Equivariant Subgraph Aggregation Networks

ICLR 2022

Department of Artificial Intelligence Korea University

Younghan Son

May 2022

KOREA UNIVERSITY MAI Lab EDICAL

MPNN is WL or less

- MPNN은 1-WL과 근사한다.

Published as a conference paper at ICLR 2019

How Powerful are Graph Neural Networks?

Keyulu Xu *†
MIT
keyulu@mit.edu

Weihua Hu *‡
Stanford University
weihuahu@stanford.edu

Jure Leskovec
Stanford University
jure@cs.stanford.edu

Stefanie Jegelka
MIT
stefje@mit.edu

Graph Isomorphism

- 두 그래프의 동형성을 확인
- if isomorphic, same connectivity, permutated nodes
- NP-hard 문제

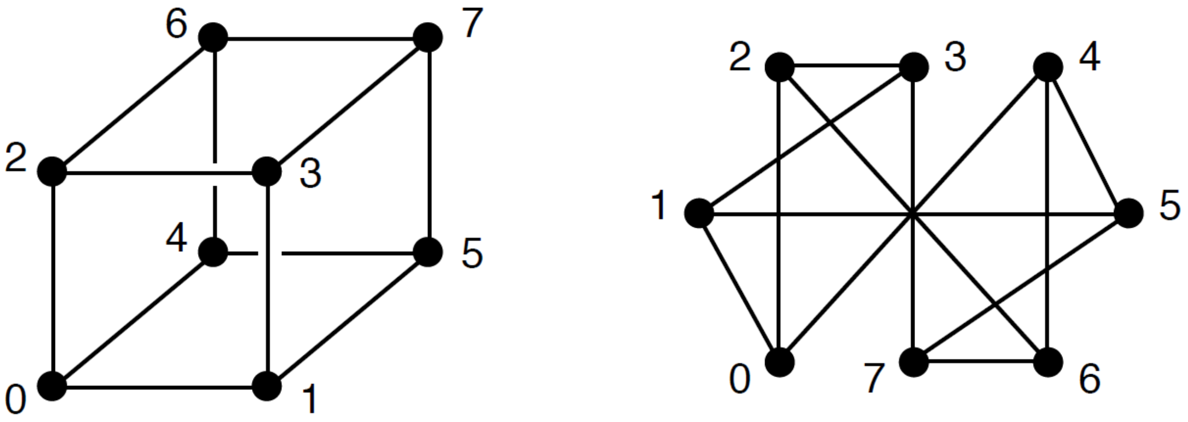


Figure 1.1: Two different drawings of the same graph.

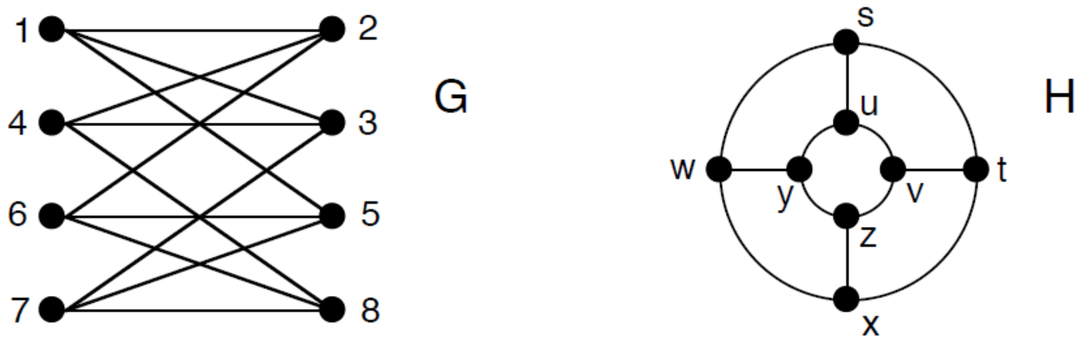


Figure 1.2: Two more drawings of that same graph.

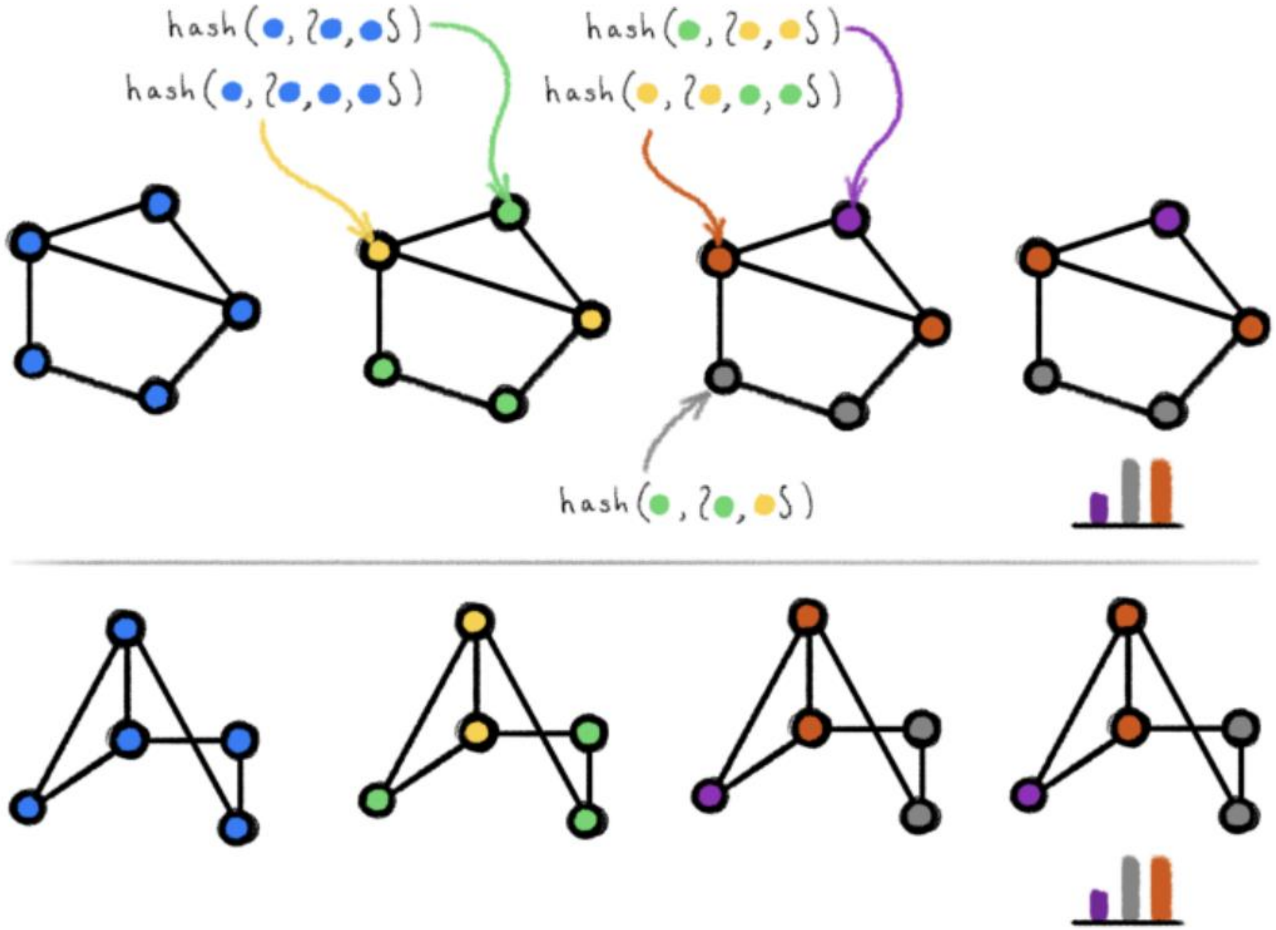
These two graphs are the "same" because, instead of having the same set of vertices, this time we have a bijection $V_G \to V_H$

$$
\begin{array}{cccc}
1 \to s & 2 \to t & 3 \to u & 4 \to v \\
5 \to w & 6 \to x & 7 \to y & 8 \to z
\end{array}
$$

WL algorithm

- Polynomial하게 계산가능
- counter example 존재, 대부분은 판별 가능
- Init: Same Color or Random color
- HASH(self, {neighbor set})
- 같은 색의 집합일때 같은 색깔

- 이형 그래프를 같다고 판단 할 수 있지만 다르다고 판단했을 때는 무조건 이형 그래프



medium - Expressive power of graph neural networks and the Weisfeiler-Lehman test

GIN, How powerful graph neural network is?

