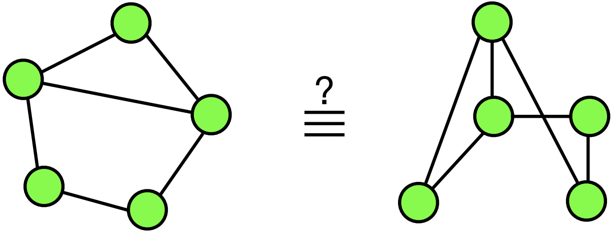
A New Perspective on ”How Graph Neural Networks Go Beyond Weisfeiler-Lehman?”

Asiri Wijesinghe & Qing Wang

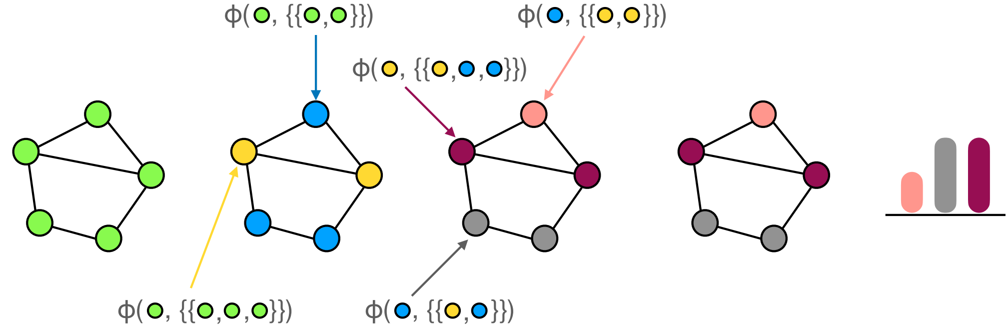
School of Computing, Australian National University
Canberra, Australia

Graph Isomorphism

- Are two graphs isomorphic?

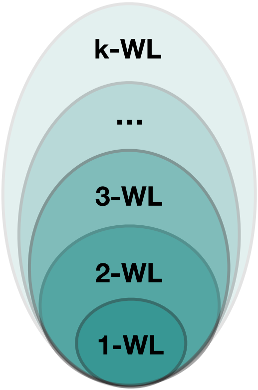


Classical Weisfeiler-Lehman (1-WL)



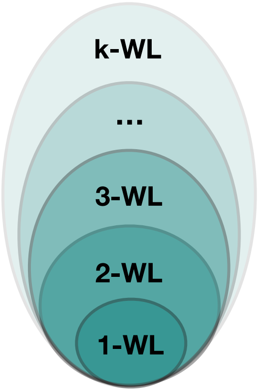
Weisfeiler-Lehman (WL) Hierarchy

- **Known results:**



Weisfeiler-Lehman (WL) Hierarchy

- **Known results:**

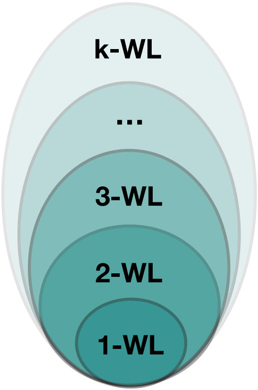


- $<$ 1-WL

  e.g., Graph Convolutional Network (GCN) [Kipf and Welling 2017]

Weisfeiler-Lehman (WL) Hierarchy

- **Known results:**



- $\equiv$ 1-WL

  e.g., Graph Isomorphism Network (GIN) [Xu et al. 2019]

- $<$ 1-WL

  e.g., Graph Convolutional Network (GCN) [Kipf and Welling 2017]

Go Beyond (Classical) Weisfeiler-Lehman?

- **Main directions:**
  1. Build higher-order variants of GNNs that are as powerful as k-WL with $k \geq 3$;

Go Beyond (Classical) Weisfeiler-Lehman?

- **Main directions:**
  1. Build higher-order variants of GNNs that are as powerful as k-WL with $k \geq 3$;
  2. Incorporate pre-defined topological features such as triangles, cliques, and rings;

Go Beyond (Classical) Weisfeiler-Lehman?

