High-order Graph Neural Networks

Prof. O-Joun Lee

Dept. of Artificial Intelligence,
The Catholic University of Korea
ojlee@catholic.ac.kr

네트워크 과학 연구실
NETWORK SCIENCE LAB

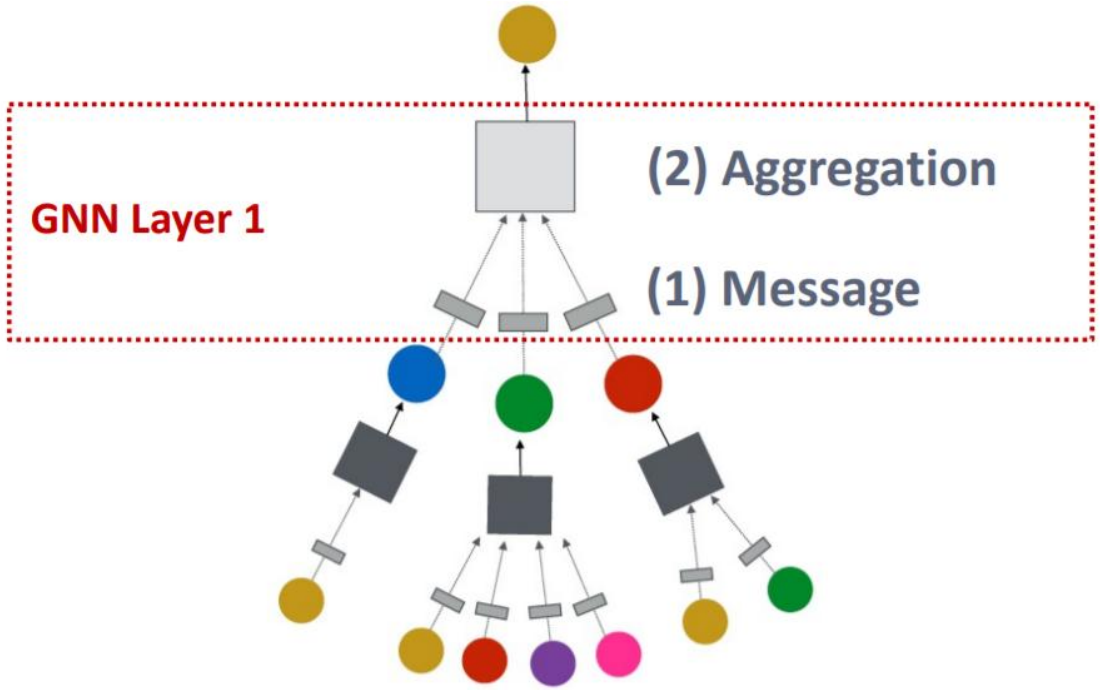
가톨릭대학교
THE CATHOLIC UNIVERSITY OF KOREA

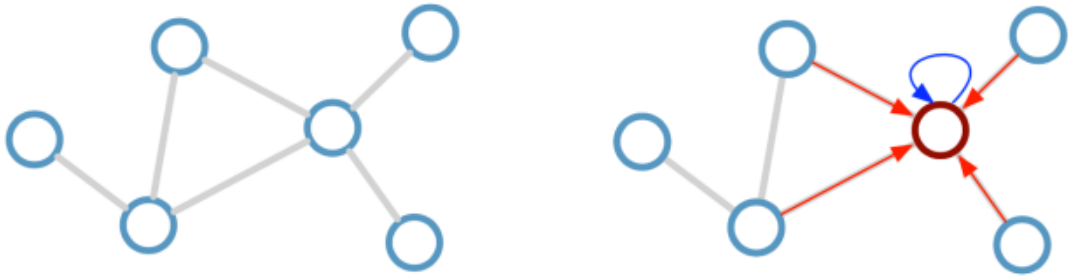
Contents

- Message Passing Neural Networks (MPNNs) issues
- MPNNs and Color refinement
- High-order MPNNs:
  - Drop Nodes/Edges
  - K-hop sampling
  - Extend the WL test to a subgraph-based variants
  - Equivariant Subgraph Aggregation Networks

Recap: Graph Convolutional Networks (GCNs)

- GNN Layer = **Message** + **Aggregation**
  - Message COMPUTATION
    - how to make each neighborhood node as embedding?
  - Message AGGERGATION
    - how to combine those embeddings?





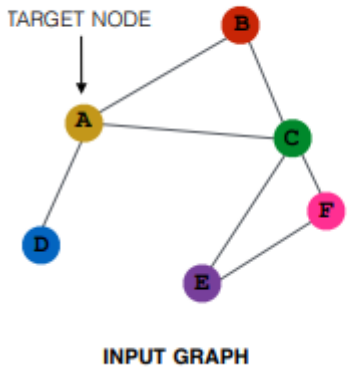
Update rule: $$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)}\mathbf{W}_0^{(l)} + \sum_{j\in\mathcal{N}_i}\frac{1}{c_{ij}}\mathbf{h}_j^{(l)}\mathbf{W}_1^{(l)}\right)$$

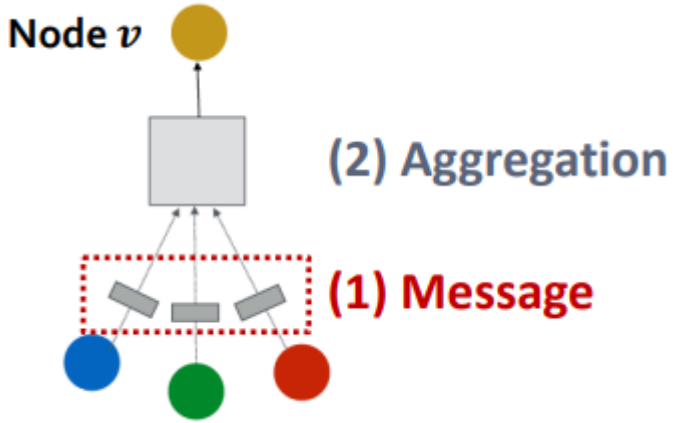
Kipf & Welling (ICLR 2017), related previous works by Duvenaud et al. (NIPS 2015) and Li et al. (ICLR 2016)

GNN Layers: Message Computation

- Intuition: Each node will create a message, which will be sent to other nodes later
- Example: A Linear layer $\mathbf{m}_u^{(l)} = \mathbf{W}^{(l)}\mathbf{h}_u^{(l-1)}$
  - Multiply node features with weight matrix $\mathbf{W}^{(l)}$

Message function: $\mathbf{m}_u^{(l)} = \mathrm{MSG}^{(l)}\left(\mathbf{h}_u^{(l-1)}\right)$





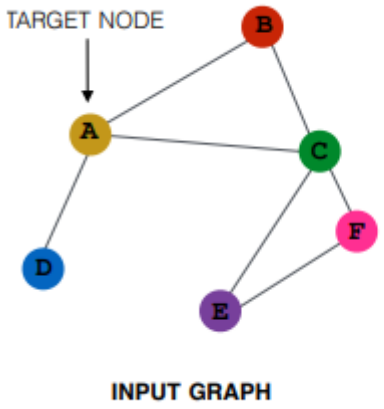
GNN Layers: Message Aggregation

- Intuition: Each node will aggregate the messages from node v's neighbors

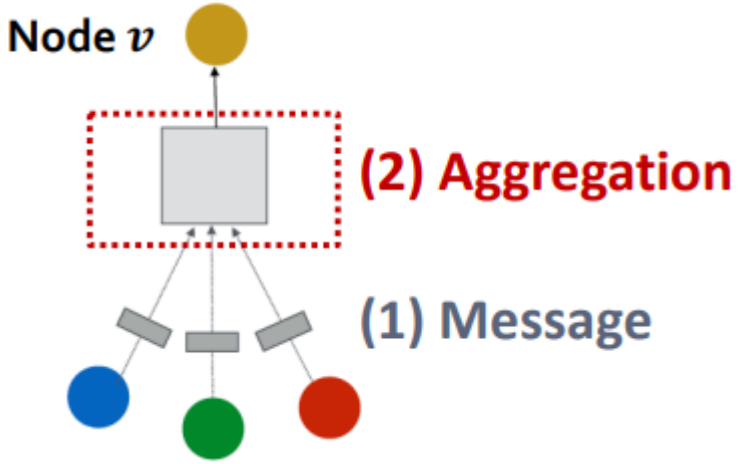
$$\mathbf{h}_v^{(l)} = \mathrm{AGG}^{(l)}\left(\left\{\mathbf{m}_u^{(l)}, u \in N(v)\right\}\right)$$