- MPNN이 어떻게 WL과 어떻게 근사 되는지 설명하는 논문
- Aggregate는 이웃들의 집합(neighborhood set)을 합침
- Combine은 HASH
- 각각 Aggregate와 Combine이 injective하다면 MPNN의 1 layer는 WL의 1 iteration과 같다.

$$h_v^{(k)} = \mathrm{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right)$$
$$= \mathrm{COMBINE}^{(k)}\left(h_v^{(k-1)}, \mathrm{AGGREGATE}^{(k)}\left(\left\{\!\!\left\{h_u^{(k-1)} : u \in N(v)\right\}\!\!\right\}\right)\right)$$

Theorem 3. *Let $\mathcal{A} : \mathcal{G} \to \mathbb{R}^d$ be a GNN. With a sufficient number of GNN layers, $\mathcal{A}$ maps any graphs G_1 and G_2 that the Weisfeiler-Lehman test of isomorphism decides as non-isomorphic, to different embeddings if the following conditions hold:*

a) $\mathcal{A}$ aggregates and updates node features iteratively with

$$h_v^{(k)} = \phi\left(h_v^{(k-1)}, f\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right)\right),$$

where the functions f, which operates on multisets, and ϕ are injective.

b) $\mathcal{A}$'s graph-level readout, which operates on the multiset of node features $\left\{h_v^{(k)}\right\}$, is injective.

GIN, How powerful graph neural network is?

- Combine은 UAT에 따라 MLP로 근사할 수 있다
- Aggregate는 Sum pooling으로 대체되었음 → 뒷 페이지

$$h_v^{(k)} = \mathrm{COMBINE}^{(k)}\left(h_v^{(k-1)}, a_v^{(k)}\right)$$
$$= \mathrm{COMBINE}^{(k)}\left(h_v^{(k-1)}, \mathrm{AGGREGATE}^{(k)}\left(\left\{\!\!\left\{h_u^{(k-1)} : u \in N(v)\right\}\!\!\right\}\right)\right)$$

universal approximation theorem

Lemma 5. *Assume $\mathcal{X}$ is countable. There exists a function $f : \mathcal{X} \to \mathbb{R}^n$ so that $h(X) = \sum_{x \in X} f(x)$ is unique for each multiset $X \subset \mathcal{X}$ of bounded size. Moreover, any multiset function g can be decomposed as $g(X) = \phi\left(\sum_{x \in X} f(x)\right)$ for some function ϕ.*

Corollary 6. *Assume $\mathcal{X}$ is countable. There exists a function $f : \mathcal{X} \to \mathbb{R}^n$ so that for infinitely many choices of ϵ, including all irrational numbers, $h(c, X) = (1+\epsilon) \cdot f(c) + \sum_{x \in X} f(x)$ is unique for each pair (c, X), where $c \in \mathcal{X}$ and $X \subset \mathcal{X}$ is a multiset of bounded size. Moreover, any function g over such pairs can be decomposed as $g(c, X) = \varphi\left((1+\epsilon) \cdot f(c) + \sum_{x \in X} f(x)\right)$ for some function φ.*

MAI Lab EDICAL

GIN, How powerful graph neural network is?

- Neighborhood aggregation에서 Sum, Max, Mean을 비교하는데 Injective 한 aggregation function을 찾아야함
- 아래 세가지 구조 모두 sum function은 구분 가능하지만 mean, max는 구분하지 못함
- Aggregation은 injective한 sum function으로 대체 가능

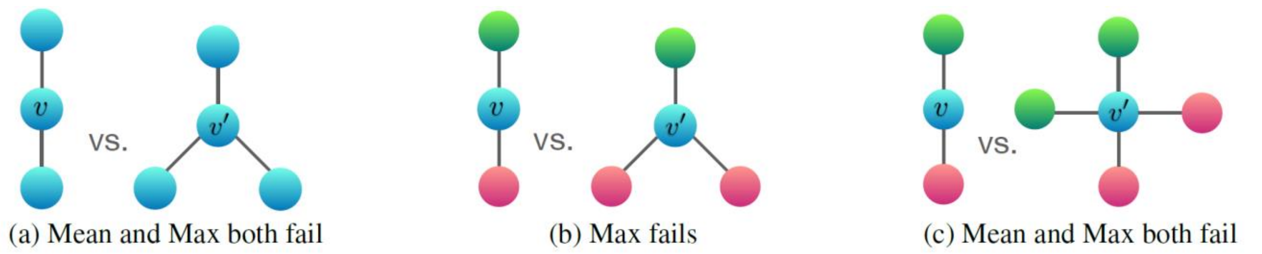


Figure 3: **Examples of graph structures that mean and max aggregators fail to distinguish.** Between the two graphs, nodes v and v' get the same embedding even though their corresponding graph structures differ. Figure 2 gives reasoning about how different aggregators "compress" different multisets and thus fail to distinguish them.

GIN, How powerful graph neural network is?

- MPNN의 1 layer는 WL의 1 iteration과 같다
- MPNN의 depth가 깊어질수록 node representation은 local → global로 가므로 모든 구조를 같이 판단하기 위해 모든 layer의 readout을 concat해서 사용한다(Inspired by JKNet).

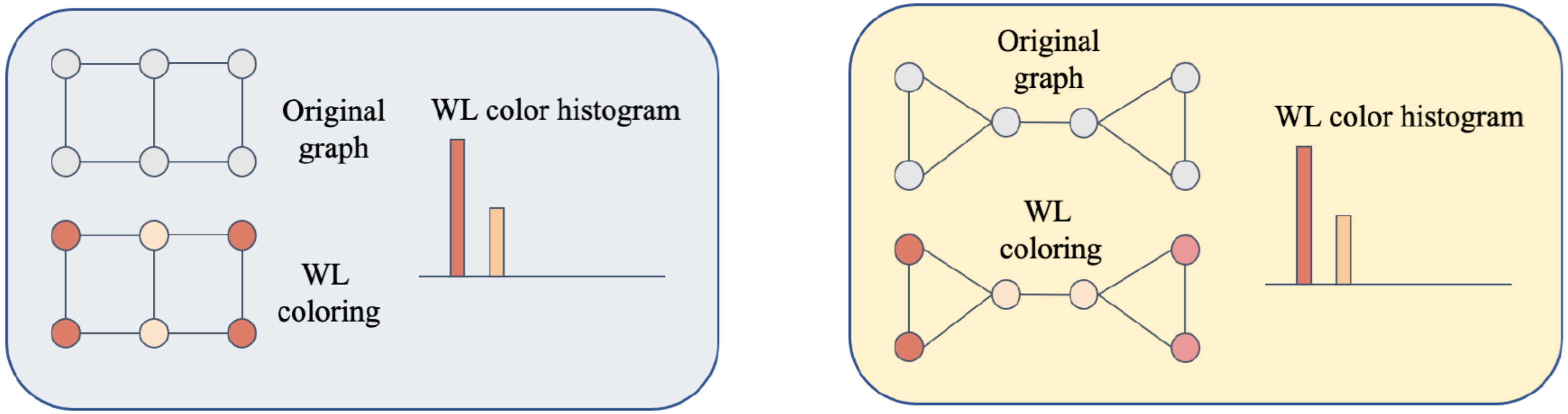
Networks (Xu et al., 2018), where we replace Eq. 2.4 with graph representations concatenated across *all iterations/layers* of GIN:

