      $\chi_G^t(v) := \text{hash}(\chi_G^{t-1}(v), \{\{\chi_G^{t-1}(u) : u \in \mathcal{N}_G(v)\}\})$ 
5 Return:  $\chi_G^T$ 
  
```

---

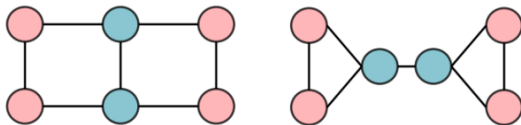
- If  $\{\{\chi_G(v) : v \in \mathcal{V}_G\}\} \neq \{\{\chi_H(v) : v \in \mathcal{V}_H\}\}$ , then  $G$  is not isomorphic to  $H$ !



Example of 1-WL (Color refinement) iterations.

# MPNNs are at Most as Expressive as 1-WL

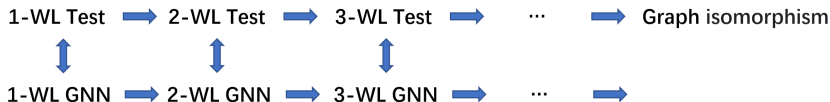
- Whenever 1-WL fails to distinguish two non-isomorphic graphs, MPNNs also fail.
- Failure cases:



- It is a central problem to study how to design more expressive GNNs beyond the 1-WL test.

# Higher-order GNNs

- A straightforward way is to leveraging *higher-order* WL variants to design provably more powerful GNNs [Morris et al., 2019, 2020, Maron et al., 2019, Geerts and Reutter, 2022].



- Given a graph  $G = (\mathcal{V}, \mathcal{E})$ ,  $k$ -FWL computes a color mapping  $\chi_G : \mathcal{V}_G^k \rightarrow \mathcal{C}$  [Cai et al., 1992].
- If  $\{\{\chi_G(v_1, \dots, v_k) : v_1, \dots, v_k \in \mathcal{V}_G\}\} \neq \{\{\chi_H(v_1, \dots, v_k) : v_1, \dots, v_k \in \mathcal{V}_H\}\}$ , then  $G$  is not isomorphic to  $H$ !

# Higher-order WL

---

## Algorithm 2: The $k$ -dimensional Folklore Weisfeiler-Lehman Algorithm

---

```

1 Initialize:  $\chi_G^0(v_1, \dots, v_k) := \text{hash}(\mathbf{A}[(v_1, \dots, v_k)])$  for all  $v_1, \dots, v_k \in \mathcal{V}_G$ 
2 for  $t \leftarrow 1$  to  $T$  do
3   for each  $v_1, \dots, v_k \in \mathcal{V}$  do
4      $\chi_G^t(v_1, \dots, v_k) := \text{hash} \left( \chi_G^{t-1}(v_1, \dots, v_k), \right.$ 
7        $\{ \chi_G^{t-1}(\mathbf{u}, v_2, \dots, v_k),$ 
8        $\chi_G^{t-1}(v_1, \mathbf{u}, \dots, v_k),$ 
9        $\dots,$ 
10       $\chi_G^{t-1}(v_1, \dots, v_{k-1}, \mathbf{u}) : u \in \mathcal{V}_G \}$ 
11     $\left. \right)$ 
5 Return:  $\chi_G^T$ 

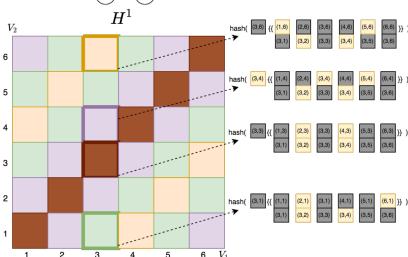
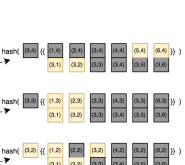
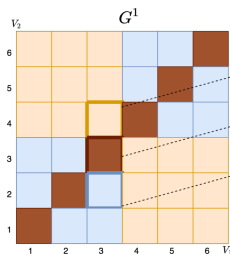
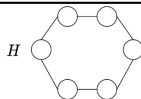
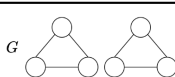
```

---

# 2-FWL

## Algorithm 3: The 2-dimensional Folklore Weisfeiler-Lehman Algorithm

- 1 **Initialize:**  $\chi_G^0(u, v) := (\mathbb{I}[u = v], A[u, v])$  for all  $u, v \in \mathcal{V}_G$
- 2 **for**  $t \leftarrow 1$  **to**  $T$  **do**
- 3     **for each**  $u, v \in \mathcal{V}$  **do**
- 4          $\chi_G^t(u, v) := \text{hash}(\chi_G^{t-1}(u, v), \{(\chi_G^{t-1}(u, w), \chi_G^{t-1}(w, v)) : w \in \mathcal{V}_G\})$
- 5 **Return:**  $\chi_G^T$



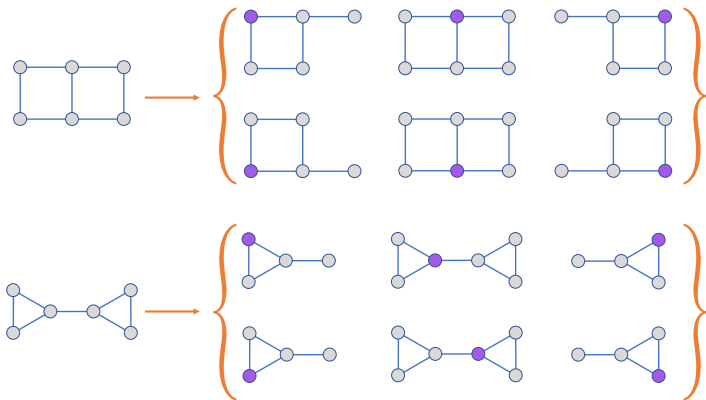
# Limitation of Higher-order GNNs

- However, higher-order GNNs suffer from several severe limitations:
  - ▶ High computation/memory costs
  - ▶ Coarse bound between 1-WL and 3-WL [Morris et al., 2022]
  - ▶ Unclear about *necessity* for real-world tasks
- Fundamental question: How can we design **simpler**, **more efficient**, **expressive**, and **practical** GNN architectures?