- Example: Sum(·), Mean(·) or Max(·) aggregator

$$\mathbf{h}_v^{(l)} = \mathrm{Sum}(\{\mathbf{m}_u^{(l)}, u \in N(v)\})$$

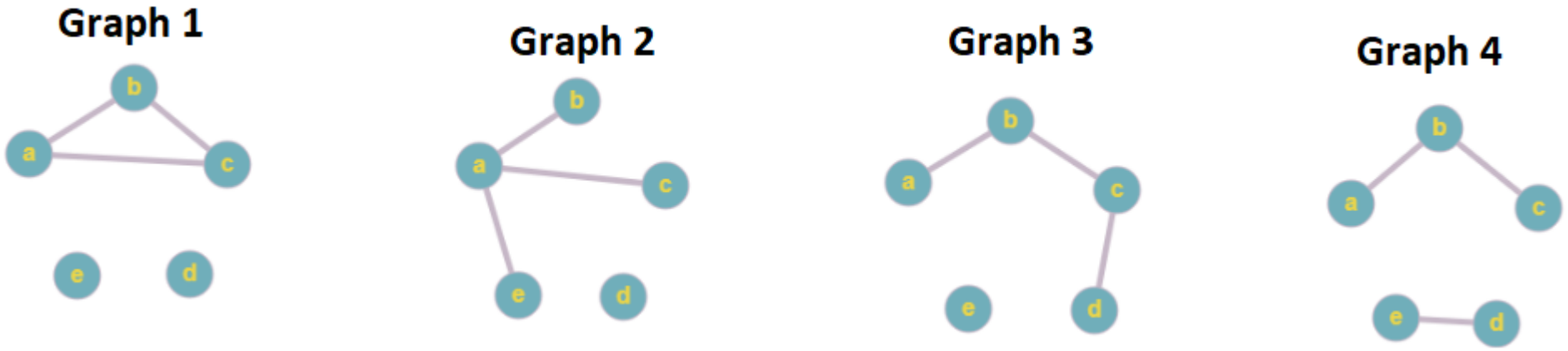


INPUT GRAPH



The fundamental issue of GNNs

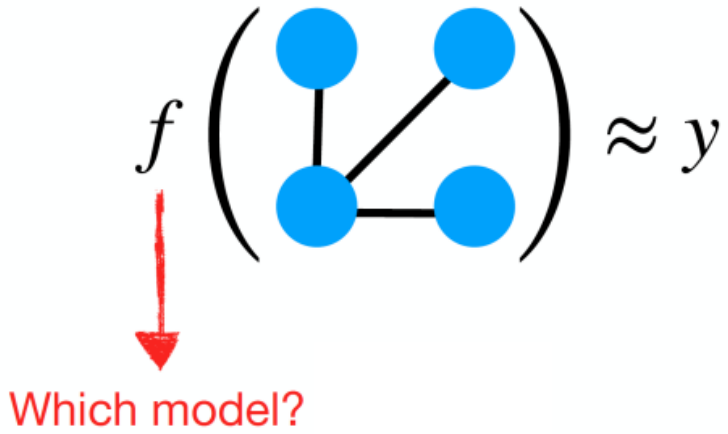
- **Do similar graphs (similar structures) really have the same label?**
  - The Graph Classification problem and the Graph Isomorphism problem are not the same.
  - Graph Isomorphism can be judged as isomorphic if the positions on the vector are close, but Graph Classification adds an element called "Label" to the graph, so that it is classified by the same Label, not by position.



Message Passing Neural Networks: Problem definition

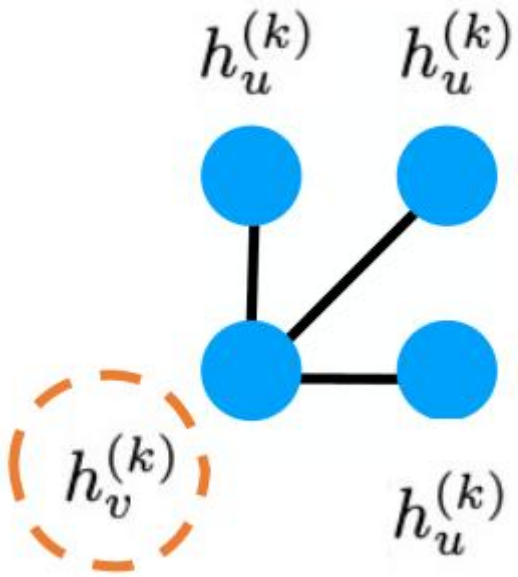
- Training data: $(G_1, y_1), \ldots, (G_m, y_m)$

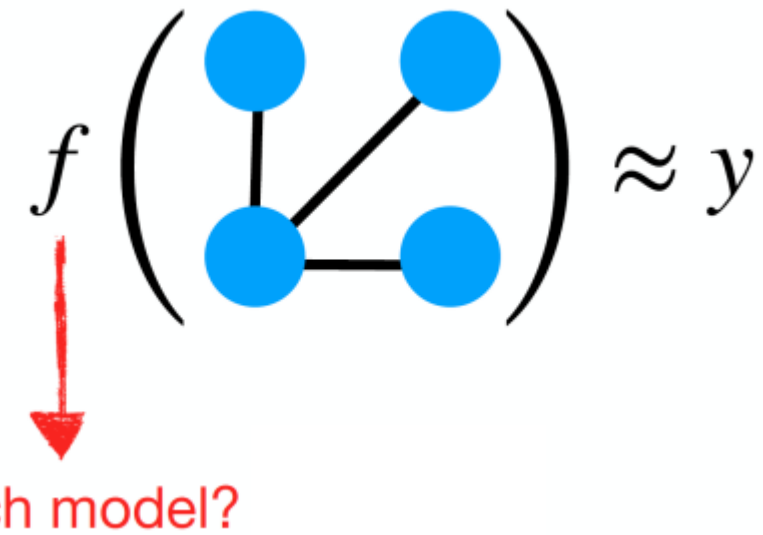
- Each graph G_i consists of:
  - Node feature $x_i \in R^d$
  - Label $y_i \in \{-1, 1\}$
- **Goal:** Find a model that maps graphs to output labels



- Parametric neighborhood aggregation layers

$$\mathrm{msg}_v^{(k)} = \textsc{Aggregate}(\{h_u^{(k)} : u \text{ neighbor of } v\}),$$
$$h_v^{(k+1)} = \textsc{Combine}\left(h_v^{(k)}, \mathrm{msg}_v^{(k)}\right).$$





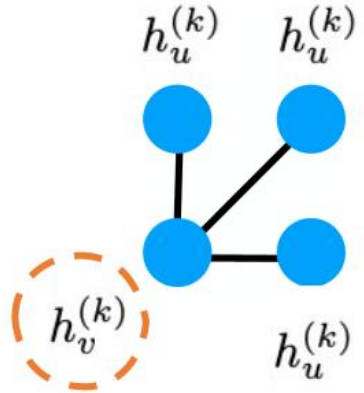
Message Passing Neural Networks

- Parametric neighborhood aggregation layers

$$\text{msg}_v^{(k)} = \text{AGGREGATE}(\{h_u^{(k)} : u \text{ neighbor of } v\}),$$
$$h_v^{(k+1)} = \text{COMBINE}\left(h_v^{(k)}, \text{msg}_v^{(k)}\right).$$

- We will aggregate all node features to generate a graph-level representation

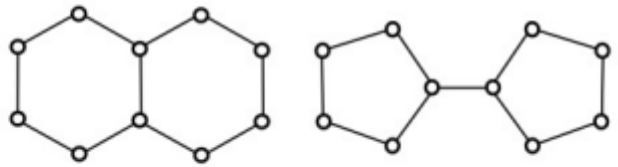
$$h_{graph} = \text{Aggregate}(\{h_u^{(K)} : k = 1, \ldots, n\})$$



- **Question:** What is the expressive power of MPNNs?
  - Given two non-isomorphic graphs G_1, G_2
  - Can we find an MPNN $f(\cdot)$ such that:

$$f(G_1) \neq f(G_2)$$

No!