- **Main directions:**
  1. Build higher-order variants of GNNs that are as powerful as k-WL with $k \geq 3$;
  2. Incorporate pre-defined topological features such as triangles, cliques, and rings;
  3. Augment GNNs with node identifiers or random features.

Go Beyond (Classical) Weisfeiler-Lehman?

Our work injects properties of structural interactions among vertices without sacrificing computational simplicity and efficiency.

Go Beyond (Classical) Weisfeiler-Lehman?

Our work injects properties of structural interactions among vertices without sacrificing computational simplicity and efficiency.

- **Key observations**

  O1: Treating a neighborhood as a multiset of feature vectors ignores the rich structure among vertices in the neighborhood;

Go Beyond (Classical) Weisfeiler-Lehman?

Our work injects properties of structural interactions among vertices without sacrificing computational simplicity and efficiency.

- **Key observations**

  O1: Treating a neighborhood as a multiset of feature vectors ignores the rich structure among vertices in the neighborhood;

  O2: There exists a natural hierarchy of local isomorphism relating to GNNs' aggregation schemes;

Go Beyond (Classical) Weisfeiler-Lehman?

Our work injects properties of structural interactions among vertices without sacrificing computational simplicity and efficiency.

- **Key observations**

  O1: Treating a neighborhood as a multiset of feature vectors ignores the rich structure among vertices in the neighborhood;

  O2: There exists a natural hierarchy of local isomorphism relating to GNNs' aggregation schemes;

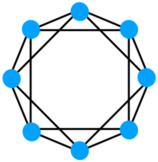
  O3: Designing an injective scheme that considers structural interactions of vertices may lead to more expressive GNNs.

Neighborhood Subgraph

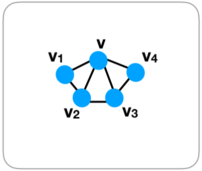
- For each vertex v, the **neighborhood subgraph** S_v is the subgraph induced by $\mathcal{N}(v) \cup \{v\}$.

Neighborhood Subgraph

- For each vertex v, the **neighborhood subgraph** S_v is the subgraph induced by $\mathcal{N}(v) \cup \{v\}$.



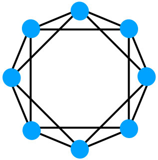
Graph



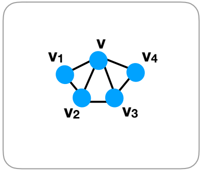
Neighborhood subgraph

Neighborhood Subgraph

- For each vertex v, the **neighborhood subgraph** S_v is the subgraph induced by $\mathcal{N}(v) \cup \{v\}$.

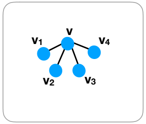


Graph



Neighborhood subgraph

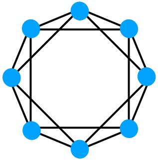
$\Longrightarrow$



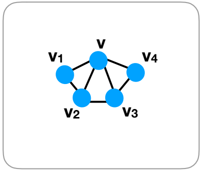
Neighborhood subtree (GIN)

Neighborhood Subgraph

- For each vertex v, the **neighborhood subgraph** S_v is the subgraph induced by $\mathcal{N}(v) \cup \{v\}$.

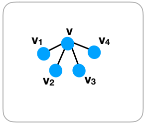


Graph

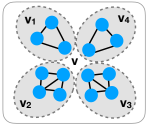


Neighborhood subgraph

$\Longrightarrow$



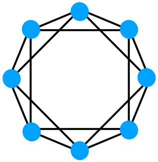
Neighborhood subtree (GIN)



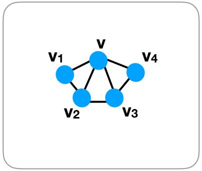
Overlap subgraphs (our work)

Neighborhood Subgraph

- For each vertex v, the **neighborhood subgraph** S_v is the subgraph induced by $\mathcal{N}(v) \cup \{v\}$.

