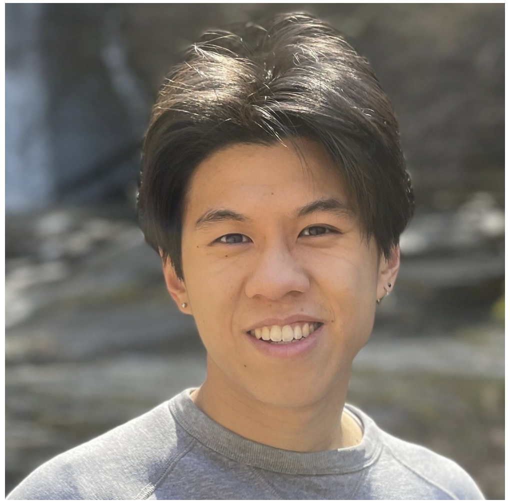
Subgraph-based networks for expressive, efficient, and domain-independent graph learning

Haggai Maron

Beatrice Bevilacqua
(Purdue)

Fabrizio Frasca
(Imperial College London)



Derek Lim
(MIT)

Equivariant Subgraph Aggregation Networks

ICLR 2022 (Spotlight presentation)

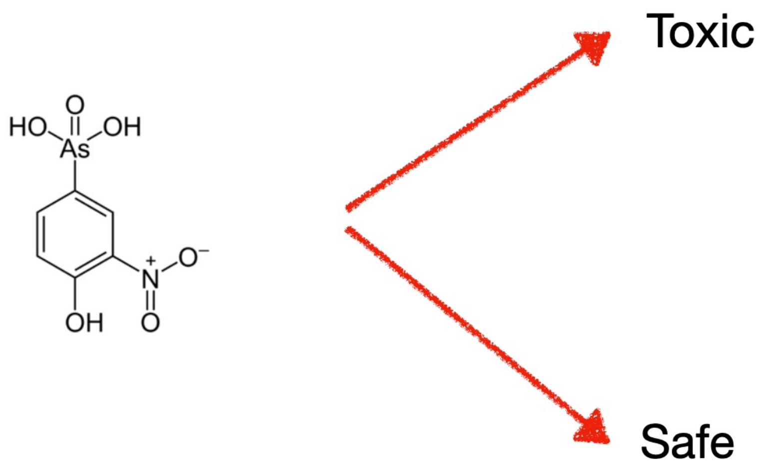
B. Bevilacqua, F. Frasca*, D. Lim*, B. Srinivasan, C. Cai, G. Balamurugan, M. M. Bronstein,* ***H. Maron***

Understanding and Extending Subgraph GNNs by Rethinking Their Symmetries

NeurIPS 2022 (oral presentation)

F. Frasca, B. Bevilacqua*, M. M. Bronstein,* ***H. Maron***

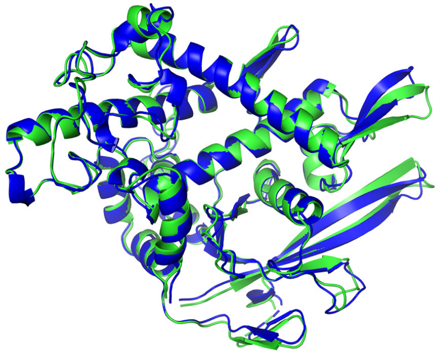
Learning on graphs



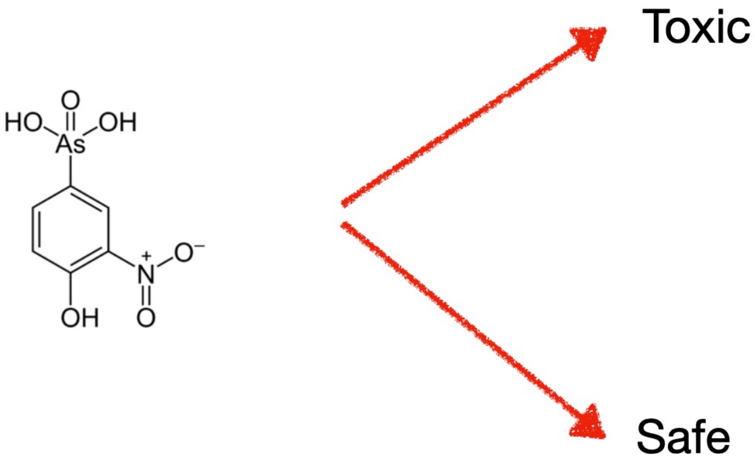
Molecule classification

Learning on graphs

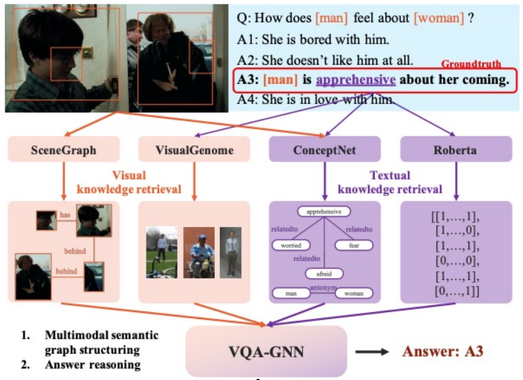
Social network analysis



Protein structure prediction

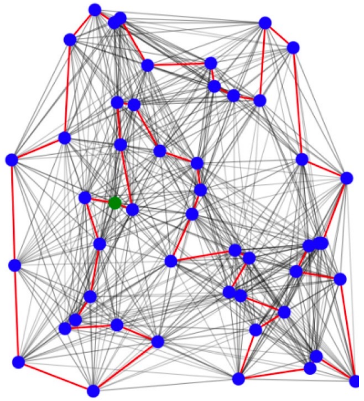


Molecule classification



Wang et al., 2022

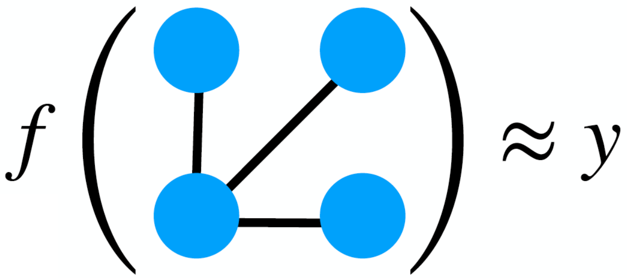
Visual question answering



Solving combinatorial optimization problems

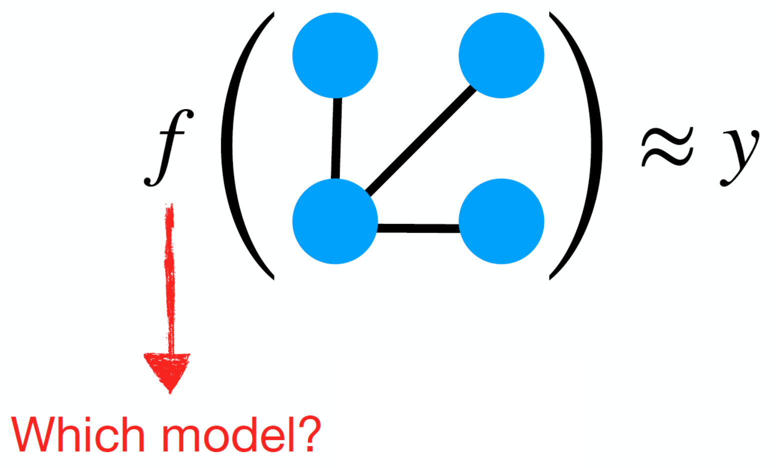
Setup

- **Training data**: $(G_1, y_1), \ldots, (G_m, y_m)$
- Each graph G_i consists of:
  - Adjacency structure A_i
  - Node features $x_j \in \mathbb{R}^d$
  - Label $y_i \in \{-1,1\}$
- **Goal**: find a model that maps graphs to output labels



Setup

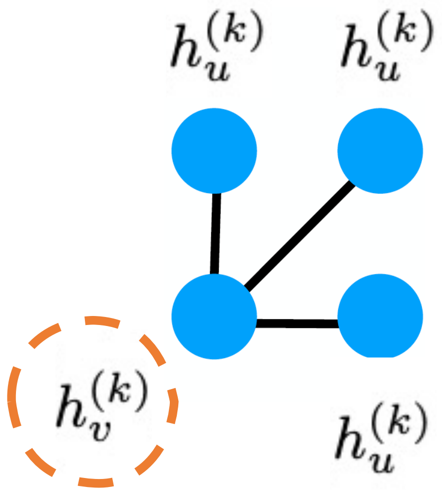
- **Training data**: $(G_1, y_1), \dots, (G_m, y_m)$
- Each graph G_i consists of:
  - Adjacency structure A_i
  - Node features $x_j \in \mathbb{R}^d$
  - Label $y_i \in \{-1,1\}$
- **Goal**: find a model that maps graphs to output labels



Message passing Neural Networks

- Parametric **neighborhood aggregation** layers

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

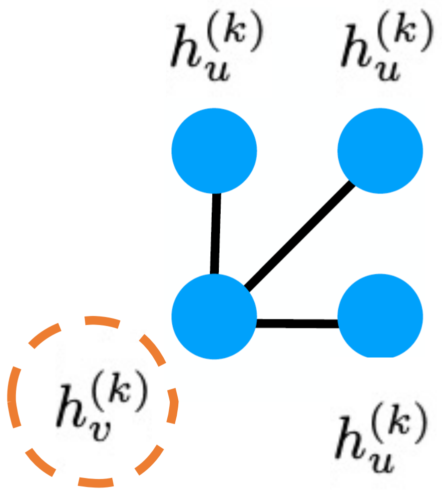


[Gilmer et al., ICML 2017]

Message passing Neural Networks

- Parametric **neighborhood aggregation** layers

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



- **final graph representation** aggregates all node features

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

[Gilmer et al., ICML 2017]

Q: What is the expressive power of MPNNs?

- Given two non-isomorphic graphs G_1, G_2
- Can we find an MPNN f such that:

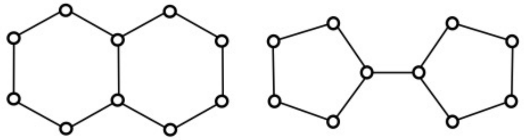
$$f(G_1) \neq f(G_2)$$

Q: What is the expressive power of MPNNs?

- Given two non-isomorphic graphs G_1, G_2
- Can we find an MPNN f such that:

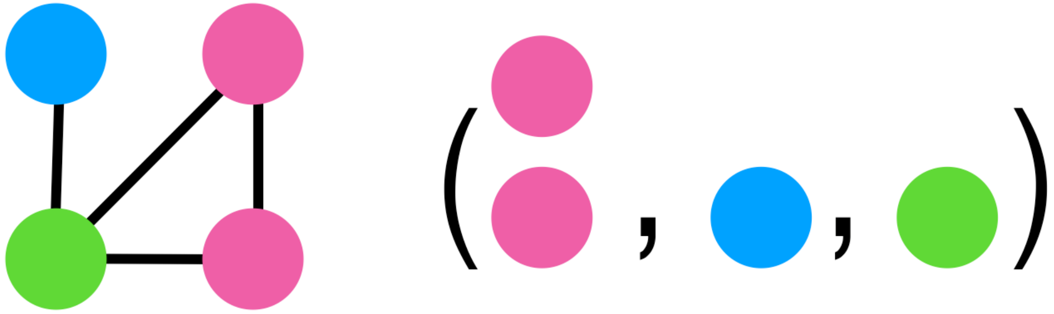
$$f(G_1) \neq f(G_2)$$

No!

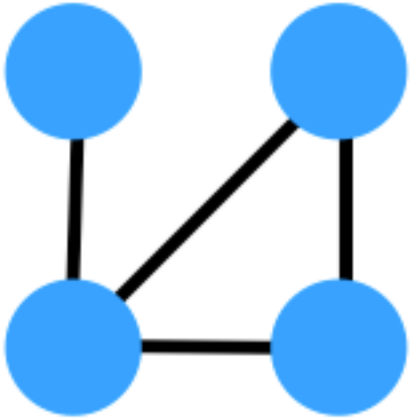


Color refinement

- MPNNs are closely related to the *color refinement* algorithm
- An efficient heuristic for graph isomorphism testing
- Also known as the **Weisfeiler-Lehman (WL)** graph isomorphism test

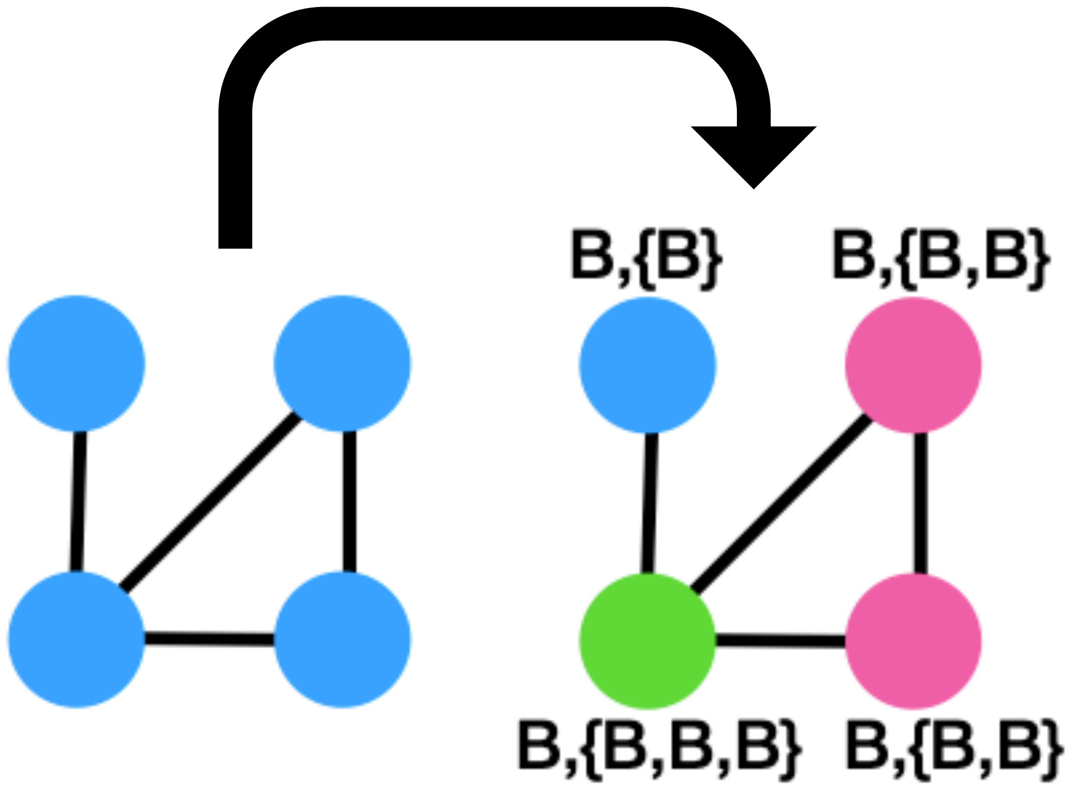


Color refinement



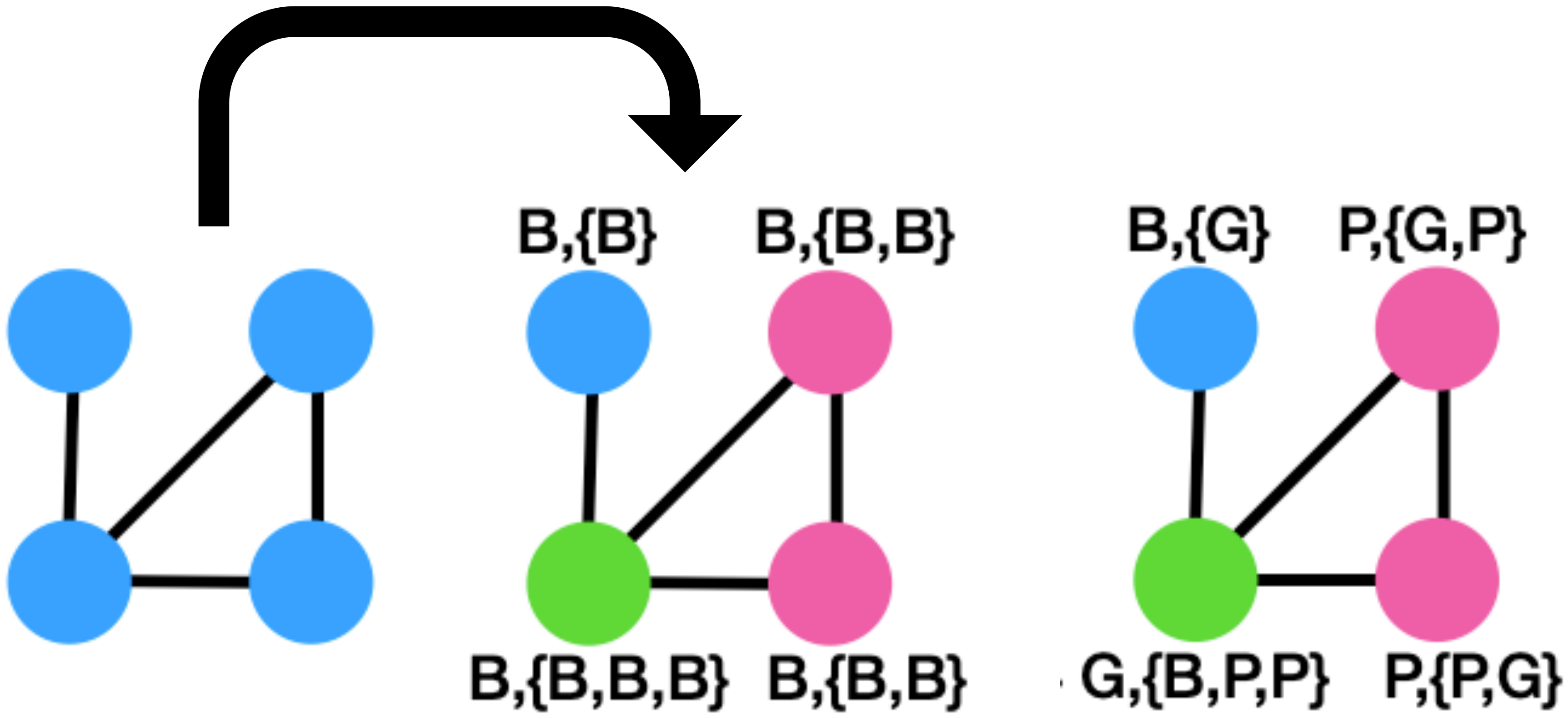
Color refinement

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



Color refinement

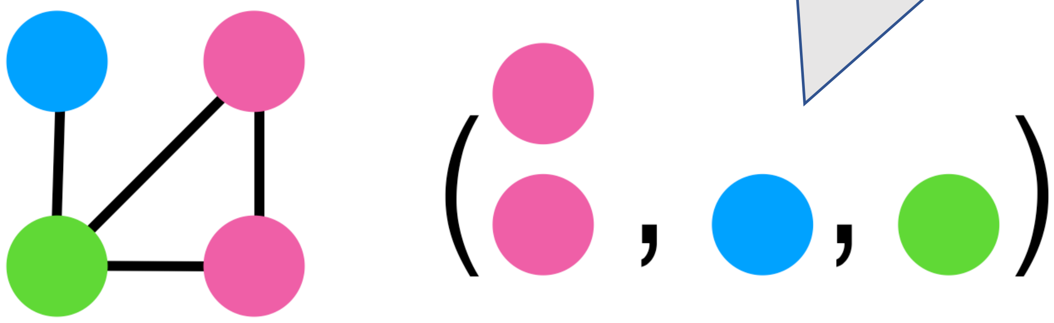
New color:= [old color; {colors of neighbors}]