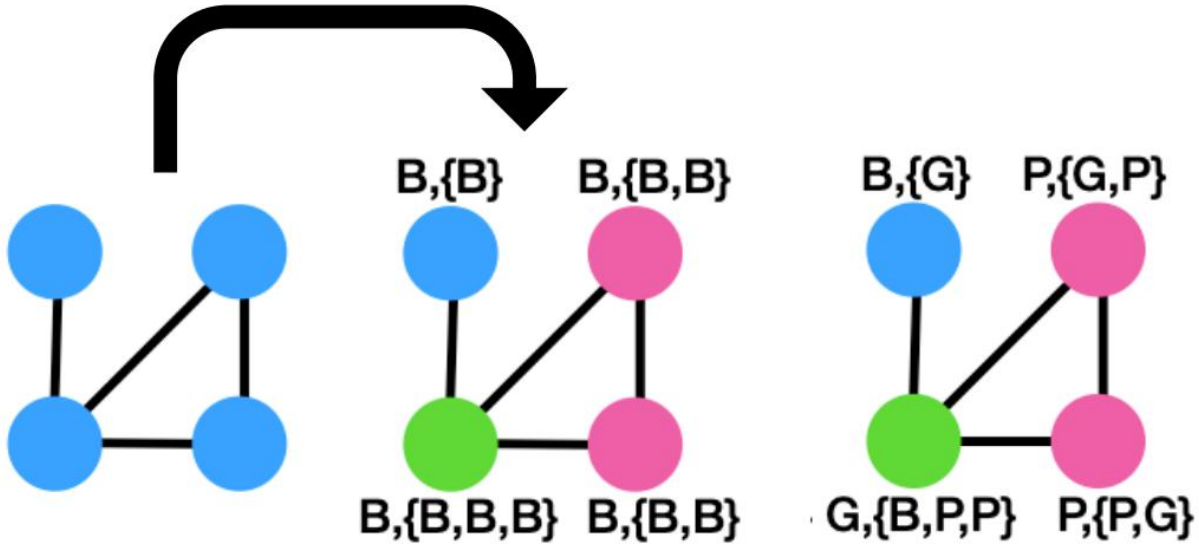
Recap: Color refinement and MPNNs

- MPNNs are closely related to the color refinement algorithm
- The color refinement (CR) algorithm, also known as the Weisfeiler-Lehman (WL) test, is an efficient heuristic for graph isomorphism testing.
  - CR works by iteratively refining the colors (labels) assigned to the vertices of the graph based on their neighborhood structures.

Recap: Color refinement and MPNNs

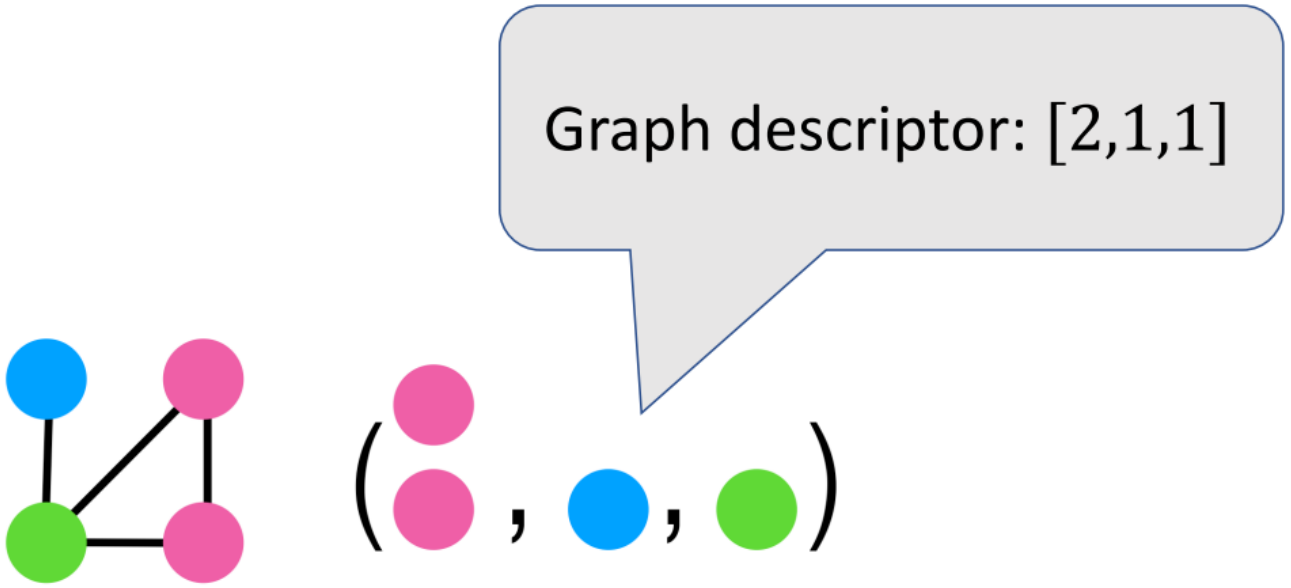
- MPNNs are closely related to the color refinement algorithm
- The color refinement (CR) algorithm, also known as the Weisfeiler-Lehman (WL) test, is an efficient heuristic for graph isomorphism testing.
  - CR works by iteratively refining the colors (labels) assigned to the vertices of the graph based on their neighborhood structures.

New color:= [old color; {colors of neighbors}]



Color refinement (CR)

- Final graph descriptor: Color histogram
- The goal is to reach a stable coloring that represents the structural distinctions between nodes in a graph.
- After refinement, a color histogram can be created as a final graph descriptor.



Recap: Color refinement and MPNNs

- [Morris et al 2019, Xu et al. 2019]: MPNNs are equivalent to CR (1-WL)

MPNNs

$$\text{msg}_v^{(k)} = \text{AGGREGATE}(\{h_u^{(k)} : u \text{ neighbor of } v\}),$$
$$h_v^{(k+1)} = \text{COMBINE}\left(h_v^{(k)}, \text{msg}_v^{(k)}\right).$$

CR

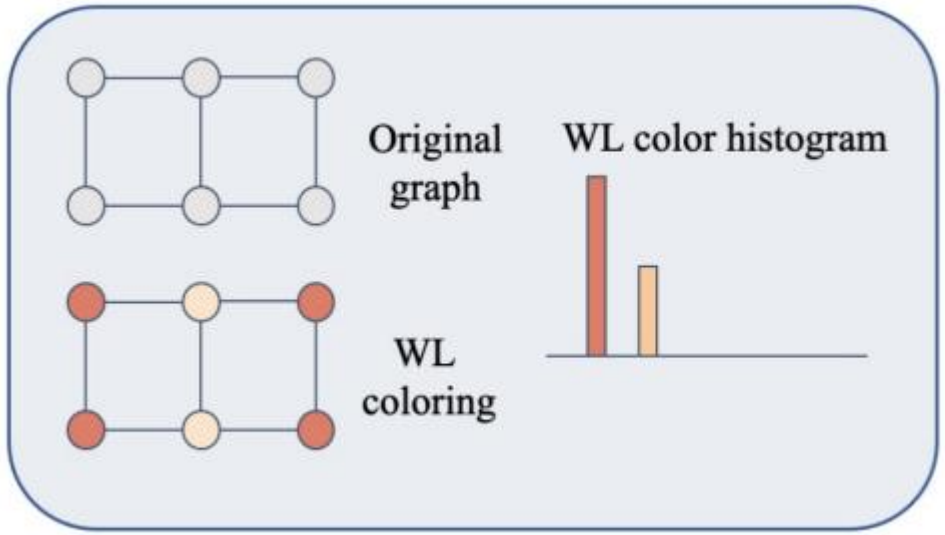
- Moving beyond 1-WL limitations: Higher-order, more powerful generalizations forming a hierarchy:

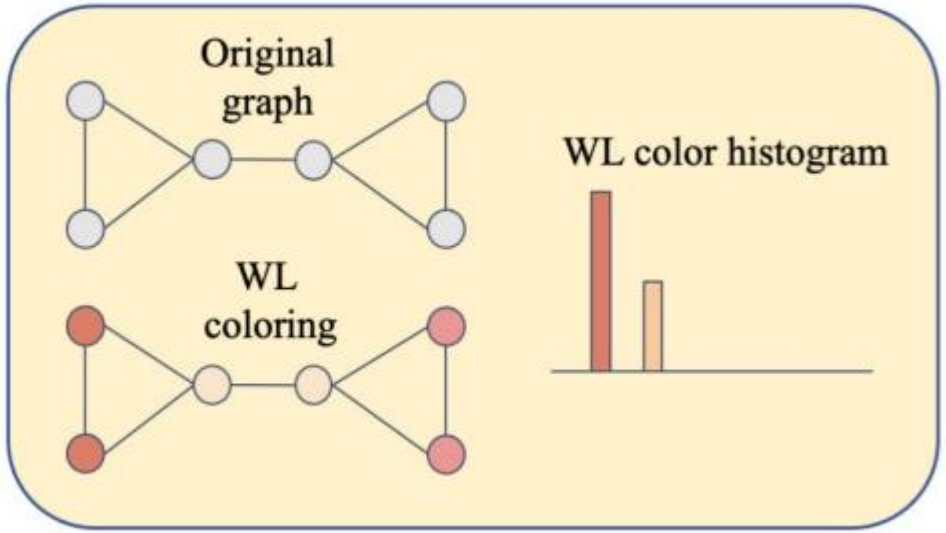
1-WL < 3-WL < 4-WL < ...