# Index

- 1 Introduction
- 2 Subgraph GNNs**
- 3 A Complete Expressiveness Hierarchy for Subgraph GNNs
- 4 Localized (Folklore) Weisfeiler-Lehman Test
- 5 Strict Expressicity Separation Results
- 6 Experiments & Conclusion

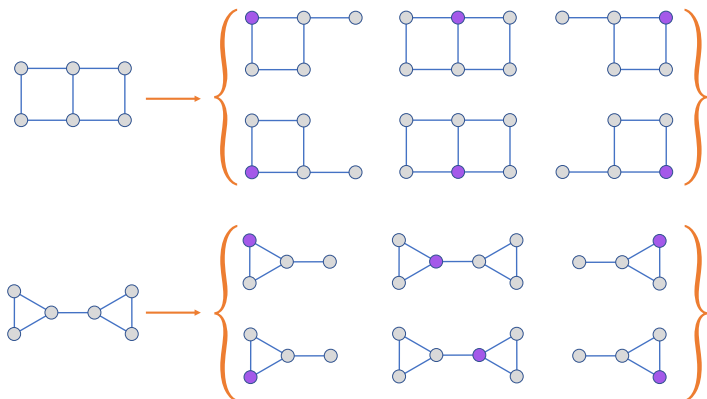
# Subgraph GNNs



- Graphs indistinguishable by MPNNs can be easily distinguished via subgraphs.
- Idea: transform a graph into a collection of subgraphs for better expressivity!

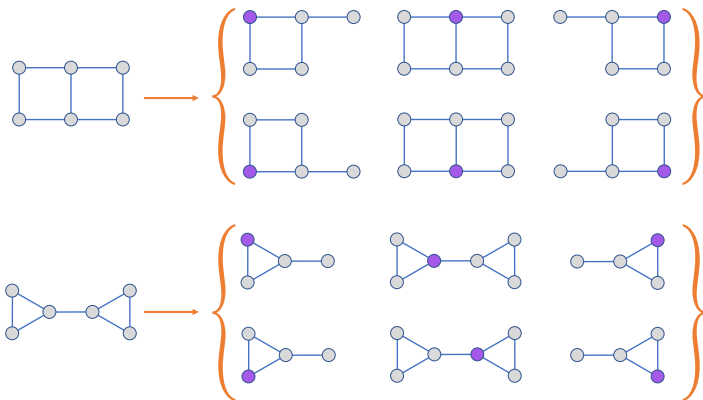


# Vanilla Subgraph GNN



- Extract  $k$ -hop ego networks for each node
- Perform MPNNs for each  $k$ -hop ego network
- Aggregate representations across all subgraphs

# General Design Space of Subgraph GNNs

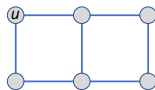


- Key question:

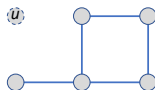
- ▶ How can we transform a graph into subgraphs?
- ▶ How can we design equivariant GNNs to process a collection of subgraphs?

# Subgraph Generation Policies

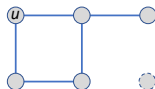
- We consider **node-based** subgraph generation policies: each subgraph is associated to a specific node of the original graph [Frasca et al., 2022].
- Commonly-used policies:
  - ▶ Node deletion [Cotta et al., 2021];
  - ▶  $k$ -hop ego network [Zhang and Li, 2021, You et al., 2021, Zhao et al., 2022, Bevilacqua et al., 2022];
  - ▶ The original graph.



The original graph



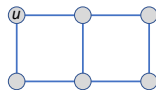
Node deletion



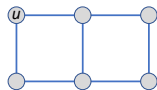
2-hop ego network

# Subgraph Generation Policies

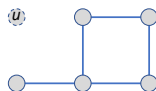
- We consider **node-based** subgraph generation policies: each subgraph is associated to a specific node of the original graph [Frasca et al., 2022].
- Commonly-used policies:
  - ▶ Note deletion [Cotta et al., 2021];
  - ▶  $k$ -hop ego network [Zhang and Li, 2021, You et al., 2021, Zhao et al., 2022, Bevilacqua et al., 2022];
  - ▶ The original graph.
- Feature initialization:
  - ▶ Constant;
  - ▶ Node marking [Qian et al., 2022];
  - ▶ Distance encoding [Zhang and Li, 2021, Zhao et al., 2022].
- Example:  $k$ -hop ego network + distance encoding.



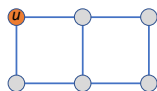
The original graph



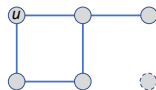
Constant



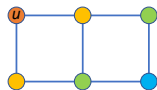
Node deletion



Node marking



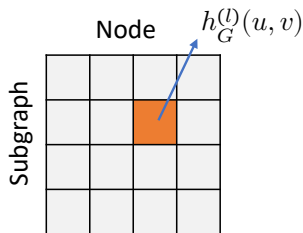
2-hop ego network



Distance Encoding

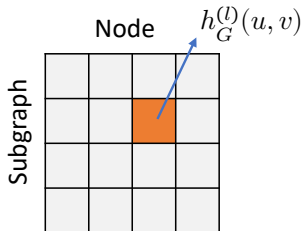
# Equivariant Message-passing Scheme

- How to design equivariant layer for a collection of subgraphs?