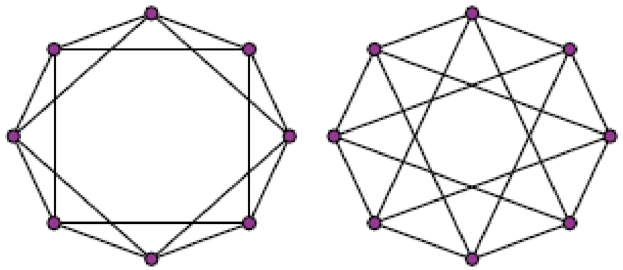
$$h_G = \mathrm{CONCAT}\Big(\mathrm{READOUT}\Big(\Big\{h_v^{(k)} | v \in G\Big\}\Big) \mid k = 0, 1, \ldots, K\Big). \tag{4.2}$$

MPNN(1-WL)의 한계

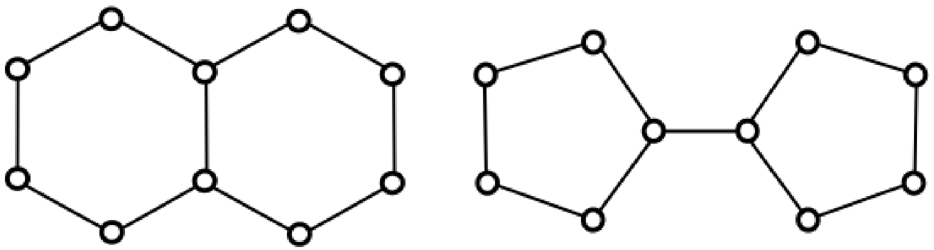
- 모델이 판별할 수 있는 그래프가 많을 때 모델의 Expressivity가 높다고 표현한다.(GNN expressivity)
- 1-WL이 판별 불가능한 simple graph다수 → GNN Expressivity를 높일 수 있는 방법을 고안하자

WL-indistiguishable non-isomorphic graphs

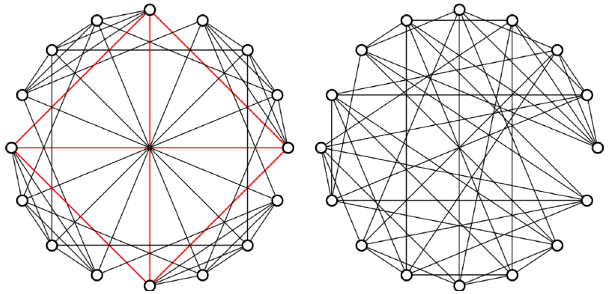




Circulant Skip Link Graph(CSL)
CSL(8,2) and CSL(8,3)



Decalin and Bicyclopentyl



Rook's 4x4 graph and Simple Regular Graph(SR)
SR(16,6,2,2)

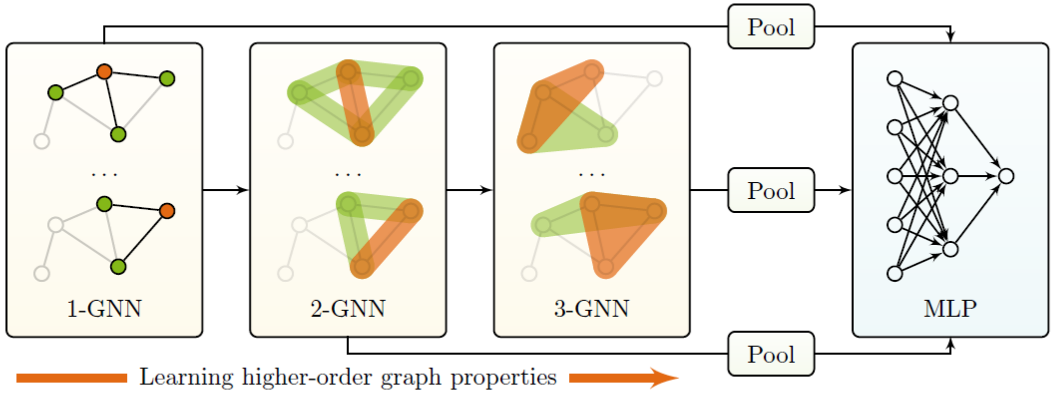
Equivariant Subgraph Aggregation Networks - ICLR 2022.pdf
Improving Graph Neural Network Expressivity via Subgraph Isomorphism Counting - TPAMI 2022

Enhancing GNN Expressivity

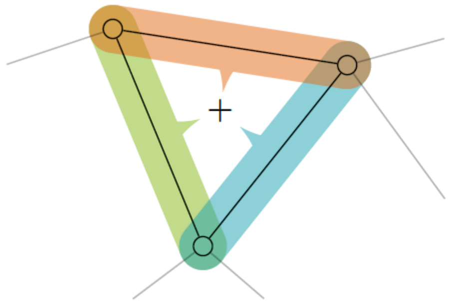
- (1) Aligning to the k-WL hierarchy (Morris et al., 2019; 2020b; Maron et al., 2019b;a);
- (2) Augmenting node features with exogenous identifiers (Sato et al., 2021; Dasoulas et al., 2020; Abboud et al., 2020);
- (3) Leveraging on structural information that cannot provably be captured by the WL test (Bouritsas et al., 2022; Thiede et al., 2021; de Haan et al., 2020; Bodnaret al., 2021b;a)

(1) k-WL hierarchy

- k-WL 방법은 노드를 k개씩 묶어서 HASH 하는 방법이다.
- 즉 k개의 노드를 묶고 해당 노드들의 이웃들을 전부다 HASH하는 것
- Pros: High expressivity
- Cons: Computation cost, 1-WL에 비해 성능이 크게 오르지 않음



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



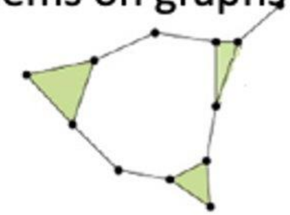
(b) Pooling from 2- to 3-GNN.

(1) k-WL hierarchy

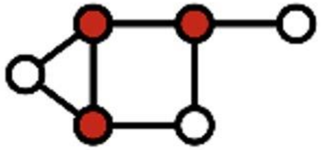
1-WL GNN (MPNN) versus k-WL GNN

- Pros:
  - Linear memory&time complexity.
  - Local update schema.
  - Natural problems consist of graphs can be distinguishable by 1-WL.
  - Their results are still competitive!
- Cons:
  - Maps 1-WL equivalent graphs to the exact the same point on latent space.
  - Cannot count some substructures that is informative many graph problems.
  - Cannot solve many combinatorial problems on graphs that may needed.

- Pros:
  - Can distinguish up to k-WL equivalent graphs.
  - Can count some substructures related to k.
  - Can solve some combinatorial problems.
- Cons:
  - O(n^(k-1)) memory, O(n^k) CPU time
  - Non-local update schema.
  - Unable to learn frequency relations.
  - Their results are not better than 1-WL GNN on many realistic problems.