Christopher Morris et al., *Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks*

1-WL Testing and MPNN Expressivity

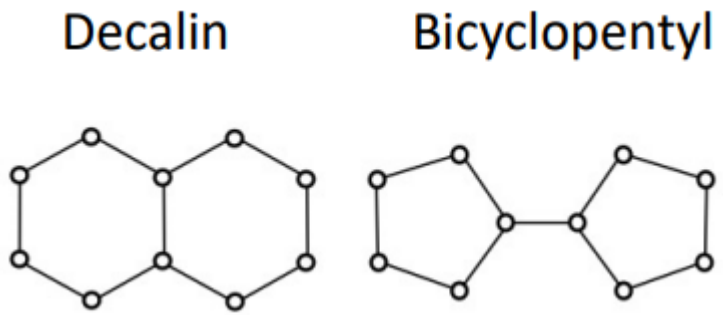
- Learning graph embeddings based on color histogram
- Message Passing Neural Networks (MPNNs) are limited expressivity due to their inherent constraints within the 1-dimensional Weisfeiler-Lehman (1-WL) graph isomorphism test





Why expressivity matters?

- This comes from the nature of graph structures:
  - Cannot assign different labels to different graphs (different graph structures)



- **Also**: we might not be able to learn the “correct” features
  - For example: MPNNs cannot detect rings

Recap: What we have learned about the MPNN expressivity?

- **GIN** model: is a type of MPNNs specifically designed to address the limited expressivity of traditional MPNNs, making it as powerful as the 1-dimensional Weisfeiler-Lehman (1-WL) test for distinguishing non-isomorphic graphs.
- Key ideas behind the GIN:
  - making the MPNN expressive enough to match the power of the 1-WL test in distinguishing non-isomorphic graphs.

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left((1+\epsilon^{(k)})\cdot h_v^{(k-1)} + \sum_{u\in\mathcal{N}(v)} h_u^{(k-1)}\right)$$

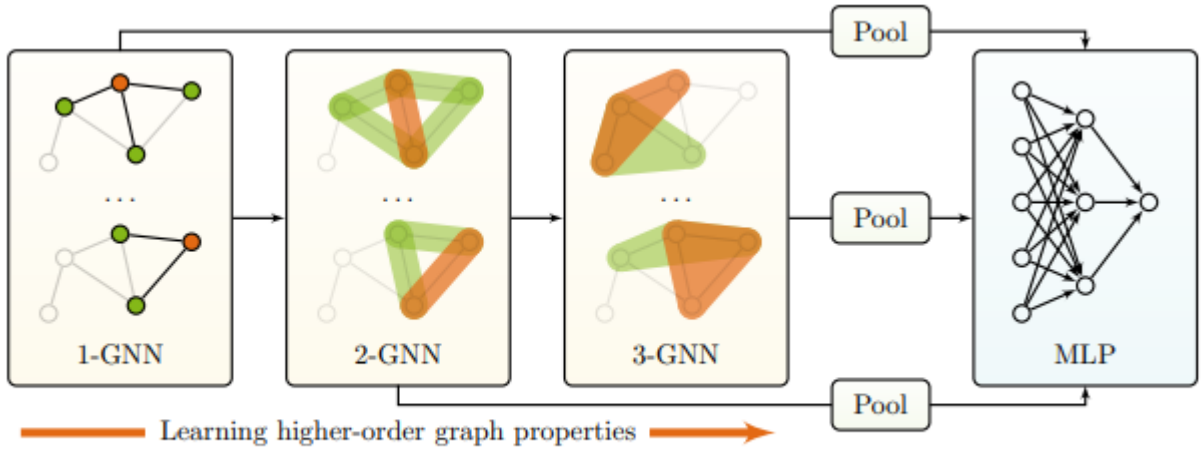
- **Problem**: Higher-Order Weisfeiler-Lehma, Extending the expressivity of GNNs by adopting the logic of higher-order WL tests (e.g., 2-WL, 3-WL) can help capture intricate structures.

1-WL < 3-WL < 4-WL < ...

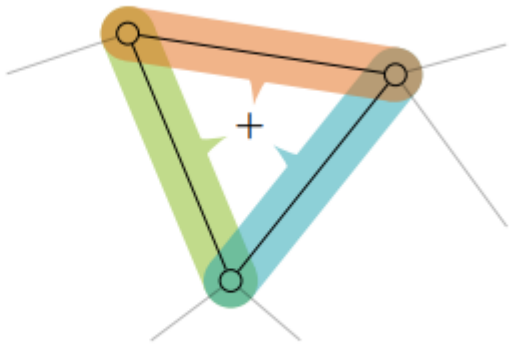
State-of-the-art (SOTA) expressive GNNs

- k-GNNs/k-IGNs/ k-hop
  - High computational complexity [Morris et al., 2019, 2020;M. et al., 2019]
- Random node features
  - Experimental results are not great [Abboud et al., 2020, Sato et al., 2021]
- Subgraph-based networks

- **k-GNNs as powerful as k-WL**
  - High computational complexity [Morris et al., 2019, 2020;M. et al., 2019]
  - Objective: aggregating the high-order information (edges, traiangle, ..)



(a) Hierarchical 1-2-3-GNN network architecture



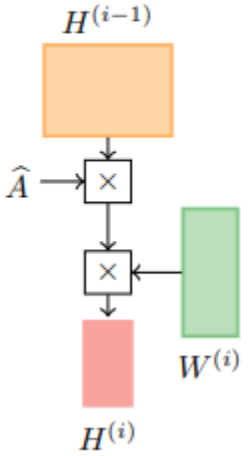
(b) Pooling from 2- to 3-GNN.

- **How about k-hop information?**
  - Objective: repeatedly mixing feature representations of neighbors at various distances.

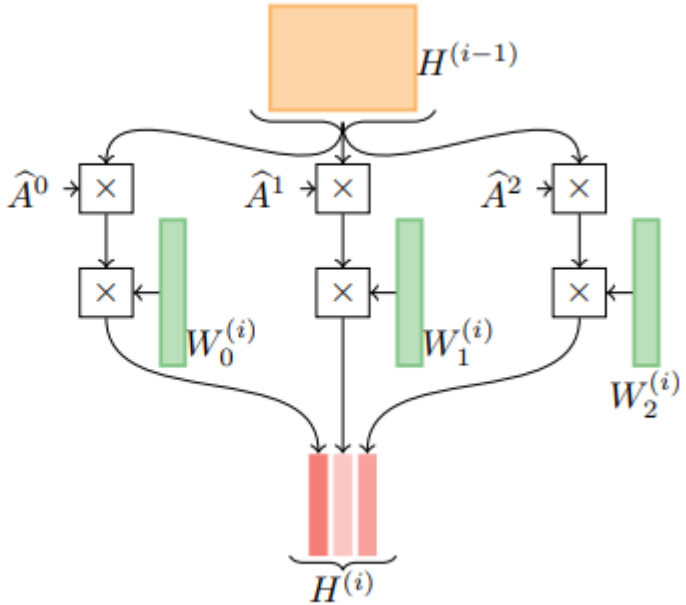
Replacing the Graph Convolution layer:

$$H^{(i+1)} = \Big\Vert_{j \in P} \sigma\left(\widehat{A}^j H^{(i)} W_j^{(i)}\right),$$