Color refinement (CR)

- Final graph descriptor: Color histogram

Graph descriptor: [2,1,1]



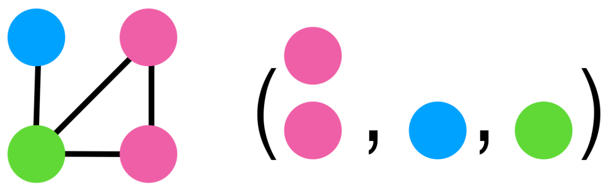
Color refinement (CR)

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

MPNNs

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

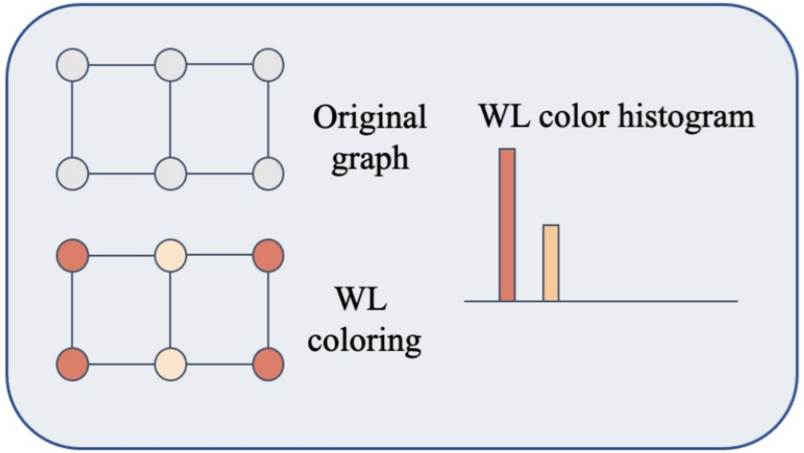
CR



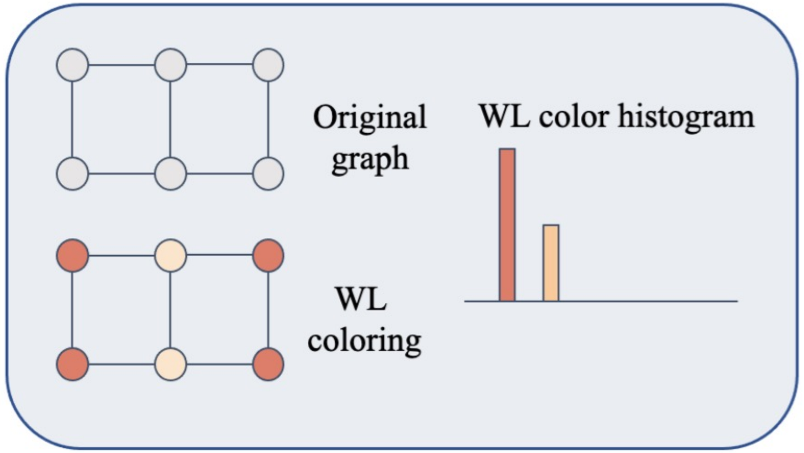
- **k-WL:** Higher-order, more powerful generalizations forming a hierarchy:

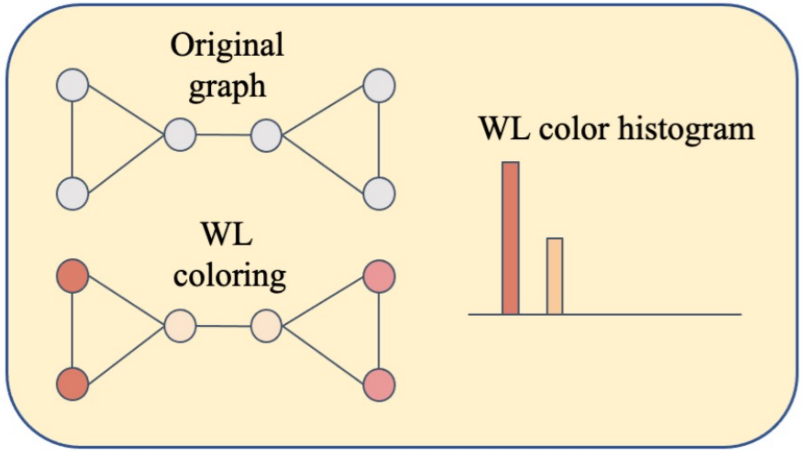
1-WL < 3-WL < 4-WL < ...

MPNNs have limited expressivity



MPNNs have limited expressivity

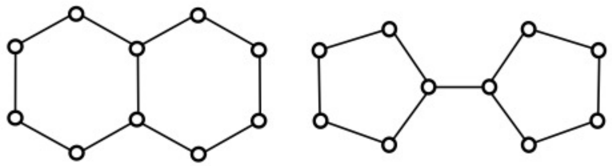




Why expressivity matters?

- Cannot assign different labels to different graphs

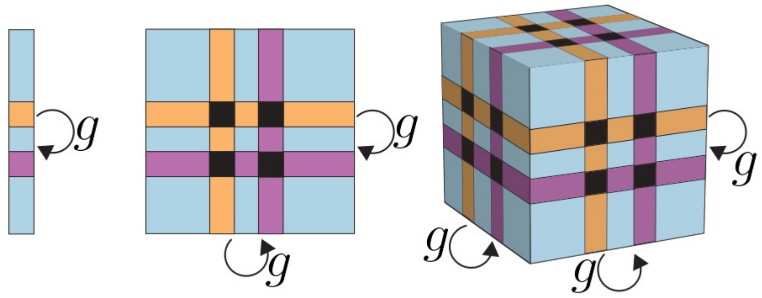
Decalin Bicyclopentyl



Taken from Bouritsas et al., 2021

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

State-of-the-art in expressive GNNs



- **k-GNNs/k-IGNs**

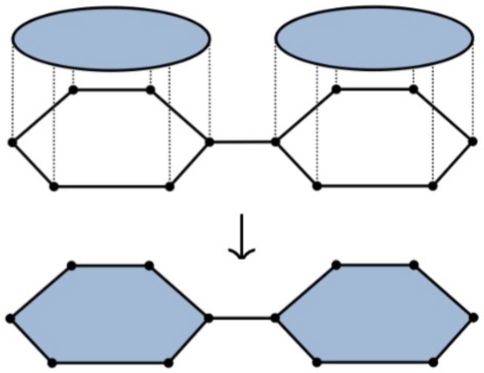
High computational complexity

[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]



- Using **domain knowledge**

Requires knowledge of meaningful structures

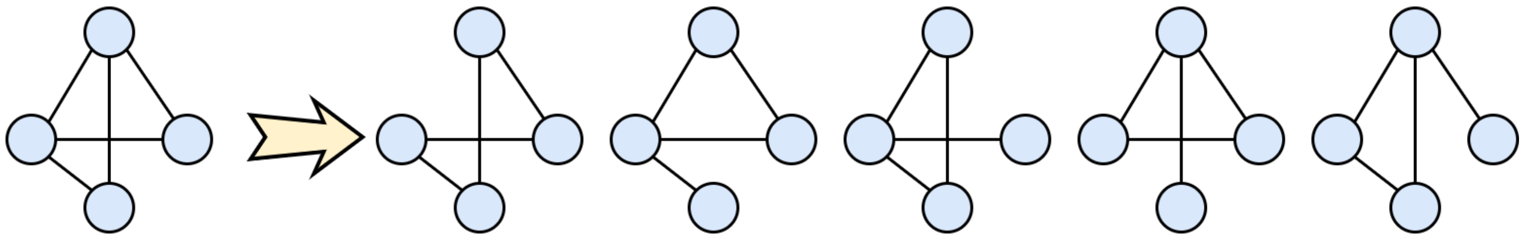
[Bouritsas et al., 2022, Bodnar et al., 2021]

Goal for today:

Domain-agnostic, Efficient, and Expressive GNN

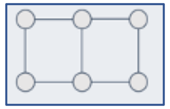
Sets of subgraphs: intuition

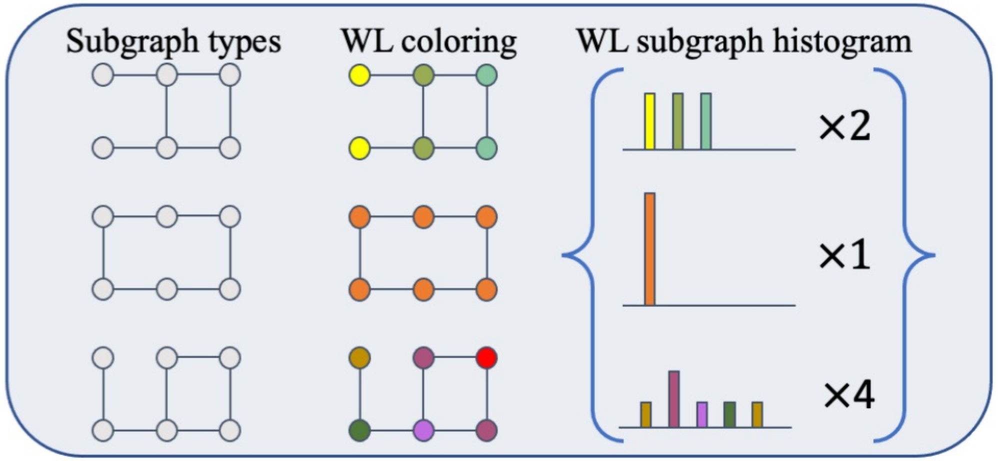
- **Main observation:** we can gain more expressive power by representing a graph as a set of subgraphs.



Sets of subgraphs: example

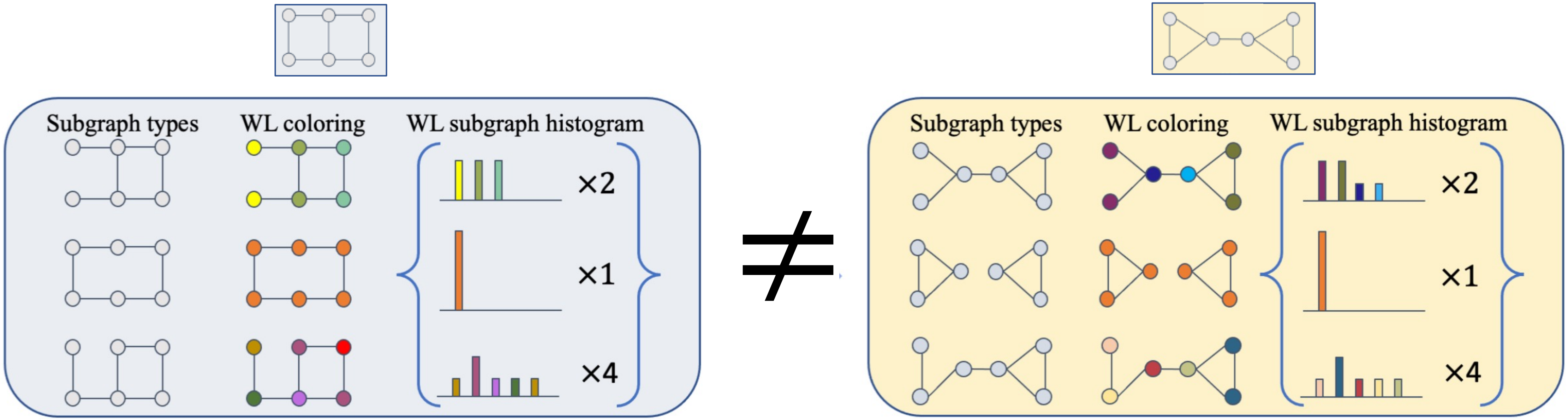
- Edge deleted subgraphs





Sets of subgraphs: example

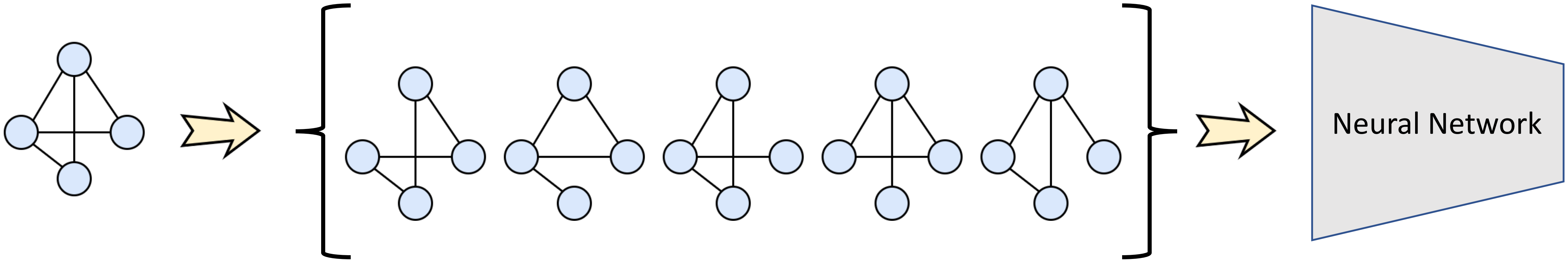
- Edge deleted subgraphs



Equivariant Subgraph Aggregation Networks (ESAN)

Recipe:

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



Equivariant Subgraph Aggregation Networks (ESAN)

Two main challenges:

- **Architecture:** How to process sets of subgraphs ?
  - We design layers that respect the resulting symmetry group
- Which **subgraph selection policies** are useful?
  - We propose four simple policies that work well

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

- Which **subgraph selection policies** are useful?
  - We propose four simple policies that work well

Equivariance as a design principle

- 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

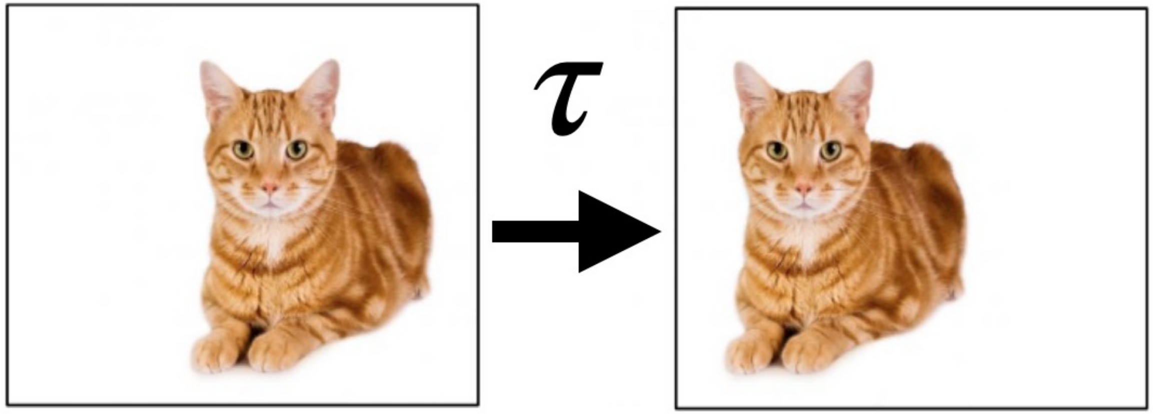


Image credit: imgur - https://imgur.com/mEIUqT8

Equivariance as a design principle

- A function f is called equivariant if :

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

- **Example**: Convolutions / image segmentation are translation equivariant

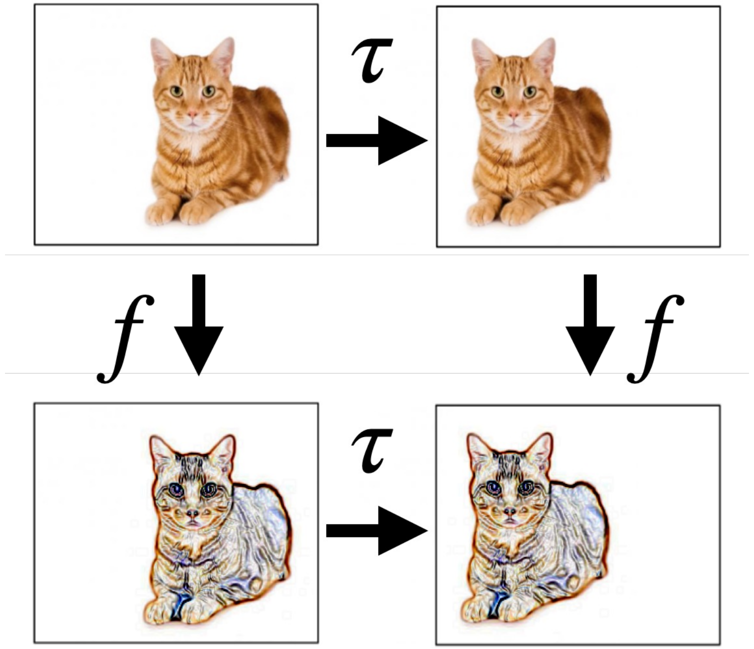
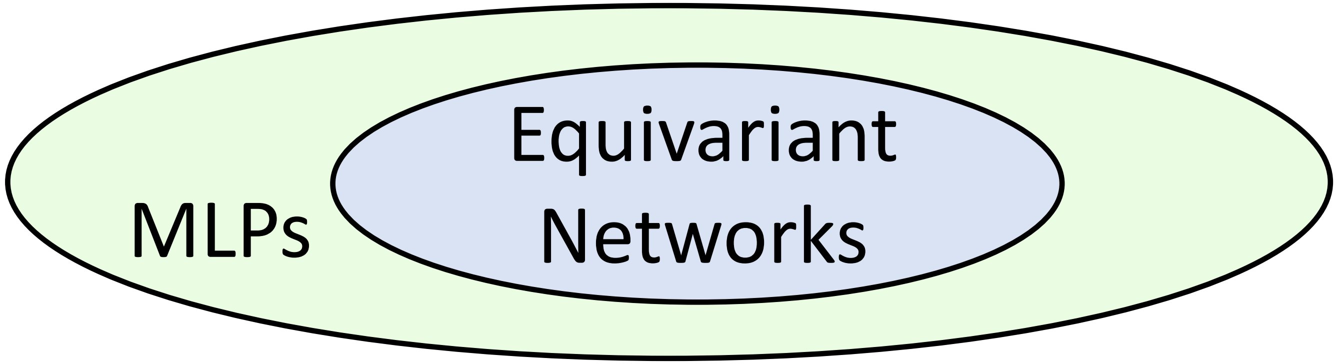