- Idea: treat all nodes features in all subgraphs as a 2D square matrix!



# Equivariant Message-passing Scheme

- How to design equivariant layer for a collection of subgraphs?
- Idea: treat all nodes features in all subgraphs as a 2D square matrix!
- Following Frasca et al. [2022], we study the following general design space:



$$h_G^{(l+1)}(u, v) = \text{MERGE}^{(l+1)}(\text{AGGR}_1(u, v, G, h_G^{(l)}), \dots, \text{AGGR}_r(u, v, G, h_G^{(l)}))$$

- Each atomic operation  $\text{AGGR}_i(u, v, G, h)$  takes any of the following form:
  - Single-point:  $h(u, v)$ ,  $h(v, u)$ ,  $h(u, u)$ , or  $h(v, v)$ ;
  - Global:  $\sum_{w \in \mathcal{V}_G} h(u, w)$  or  $\sum_{w \in \mathcal{V}_G} h(w, v)$ ;
  - Local:  $\sum_{w \in \mathcal{N}_G^u(v)} h(u, w)$  or  $\sum_{w \in \mathcal{N}_G^v(u)} h(w, v)$ .

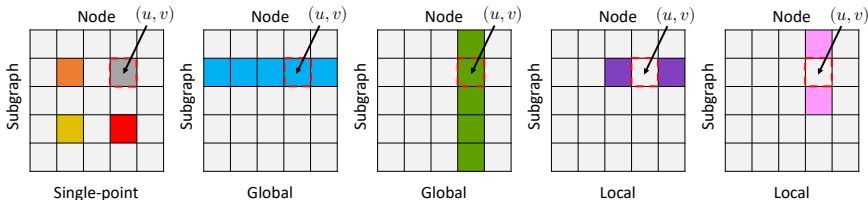
# Equivariant Message-passing Scheme

- Each atomic operation  $\text{AGGR}_i(u, v, G, h)$  takes any of the following form:

- ▶ Single-point:  $h(u, v)$ ,  $h(v, u)$ ,  $h(u, u)$ , or  $h(v, v)$ ;

- ▶ Global:  $\sum_{w \in \mathcal{V}_G} h(u, w)$  or  $\sum_{w \in \mathcal{V}_G} h(w, v)$ ;

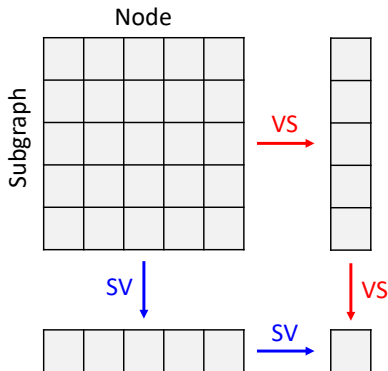
- ▶ Local:  $\sum_{w \in \mathcal{N}_{G^u}(v)} h(u, w)$  or  $\sum_{w \in \mathcal{N}_{G^v}(u)} h(w, v)$ .



- Examples: Vanilla subgraph GNNs, ESAN [Bevilacqua et al., 2022], GNN-AK [Zhao et al., 2022], SUN [Frasca et al., 2022].

# Pooling Paradigm

- How to compute a graph representation based on these subgraph node features?
- Vertex-subgraph (VS) pooling v.s. Subgraph-vertex (SV) pooling:





# Fundamental Problems in This Area

- As in previous slides, there are a huge number of combinatorial ways to design subgraph GNNs.

# Fundamental Problems in This Area

- As in previous slides, there are a huge number of combinatorial ways to design subgraph GNNs.
- **Problem 1:** How do various design paradigms differ in **expressiveness**?
  - ▶ Related to a series of open questions [Bevilacqua et al., 2022, Frasca et al., 2022, Qian et al., 2022, Zhao et al., 2022]

# Fundamental Problems in This Area

- As in previous slides, there are a huge number of combinatorial ways to design subgraph GNNs.
- **Problem 1:** How do various design paradigms differ in **expressiveness**?
  - ▶ Related to a series of open questions [Bevilacqua et al., 2022, Frasca et al., 2022, Qian et al., 2022, Zhao et al., 2022]
- **Problem 2:** What design principle achieves the **maximal** expressiveness with the **least** architectural complexity?
  - ▶ Important for the practical design of subgraph GNNs

# Fundamental Problems in This Area

- As in previous slides, there are a huge number of combinatorial ways to design subgraph GNNs.
- **Problem 1:** How do various design paradigms differ in **expressiveness**?
  - ▶ Related to a series of open questions [Bevilacqua et al., 2022, Frasca et al., 2022, Qian et al., 2022, Zhao et al., 2022]
- **Problem 2:** What design principle achieves the **maximal** expressiveness with the **least** architectural complexity?
  - ▶ Important for the practical design of subgraph GNNs
- **Problem 3:** Limitation of the subgraph GNN model class: Can we give a **tight** expressivity **upper bound** for all subgraph GNNs?
  - ▶ Frasca et al. [2022] recently bounded subgraph GNNs to be 2-FWL.
  - ▶ Whether an inherent gap exists remains a central open problem.

# Index

- 1 Introduction
- 2 Subgraph GNNs
- 3 A Complete Expressiveness Hierarchy for Subgraph GNNs**
- 4 Localized (Folklore) Weisfeiler-Lehman Test
- 5 Strict Expressicity Separation Results
- 6 Experiments & Conclusion

# Subgraph Weisfeiler-Lehman Test (SWL)

- Maintain a color for each subgraph-node pair  $(u, v)$ .
- Initially, the color  $\chi_G^0(u, v)$  is determined by the subgraph generation policy.