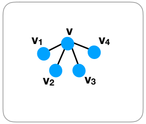


Graph

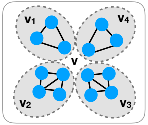


Neighborhood subgraph

$\Longrightarrow$



Neighborhood subtree (GIN)



Overlap subgraphs (our work)

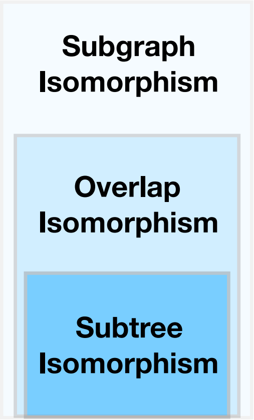
- For two adjacent vertices v and u, the **overlap subgraph** is $S_{vu} = S_v \cap S_u$.

A New Hierarchy of Local Isomorphism

Neighborhood Subgraph

Overlap Subgraphs

Neighborhood Subtree

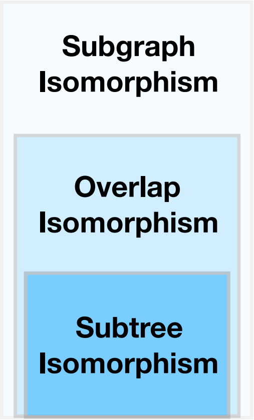


A New Hierarchy of Local Isomorphism

Neighborhood Subgraph

Overlap Subgraphs

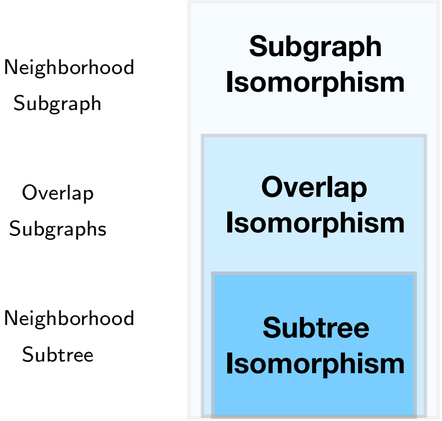
Neighborhood Subtree



Theorem

If $S_i \simeq_{subgraph} S_j$, then $S_i \simeq_{overlap} S_j$, but not vice versa.

A New Hierarchy of Local Isomorphism



Theorem

If $S_i \simeq_{subgraph} S_j$, then $S_i \simeq_{overlap} S_j$, but not vice versa.

Theorem

If $S_i \simeq_{overlap} S_j$, then $S_i \simeq_{subtree} S_j$, but not vice versa.

A New Hierarchy of Local Isomorphism

Theorem

If $S_i \simeq_{subgraph} S_j$, then $S_i \simeq_{overlap} S_j$, but not vice versa.

Neighborhood
Subgraph

Overlap
Subgraphs

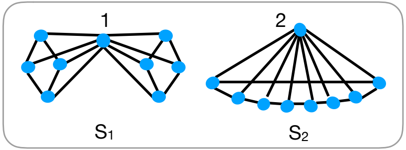
Neighborhood
Subtree

A New Hierarchy of Local Isomorphism

Theorem

If $S_i \simeq_{subgraph} S_j$, then $S_i \simeq_{overlap} S_j$, but not vice versa.

Neighborhood Subgraph



Overlap Subgraphs

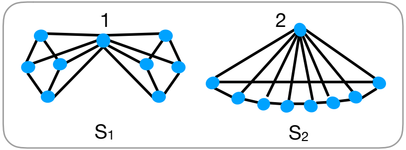
Neighborhood Subtree

A New Hierarchy of Local Isomorphism

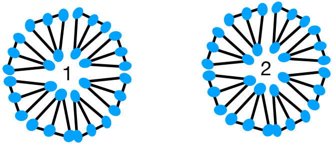
Theorem

If $S_i \simeq_{subgraph} S_j$, then $S_i \simeq_{overlap} S_j$, but not vice versa.

