From Stars to Subgraphs:

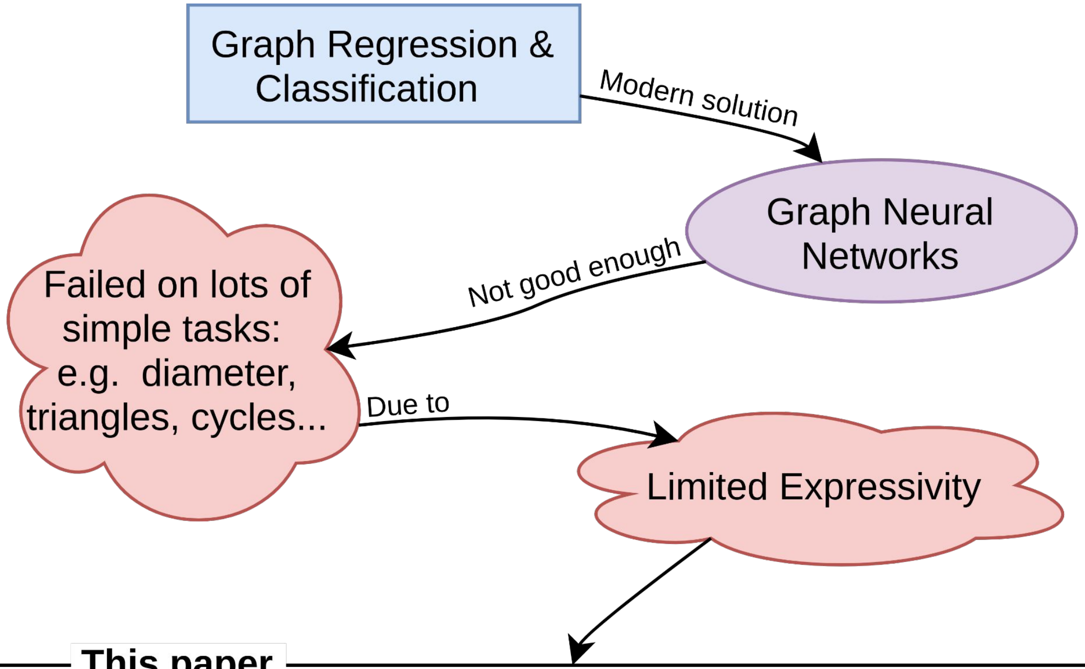
Uplifting any GNN with local structure awareness

Accepted at ICLR 2022:
International Conference on Learning Representations

Lingxiao Zhao, Wei Jin, Leman Akoglu, Neil Shah

Carnegie Mellon University

The Big Picture



This paper

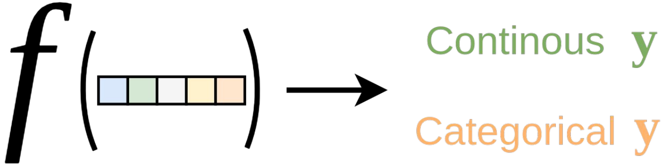
How to *efficiently* boost the expressivity for any GNNs?

Agenda

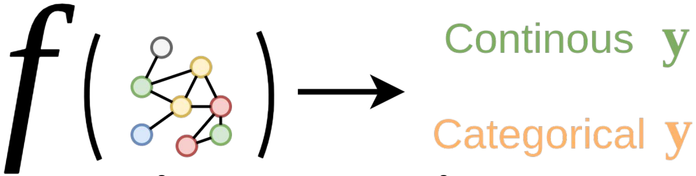
- ➢ Background
  - Graph Regression & Classification
  - Graph Neural Network (GNN)
  - Expressivity & Universality
- From Stars to Subgraphs
- GNN-AK: Boosting Expressivity for Any GNN
- SubgraphDrop: Improving Scalability
- Experiments