- Iteration:

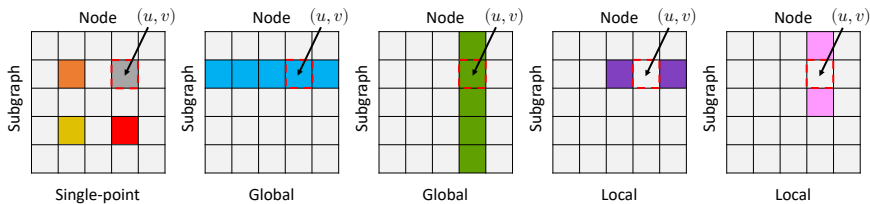
$$\chi_G^{(t+1)}(u, v) = \text{hash}(\text{agg}_1(u, v, G, \chi_G^{(t)}), \dots, \text{agg}_r(u, v, G, \chi_G^{(t)})).$$

Each  $\text{agg}_i(u, v, G, \chi)$  can take any of the following expressions:

- ▶ Single-point:  $\chi(u, v)$ ,  $\chi(v, u)$ ,  $\chi(u, u)$ , or  $\chi(v, v)$ ;
- ▶ Global:  $\{\{\chi(u, w) : w \in \mathcal{V}_G\}\}$  or  $\{\{\chi(w, v) : w \in \mathcal{V}_G\}\}$ .
- ▶ Local:  $\{\{\chi(u, w) : w \in \mathcal{N}_{G^u}(v)\}\}$  or  $\{\{\chi(w, v) : w \in \mathcal{N}_{G^v}(u)\}\}$ .

# Subgraph Weisfeiler-Lehman Test (SWL)

- Symbols for the 8 atomic aggregations:  $\text{agg}_{uv}^P$ ,  $\text{agg}_{vu}^P$ ,  $\text{agg}_{uu}^P$ ,  $\text{agg}_{vv}^P$ ,  $\text{agg}_u^G$ ,  $\text{agg}_v^G$ ,  $\text{agg}_u^L$ ,  $\text{agg}_v^L$ .



- Denote the stable color of  $(u, v)$  as  $\chi_G(u, v)$ .
  - VS pooling:**  $c(G) = \text{hash}(\{\{\text{hash}(\{\chi_G(u, v) : v \in \mathcal{V}_G\}) : u \in \mathcal{V}_G\})$ ;
  - SV pooling:**  $c(G) = \text{hash}(\{\{\text{hash}(\{\chi_G(u, v) : u \in \mathcal{V}_G\}) : v \in \mathcal{V}_G\})$ .

# Equivalence between Subgraph GNNs and SWL

## Proposition (informal)

SWL is **as powerful as** Subgraph GNNs in distinguishing non-isomorphic graphs when matching the subgraph generation policy, the aggregation scheme, and the pooling paradigm.



# Equivalence between Subgraph GNNs and SWL

## Proposition (informal)

SWL is **as powerful as** Subgraph GNNs in distinguishing non-isomorphic graphs when matching the subgraph generation policy, the aggregation scheme, and the pooling paradigm.

- Notations for “powerful”:
  - ▶  $A_1 \preceq A_2$ :  $A_2$  is **more powerful** than  $A_1$ ;
  - ▶  $A_1 \prec A_2$ :  $A_2$  is **strictly more powerful** than  $A_1$ ;
  - ▶  $A_1 \simeq A_2$ :  $A_2$  is **as powerful as**  $A_1$ ;
  - ▶  $A_1 \not\sim A_2$ :  $A_2$  is **incomparable** to  $A_1$ .

# The canonical form: node marking SWL

- Subgraph generation policy is the trickiest part in SWL.
- Surprisingly, the simple **node marking policy (on the original graph) achieves the maximal power** among other policies! (see also [Qian et al., 2022, Huang et al., 2023])
- Insight: when the special node mark is propagated
  - ▶ the color of each node pair  $(u, v)$  can encode its distance  $\text{dis}_G(u, v)$
  - ▶ the structure of  $k$ -hop ego network is also encoded

# The canonical form: node marking SWL

- Subgraph generation policy is the trickiest part in SWL.
- Surprisingly, the simple **node marking policy (on the original graph)** achieves **the maximal power** among other policies! (see also [Qian et al., 2022, Huang et al., 2023])
- Insight: when the special node mark is propagated
  - ▶ the color of each node pair  $(u, v)$  can encode its distance  $\text{dis}_G(u, v)$
  - ▶ the structure of  $k$ -hop ego network is also encoded
- Notation:  $\text{SWL}(\mathcal{A}, \text{Pool})$  denotes node marking SWL with **aggregation scheme**

$$\mathcal{A} \subset \{\text{agg}_{uu}^P, \text{agg}_{vv}^P, \text{agg}_{vu}^P, \text{agg}_u^G, \text{agg}_v^G, \text{agg}_u^L, \text{agg}_v^L\}$$

and **pooling paradigm**  $\text{Pool} \in \{\text{VS}, \text{SV}\}$ . We omit explicitly writing  $\text{agg}_{uv}^P$ .

# Analyzing Aggregation Schemes

## Theorem

For any  $\mathcal{A}$  and Pool, the following hold:

- $\text{SWL}(\mathcal{A} \cup \{\text{agg}_u^G\}, \text{Pool}) \preceq \text{SWL}(\mathcal{A} \cup \{\text{agg}_u^L\}, \text{Pool})$  and  $\text{SWL}(\mathcal{A} \cup \{\text{agg}_u^L\}, \text{Pool}) \simeq \text{SWL}(\mathcal{A} \cup \{\text{agg}_u^L, \text{agg}_u^G\}, \text{Pool})$ ;
- $\text{SWL}(\mathcal{A} \cup \{\text{agg}_{uu}^P\}, \text{Pool}) \preceq \text{SWL}(\mathcal{A} \cup \{\text{agg}_u^G\}, \text{Pool})$  and  $\text{SWL}(\mathcal{A} \cup \{\text{agg}_u^G\}, \text{Pool}) \simeq \text{SWL}(\mathcal{A} \cup \{\text{agg}_u^G, \text{agg}_{uu}^P\}, \text{Pool})$ ;
- $\text{SWL}(\{\text{agg}_u^L, \text{agg}_{vu}^P\}, \text{Pool}) \simeq \text{SWL}(\{\text{agg}_u^L, \text{agg}_v^L\}, \text{Pool}) \simeq \text{SWL}(\{\text{agg}_u^L, \text{agg}_v^L, \text{agg}_{vu}^P\}, \text{Pool})$ .