Neighborhood Subgraph



Overlap Subgraphs



Neighborhood Subtree

A New Hierarchy of Local Isomorphism

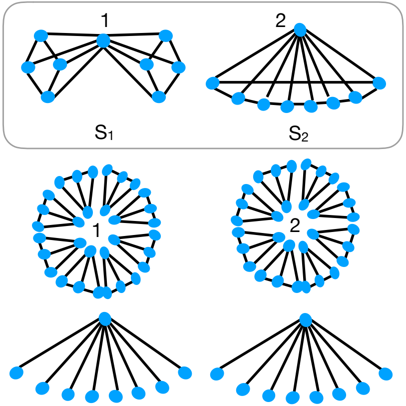
Theorem

If $S_i \simeq_{subgraph} S_j$, then $S_i \simeq_{overlap} S_j$, but not vice versa.

Neighborhood Subgraph

Overlap Subgraphs

Neighborhood Subtree



O2. A New Hierarchy of Local Isomorphism

Theorem

If $S_i \simeq_{overlap} S_j$, then $S_i \simeq_{subtree} S_j$, but not vice versa.

Neighborhood
Subgraph

Overlap
Subgraphs

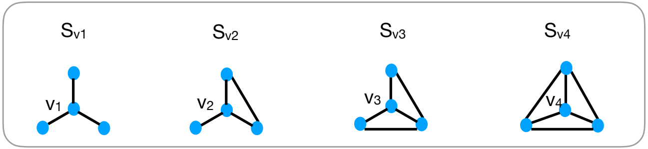
Neighborhood
Subtree

O2. A New Hierarchy of Local Isomorphism

Theorem

If $S_i \simeq_{overlap} S_j$, then $S_i \simeq_{subtree} S_j$, but not vice versa.

Neighborhood
Subgraph



Overlap
Subgraphs

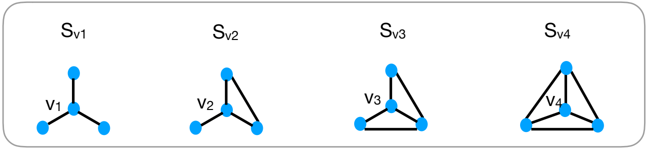
Neighborhood
Subtree

O2. A New Hierarchy of Local Isomorphism

Theorem

If $S_i \simeq_{overlap} S_j$, then $S_i \simeq_{subtree} S_j$, but not vice versa.

Neighborhood Subgraph



Overlap Subgraphs

Neighborhood Subtree

O2. A New Hierarchy of Local Isomorphism

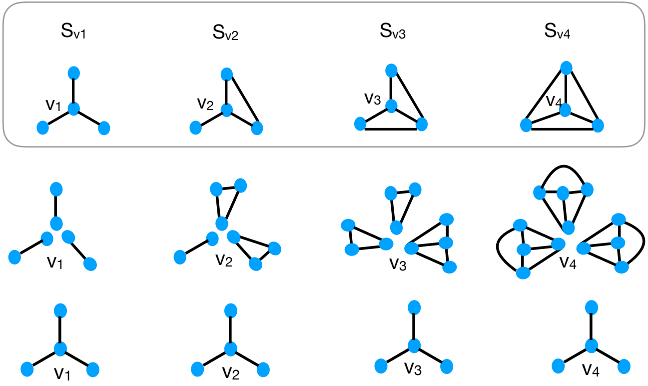
Theorem

If $S_i \simeq_{overlap} S_j$, then $S_i \simeq_{subtree} S_j$, but not vice versa.

Neighborhood Subgraph

Overlap Subgraphs

Neighborhood Subtree



A New Hierarchy of Local Isomorphism

- How does this relate to the existing GNNs?

A New Hierarchy of Local Isomorphism

- How does this relate to the existing GNNs?

Theorem

A GNN is as powerful as 1-WL if two conditions are met:

- *A sufficient number of layers;*
- *Each layer can map any two neighorhood subgraphs S_i and S_j into two different embeddings iff $S_i \not\simeq_{subtree} S_j$.*

A New Hierarchy of Local Isomorphism

- How does this help design more powerful GNNs?

A New Hierarchy of Local Isomorphism

- How does this help design more powerful GNNs?