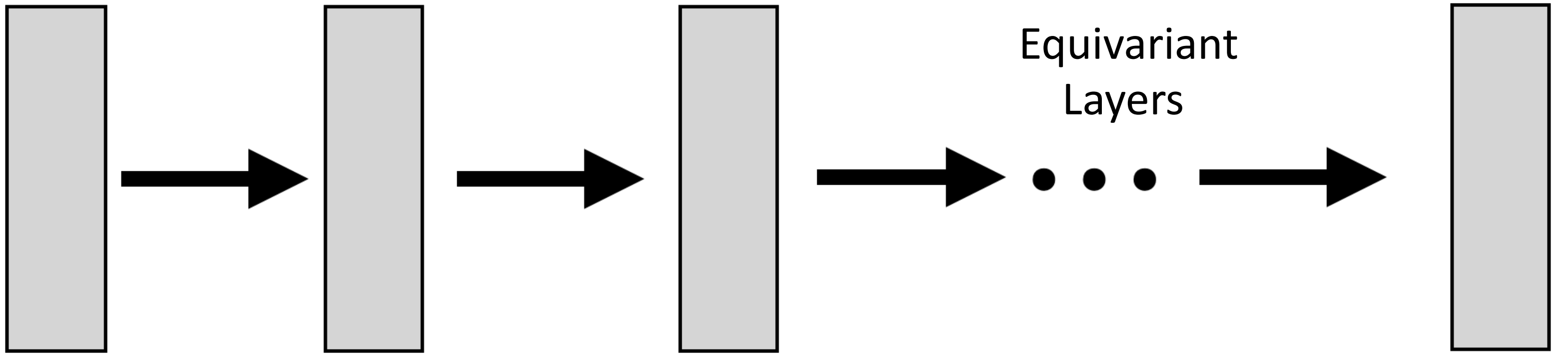
Image credit: imgur - https://imgur.com/mEIUqT8

Equivariance as a design principle

- **Common principle:** if the target function is equivariant, restrict hypothesis class to equivariant functions



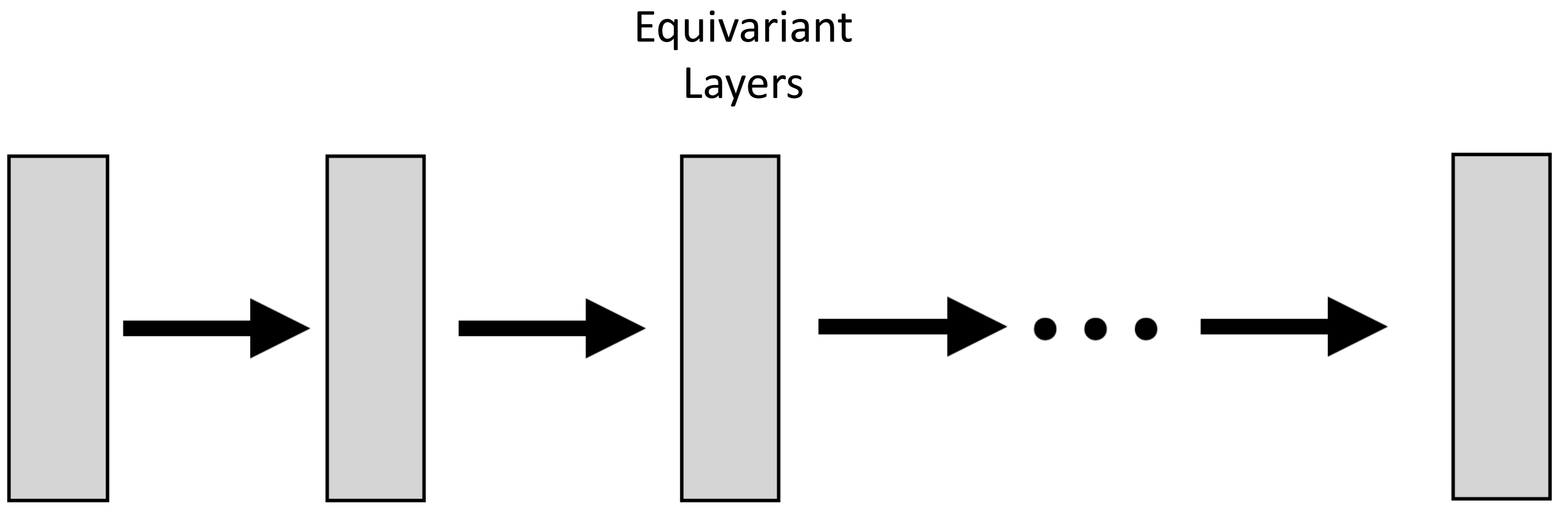
- Usually implemented by a sequence of simple equivariant functions



Equivariance as a design principle

Lots of theoretical and paractical benefits:

- Less parameters
- Better generalization
- Lower computational complexity



Equivariance as a design principle

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

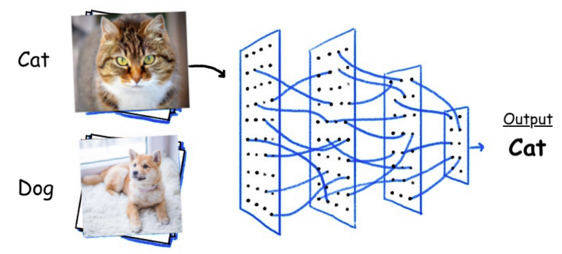


Image credit - Ethan Yanjia Li

Equivariance as a design principle

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

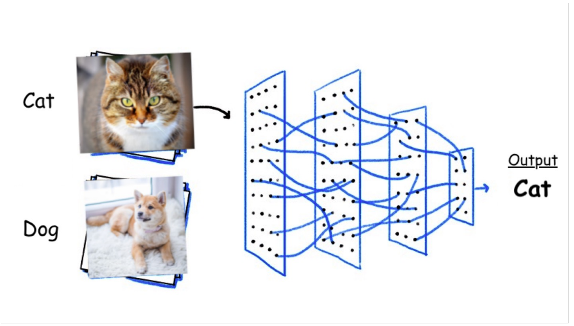
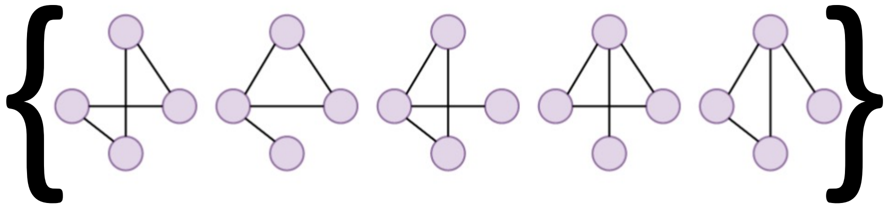
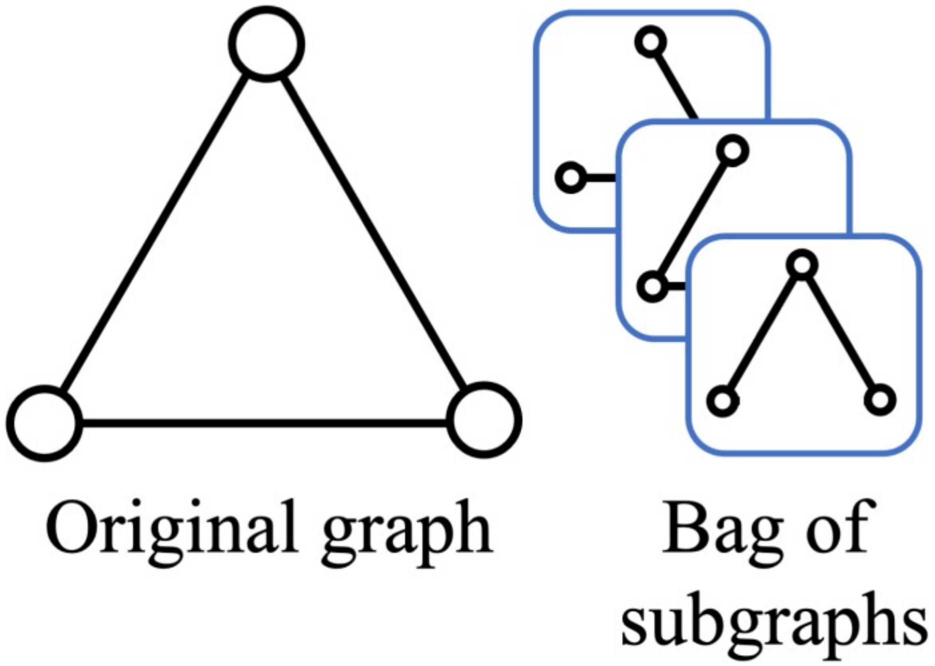


Image credit - Ethan Yanjia Li

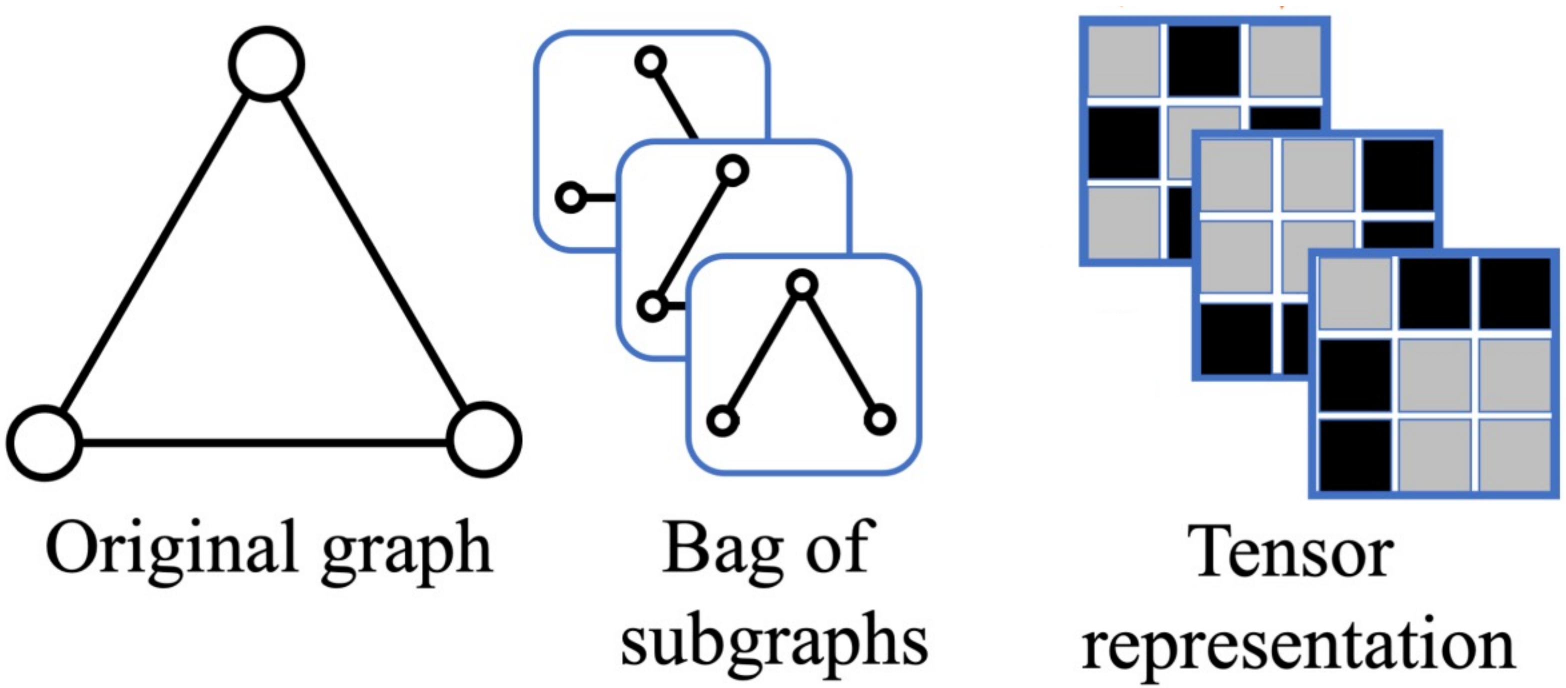
- In our case:
  - Input: sets of subgraphs
  - Symmetry group: ?
  - Basic layers: ?
  - Resulting architecture : ?



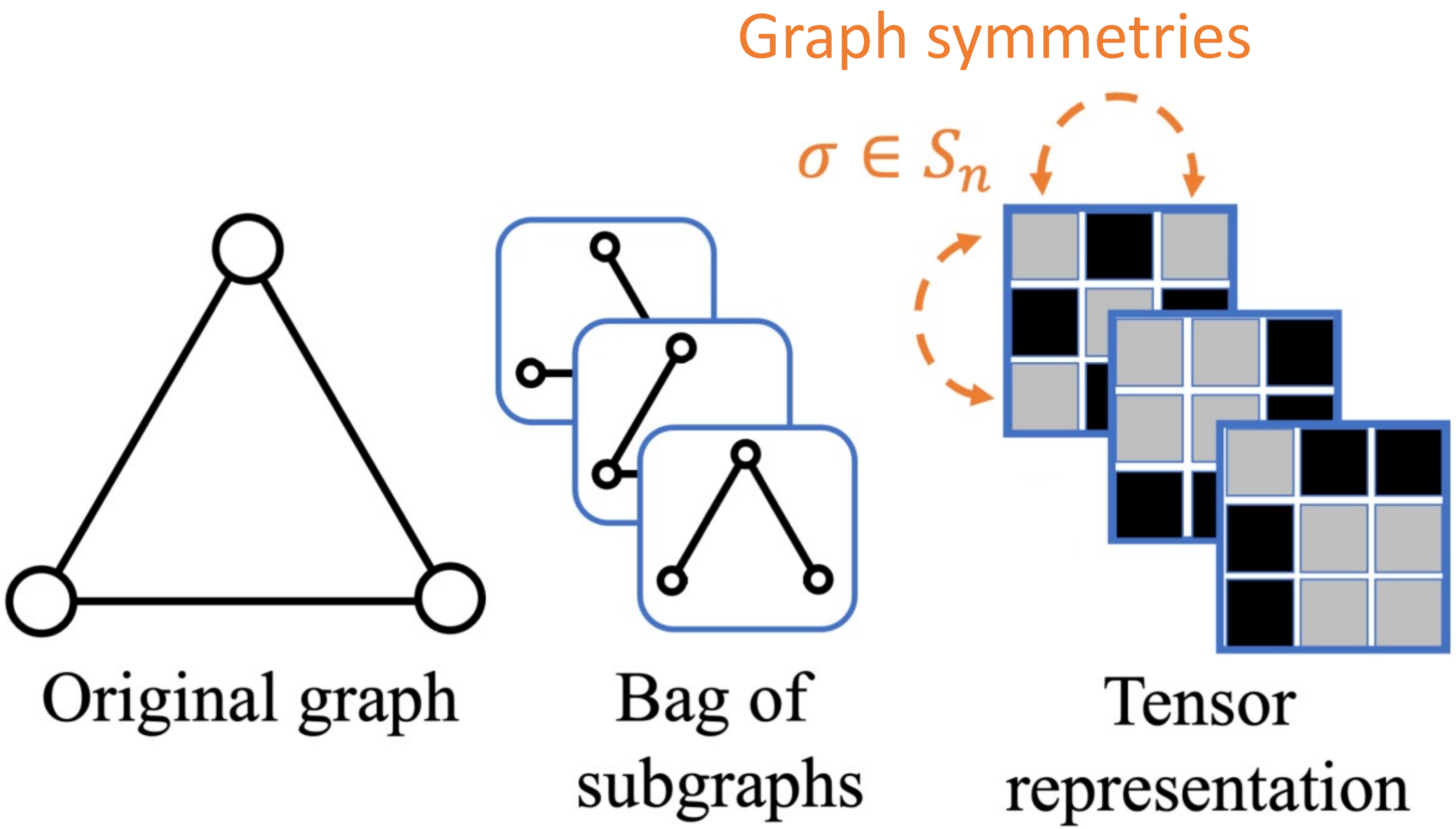
Symmetry for sets of subgraphs



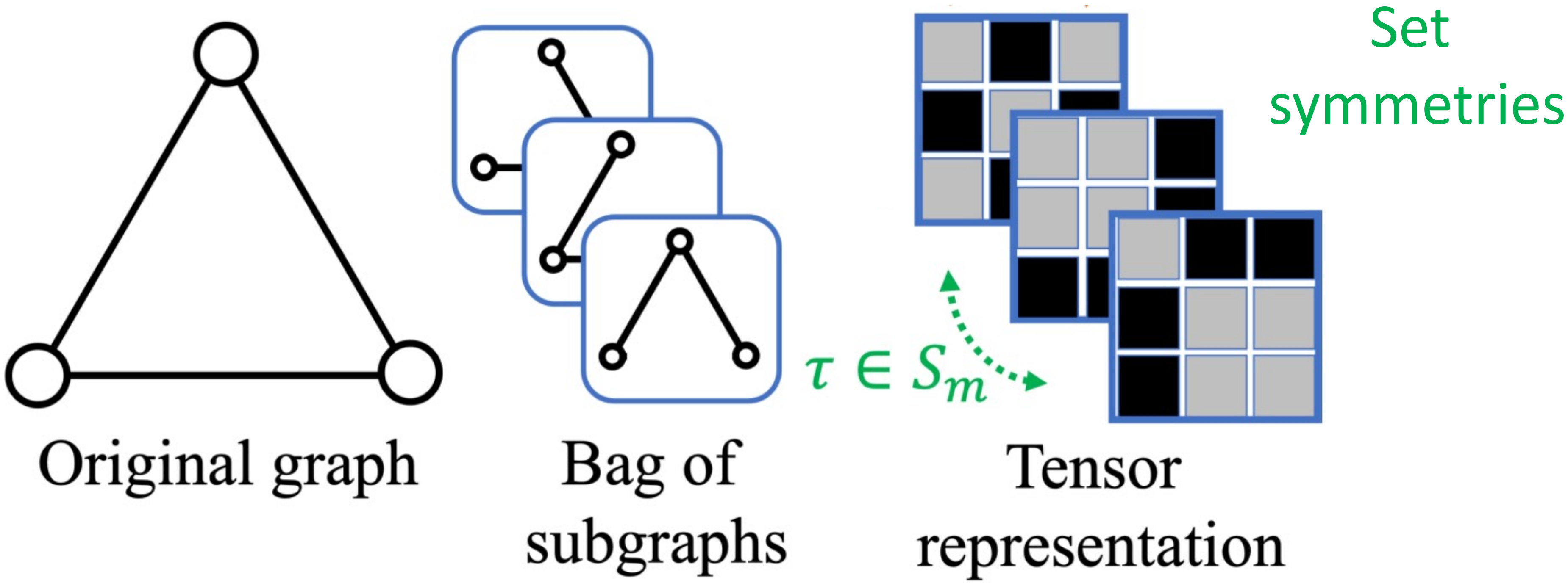
Symmetry for sets of subgraphs



Symmetry for sets of subgraphs

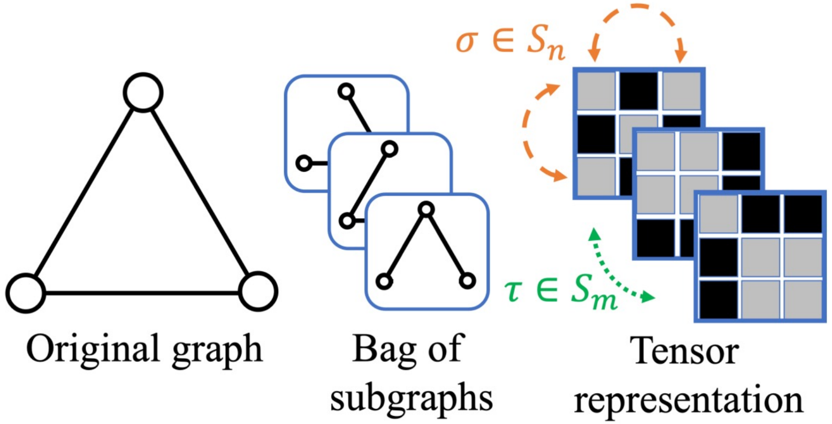


Symmetry for sets of subgraphs



Symmetry for sets of subgraphs

- We have two types of symmetries:
  - **Internal** graph symmetry
  - **External** set symmetry
- We know how to handle each one. What about their combination?