P: is the p-strep transition power
A^j: The normalized matrix at j-step



GCN layer



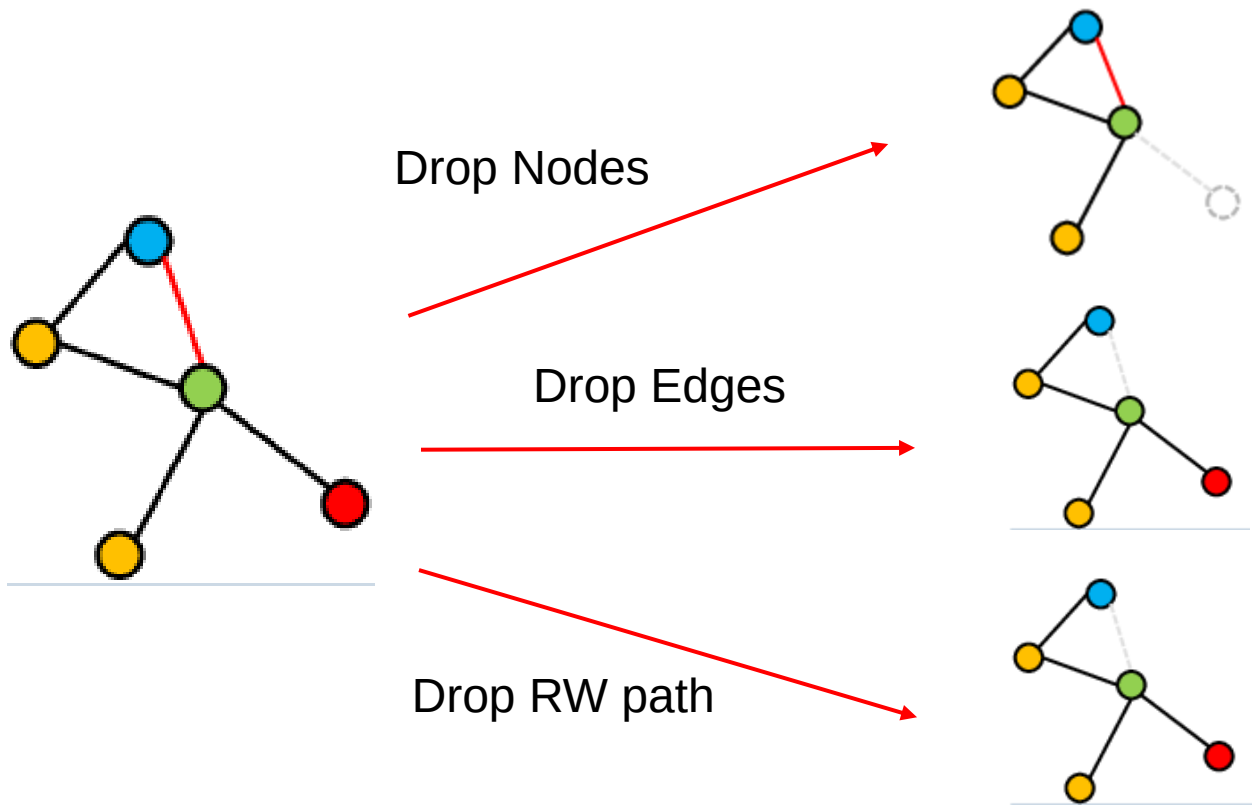
MixHop

MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing; ICML 2019

State-of-the-art in expressive GNNs

- k-GNNs/k-IGNs/ k-hop
  - High computational complexity [Morris et al., 2019, 2020;M. et al., 2019]
- Random node features
  - Experimental results are not great [Abboud et al., 2020, Sato et al., 2021]
- **Subgraph-based networks**

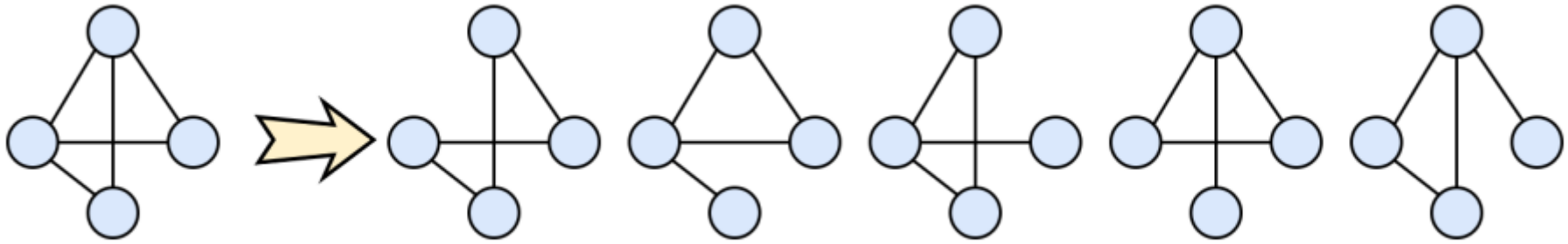
- **Simple methods:** Drop Nodes/Edges/Random walk path in Graphs
  - A simple, yet surprisingly effective, method which randomly removes nodes from the computation graph at each layer of the MPNNs.
  - By doing so, each node sees a different neighborhood of the graph, thereby implicitly creating different subgraph embeddings.



Set of subgraphs: intuition

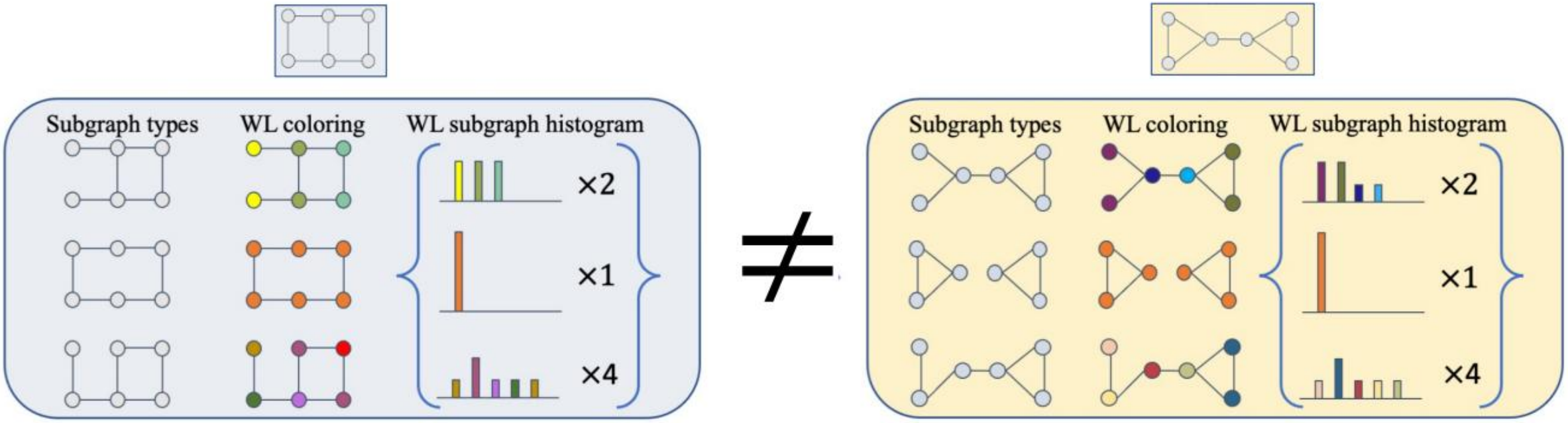
- **Main observation:**

  we can gain more expressive power by representing a graph as a set of subgraphs.



Set of subgraphs: example

- **Edge deleted subgraphs:** After deleting edges, we can obtain different WL subgraph histogram



Increasing the expressive power of GNNs using subgraphs *

- **Objective**: extend the WL test to a subgraph-based variants

- By extending *stars* (subtrees formed by root nodes with their neighbors) to *subgraphs* (the *k*-hop neighborhoods of root nodes), the power of the graph isomorphism test greatly improves, even for small values of *k.*

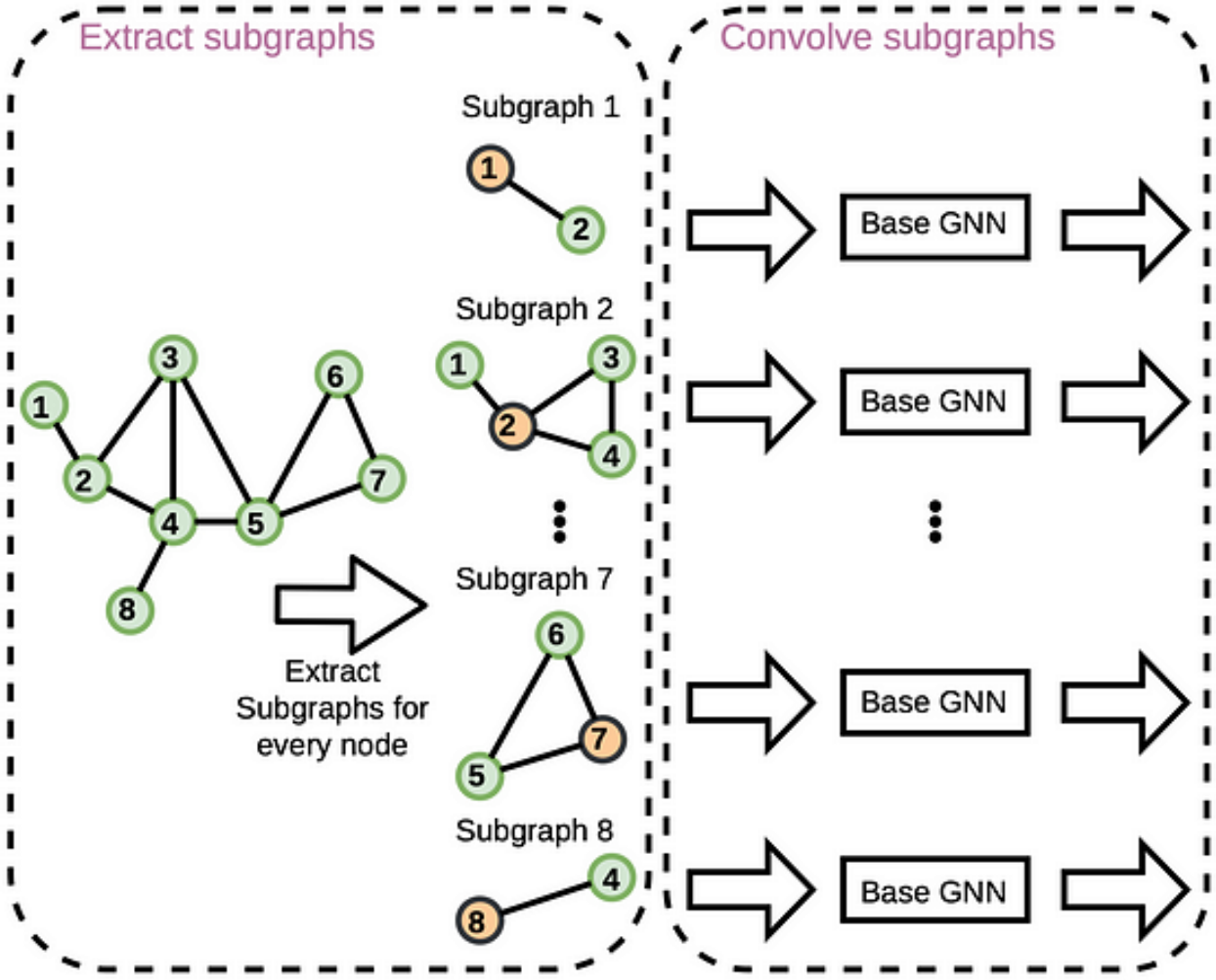
From Start (Vanilla GNNs):