  $\hookrightarrow$ **Two design choices:**

A New Hierarchy of Local Isomorphism

- How does this help design more powerful GNNs?

  $\hookrightarrow$ **Two design choices:**

  1. Structural coefficients

     $\{\{A_{vu} | u \in \mathcal{N}(v)\}\}$ for all $v \in V$

  2. Model injectivity

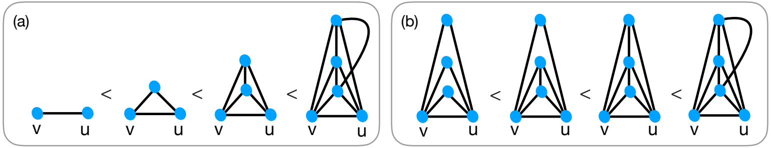
     $f(h_v) = f(h_u) \to h_v = h_u$

GraphSNN - A GNN Model Beyond 1-WL

- Let $\mathcal{S}^* = \{S_{vu} | (v, u) \in E\}$ be the set of overlap subgraphs in G. We define a function $\omega : \mathcal{S} \times \mathcal{S}^* \rightarrow \mathbb{R}$ for **structural coefficients**.

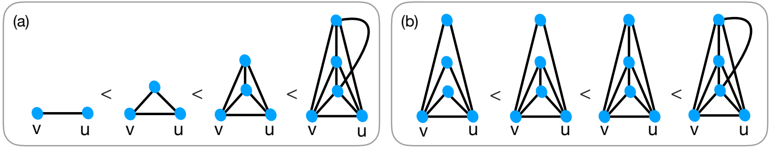
GraphSNN - A GNN Model Beyond 1-WL

- Let $\mathcal{S}^* = \{S_{vu} | (v, u) \in E\}$ be the set of overlap subgraphs in G. We define a function $\omega : \mathcal{S} \times \mathcal{S}^* \to \mathbb{R}$ for **structural coefficients**.
- What are the desirable properties of such a function ω?
  - (a) Local closeness
  - (b) Local denseness
  - (c) Isomorphic invariant



GraphSNN - A GNN Model Beyond 1-WL

- Let $\mathcal{S}^* = \{S_{vu} | (v, u) \in E\}$ be the set of overlap subgraphs in G. We define a function $\omega : \mathcal{S} \times \mathcal{S}^* \to \mathbb{R}$ for **structural coefficients**.
- What are the desirable properties of such a function ω?
  (a) Local closeness
  (b) Local denseness
  (c) Isomorphic invariant



- **An instance:**

$$A_{vu} = \frac{|E_{vu}|}{|V_{vu}| \cdot |V_{vu} - 1|} |V_{vu}|^{\lambda}, \; \lambda > 0$$

GraphSNN - A GNN Model Beyond 1-WL

- A single layer:

$$h_v^{(t)} = \textsc{Mlp}\Big(\gamma^{(t)}\Big(\sum_{u\in\mathcal{N}(v)} \tilde{A}_{vu} + 1\Big)h_v^{(t-1)} + \sum_{u\in\mathcal{N}(v)} \Big(\tilde{A}_{vu} + 1\Big)h_u^{(t-1)}\Big)$$

GraphSNN - A GNN Model Beyond 1-WL

- A single layer:

$$h_v^{(t)} = \textsc{Mlp}\Big(\gamma^{(t)}\Big(\sum_{u\in\mathcal{N}(v)} \tilde{A}_{vu} + 1\Big)h_v^{(t-1)} + \sum_{u\in\mathcal{N}(v)} \Big(\tilde{A}_{vu} + 1\Big)h_u^{(t-1)}\Big)$$

- Multiple layers (same as GIN):

$$h_G = \textsc{Concat}(\textsc{Readout}(\{\{h_v^{(t)}|v \in V\}\})|t = 1, \ldots, k)$$