Detour: Deep Sets for Symmetric Elements

Input: a set whose elements have symmetry group H (e.g., set of graphs)

- $Z = [z_1, \dots, z_n]$ is a set with symmetry group S_n
- Each z_i has symmetry group H
- **Symmetry group of the whole thing is $\boldsymbol{G = S_n \times H}$**

Theorem. G –Equivariant *linear* layers are of the following form:

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

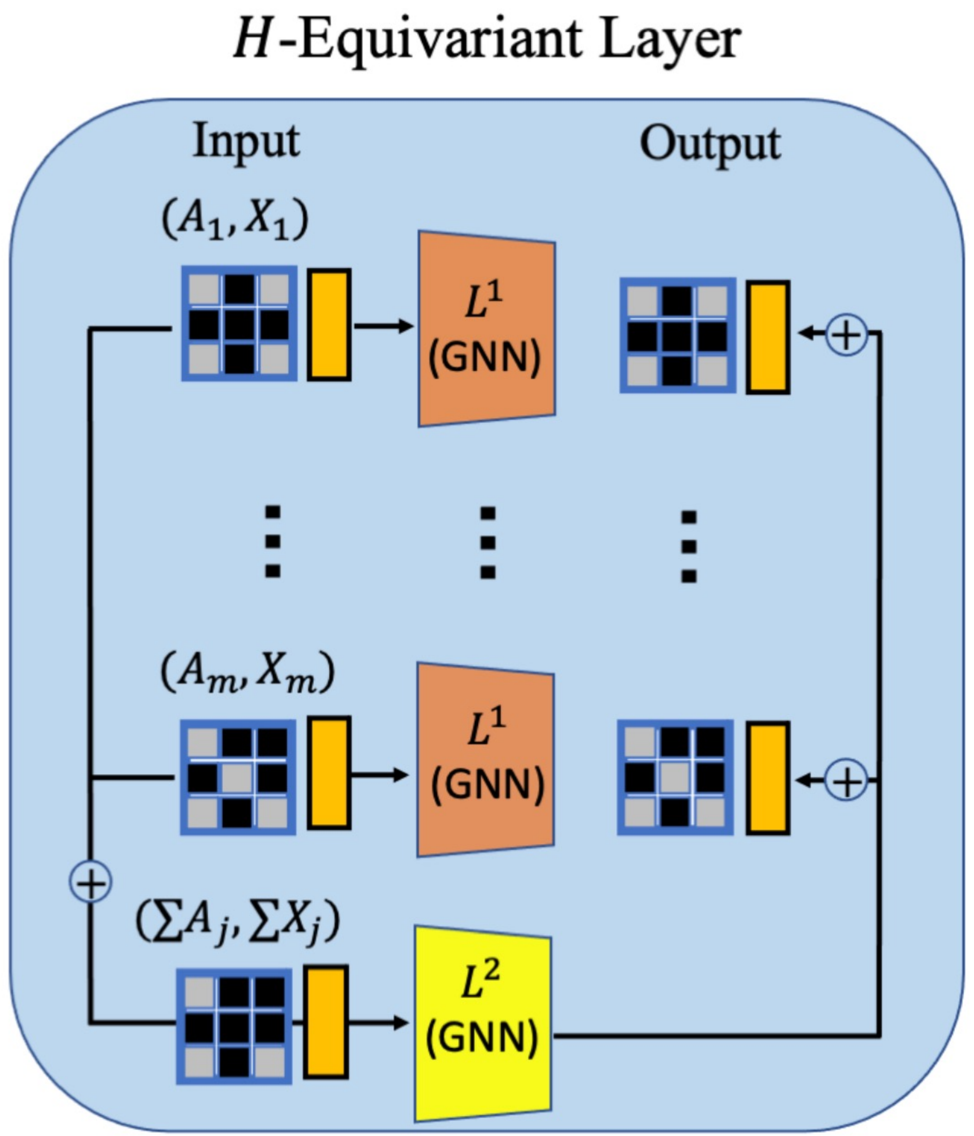
- L_1, L_2 are H –equivariant

[M. et al., ICML 2020]

Equivariant layer

DSS-GNN:

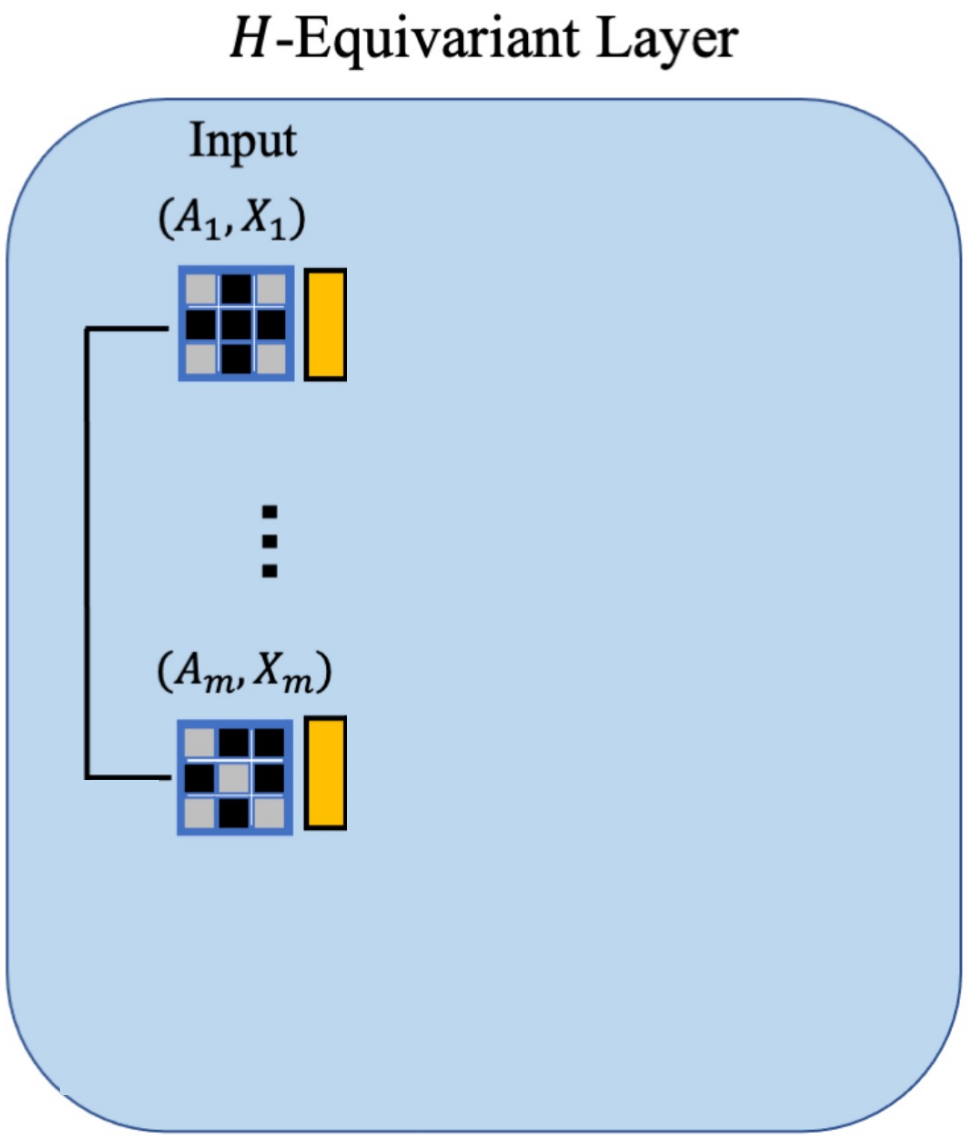
- L_1, L_2 are called the **base encoders**
  - Usually, we use MPNNs
- DSS preserves **node alignment**
- $\sum A_j, \sum X_j$ can be replaced with any invariant aggregation like max and mean



Equivariant layer

DSS-GNN:

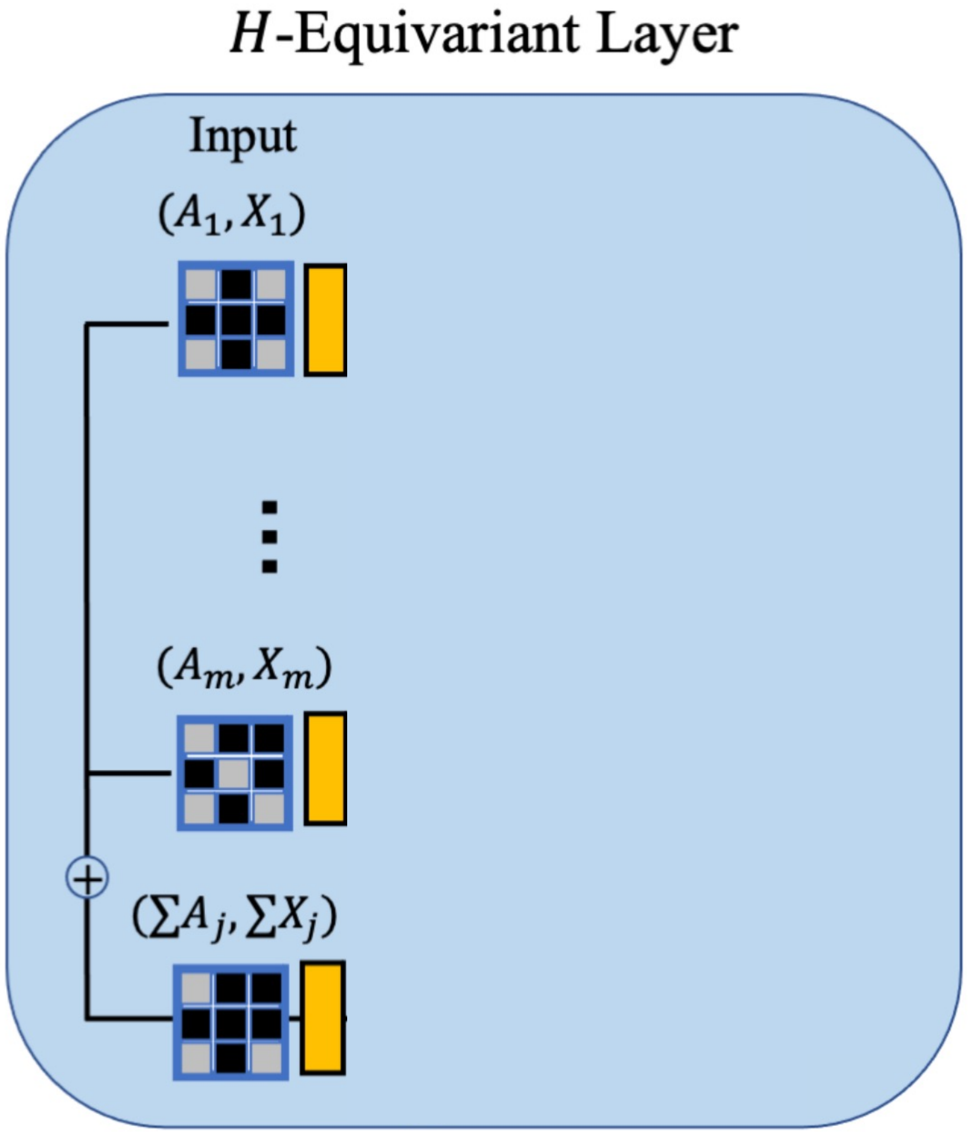
- L_1, L_2 are called the **base encoders**
  - Usually, we use MPNNs
- DSS preserves **node alignment**
- $\sum A_j, \sum X_j$ can be replaced with any invariant aggregation like max and mean



Equivariant layer

DSS-GNN:

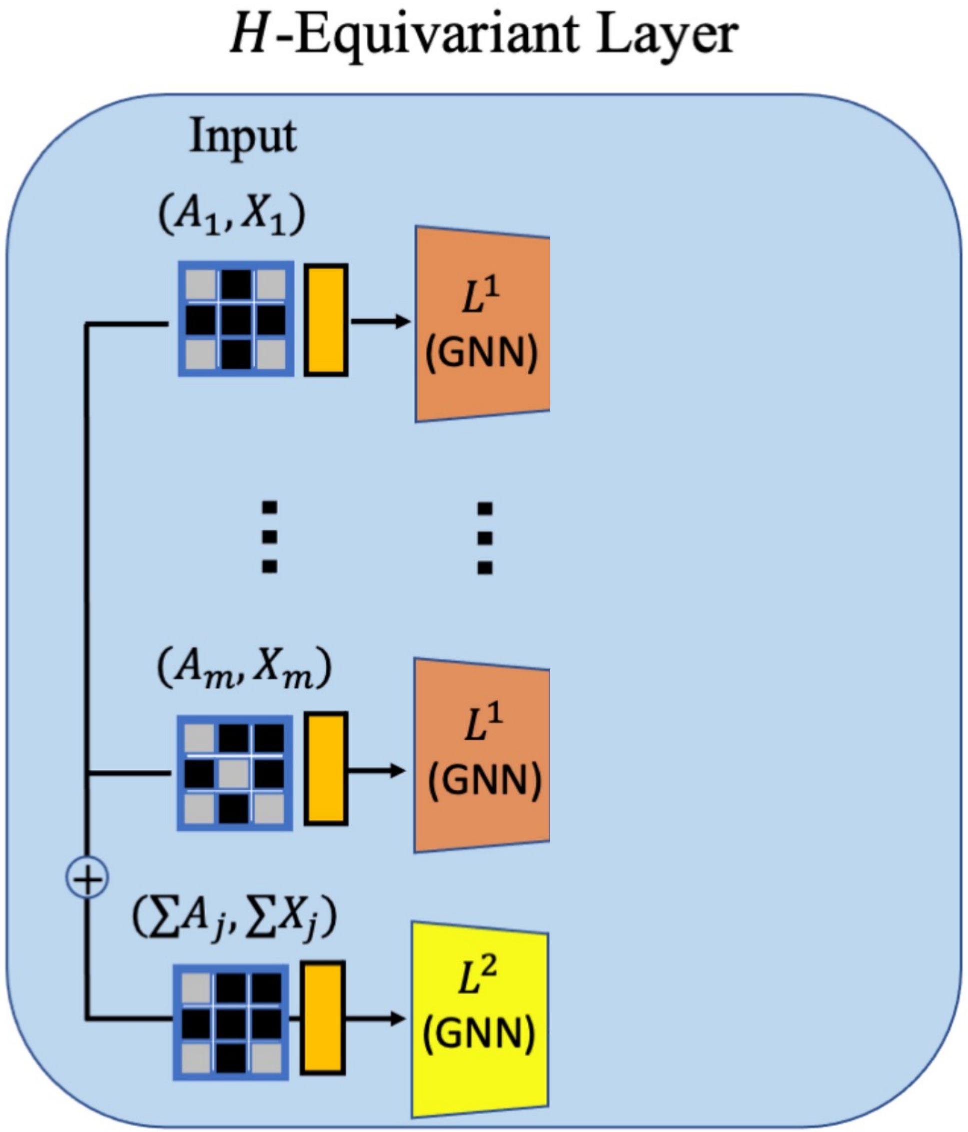
- L_1, L_2 are called the **base encoders**
  - Usually, we use MPNNs
- DSS preserves **node alignment**
- $\sum A_j, \sum X_j$ can be replaced with any invariant aggregation like max and mean



Equivariant layer

DSS-GNN:

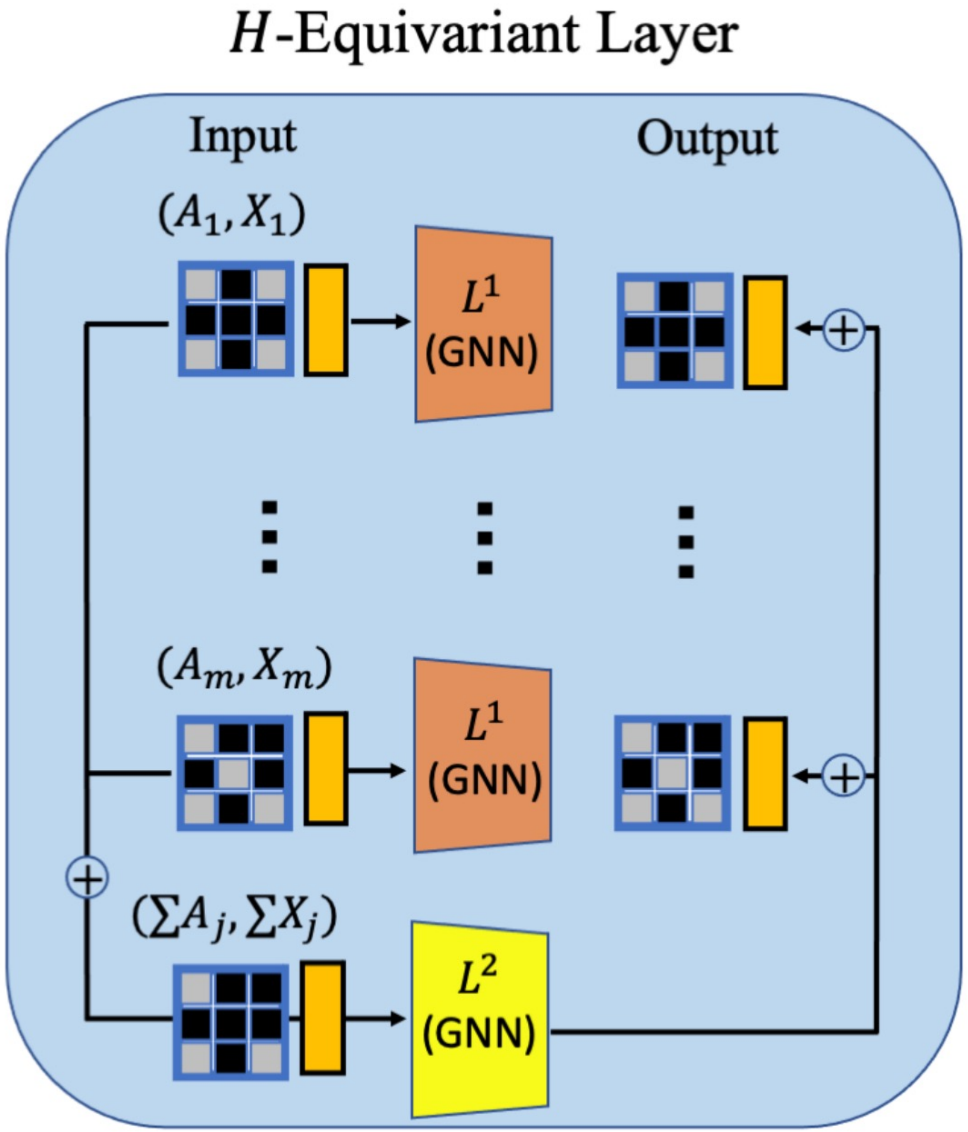
- L_1, L_2 are called the **base encoders**
  - Usually, we use MPNNs
- DSS preserves **node alignment**
- $\sum A_j, \sum X_j$ can be replaced with any invariant aggregation like max and mean