- Implication:

- ▶ Local aggregation is more powerful than global aggregation;
- ▶ Global aggregation is more powerful than single-point aggregation;
- ▶ The “transpose” aggregation  $\text{agg}_{vu}^P$  combined with one local aggregation can express the other local aggregation.

# Analyzing Pooling Paradigms

## Theorem

Let  $\text{agg}_u^L \in \mathcal{A}$ . Then,

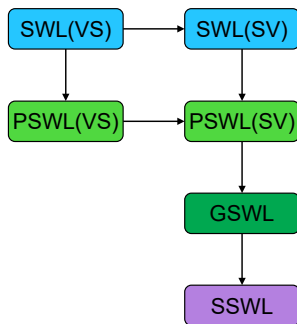
- $\text{SWL}(\mathcal{A}, \text{VS}) \preceq \text{SWL}(\mathcal{A}, \text{SV})$ ;
  - If  $\{\text{agg}_v^G, \text{agg}_v^L\} \cap \mathcal{A} \neq \emptyset$ , then  $\text{SWL}(\mathcal{A}, \text{VS}) \simeq \text{SWL}(\mathcal{A}, \text{SV})$ .
- 
- SV pooling is more powerful than VS pooling, especially when the aggregation scheme is weak (e.g, the vanilla SWL).
  - SV pooling is as powerful as VS pooling for SWL with strong aggregation schemes.

# SWL Hierarchy

## Corollary

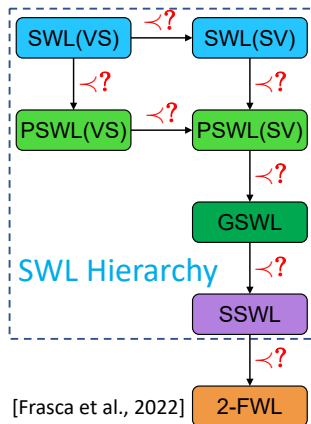
Any  $\text{SWL}(\mathcal{A}, \text{Pool})$  must be as expressive as one of the 6 SWL algorithms:

- (Vanilla SWL)  
 $\text{SWL}(\text{VS}) := \text{SWL}(\{\text{agg}_u^L\}, \text{VS}),$   
 $\text{SWL}(\text{SV}) := \text{SWL}(\{\text{agg}_u^L\}, \text{SV});$
- (SWL with additional single-point aggregation)  
 $\text{PSWL}(\text{VS}) := \text{SWL}(\{\text{agg}_u^L, \text{agg}_{vv}^P\}, \text{VS}),$   
 $\text{PSWL}(\text{SV}) := \text{SWL}(\{\text{agg}_u^L, \text{agg}_{vv}^P\}, \text{SV});$
- (SWL with additional global aggregation)  
 $\text{GSWL} := \text{SWL}(\{\text{agg}_u^L, \text{agg}_v^G\}, \text{VS});$
- (Symmetrized SWL)  
 $\text{SSWL} := \text{SWL}(\{\text{agg}_u^L, \text{agg}_v^L\}, \text{VS}).$



# What's Next?

- Strict separation of different equivalence classes?
- Expressivity upper bound?
  - ▶ All SWL algorithms have  $O(nm)$  complexity for a graph with  $n$  nodes and  $m$  edges
  - ▶ 2-FWL requires  $O(n^3)$  complexity
- Does SWL achieve the maximal expressiveness among all CR algorithms with  $O(nm)$  complexity?



# Index

- 1 Introduction
- 2 Subgraph GNNs
- 3 A Complete Expressiveness Hierarchy for Subgraph GNNs
- 4 Localized (Folklore) Weisfeiler-Lehman Test**
- 5 Strict Expressicity Separation Results
- 6 Experiments & Conclusion



# Localized Folklore WL Tests

- 2-FWL iteration:

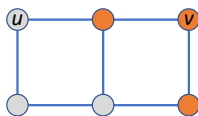
- $$\chi_G^{(t+1)}(u, v) = \text{hash} \left( \chi_G^{(t)}(u, v), \{ \{ \chi_G^{(t)}(u, w), \chi_G^{(t)}(w, v) \} : w \in \mathcal{V}_G \} \right)$$

- Can we develop an “efficient” version of 2-FWL to improve the  $O(n^3)$  complexity? (similar to the idea in Morris et al. [2020])

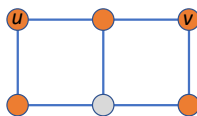
- Localized 2-FWL iteration:

- $$\chi_G^{(t+1)}(u, v) = \text{hash} \left( \chi_G^{(t)}(u, v), \{ \{ \chi_G^{(t)}(u, w), \chi_G^{(t)}(w, v) \} : w \in \mathcal{N}_G^1(v) \} \right)$$

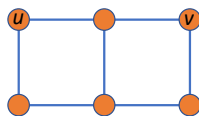
- $$\chi_G^{(t+1)}(u, v) = \text{hash} \left( \chi_G^{(t)}(u, v), \{ \{ \chi_G^{(t)}(u, w), \chi_G^{(t)}(w, v) \} : w \in \mathcal{N}_G^1(u) \cup \mathcal{N}_G^1(v) \} \right)$$



LFWL(2)



SLFWL(2)

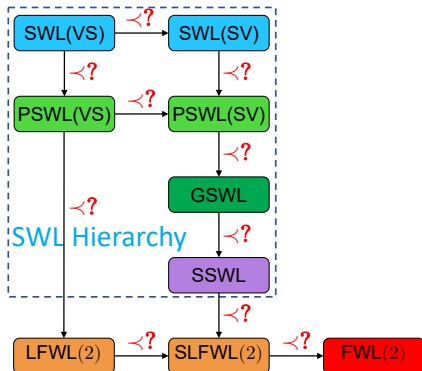


FWL(2)

# Localized Folklore WL Tests

## Theorem

- $\text{LFWL}(2) \preceq \text{SLFWL}(2) \preceq \text{FWL}(2)$ ;
- $\text{PSWL}(\text{VS}) \preceq \text{LFWL}(2)$ ;
- $\text{SSWL} \preceq \text{SLFWL}(2)$  (improving Frasca et al. [2022]).