GraphSNN - A GNN Model Beyond 1-WL

- A single layer:

$$h_v^{(t)} = \textsc{Mlp}\Big(\gamma^{(t)}\Big(\sum_{u\in\mathcal{N}(v)} \tilde{A}_{vu} + 1\Big)h_v^{(t-1)} + \sum_{u\in\mathcal{N}(v)} \Big(\tilde{A}_{vu} + 1\Big)h_u^{(t-1)}\Big)$$

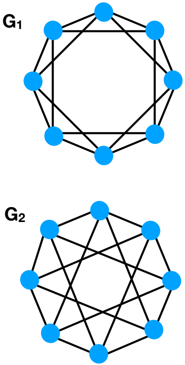
- Multiple layers (same as GIN):

$$h_G = \textsc{Concat}(\textsc{Readout}(\{\{h_v^{(t)}|v \in V\}\})|t = 1, \dots, k)$$

Theorem

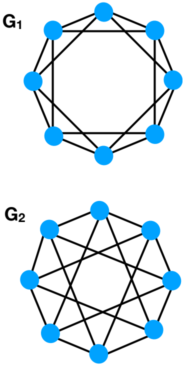
GraphSNN is strictly more expressive than 1-WL.

GraphSNN - A GNN Model Beyond 1-WL



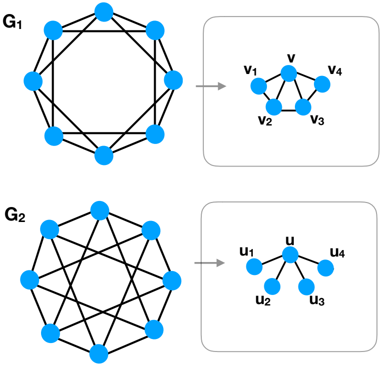
Graphs

GraphSNN - A GNN Model Beyond 1-WL



Graphs

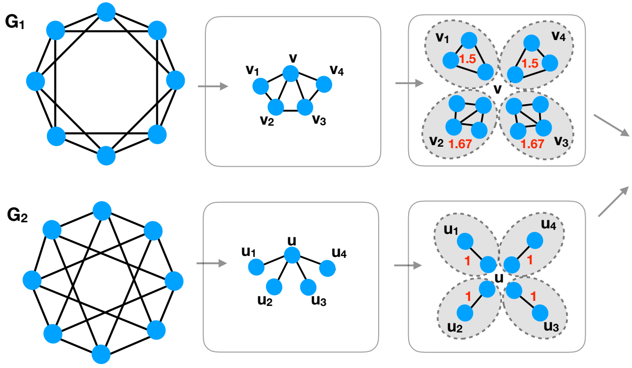
GraphSNN - A GNN Model Beyond 1-WL



Graphs

Neighborhood Subgraphs

GraphSNN - A GNN Model Beyond 1-WL

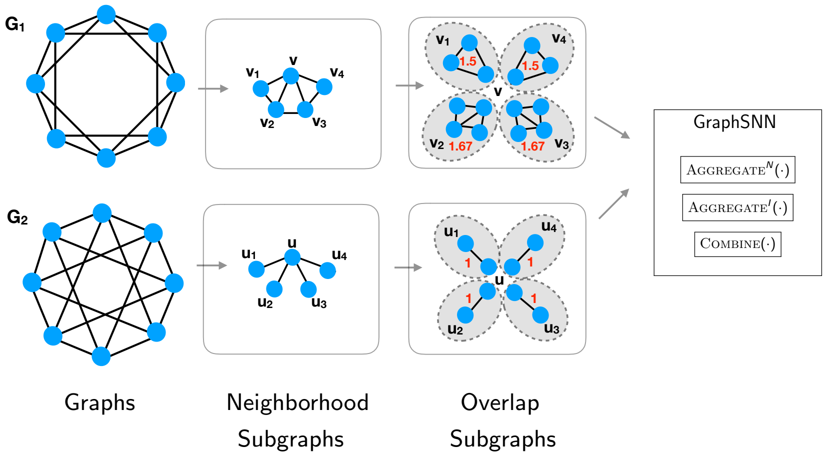


Graphs

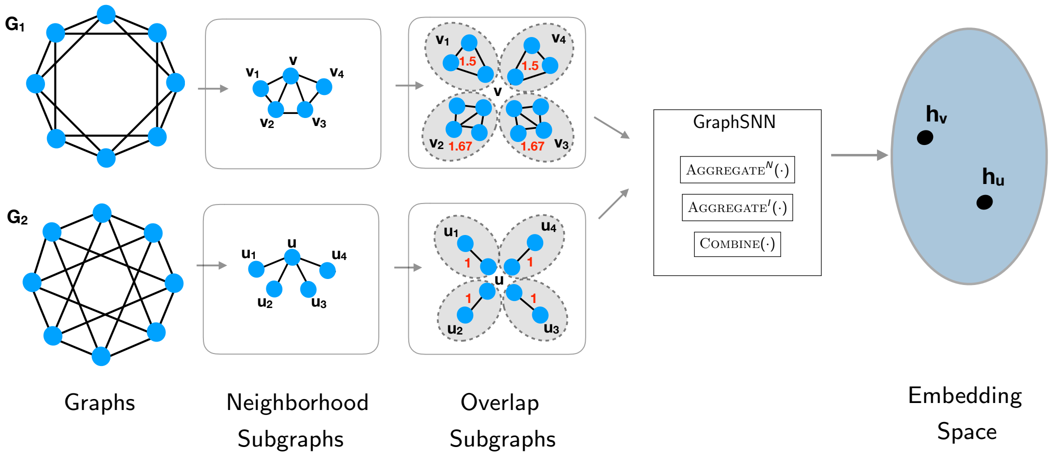
Neighborhood Subgraphs

Overlap Subgraphs

GraphSNN - A GNN Model Beyond 1-WL



GraphSNN - A GNN Model Beyond 1-WL



A Generalised Message Passing GNN

- A layer of a **Message-Passing GNN** is defined as:

   1. Aggregate "messages" from neighbors $\mathcal{N}(v)$

      $\cancel{h^{(t)} = \textsc{Aggregate}\left(\{\{h_u^{(t)} | u \in \mathcal{N}(v)\}\}\right)}$

      $\hookrightarrow m_a^{(t)} = \textsc{Aggregate}^N\left(\{\{(\tilde{A}_{vu}, h_u^{(t)}) | u \in \mathcal{N}(v)\}\}\right)$