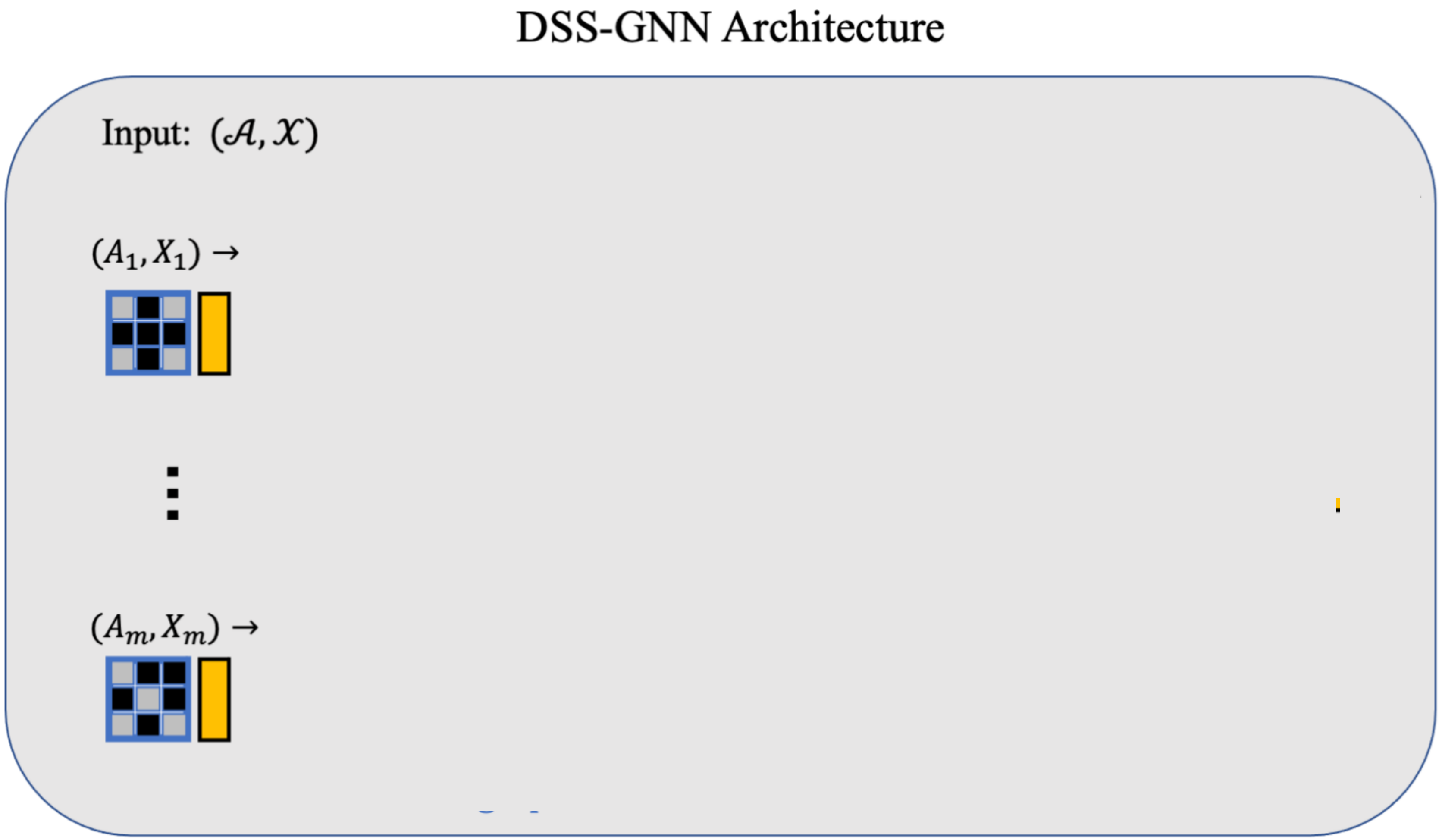
Equivariant layer

DSS-GNN:

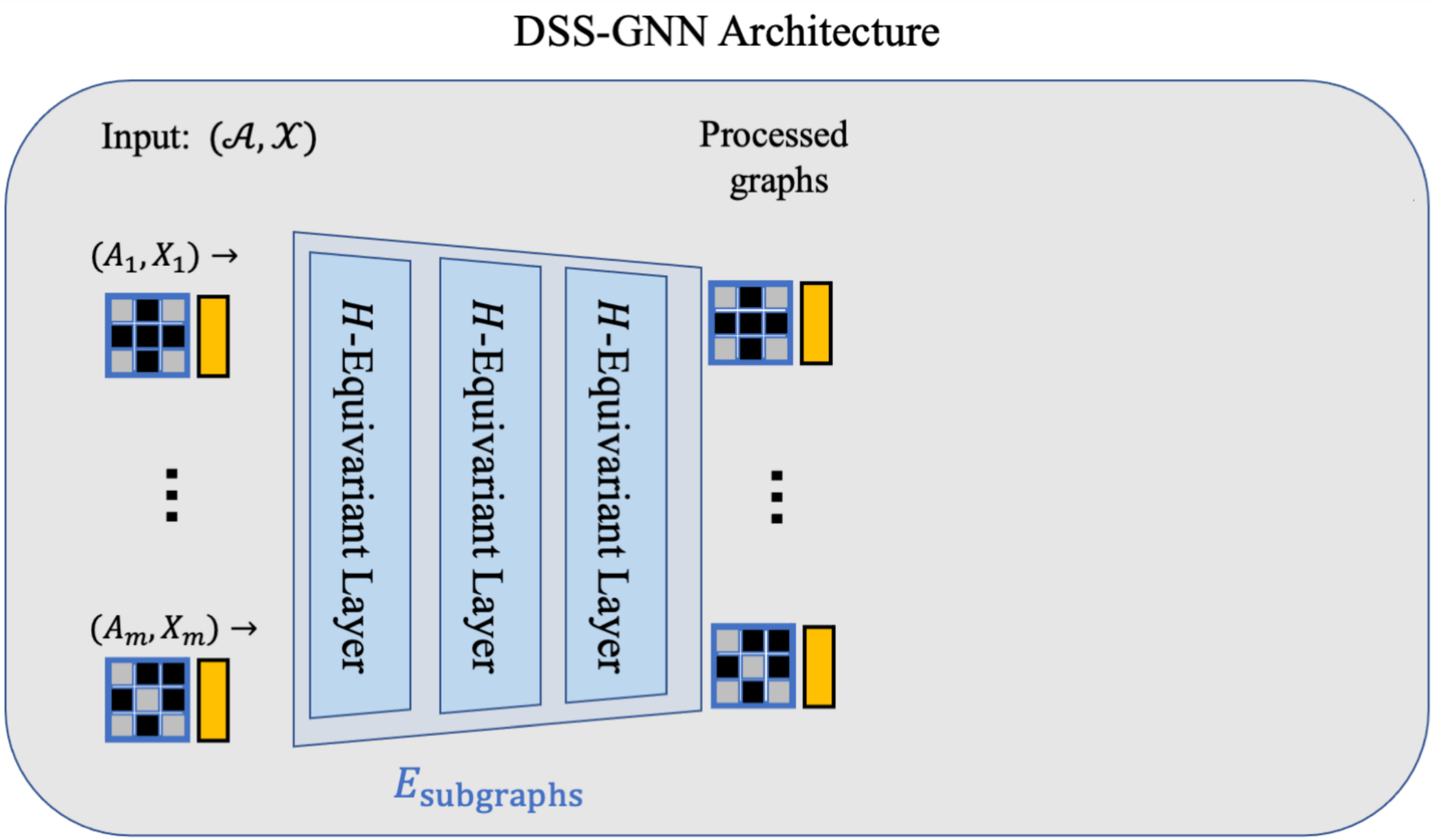
- L_1, L_2 are called the **base encoders**
  - Usually, we use MPNNs
- DSS preserves **node alignment**
- $\sum A_j, \sum X_j$ can be replaced with any invariant aggregation like max and mean



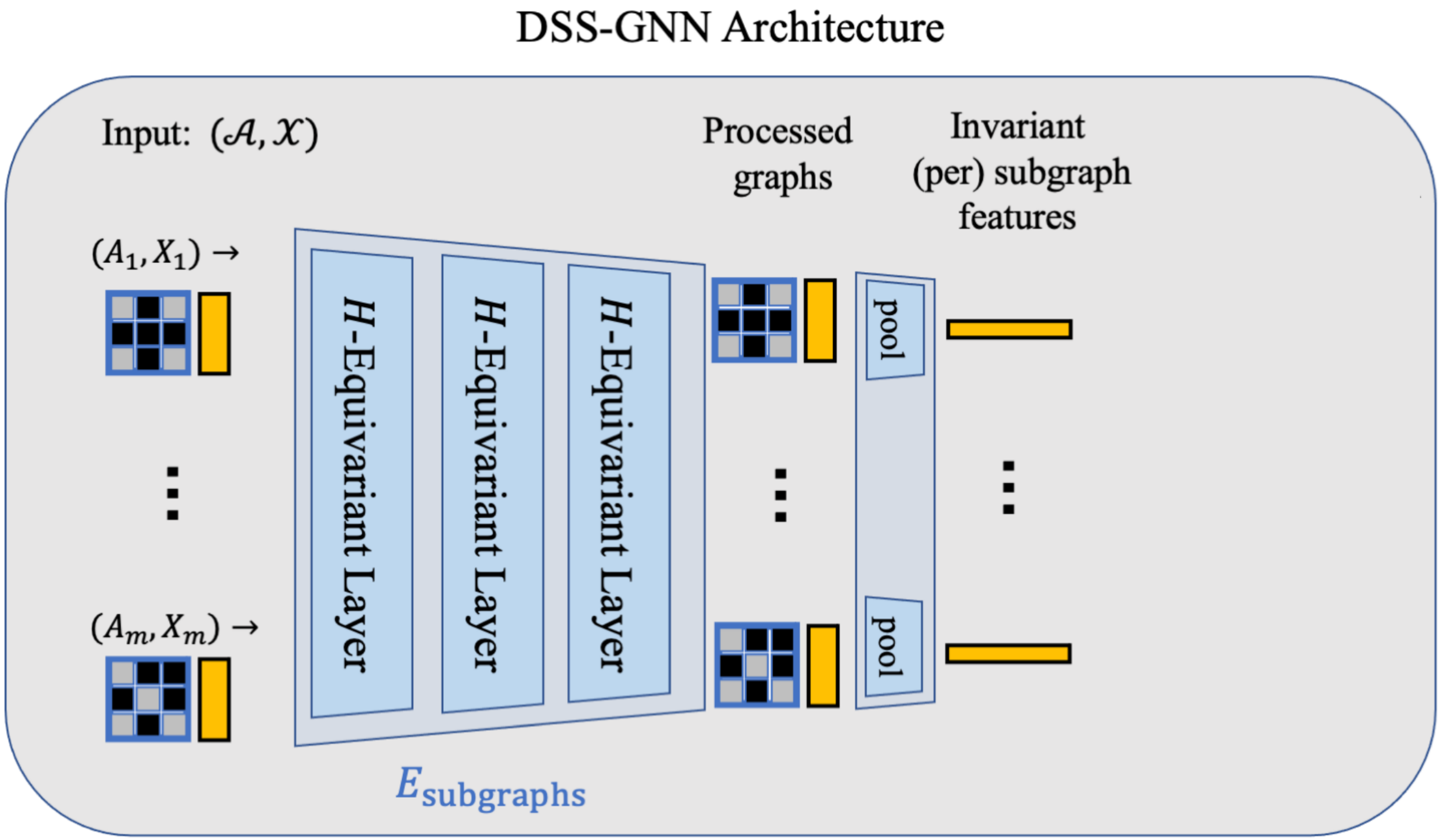
Architecture



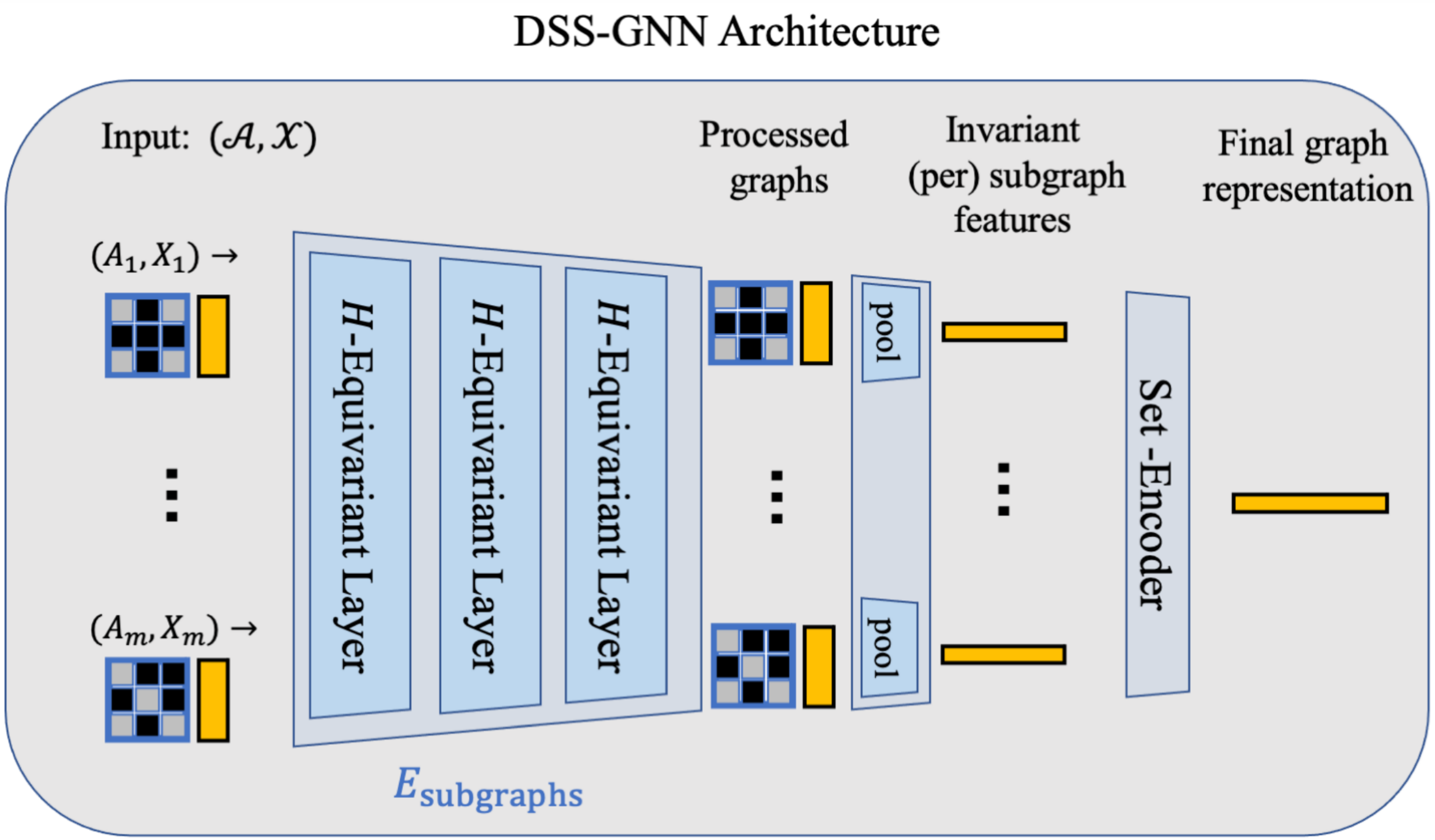
Architecture



Architecture



Architecture



Equivariant Subgraph Aggregation Networks (ESAN)

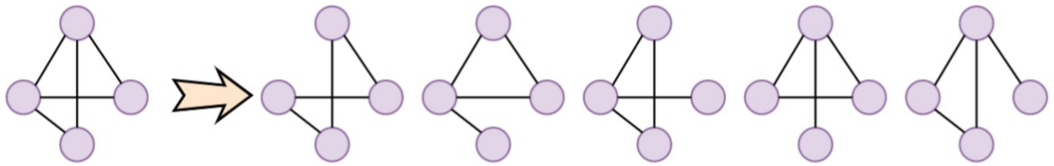
Two main challenges:

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

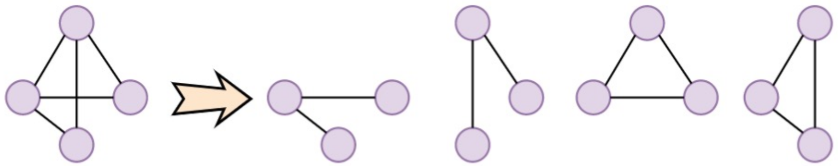
- Which **subgraph selection policies** are useful?
    - We propose four simple policies that work well

Subpraph selection policies

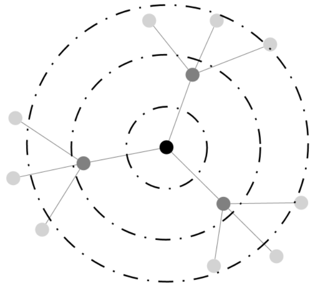
- Edge-deleted subgraphs



- Node-deleted subgraphs



- Ego-networks (with and without root identification)



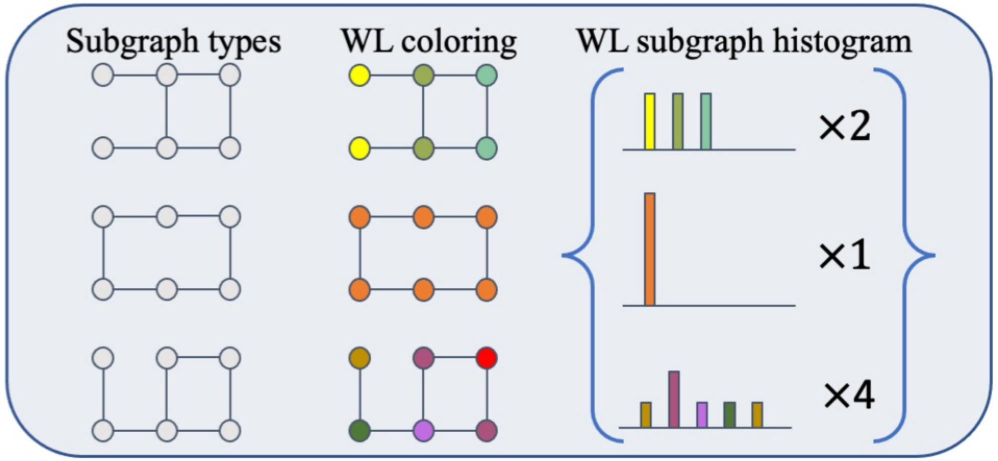
Stochastic subgraph sampling

- Full policies might generate too many subgraphs
- Solution: sample subsets of the policies
  - We tried 5%,20%,50%
- Two benefits:
  - Can process larger graphs
  - Improves training time

Experiments demostrate that subgraph sampling maintains high expressive power and performs well on real data

Comparison to WL

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



Design choices and expressivity

- **Proposition 2 (policy matters):** On the family of strongly regular graphs:

  Edge-deletion > Node-deletion = Depth-n ego-nets = 3-WL.