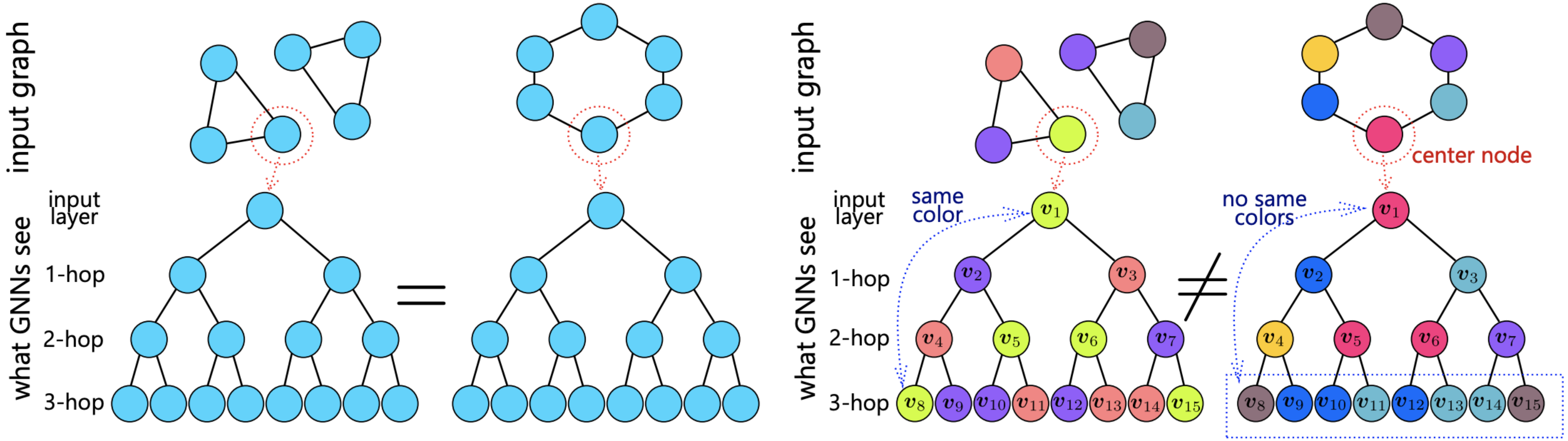
Counting triangles



Minimum vertex covering

(2) Augmenting node features with exogenous identifiers

- WL의 Random coloring methods 에서 영감 받은 방법



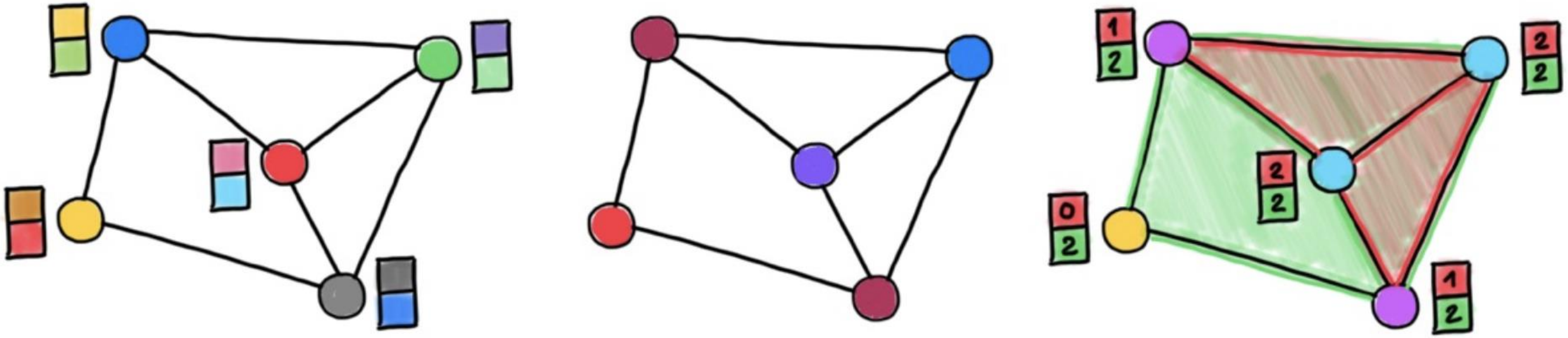
(a) Identical Features. (b) Random Features.

Figure 1: Illustrative example: GNNs with identical features (such as degree features) cannot distinguish a node in a cycle of three nodes with a node in a cycle of six nodes, whereas GNNs with random features can.

Random features strengthen graph neural networks - SDM 2021

(3) Leveraging on structural information that cannot captured by the WL test

- 각 노드에 위치정보(coloring in WL)을 부여하는 것
- Transformer의 Positional Encoding 에서 Inspired됨



Positional encoding examples. Shown left-to-right: random features, Laplacian eigenvectors (analogy of sinusoids used in Transformers), structural features (counts of triangles and rectangles).

Michael Bronstein Medium Graph Neural Networks beyond Weisfeiler-Lehman and vanilla Message Passing

Leveraging on structural information

- 각 노드와 엣지에 clique, triangle, cycle 정보(structural information)을 넣어줌
- 각 노드 별 subgraph counting을 사용해야함
- 어떤 subgraph를 사용할지 결정해야함

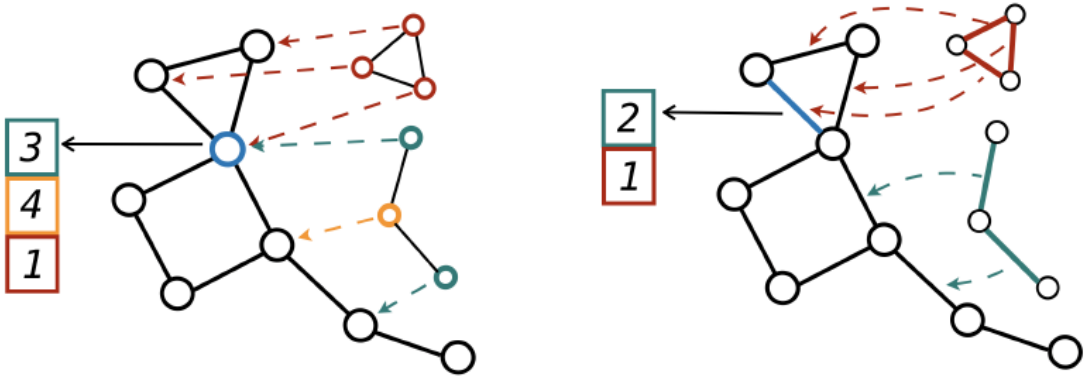
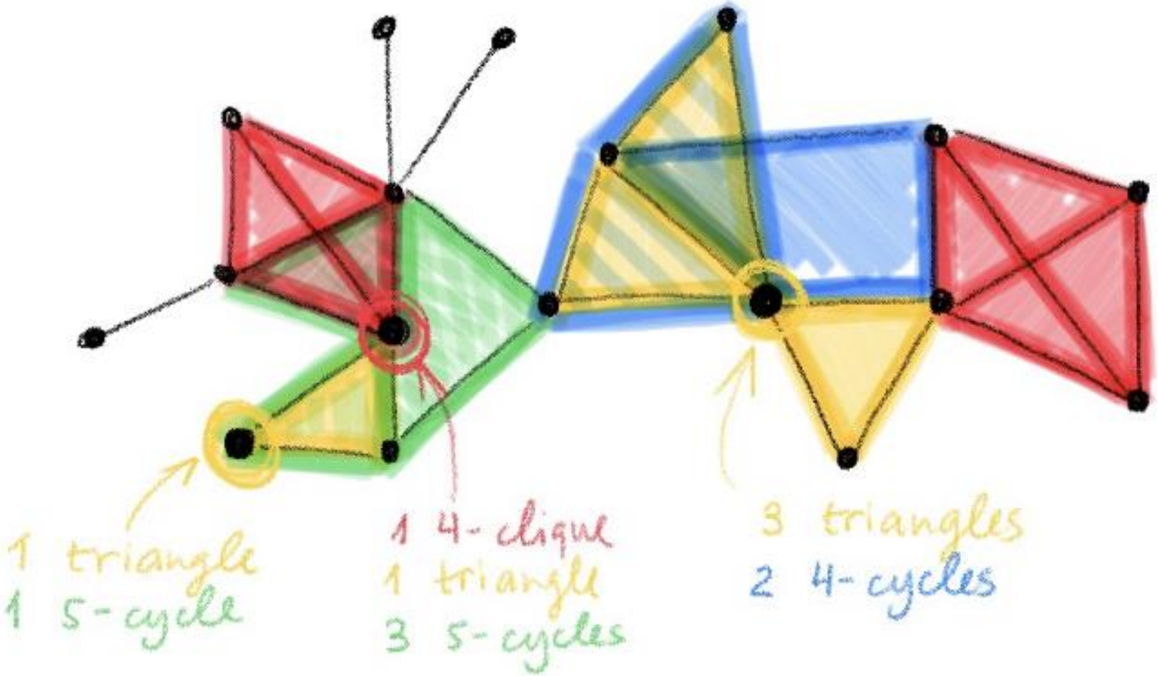
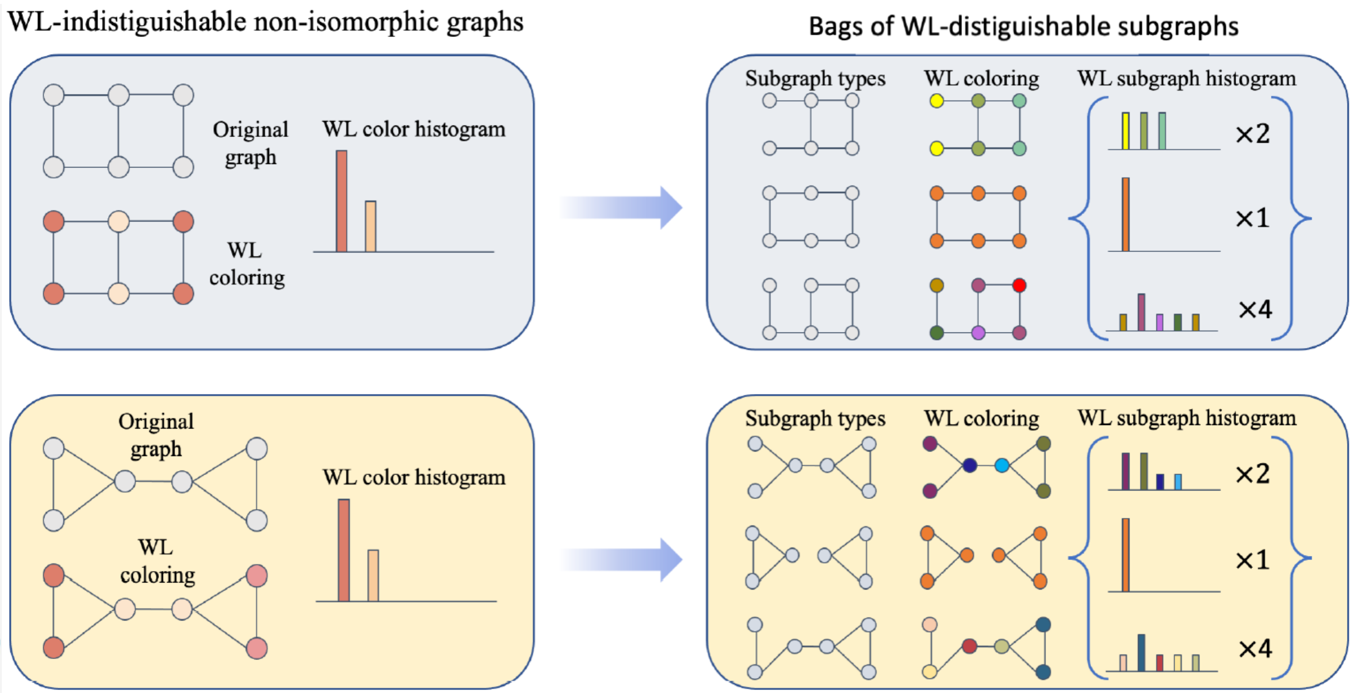