A Generalised Message Passing GNN

- A layer of a **Message-Passing GNN** is defined as:
  1. Aggregate "messages" from neighbors $\mathcal{N}(v)$

$$\cancel{h^{(t)} = \textsc{Aggregate}\left(\{\!\{h_u^{(t)} | u \in \mathcal{N}(v)\}\!\}\right)}$$

$$\hookrightarrow m_a^{(t)} = \textsc{Aggregate}^N\left(\{\!\{(\tilde{A}_{vu}, h_u^{(t)}) | u \in \mathcal{N}(v)\}\!\}\right)$$

$$\hookrightarrow m_v^{(t)} = \textsc{Aggregate}^I\left(\{\!\{\tilde{A}_{vu} | u \in \mathcal{N}(v)\}\!\}\right) h_v^{(t)}$$

A Generalised Message Passing GNN

- A layer of a **Message-Passing GNN** is defined as:

1. Aggregate "messages" from neighbors $\mathcal{N}(v)$

$$\cancel{h^{(t)} = \textsc{Aggregate}\left(\{\!\{h_u^{(t)} | u \in \mathcal{N}(v)\}\!\}\right)}$$

$$\hookrightarrow m_a^{(t)} = \textsc{Aggregate}^N\left(\{\!\{(\tilde{A}_{vu}, h_u^{(t)}) | u \in \mathcal{N}(v)\}\!\}\right)$$

$$\hookrightarrow m_v^{(t)} = \textsc{Aggregate}^I\left(\{\!\{\tilde{A}_{vu} | u \in \mathcal{N}(v)\}\!\}\right) h_v^{(t)}$$

2. Combine with its own "message" $h_v^{(t)}$

$$\cancel{h_v^{(t+1)} = \textsc{Combine}\left(h_v^{(t)}, h^{(t)}\right)}$$

$$\hookrightarrow h_v^{(t+1)} = \textsc{Combine}\left(m_v^{(t)}, m_a^{(t)}\right)$$

Numerical Experiments

- Classification on Open Graph Benchmark (OGB) datasets, including four molecular graph datasets and one protein-protein association network.