- **Proposition 3 (base graph encoder matters):** Our architecture with 3-WL base encoder is strictly stronger than our architecture with a 1-WL base encoder (MPNN)

- Many more results in the paper

Experiments

- ESAN is SOTA among *domain agnostic* methods on multiple important datasets
- For example, on the ZINC molecule property prediction
  - Target property: *logP (water-octanol partition coefficient)*

Method	ZINC (MAE ↓)
PNA (Corso et al., 2020)	0.188±0.004
DGN (Beaini et al., 2021)	0.168±0.003
SMP (Vignac et al., 2020)	0.138±?
GIN (Xu et al., 2019)	0.252±0.017
HIMP (Fey et al., 2020)	0.151±0.006
GSN (Bouritsas et al., 2022)	0.108±0.018
CIN-SMALL (Bodnar et al., 2021a)	0.094±0.004
DS-GNN (GIN) (ED)	0.172±0.008
DS-GNN (GIN) (ND)	0.171±0.010
DS-GNN (GIN) (EGO)	0.126±0.006
DS-GNN (GIN) (EGO+)	0.116±0.009
DSS-GNN (GIN) (ED)	0.172±0.005
DSS-GNN (GIN) (ND)	0.166±0.004
DSS-GNN (GIN) (EGO)	0.107±0.005
DSS-GNN (GIN) (EGO+)	0.102±0.003

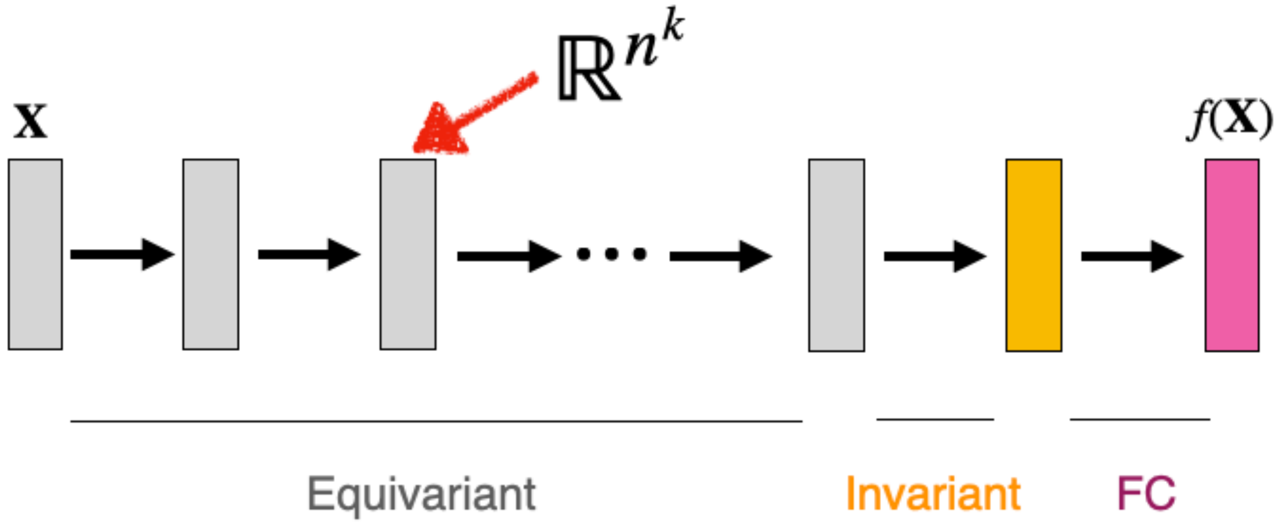
Is this the only way to design subgraph GNNs?

- A surge of recent subgraph GNNs in top ML conferences:
  - Drop GNN [Papp et al., 2021]
  - Reconstruction GNN [Cotta et al., 2021]
  - Nested-GNN [Zhang and Li, 2021]
  - ID-GNN [You et al, 2021]
  - GNN-AK [Zhao et al., 2022]
  - …
- A **zoo** of aggregation/sharing rules between subgraphs

Q: How can we compare/understand them? Is there a general framework?

Detour: Invariant Graph Networks (IGNs)

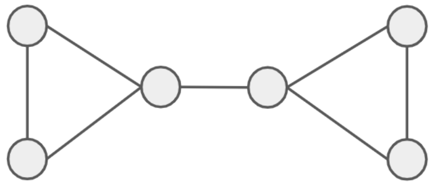
- Start with an adjacency representation in $\mathbb{R}^{n \times n}$
- Map to k order tensors $\mathbb{R}^{n^k}$ using linear equivariant maps
- k-IGNs are equivalent to k-WL

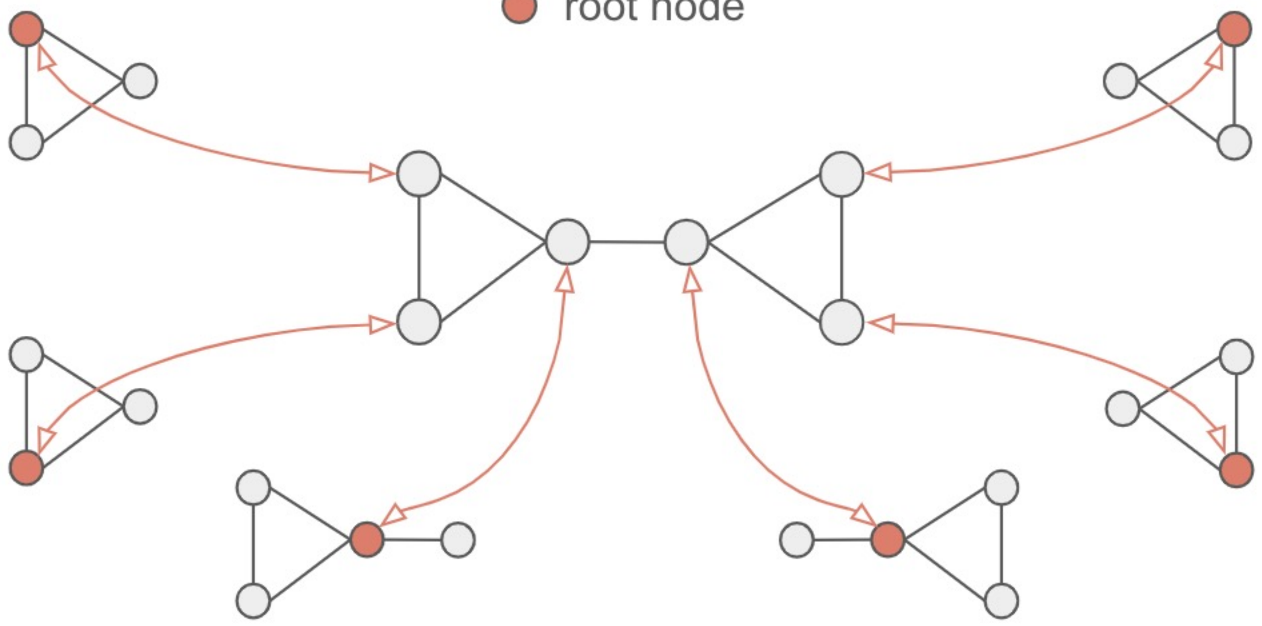


[M. et al., 2019, Girts 2020, Azizian and Lelarge 2020]

Node-based policies

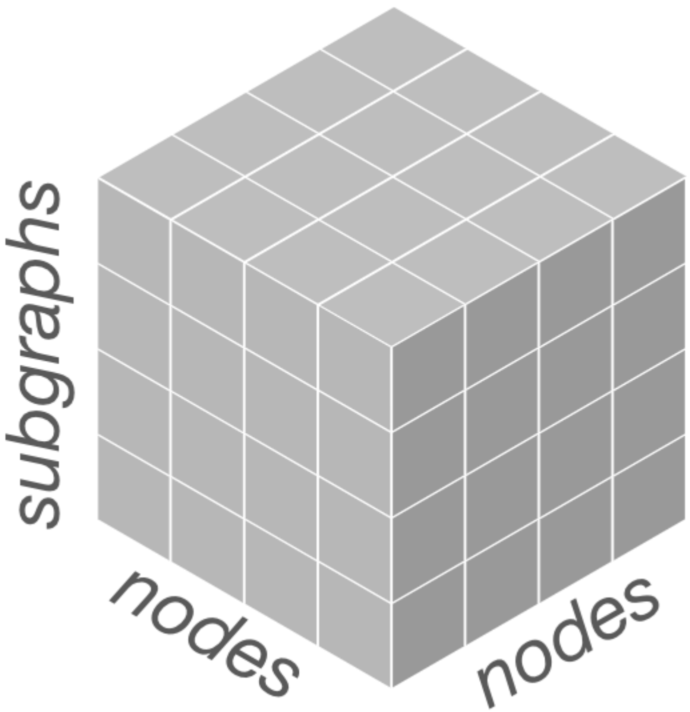
- We focus on node-based policies
  - Each subgraph is <u>tied</u> to a specific node
  - Examples: node-deletion, ego-networks, node-marking…





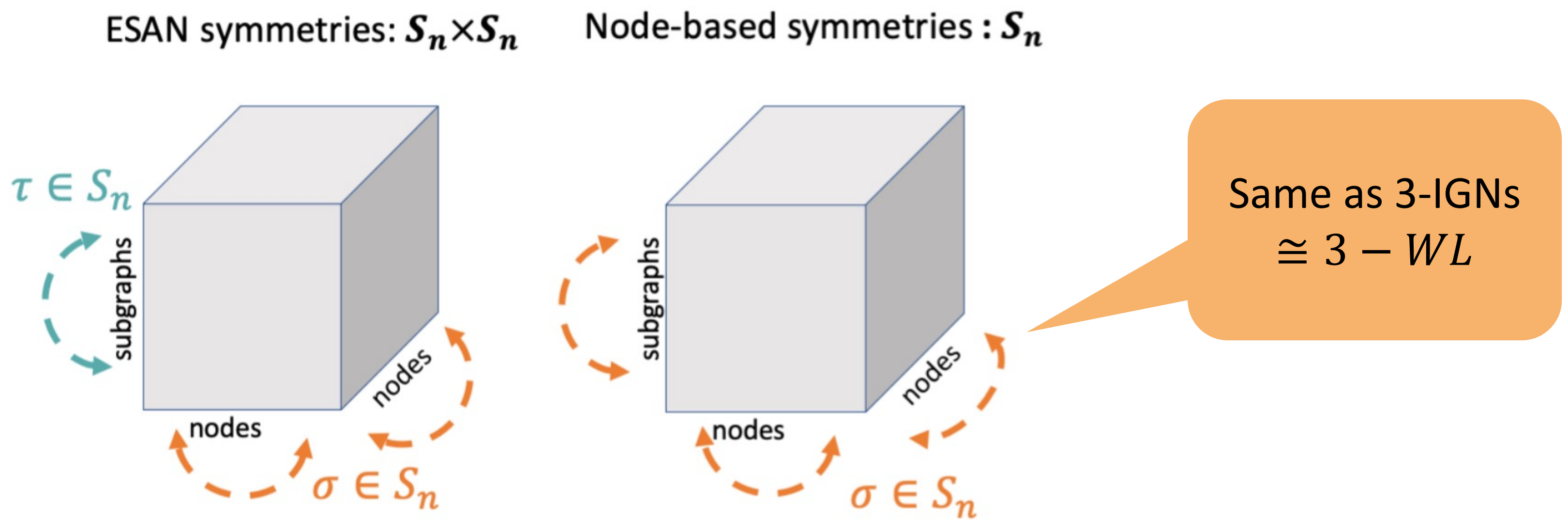
Symmetries of node-based policies

We represent the set of subgraphs as a 3-tensor



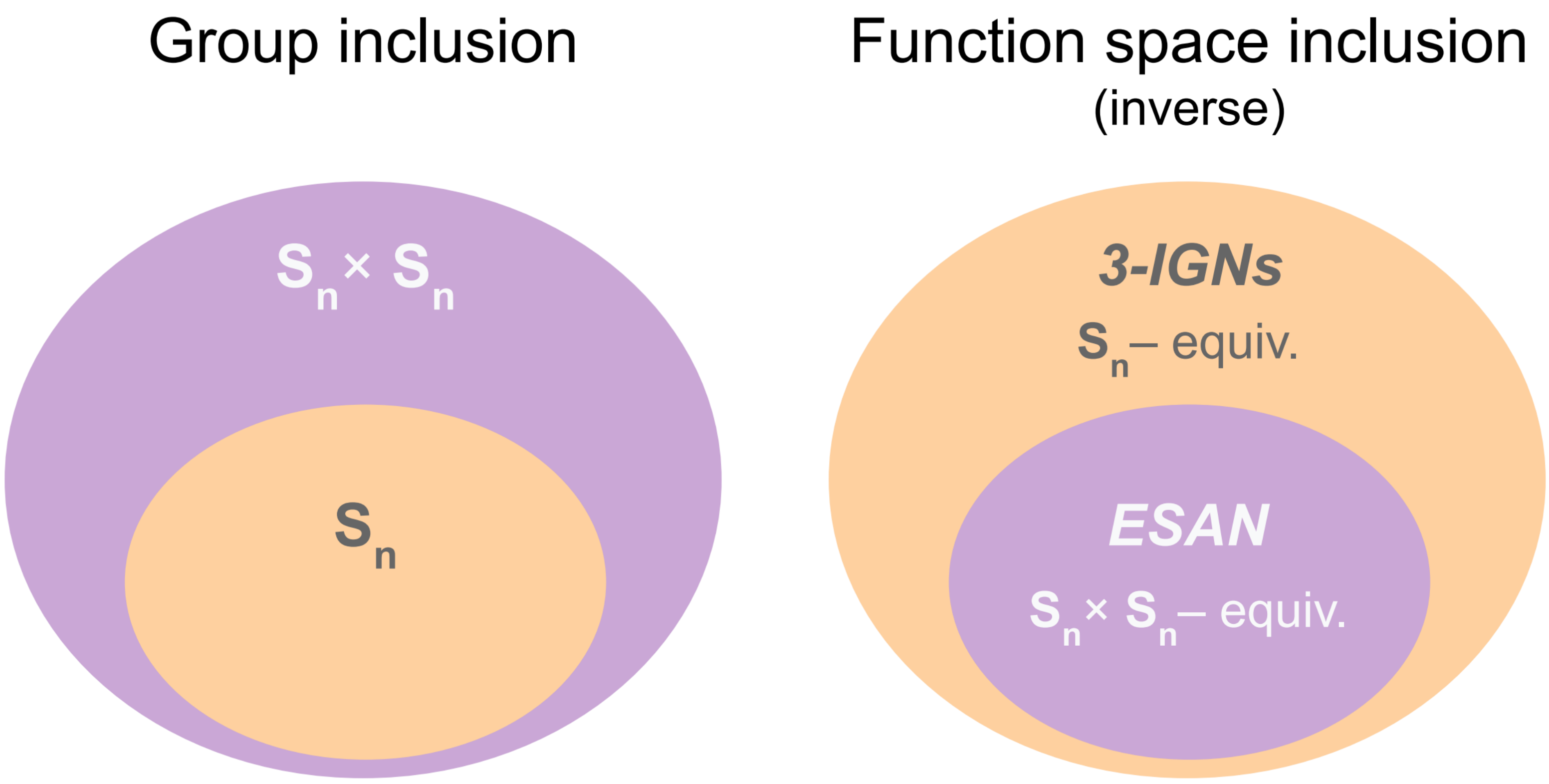
Symmetries of node-based policies

- **Observation:** subgraphs can be *ordered according to nodes*
- We can use a significantly smaller symmetry group compared to ESAN



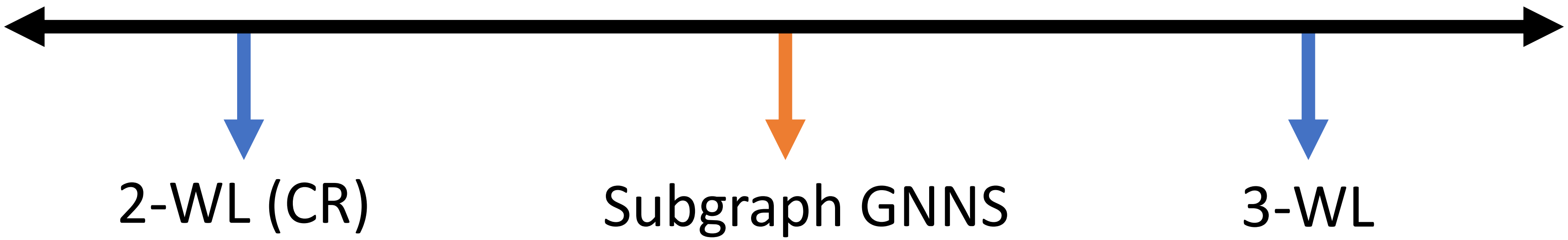
Symmetries of node-based policies

Outcome: the resulting equivariant function space is larger



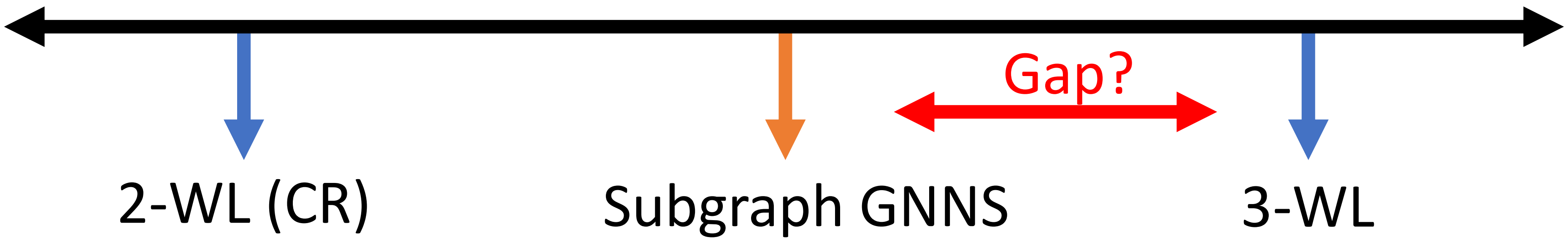
Results (1)