Fig. 1: *Vertex* (left) and *edge* (right) induced subgraph counting for a 3-cycle and a 3-path. Counts are reported for the blue vertex on the left and for the blue edge on the right. Different colors depict orbits.



왜 bag of subgraph인가?

- 1) k-WL은 졸라 비쌈
- 2) Node augmentation은 Equivariance가 안됨(Random coloring)
- 3) subgraph counting도 졸라 비쌈…
- 위 방법들의 문제점은 WL이든 subgraph든 각 노드마다 counting 을 수행해야 한다는 점이다.
- → 각각의 그래프를 subgraph 들의 집합으로 표현하면 노드마다 계산하지 않아도 됨
- bag of subgraph → 집합 데이터들은 equivariance를 충족하는가?



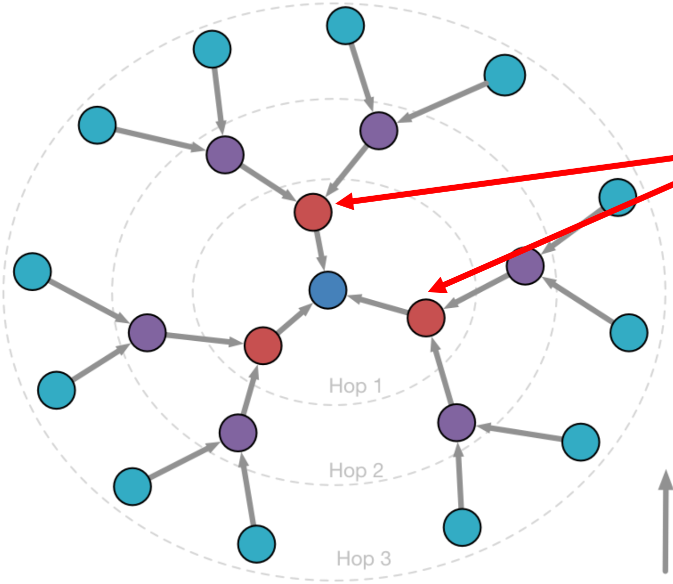
Equivariance

- GNN은 노드의 순서가 없으므로 Equivariance or Invariant가 보장 되어야함
- 즉 노드의 순서가 바뀌어도(permutation) 결과가 같아야 함

$$f(h \cdot x) = h \cdot f(x) \qquad\qquad f(h \cdot x) = f(x)$$

Permutation

Equivariant　　　　　　　　　　　　Invariant

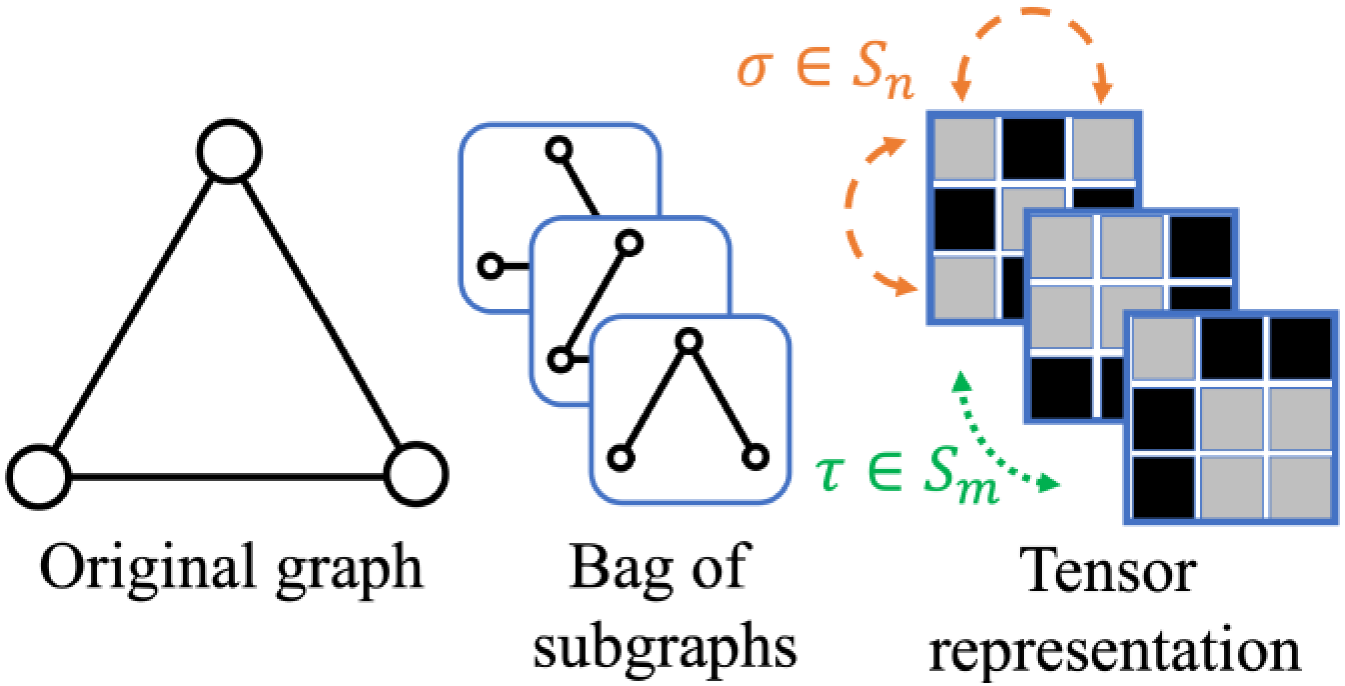


No order

Graph Convolution

Equivariance

- 오른쪽처럼 node permutation and graph permutation 해도 괜찮다



Deep Set(DS)

- 집합 데이터를 다루는 모델
- 집합에는 순서가 없음 --> 순서에 영향을 받지 않아야 하는 Permutation Invariant / Equivariant 가 자연스럽게 중요해짐
- 그래프의 노드 역시 순서가 없음 --> 똑같이 Permutation Equivariant 해야함

Definition

f is **invariant** if $f(h \cdot x) = f(x)$ for all $h \in H$.
f is **equivariant** if $f(h \cdot x) = h \cdot f(x)$ for all $h \in H$.