Method	ogbg-molhiv	ogbg-moltox21	ogbg-moltoxcast	ogbg-ppa	ogbg-molpcba
GIN	75.58±1.40	74.91±0.51	63.41±0.74	68.92±1.00	22.66±0.28
GIN+VN	75.20±1.30	76.21±0.82	66.18±0.68	70.37±1.07	27.03±0.23
GSN	77.99±1.00	-	-	-	-
PNA	79.05±1.30	-	-	-	28.38±0.35
ID-GNN	78.30±2.00	-	-	-	-
Deep LRP	77.19±1.40	-	-	-	-
GraphSNN	78.51±1.70	75.45±1.10	65.40±0.71	70.66±1.65	24.96±1.50
GraphSNN+VN	**79.72±1.83**	**76.78±1.27**	**67.68±0.92**	**72.02±1.48**	**28.50±1.68**

Table: Classification accuracy on large graph classification.

Numerical Experiments

- Classification w.r.t GraphSNN$_M$ models by replacing GCN, GAT, GIN, and GraphSAGE aggregation schemes by our aggregation scheme.

Method	Cora	Citeseer	Pubmed	NELL	ogbn-arxiv
GCN	81.5 ± 0.4	70.3 ± 0.5	79.0 ± 0.5	66.0 ± 1.7	71.74 ± 0.29
GraphSNN$_{GCN}$	**83.1 ± 1.8**	**72.3 ± 1.5**	**79.8 ± 1.2**	**68.3 ± 1.6**	**72.20 ± 0.90**
GAT	83.0 ± 0.6	72.6 ± 0.6	78.5 ± 0.3	-	-
GraphSNN$_{GAT}$	**83.8 ± 1.2**	**73.5 ± 1.6**	**79.6 ± 1.4**	-	-
GIN	77.6 ± 1.1	66.1 ± 1.5	77.0 ± 1.2	61.5 ± 2.3	-
GraphSNN$_{GIN}$	**79.2 ± 1.7**	**68.3 ± 1.5**	**78.8 ± 1.3**	**63.8 ± 2.7**	-
GraphSAGE	79.2 ± 3.7	71.6 ± 1.9	77.4 ± 2.2	63.7 ± 5.2	71.49 ± 0.27
GraphSNN$_{GraphSAGE}$	**80.5 ± 2.5**	**72.7 ± 3.2**	**79.0 ± 3.5**	**66.3 ± 5.6**	**71.80 ± 0.70**

Table: Classification accuracy on semi-supervised node classification.

Numerical Experiments

- Oversmoothing analysis of GCN and GraphSNN$_{GCN}$ on the datasets Cora, Citeseer and Pubmed.

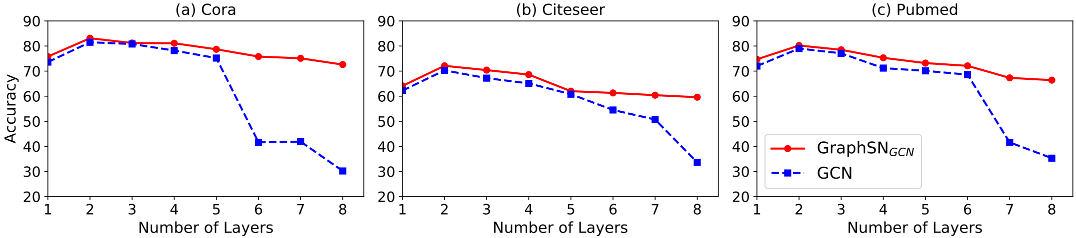


Figure: Oversmoothing analysis w.r.t. the model depth for node classification.

Thank You