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



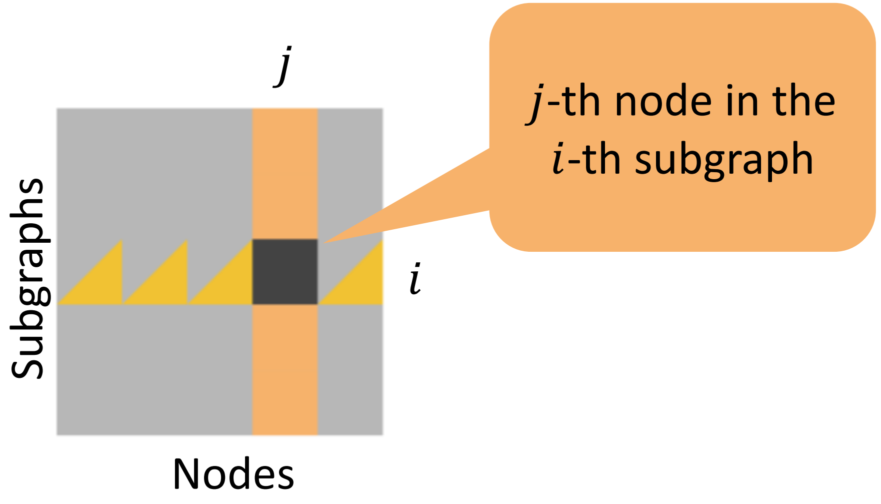
Results (1)

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



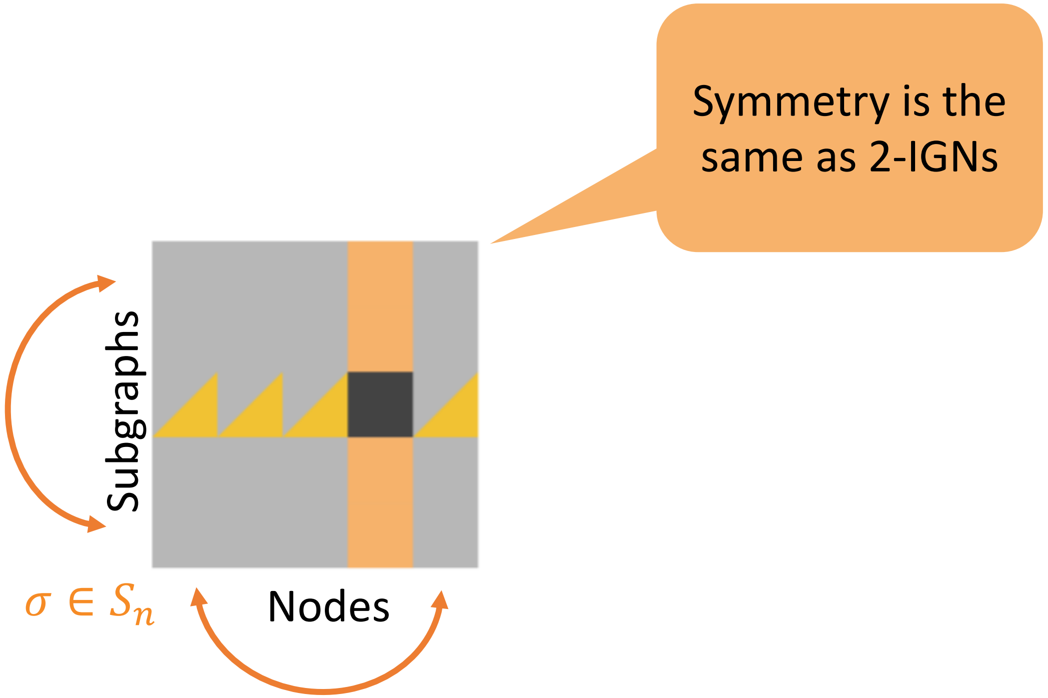
Results (2)

- A shared layer space inspired by 2-IGNs
  - Understand differences
  - Unify and extend architectures



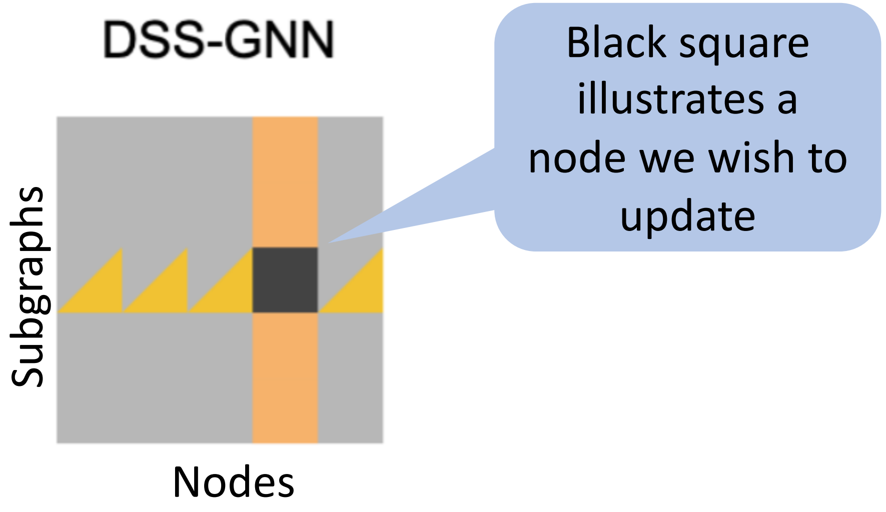
Results (2)

- A shared layer space inspired by 2-IGNs
  - Understand differences
  - Unify and extend architectures



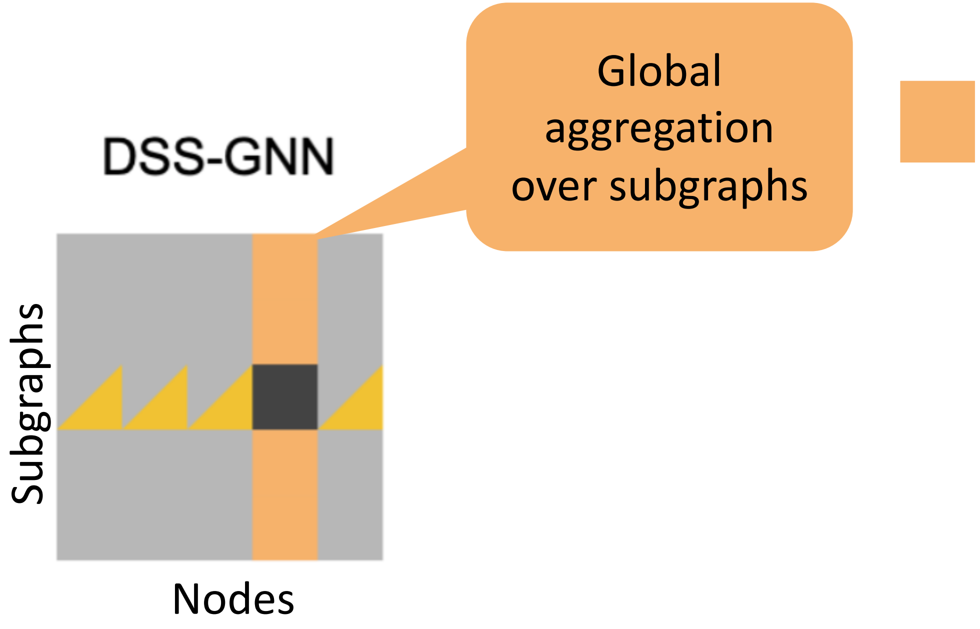
Results (2)

- A shared layer space inspired by 2-IGNs
  - Understand differences
  - Unify and extend architectures



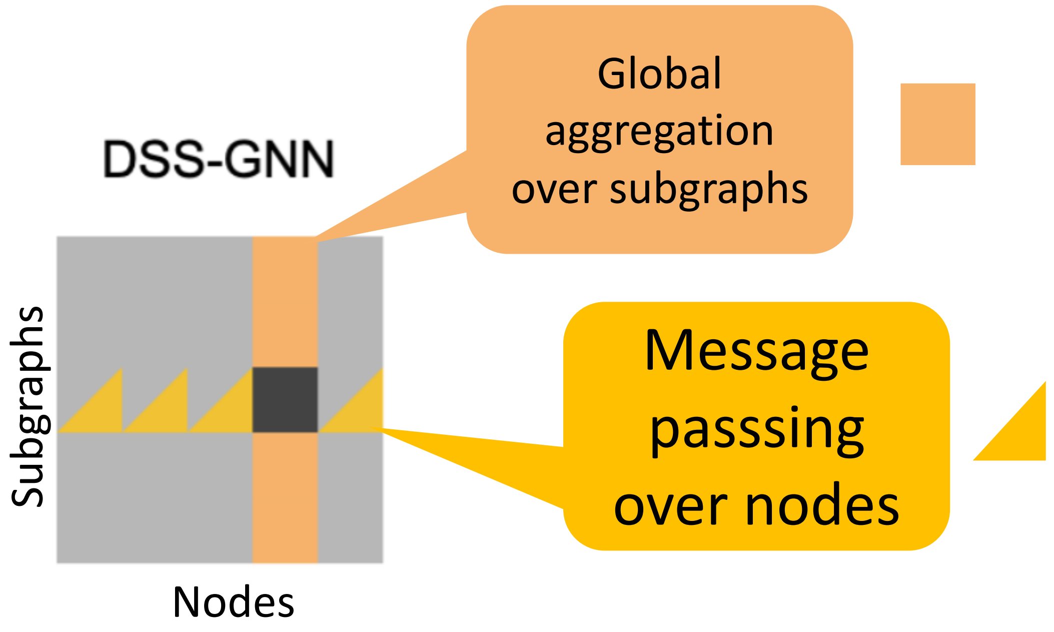
Results (2)

- A shared layer space inspired by 2-IGNs
  - Understand differences
  - Unify and extend architectures



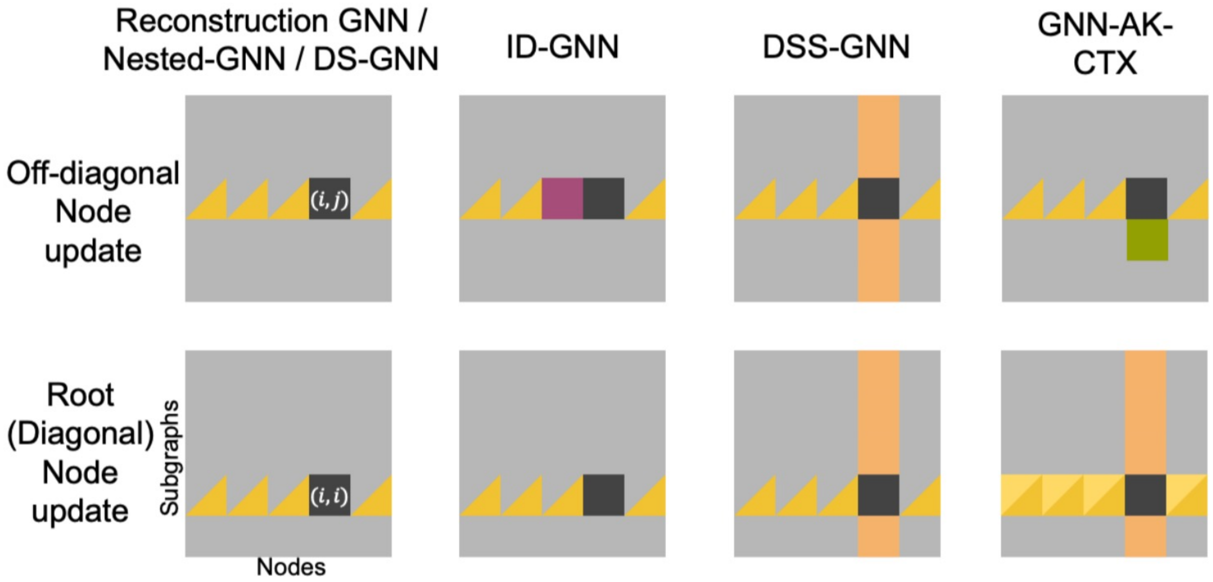
Results (2)

- A shared layer space inspired by 2-IGNs
  - Understand differences
  - Unify and extend architectures



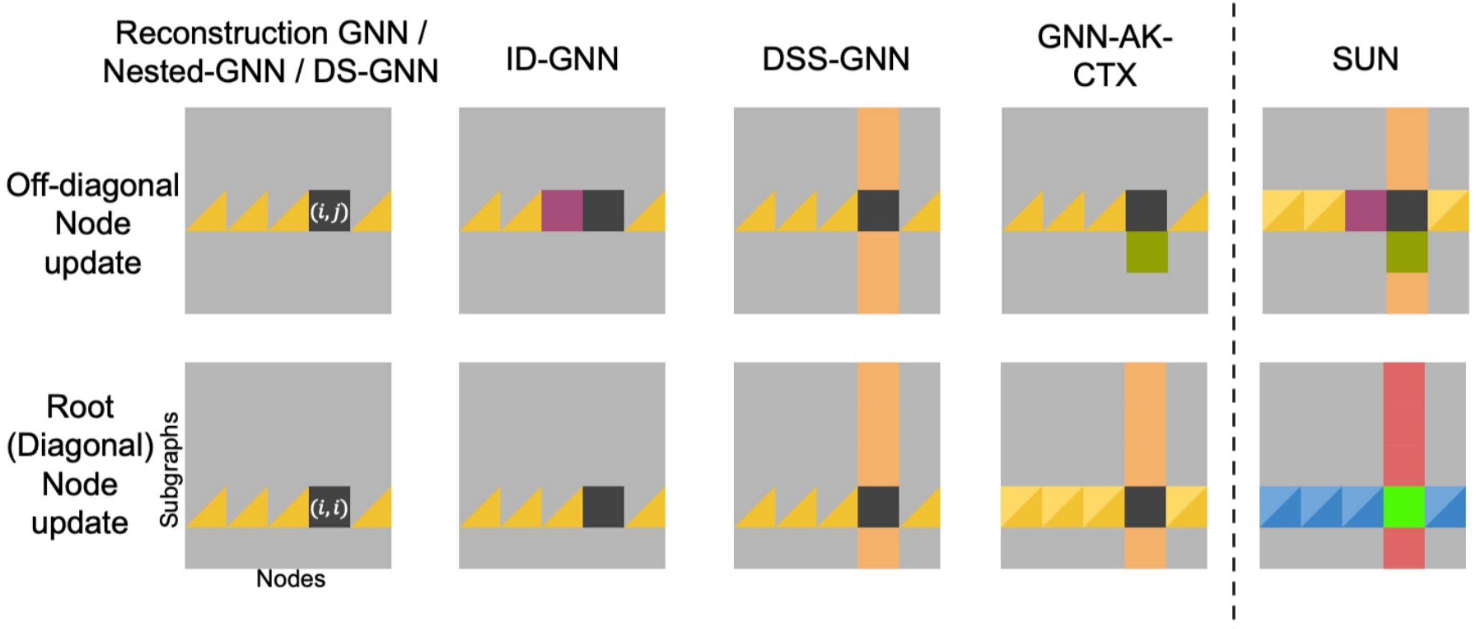
Results (2)

- A shared layer space inspired by 2-IGNs
  - Understand differences
  - Unify and extend architectures



Results (2)

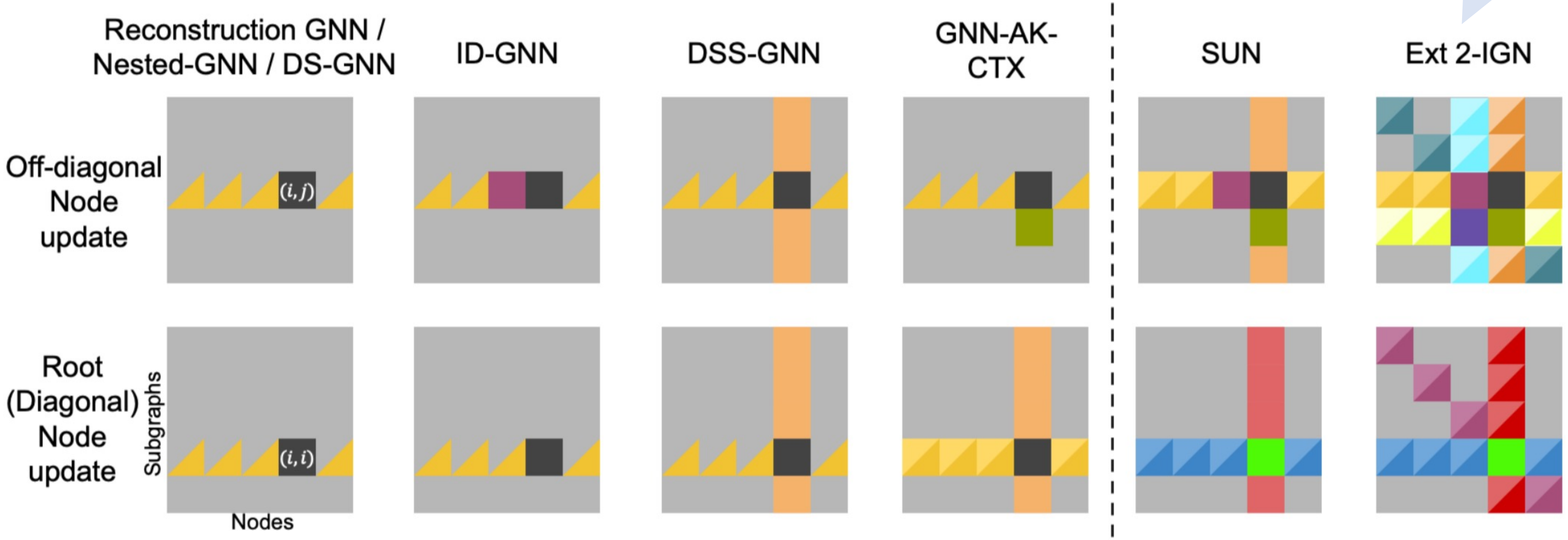
- A shared layer space inspired by 2-IGNs
  - Understand differences
  - Unify and extend architectures



Results (2)

- A shared layer space inspired by 2-IGNs
  - Understand differences
  - Unify and extend architectures