# Localized WL Tests

- Another highly related algorithm is the localized 2-WL test [Morris et al., 2020]:

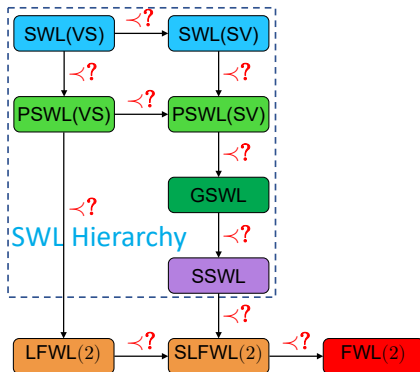
$$\begin{aligned} \blacktriangleright \chi_G^{(t+1)}(u, v) = \\ \text{hash} \left( \chi_G^{(t)}(u, v), \{ \chi_G^{(t)}(u, w) : w \in \mathcal{N}_G(v) \}, \{ \chi_G^{(t)}(w, v) : w \in \mathcal{N}_G(u) \} \right) \end{aligned}$$

## Theorem

- $\text{LFWL}(2) \preceq \text{SLFWL}(2) \preceq \text{FWL}(2)$ ;
- $\text{PSWL}(\text{VS}) \preceq \text{LFWL}(2)$ ;
- $\text{SSWL} \preceq \text{SLFWL}(2)$  (improving Frasca et al. [2022]);
- $\text{SSWL} \simeq \text{LWL}(2)$ .

# Open Questions

- Gap between 2-FWL and localized variants?
- Gap between localized FWL and localized WL?
- Gap between SLFWL(2) and subgraph GNNs?



# Index

- 1 Introduction
- 2 Subgraph GNNs
- 3 A Complete Expressiveness Hierarchy for Subgraph GNNs
- 4 Localized (Folklore) Weisfeiler-Lehman Test
- 5 Strict Expressivity Separation Results**
- 6 Experiments & Conclusion

# A Unified Pebbling Game Framework

- To prove strict separation results, we develop an analyzing framework based on pebbling games [Cai et al., 1992].
- Pebbling game:
  - ▶ Two players: Spoiler and Duplicator;
  - ▶ Two graphs:  $G$  and  $H$  (with the same number of nodes).
  - ▶ Each graph is equipped with two pebbles:  $u$  and  $v$ .
  - ▶ Initially, pebbles are outside the graphs.

# Subgraph Pebbling Game (Initialization)

- For VS pooling:
  - 1 Duplicator chooses an arbitrary bijection  $f: \mathcal{V}_G \rightarrow \mathcal{V}_H$ .
  - 2 Spoiler picks pebbles  $u$  of the two graphs on arbitrary  $x \in \mathcal{V}_G$  and  $f(x) \in \mathcal{V}_H$ , respectively.
  - 3 Duplicator chooses another arbitrary bijection  $g: \mathcal{V}_G \rightarrow \mathcal{V}_H$ .
  - 4 Spoiler picks pebbles  $v$  of the two graphs on arbitrary  $y \in \mathcal{V}_G$  and  $g(y) \in \mathcal{V}_H$ , respectively.
- For SV pooling: first pick pebbles  $v$  and then pebbles  $u$ .

# Subgraph Pebbling Game (Iteration)

- For each iteration:
  - ▶ Spoiler selects an aggregation  $\text{agg} \in \mathcal{A}$
  - ▶ For  $\text{agg}_{uu}^P$ , move pebble  $v$  to the position of pebble  $u$  for both graph
  - ▶ For  $\text{agg}_{vu}^P$ , swap pebble  $v$  with  $u$  for both graph
  - ▶ For  $\text{agg}_u^G$ :
    - ① Duplicator chooses an arbitrary bijection  $g : \mathcal{V}_G \rightarrow \mathcal{V}_H$ .
    - ② Spoiler chooses on arbitrary  $x \in \mathcal{V}_G$  and the corresponding  $g(x) \in \mathcal{V}_H$ , and moves pebbles  $v$  of the two graphs to  $x$  and  $g(x)$ , respectively.
  - ▶ For  $\text{agg}_u^L$ :
    - ① Duplicator chooses an arbitrary bijection  $g : \mathcal{N}_G(v) \rightarrow \mathcal{N}_H(v)$  (losing the game if  $|\mathcal{N}_G(v)| \neq |\mathcal{N}_H(v)|$ ).
    - ② Spoiler chooses on arbitrary  $x \in \mathcal{N}_G(v)$  and the corresponding  $g(x) \in \mathcal{N}_H(v)$ , and moves pebbles  $v$  of the two graphs to  $x$  and  $g(x)$ , respectively.
  - ▶ Similar for  $\text{agg}_{vv}^P$ ,  $\text{agg}_v^G$ , and  $\text{agg}_v^L$ .



# Subgraph Pebbling Game (Winning Judgement)

- After each iteration, Spoiler wins if the isomorphism type of  $u, v$  differs in  $G$  and  $H$ .