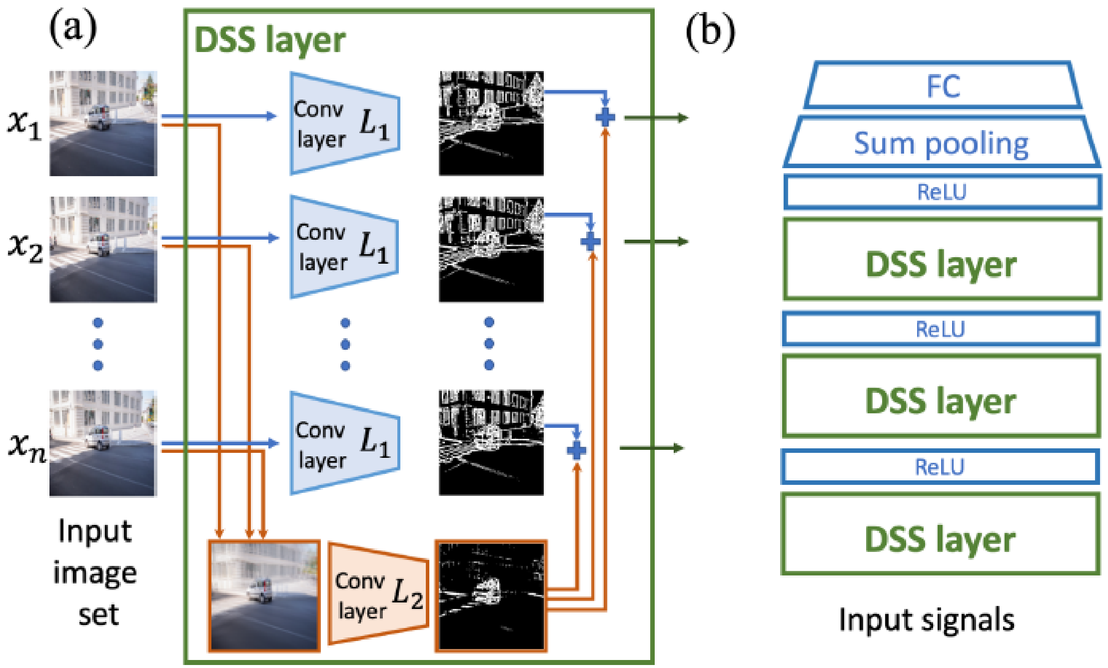
$$\Theta = \underbrace{\lambda \mathbf{I}}_{\text{자기자신}} + \underbrace{\gamma\,(\mathbf{1}\mathbf{1}^{\mathsf{T}})}_{\text{집합}} \qquad \underbrace{\lambda, \gamma \in \mathbb{R}}_{\text{matrix}} \quad \mathbf{1} = [1, \ldots, 1]^{\mathsf{T}} \in \mathbb{R}^{M} \qquad \mathbf{I} \in \mathbb{R}^{M \times M} \textit{ is the identity matrix}$$

$$x_i^t = W_1^t x_i^{t-1} + W_2^t \sum_{j=1}^{n} x_j^{t-1} + b^t$$

누가 봐도 GNN과 굉장히 유사

Deep Sets for Symmetric elements(DSS)

- Each element has its own symmetries(=Translations)
- 각 집합 원소를 embedding하고 Sum pooling하면 집합 순서에 상관없음
- 똑같이 UAT를 사용하여 L을 CNN으로 사용함 → 이 논문에서는 MPNN
- Deep Set에서 Information sharing layer 를 추가한 논문



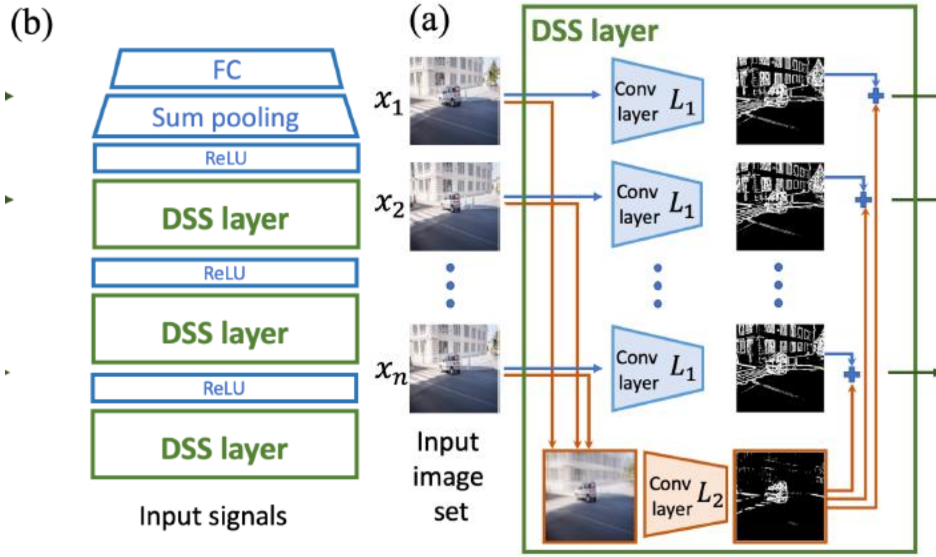
Theorem 1. *Any linear G−equivariant layer $L : \mathbb{R}^{n\times d} \to \mathbb{R}^{n\times d}$ is of the form*

$$L(X)_i = \underbrace{L_1^H(x_i) +}_{\text{Siamese network}} \underbrace{L_2^H\left(\sum_{j\neq i}^{n} x_j\right)}_{\text{Information sharing}},$$