General layer space with Many possible layers to explore in the future



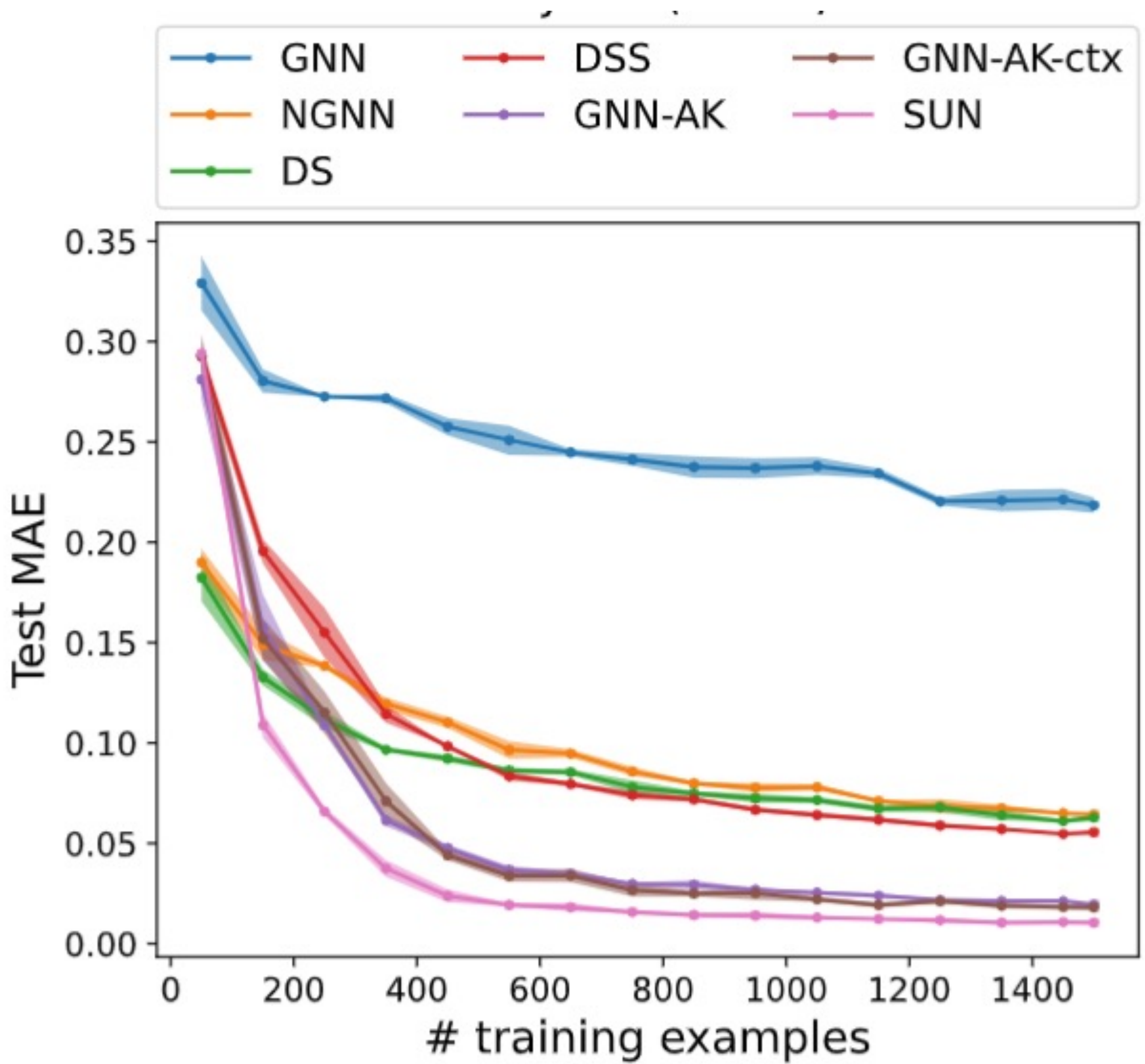
Experiments

- Better performance than previous methods on most benchmarks

Method	ZINC (MAE ↓)
GCN [26]	0.321 ± 0.009
GIN [51]	0.163 ± 0.004
PNA [13]	0.133 ± 0.011
GSN [11]	0.101 ± 0.010
CIN [9]	**0.079** ± 0.006
NGNN [53]	0.111 ± 0.003
DS-GNN (EGO) [7]	0.115 ± 0.004
DS-GNN (EGO+) [7]	0.105 ± 0.003
DSS-GNN (EGO) [7]	0.099 ± 0.003
DSS-GNN (EGO+) [7]	0.097 ± 0.006
GNN-AK [57]	0.105 ± 0.010
GNN-AK-CTX [57]	0.093 ± 0.002
GNN-AK+ [57]	0.086 ± ???
SUN (EGO)	0.083 ± 0.003
SUN (EGO+)	0.084 ± 0.002

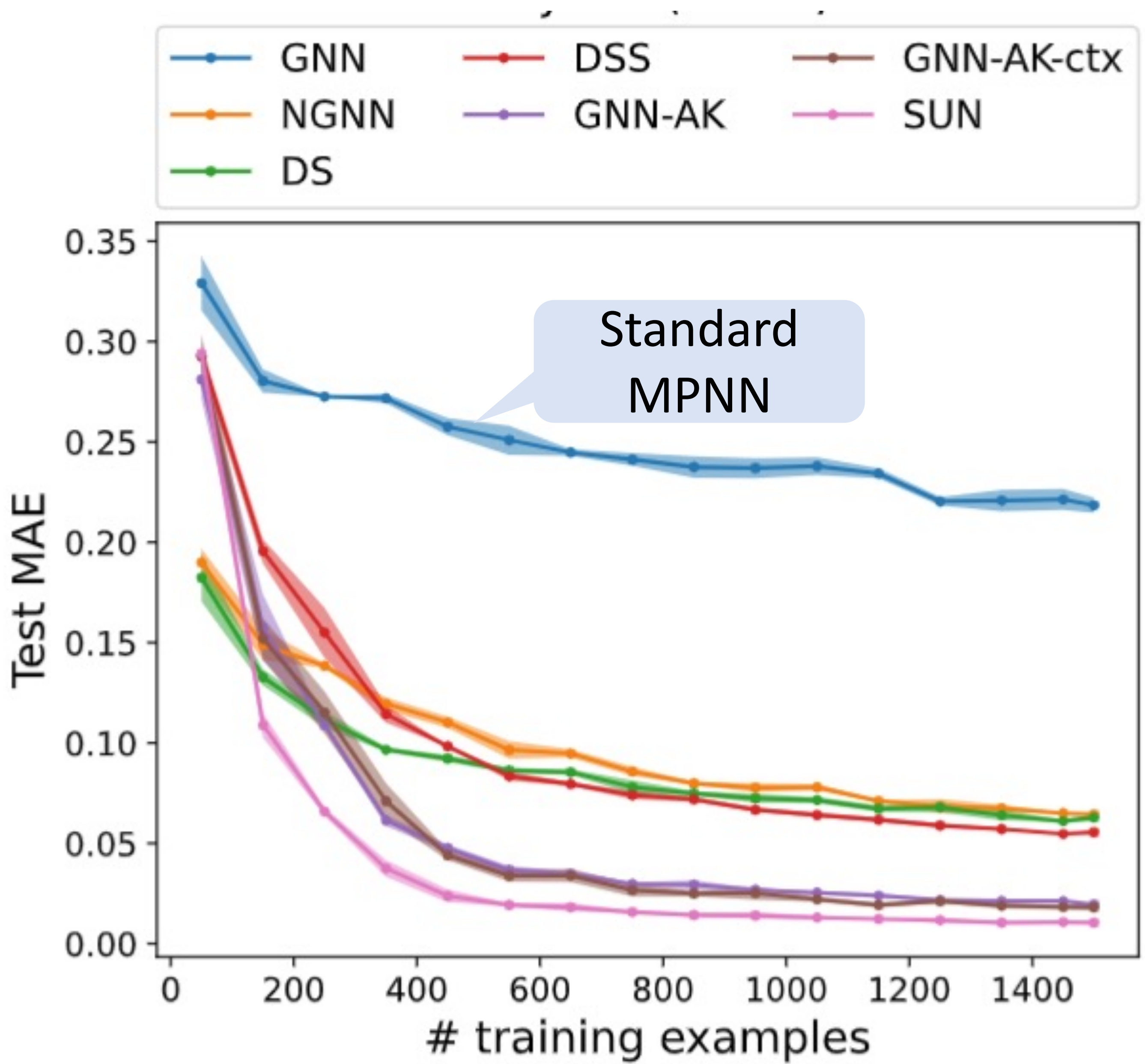
Experiments

- Adding sharing / aggregation operations improves generalization



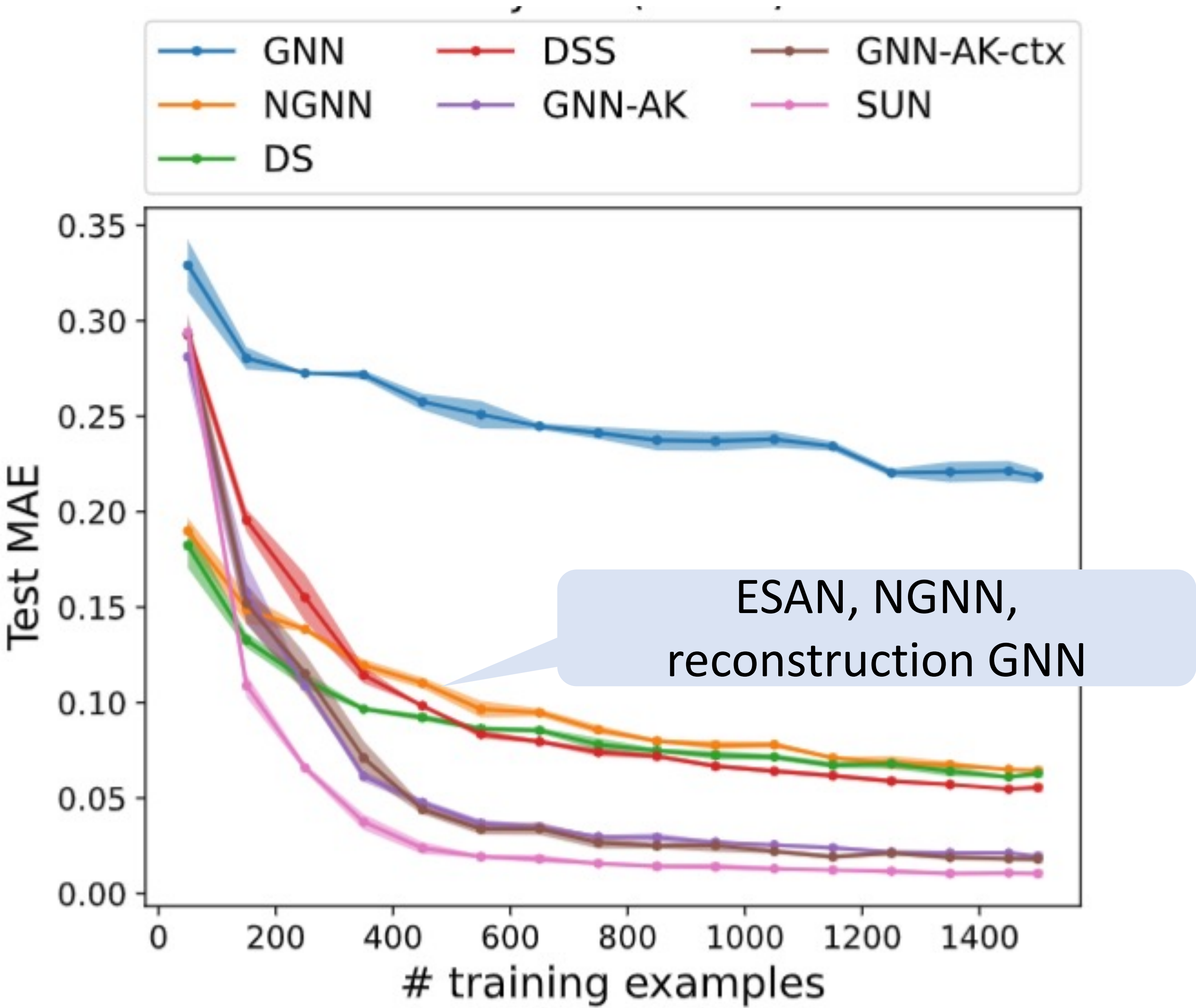
Experiments

- Adding sharing / aggregation operations improves generalization



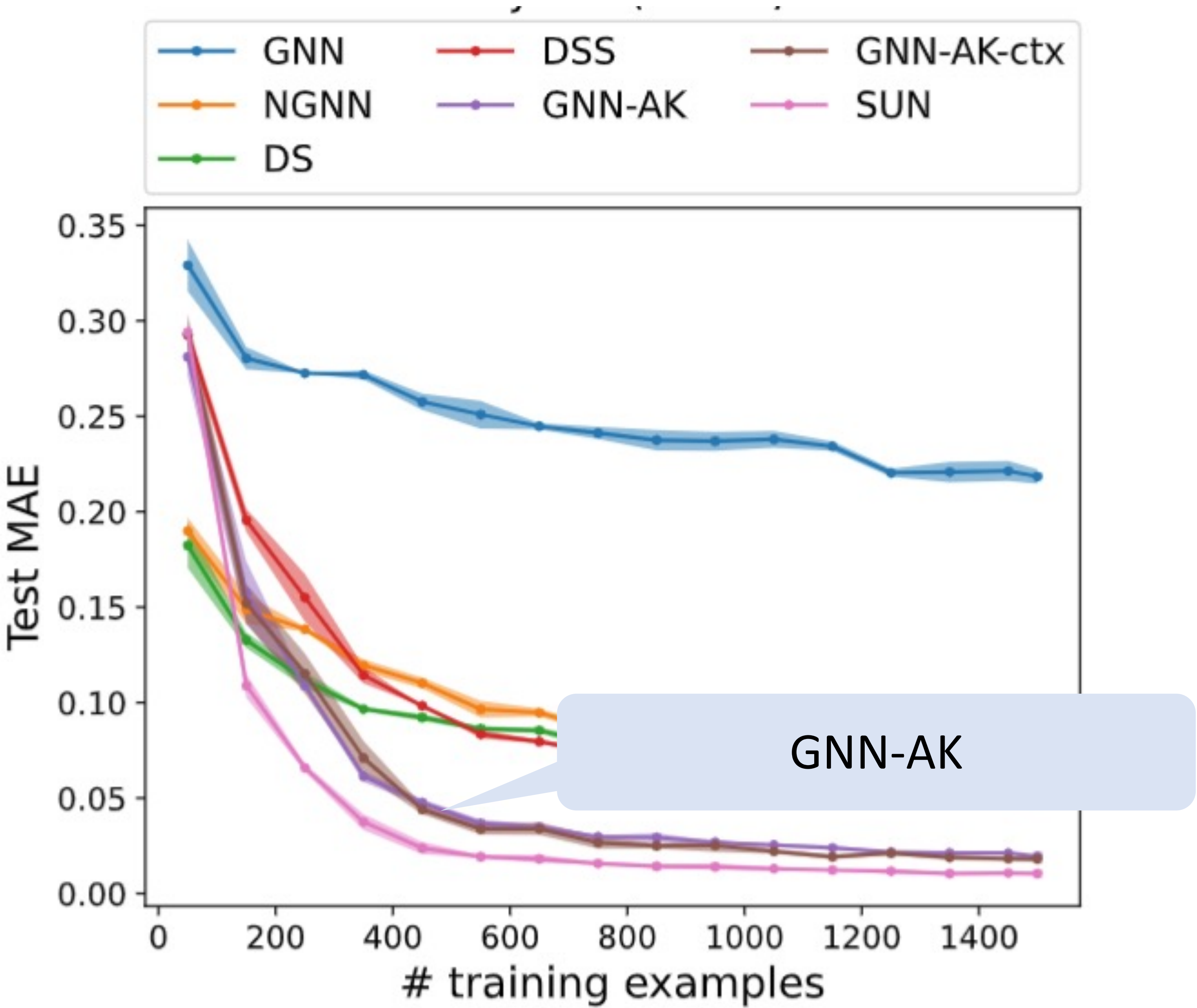
Experiments

- Adding sharing / aggregation operations improves generalization



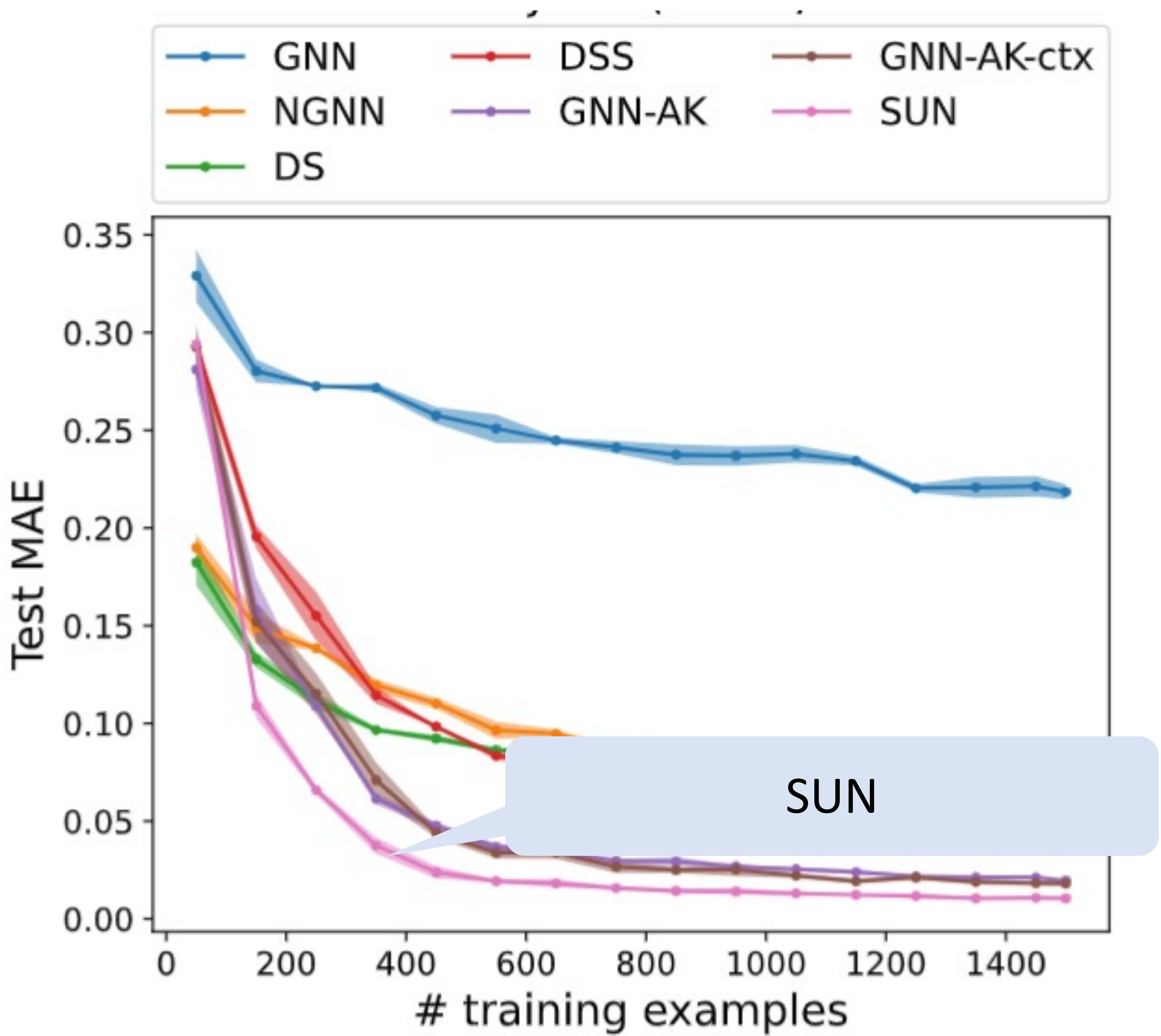
Experiments

- Adding sharing / aggregation operations improves generalization



Experiments

- Adding sharing / aggregation operations improves generalization



Take home messages:

- **GNN expressivity** is an interesting and important research direction

Take home messages:

- **GNN expressivity** is an interesting and important research direction
- **Subgraph GNNs** seem to strike a good balance between expressive power, generalization and computational complexity

Take home messages:

- **Symmetry analysis** is an *elegant* and *effective* way to design neural architectures according to the data they process.
- Meta-algorithm:
    1. Understand data symmetries
    2. Construct basic equivariant layers
    3. Use them to build invariant/equivariant network
    4. Understand expressive power

The end

Looking for PhD students and a postdoc for Oct. 2023!