$$c_v^{(t+1)} = \text{HASH}(Star^{(t)}(v))$$

To Subgraph-1-WL:

$$c_v^{(t+1)} = \text{HASH}(G^{(t)}[\mathcal{N}_k(v)]), \forall v \in \mathcal{V}$$

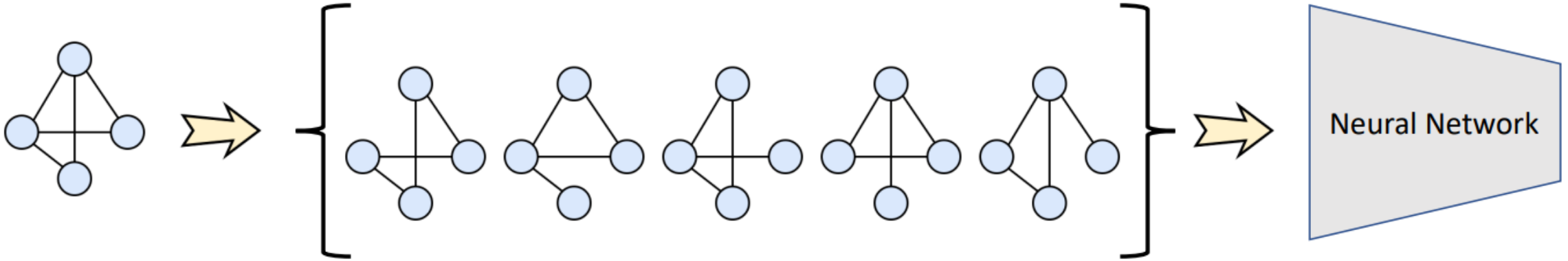
where HASH(·) is an injective function on graphs.



* L. Zhao, . From stars to subgraphs: Uplifting any GNN with local structure awareness, ICLR 2022.

Equivariant Subgraph Aggregation Networks (ESAN)

- Map a graph into a set (bag) of subgraphs
- Process the bag with a neural network



B. Bevilacqua, Equivariant Subgraph Aggregation Networks ICLR 2022

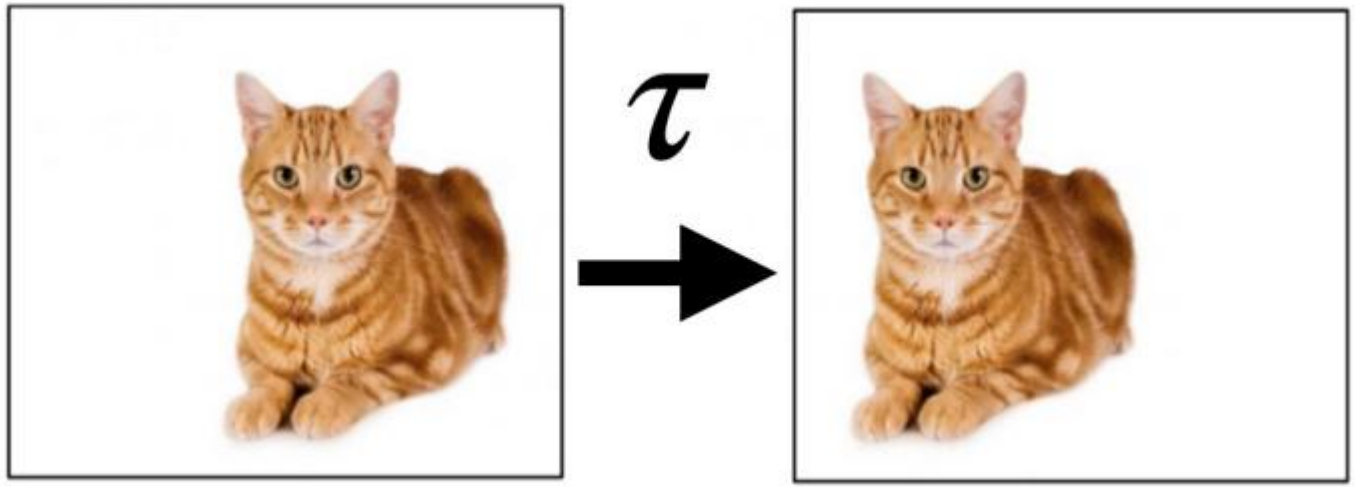
Equivariant Subgraph Aggregation Networks (ESAN)

Two main challenges:

- Architecture: How to process sets of subgraphs?
  - Design layers that respect the resulting symmetry group
- Which subgraph selection policies are useful?
  - ESAN model proposes four simple policies that work well

Equivariance as a design principle (ESAN)

- Let G be a group of transformations on our inputs
- A symmetry group G models transformations that do not change the underlying object, or that we do not care about
- Example: tranlations of images

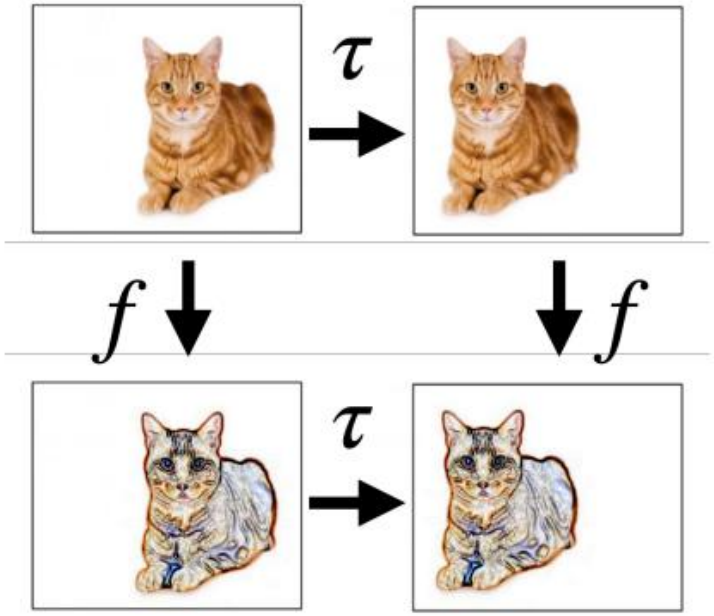


Equivariance as a design principle (ESAN)

- Equivariance as a design principle
  - A function f is called equivariant if:

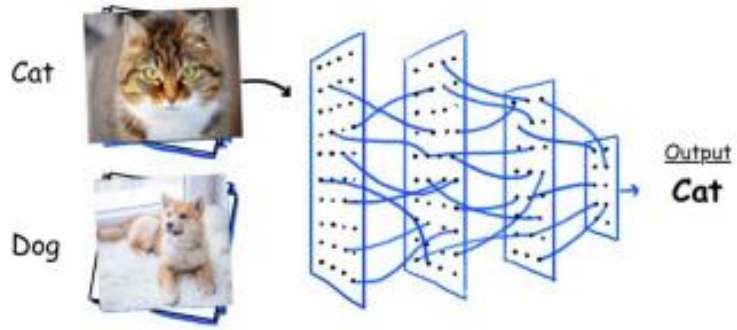
$$f(\tau x) = \tau f(x), \qquad \tau \in G$$