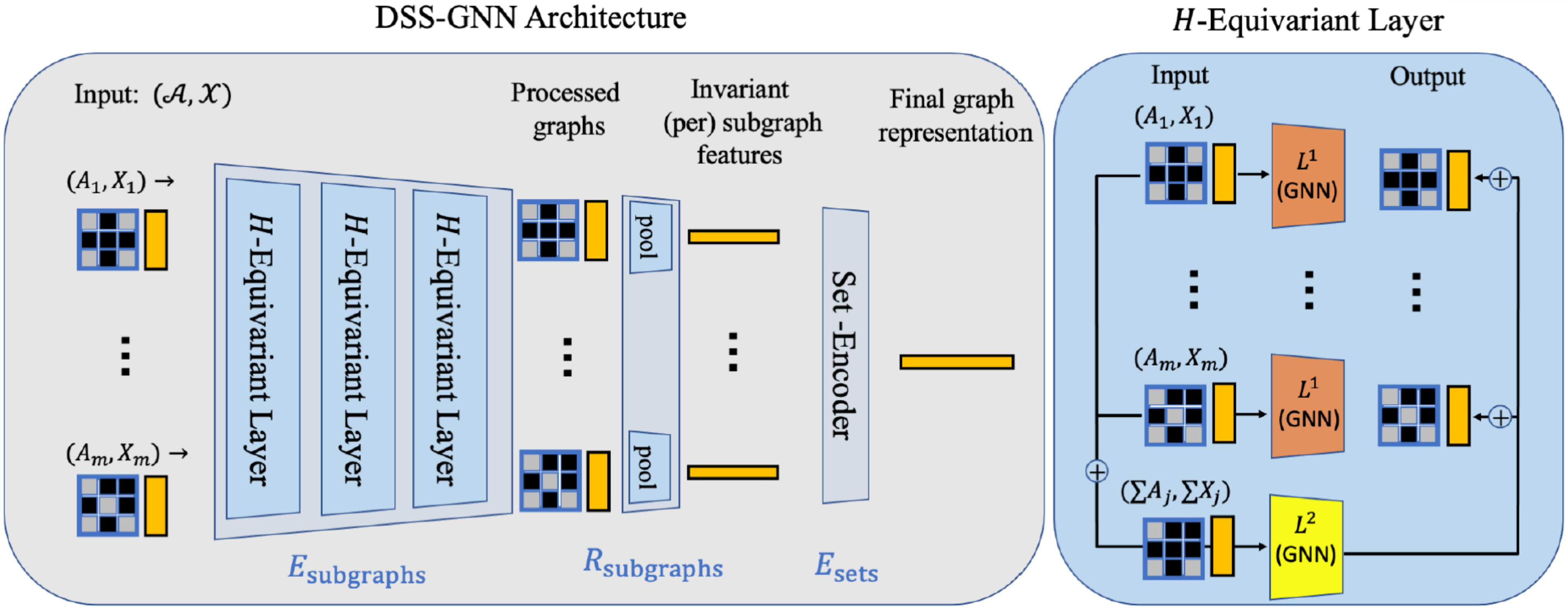
where L_1^H, L_2^H are linear H-equivariant functions

MAI Lab EDICAL

DSS-GNN



$$(L(\mathcal{A},\mathcal{X}))_i = \underbrace{L^1(A_i, X_i)}_{\text{Siamese network}} + \underbrace{L^2\left(\sum_{j=1}^m A_j, \sum_{j=1}^m X_j\right)}_{\text{Information sharing}}$$



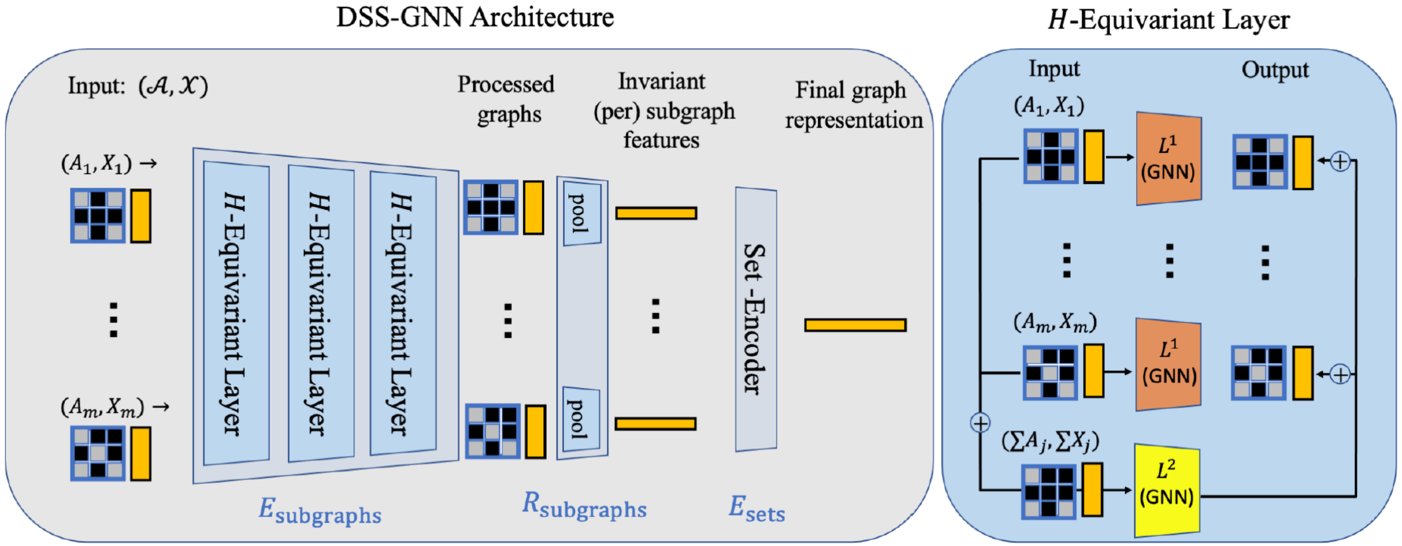
MAI Lab EDICAL

DSS-GNN

- $E_{\text{subgraphs}}$ Equivariant layer
- $R_{\text{subgraphs}}$ Sum pooling layer
- E_{sets} Sum pooling or max pooling
- Information sharing 이 없을 때 DS-GNN(Deep Set과 같음)

$$F_{\text{DSS-GNN}} = E_{\text{sets}} \circ R_{\text{subgraphs}} \circ E_{\text{subgraphs}}$$

$$(L(\mathcal{A}, \mathcal{X}))_i = L^1(A_i, X_i) + L^2\left(\sum_{j=1}^m A_j, \sum_{j=1}^m X_j\right)$$



Subgraph selection policy

- ND: 노드를 하나씩 제거
- ED: 엣지를 하나씩 제거
- EGO: Ego network
- EGO+: Ego network variants(root node holds an identifying feature???)

Ego networks

- Ego-network란 자기 자신과 연결된 네트워크만을 표현하는 것이다.
- K-Ego-networks 란 각 노드(ego) 의 k-hop network를 표현한 것이다.

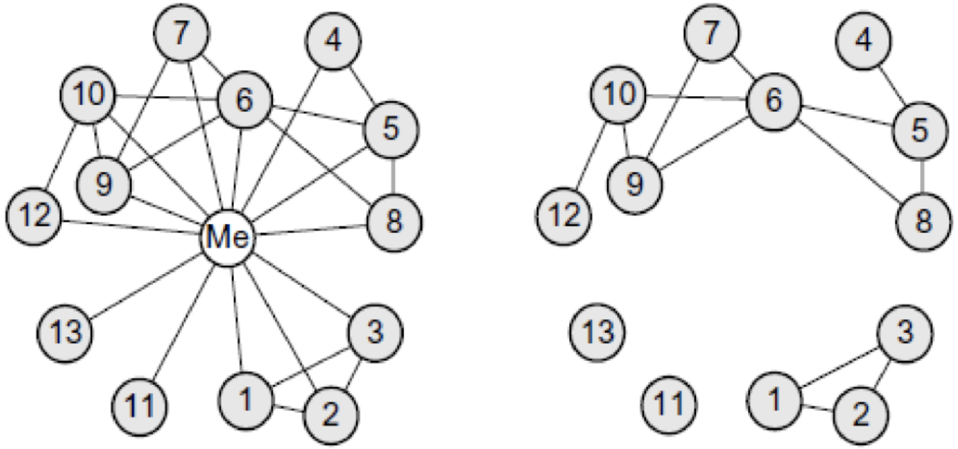


그림 3. 관계를 명확하기 위한 나 빼기

Fig. 3. The same eog network with and without ego. The second visualization (excluding ego) show the structure of the network more clearly

Stochastic sampling

- In large graphs, We cannot use all subgraph bags

$$|\overline{S}_G^{\pi}|/|S_G^{\pi}| \in \{0.05, 0.2, 0.5\}$$

- 미리 모든 subgraph bag를 만들어 놓고 뽑아서 사용하겠다

- Inference: L(5)개의 subgraph bag set를 뽑아서 majority voting을 수행하였다

GNN Expressivity 비교(WL test)

- (1) 기존 WL 방법이 할 수 있는걸 할 수 있는지
- (2) WL이 안되는 걸 우리가 할 수 있는지

자기 자신 이웃들의 집합 Subgraph bag에서 이웃들의 집합

- 기존 WL $\mathrm{HASH}(c^t_{v,S}, N^t_{v,S}, \cancel{C^t_v, M^t_v})$

Subgraph bag에서 자기 자신들의 집합

- 현 모델 $\mathrm{HASH}(c^t_{v,S}, N^t_{v,S}, C^t_v, M^t_v)$

- DSS-WL은 (1) (2) 둘 다 만족

GNN Expressivity 비교(WL test)

- DSS-WL이 잘하는 건 알겠는데, DSS-GNN이 DSS-WL 만큼 잘함?
- 실험 해보니 잘됨 + WL이 할 수 있는 건 GNN이 똑같이 할 수 있다
- EXP, CEXP, CSL: WL이 잘 못하는 그래프 데이터 모음

Table 6: Results for RNI datasets. Gray background indicates that ESAN outperforms the base encoder. DS/DSS-GNN can boost the performance of both base graph encoders, GIN and GraphConv, from random to perfect.