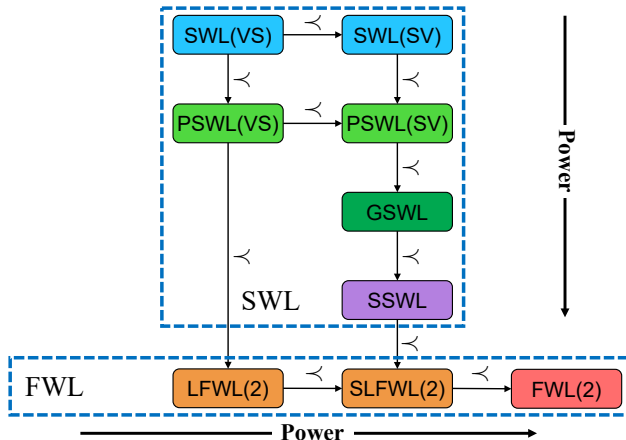
## Theorem

Any node marking SWL algorithm can distinguish a pair of graphs  $G$  and  $H$  if and only if Spoiler can win the corresponding pebbling game on  $G$  and  $H$ .

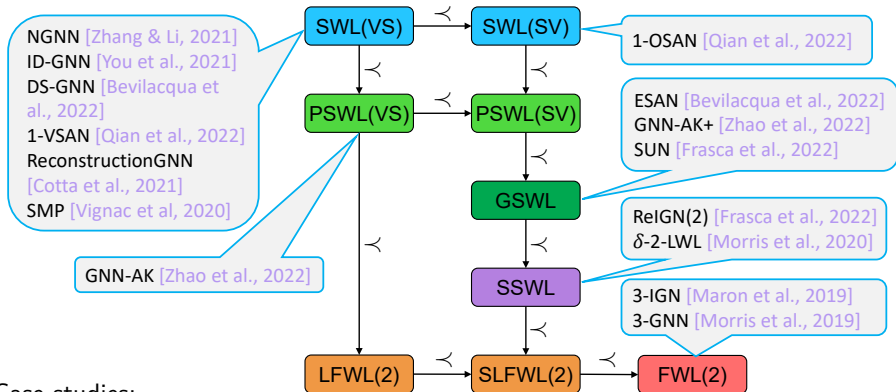
- What about localized FWL?

# Strict Separation Results

- All relations “ $\preceq$ ” in previous slides can be replaced by  $\prec$ !
- Proofs are based on skillfully constructing non-trivial counterexample graphs [Fürer, 2001] and study pebbling games on these graphs [Cai et al., 1992].



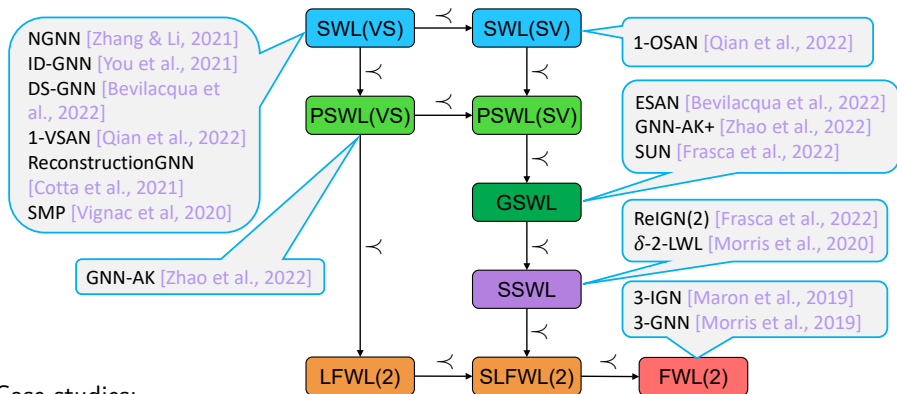
# Discussions with Prior Works



Case studies:

- DS-GNN v.s. DSS-GNN [Bevilacqua et al., 2022]
- GNN-AK v.s. GNN-AK-ctx [Zhao et al., 2022]
- OSAN v.s. VSAN [Qian et al., 2022]

# Discussions with Prior Works



Case studies:

- RelGN(2) v.s. SUN [Frasca et al., 2022]
- RelGN(2) v.s. 3-WL [Frasca et al., 2022]
- RelGN(2) v.s.  $\delta$ -2-LWL [Frasca et al., 2022, Morris et al., 2020]

# Index

- 1 Introduction
- 2 Subgraph GNNs
- 3 A Complete Expressiveness Hierarchy for Subgraph GNNs
- 4 Localized (Folklore) Weisfeiler-Lehman Test
- 5 Strict Expressicity Separation Results
- 6 Experiments & Conclusion

# Experiments on Counting Substructures

- We adopt the elegant SSWL-based subgraph GNN design principle.
- Two models:
  - ▶ GNN-SSWL:  $\text{SWL}(\text{agg}_u^L, \text{agg}_v^L)$
  - ▶ GNN-SSWL+:  $\text{SWL}(\text{agg}_u^L, \text{agg}_v^L, \text{agg}_{vv}^P)$

Performance comparison of different subgraph GNNs on ZINC benchmark.

Model	Reference	Triangle	Tailed Tri.	Star	4-Cycle	5-Cycle	6-Cycle
PPGN	Maron et al. [2019]	0.0089	0.0096	0.0148	0.0090	0.0137	0.0167
GNN-AK	Zhao et al. [2022]	0.0934	0.0751	0.0168	0.0726	0.1102	0.1063
GNN-AK+	Zhao et al. [2022]	0.0123	0.0112	0.0150	0.0126	0.0268	0.0584
SUN (EGO+)	Frasca et al. [2022]	0.0079	0.0080	<b>0.0064</b>	0.0105	0.0170	0.0550
GNN-SSWL	This paper	0.0098	0.0090	0.0089	0.0107	0.0142	0.0189
GNN-SSWL+	This paper	<b>0.0064</b>	<b>0.0067</b>	0.0078	<b>0.0079</b>	<b>0.0108</b>	<b>0.0154</b>

# Experiments on ZINC

- We adopt the elegant SSWL-based subgraph GNN design principle.