  - Example: Convolutions / image segmentation are translation equivariant

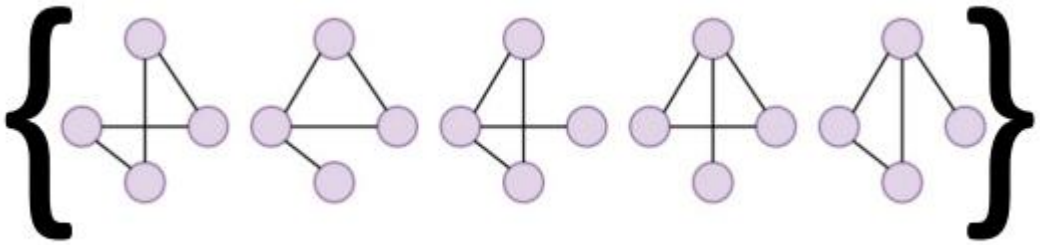


Equivariance as a design principle (ESAN)

- Prototypical Example: Convolutional Neural Networks
  - Input: images
  - Symmetry group: 2D translations
  - Basic layers: convolutions
  - Resulting architecture: CNN



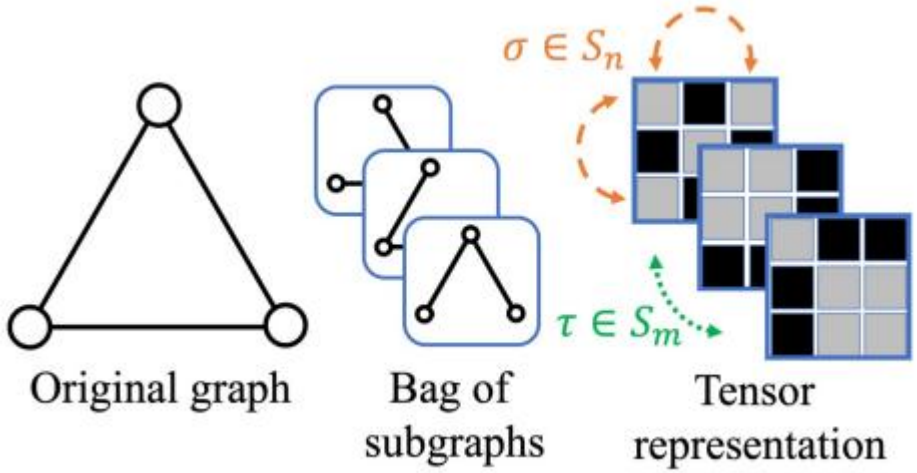
- **In graphs:**
  - Input: sets of subgraphs
  - Symmetry group: ?
  - Basic layers: ?
  - Resulting architecture : ?



Equivariance as a design principle (ESAN)

Symmetry for sets of subgraphs:

- We have two types of symmetries:
  - Internal graph symmetry
  - External set symmetry
- We know how to handle each one.
  - **Idea: The combination between subgraphs can solve the Equivariance problem**

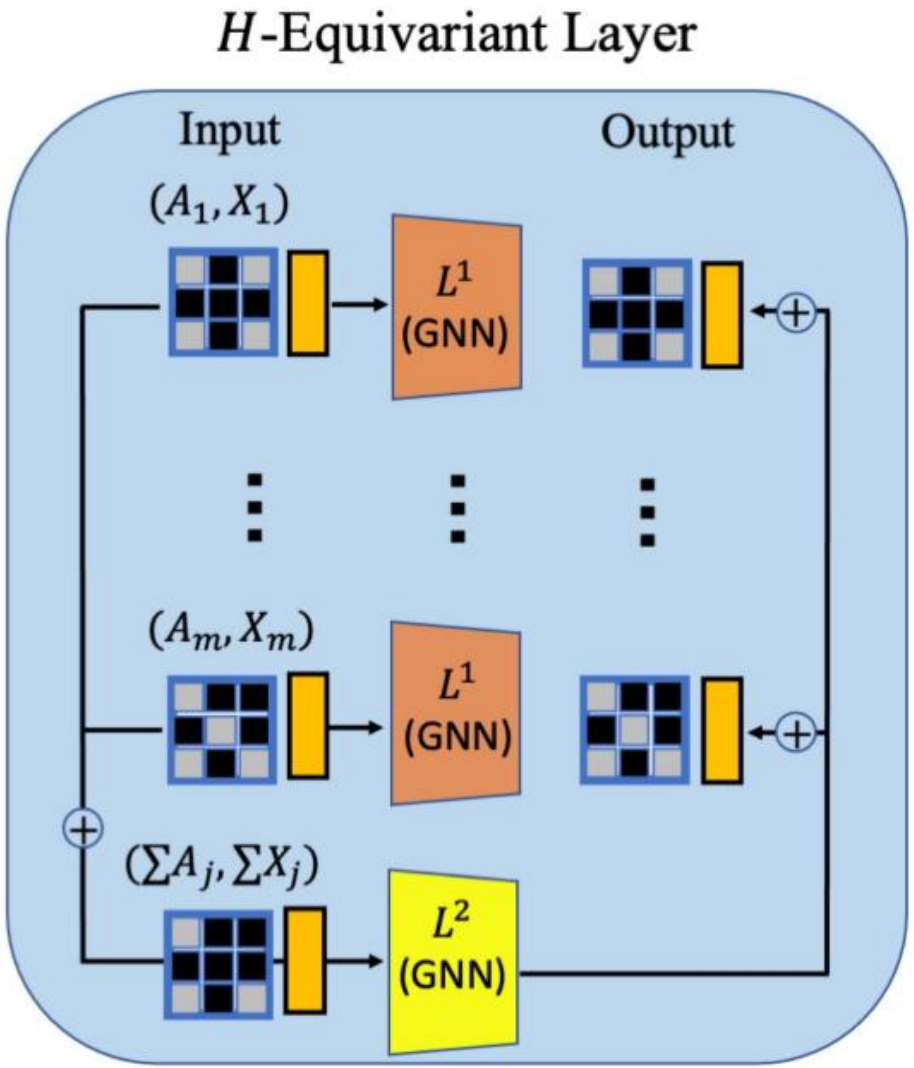


Equivariant layer (ESAN)

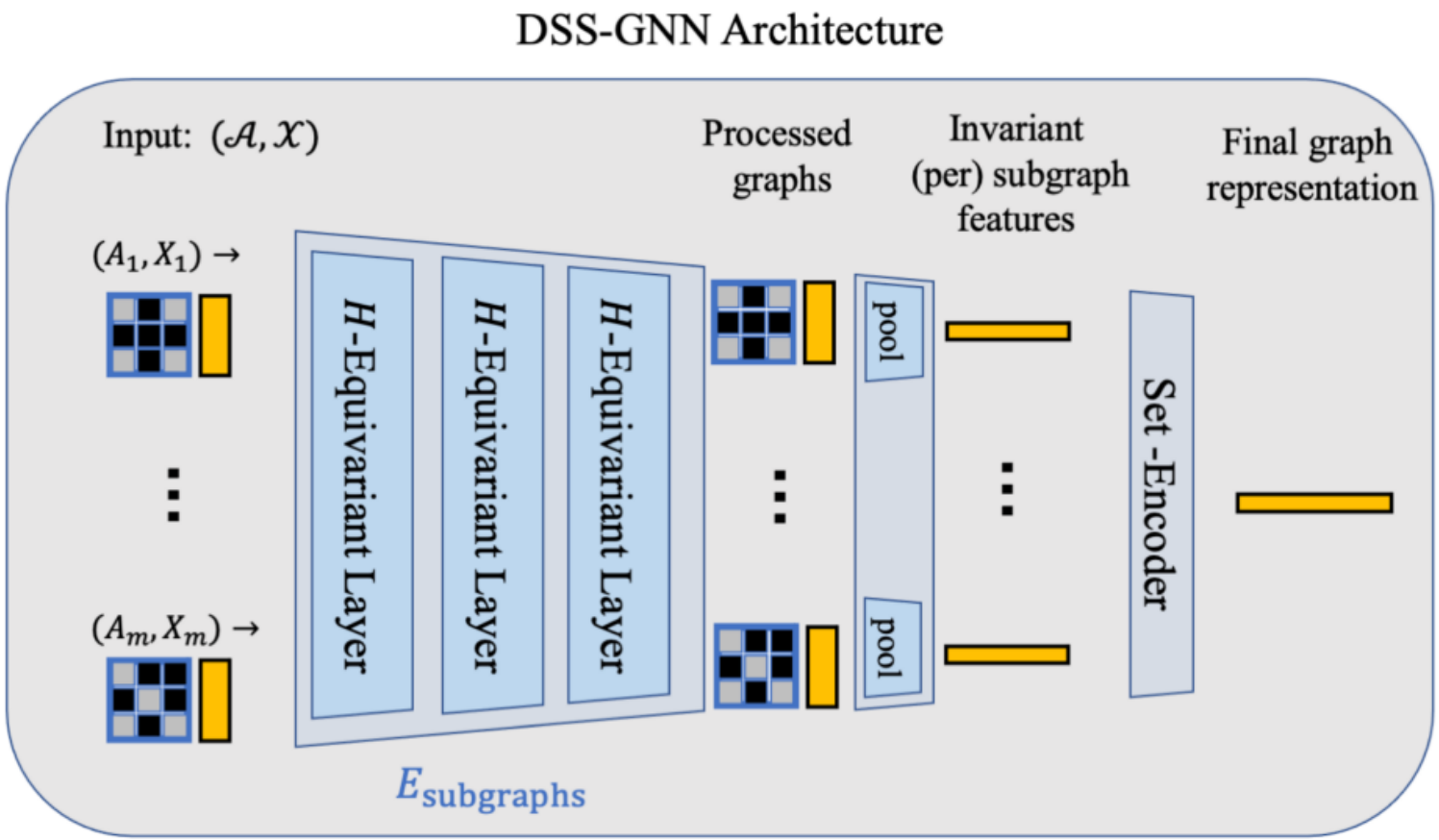
- **DSS-GNN**:
  - L_1, L_2 are called the base encoders
  - Usually, we use MPNNs
- **DSS** preserves node alignment