	EXP	CEXP
GIN (Xu et al., 2019)	51.1±2.1	70.2±4.1
GIN + ID-GNN (You et al., 2021)	100±0.0	100±0.0
DS-GNN (GIN) (ED/ND/EGO/EGO+)	100±0.0	100±0.0
DSS-GNN (GIN) (ED/ND/EGO/EGO+)	100±0.0	100±0.0
GraphConv (Morris et al., 2019)	50.3±2.6	72.9±3.6
GraphConv + ID-GNN (You et al., 2021)	100±0.0	100±0.0
DS-GNN (GraphConv) (ED/ND/EGO/EGO+)	100±0.0	100±0.0
DSS-GNN (GraphConv) (ED/ND/EGO/EGO+)	100±0.0	100±0.0

Results

- EGO network가 잘되는듯

Table 1: TUDatasets. Gray background indicates that ESAN outperforms the base encoder. SoTA line reports results for the best performing model for each dataset.

Method ↓ / Dataset →	MUTAG	PTC	PROTEINS	NCI1	NCI109	IMDB-B	IMDB-M
SoTA	92.7±6.1	68.2±7.2	77.2±4.7	83.6±1.4	84.0±1.6	77.8±3.3	54.3±3.3
GIN (Xu et al., 2019)	89.4±5.6	64.6±7.0	76.2±2.8	82.7±1.7	82.2±1.6	75.1±5.1	52.3±2.8
DS-GNN (GIN) (ED)	89.9±3.7	66.0±7.2	76.8±4.6	83.3±2.5	83.0±1.7	76.1±2.6	52.9±2.4
DS-GNN (GIN) (ND)	89.4±4.8	66.3±7.0	77.1±4.6	83.8±2.4	82.4±1.3	75.4±2.9	52.7±2.0
DS-GNN (GIN) (EGO)	89.9±6.5	68.6±5.8	76.7±5.8	81.4±0.7	79.5±1.0	76.1±2.8	52.6±2.8
DS-GNN (GIN) (EGO+)	91.0±4.8	68.7±7.0	76.7±4.4	82.0±1.4	80.3±0.9	77.1±2.6	53.2±2.8
DSS-GNN (GIN) (ED)	91.0±4.8	66.6±7.3	75.8±4.5	83.4±2.5	82.8±0.9	76.8±4.3	53.5±3.4
DSS-GNN (GIN) (ND)	91.0±3.5	66.3±5.9	76.1±3.4	83.6±1.5	83.1±0.8	76.1±2.9	53.3±1.9
DSS-GNN (GIN) (EGO)	91.0±4.7	68.2±5.8	76.7±4.1	83.6±1.8	82.5±1.6	76.5±2.8	53.3±3.1
DSS-GNN (GIN) (EGO+)	91.1±7.0	69.2±6.5	75.9±4.3	83.7±1.8	82.8±1.2	77.1±3.0	53.2±2.4

MAI Lab
EDICAL

Results

- GIN 말고 GCN이랑도 잘 되는지 비교, 잘됨
- Mol-HIV, MolTox 모두 DSS-GNN이 우수

Method	OGBG-MOLHIV ROC-AUC (%) Training	Validation	Test	OGBG-MOLTOX21 ROC-AUC (%) Training	Validation	Test
GCN (Kipf & Welling, 2017)	88.65±2.19	82.04±1.41	76.06±0.97	92.06±1.81	79.04±0.19	75.29±0.69
DS-GNN (GCN) (ED)	86.25±2.77	82.36±0.75	74.70±1.94	90.97±1.70	80.03±0.59	74.86±0.92
DS-GNN (GCN) (ND)	86.82±4.26	81.90±0.82	74.40±2.48	89.28±0.99	80.56±0.61	75.79±0.30
DS-GNN (GCN) (EGO)	91.91±3.83	83.51±0.95	74.00±2.38	89.29±1.64	80.48±0.52	75.41±0.72
DS-GNN (GCN) (EGO+)	86.85±3.57	81.95±0.69	73.84±2.58	90.24±1.20	80.77±0.51	74.74±0.96
DSS-GNN (GCN) (ED)	99.52±0.34	83.44±1.10	76.00±1.41	94.60±1.10	80.05±0.40	75.34±0.69
DSS-GNN (GCN) (ND)	97.40±3.52	82.88±1.29	75.17±1.35	93.82±2.47	81.22±0.52	75.56±0.59
DSS-GNN (GCN) (EGO)	98.56±2.08	84.34±1.02	76.16±1.02	93.06±2.54	81.51±0.43	76.14±0.53
DSS-GNN (GCN) (EGO+)	98.47±2.20	84.45±0.65	76.50±1.38	91.60±1.81	81.55±0.63	76.29±0.78
GIN (Xu et al., 2019)	88.64±2.54	82.32±0.90	75.58±1.40	93.06±0.88	78.32±0.48	74.91±0.51
DS-GNN (GIN) (ED)	91.71±3.50	83.32±0.83	76.43±2.12	92.38±1.57	78.98±0.45	75.12±0.50
DS-GNN (GIN) (ND)	89.70±3.20	83.21±0.87	76.19±0.96	91.23±2.15	79.61±0.59	75.34±1.21
DS-GNN (GIN) (EGO)	93.43±2.17	84.28±0.90	78.00±1.42	92.08±1.79	81.28±0.54	76.22±0.62
DS-GNN (GIN) (EGO+)	90.53±3.01	84.63±0.83	77.40±2.19	90.07±1.65	81.29±0.57	76.39±1.18
DSS-GNN (GIN) (ED)	91.69±3.47	83.33±0.98	77.03±1.81	95.85±2.08	80.83±0.41	76.71±0.67
DSS-GNN (GIN) (ND)	90.71±4.28	83.62±1.20	76.63±1.52	96.90±1.45	81.22±0.23	77.21±0.70
DSS-GNN (GIN) (EGO)	97.05±3.50	85.72±1.21	77.19±1.27	95.58±1.61	81.80±0.20	77.45±0.41
DSS-GNN (GIN) (EGO+)	94.47±2.13	85.51±0.79	76.78±1.66	96.06±1.61	81.82±0.21	77.95±0.40

MAI Lab
EDICAL

Future works

- (1) Is it possible to learn useful subgraph selection policies? → RL
- (2) Can higher order structures on subgraphs (e.g., a graph of subgraphs) be leveraged for better expressive power?
- (3) A theoretical analysis of the stochastic version and of different policies/aggregation functions.

- 총평
- Equivariant라고는 하나 딱히 Equivariant가 돋보이는 부분은 없었음.
- 그래프 별로 subgraph bag수가 다름(노드 수 만큼 존재) → Equivariant가 도움이 됐을 듯
- DSS를 DSS-GNN으로 잘 적용하였음,
- 방대한 실험 및 이론: Proposition 8, Corollary 3 Lemma 16 → Subgraph bag가 굉장히 괜찮은 아이디어, 그러나 조금 부족해보임

MAI Lab EDICAL

Thank you