- Two models:
  - ▶ GNN-SSWL:  $\text{SWL}(\text{agg}_u^L, \text{agg}_v^L)$
  - ▶ GNN-SSWL+:  $\text{SWL}(\text{agg}_u^L, \text{agg}_v^L, \text{agg}_{vv}^P)$

Performance comparison of different subgraph GNNs on ZINC benchmark.

Model	Reference	WL	# Param.	# Agg.	ZINC Test MAE	
					Subset	Full
GSN	Bouritsas et al. [2022]	-	~500k	-	0.101±0.010	-
CIN (small)	Bodnar et al. [2021]	-	~100k	-	0.094±0.004	0.044±0.003
NGNN	Zhang and Li [2021]	SWL(VS)	~500k	2	0.111±0.003	0.029±0.001
GNN-AK	Zhao et al. [2022]	PSWL(VS)	~500k	4	0.105±0.010	-
GNN-AK-ctx	Zhao et al. [2022]	GSWL	~500k	5	0.093±0.002	-
ESAN	Bevilacqua et al. [2022]	GSWL	~100k	4	0.102±0.003	0.029±0.003
ESAN	Frasca et al. [2022]	GSWL	446k	4	0.097±0.006	0.025±0.003
SUN	Frasca et al. [2022]	GSWL	526k	12	0.083±0.003	0.024±0.003
GNN-SSWL	This paper	SSWL	274k	3	0.082±0.003	0.026±0.001
GNN-SSWL+	This paper	SSWL	387k	4	<b>0.070±0.005</b>	<b>0.022±0.002</b>

# Take Aways

- Different subgraph GNN design approaches vary significantly in their expressive power and also the practical ability to encode fundamental graph properties.
- Subgraphs GNNs is highly related localized Folklore WL test.
- There is an inherent gap between subgraph GNNs and 2-FWL.



# Open Directions

- Expressiveness hierarchy of higher-order subgraph GNNs
- Edge-based subgraph GNNs
- Localized FWL
- Practical expressiveness of GSWL and SSWL

Paper can be found at arxiv 2302.07090 or at ICML 2023  
(<https://openreview.net/forum?id=2Hp7U3k5Ph>)

Joint work with Guhao Feng, Yiheng Du, Di He, and Liwei Wang

# References I

- Beatrice Bevilacqua, Fabrizio Frasca, Derek Lim, Balasubramaniam Srinivasan, Chen Cai, Gopinath Balamurugan, Michael M Bronstein, and Haggai Maron. Equivariant subgraph aggregation networks. In *International Conference on Learning Representations*, 2022.
- Cristian Bodnar, Fabrizio Frasca, Nina Otter, Yu Guang Wang, Pietro Liò, Guido Montufar, and Michael M. Bronstein. Weisfeiler and lehman go cellular: CW networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021.
- Giorgos Bouritsas, Fabrizio Frasca, Stefanos P Zafeiriou, and Michael Bronstein. Improving graph neural network expressivity via subgraph isomorphism counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- Jin-Yi Cai, Martin Fürer, and Neil Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.

# References II

- Leonardo Cotta, Christopher Morris, and Bruno Ribeiro. Reconstruction for powerful graph representations. In *Advances in Neural Information Processing Systems*, volume 34, pages 1713–1726, 2021.
- Fabrizio Frasca, Beatrice Bevilacqua, Michael Bronstein, and Haggai Maron. Understanding and extending subgraph gnns by rethinking their symmetries. *ArXiv*, abs/2206.11140, 2022.
- Martin Fürer. Weisfeiler-lehman refinement requires at least a linear number of iterations. In *International Colloquium on Automata, Languages, and Programming*, pages 322–333. Springer, 2001.
- Floris Geerts and Juan L Reutter. Expressiveness and approximation properties of graph neural networks. In *International Conference on Learning Representations*, 2022.
- Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

# References III

- William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, volume 30, pages 1025–1035, 2017.
- Yinan Huang, Xingang Peng, Jianzhu Ma, and Muhan Zhang. Boosting the cycle counting power of graph neural networks with  $i\hat{s}^2$ -GNNs. In *The Eleventh International Conference on Learning Representations*, 2023.
- Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.
- Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. In *Advances in neural information processing systems*, volume 32, pages 2156–2167, 2019.
- Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4602–4609, 2019.

# References IV

- Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: towards scalable higher-order graph embeddings. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 21824–21840, 2020.
- Christopher Morris, Gaurav Rattan, Sandra Kiefer, and Siamak Ravanbakhsh. Speqnets: Sparsity-aware permutation-equivariant graph networks. In *International Conference on Machine Learning*, pages 16017–16042. PMLR, 2022.
- Chendi Qian, Gaurav Rattan, Floris Geerts, Christopher Morris, and Mathias Niepert. Ordered subgraph aggregation networks. *arXiv preprint arXiv:2206.11168*, 2022.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.
- Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *NTI, Series*, 2(9):12–16, 1968.

# References V

- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019.
- Jiaxuan You, Jonathan M Gomes-Selman, Rex Ying, and Jure Leskovec. Identity-aware graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10737–10745, 2021.
- Muhan Zhang and Pan Li. Nested graph neural networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 15734–15747, 2021.
- Lingxiao Zhao, Wei Jin, Leman Akoglu, and Neil Shah. From stars to subgraphs: Uplifting any gnn with local structure awareness. In *International Conference on Learning Representations*, 2022.

# Thank You!