$$L(Z)_i = L_1(z_i) + L_2\left(\sum_{j=1}^{n} z_j\right)$$

- $\sum A_j$, $\sum X_j$ can be replaced with any invariant aggregation like max and mean

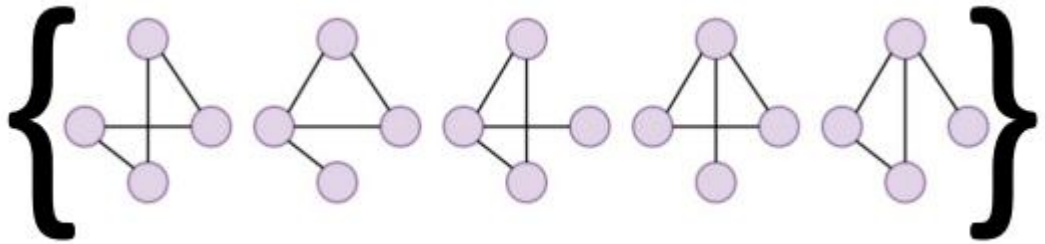


- Learning subgraphs with three Equivariant layers



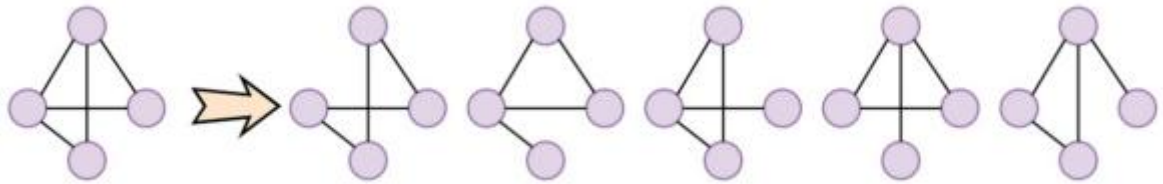
Equivariant Subgraph Aggregation Networks (ESAN)

- Two main challenges:
  - Architecture: How to process sets of subgraphs ?
  - We design layers that respect the symmetry of sets of graphs

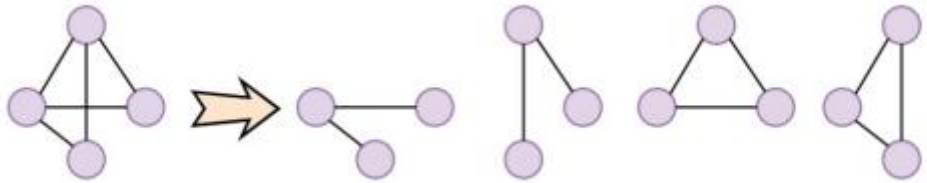


Equivariant Subgraph Aggregation Networks (ESAN)

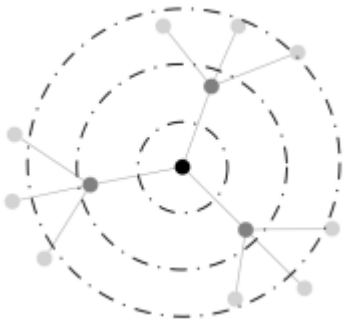
- Subpraph selection policies
  - Edge-deleted subgraphs



  - Node-deleted subgraphs

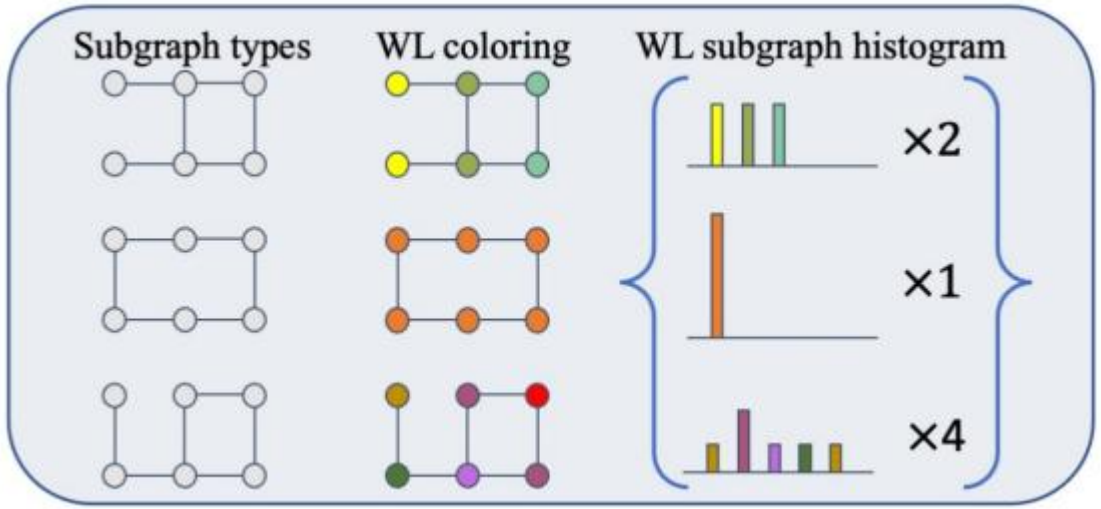


  - Ego-networks (with and without root identification)



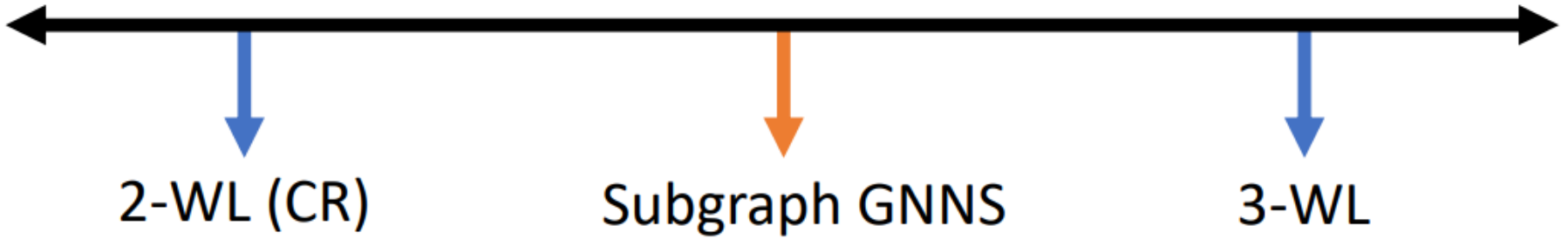
Comparison to WL

- Proposition 1 (new WL variant):
  - The architecture can implement a stronger variant of WL (DSS-WL)



Results: The power of subgraph-based GNNs (ESAN)

- **Theorem**: Subgraph GNNs are bounded by 3-WL expressive power
- Proof: Simulate subgraph GNNs with 3-IGN



B. Bevilacqua , et. al., Equivariant Subgraph Aggregation Networks ICLR 2022

네트워크 과학 연구실
NETWORK SCIENCE LAB

가톨릭대학교
THE CATHOLIC UNIVERSITY OF KOREA