Graph Regression & Classification

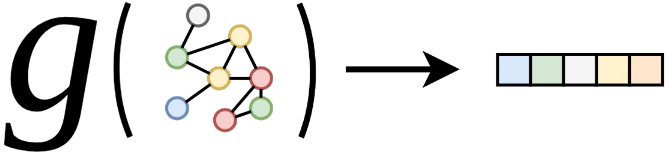
- Regression & Classification



- Graph Regression & Classification

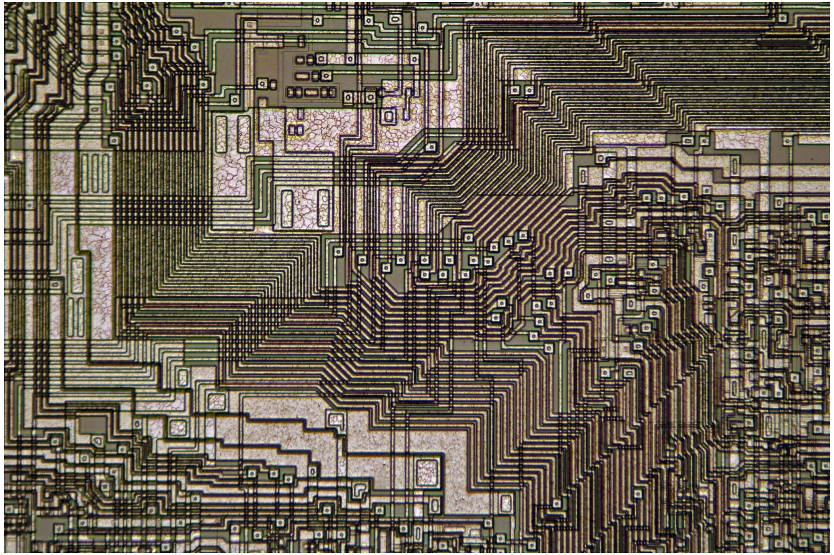


- Graph Representation Learning



Graph Regression & Classification

- Real-world Problems



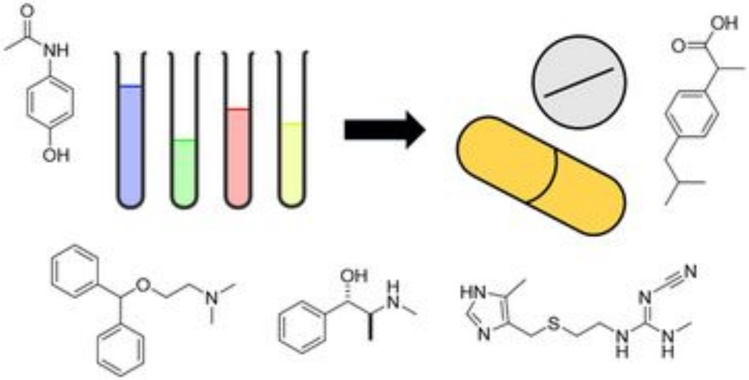
Chip Placement

[Mirhoseini et. al 2021]

Mirhoseini, Azalia, et al. "A graph placement methodology for fast chip design." Nature (2021)

Graph Regression & Classification

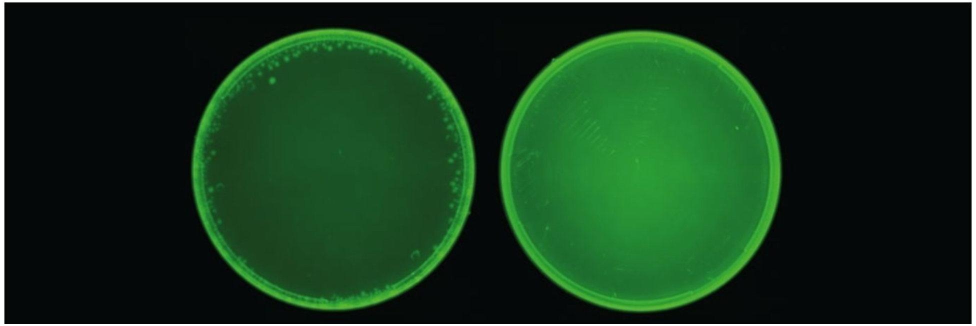
- Real-world Problems



Artificial intelligence yields new antibiotic

A deep-learning model identifies a powerful new drug that can kill many species of antibiotic-resistant bacteria.

Anne Trafton | MIT News Office
February 20, 2020

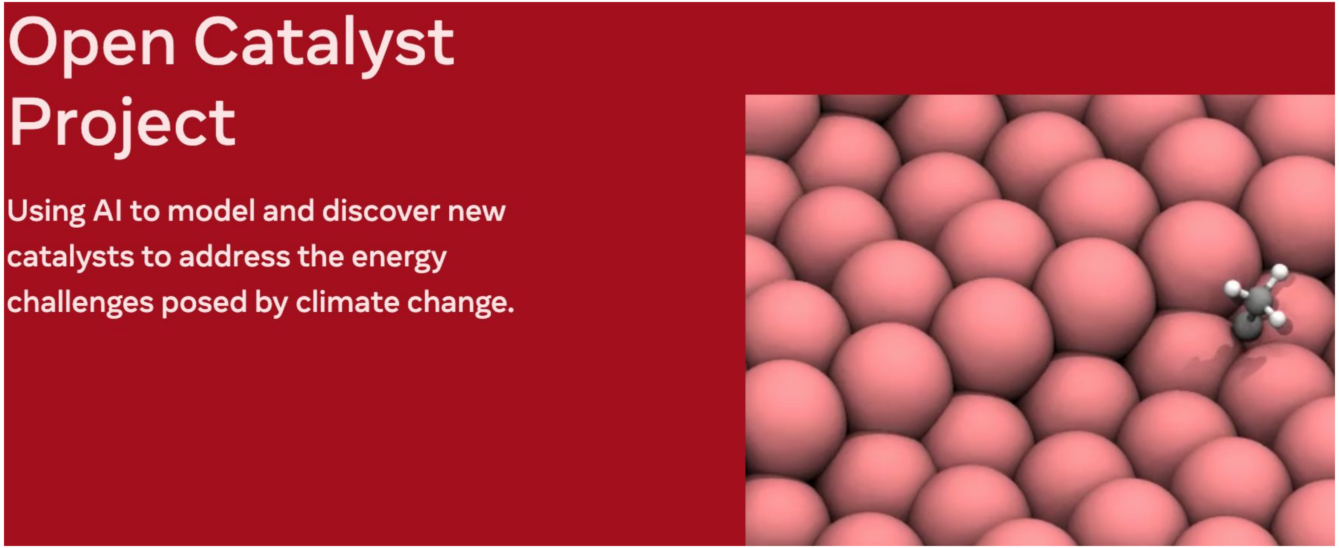


Halicin

Drug Discovery

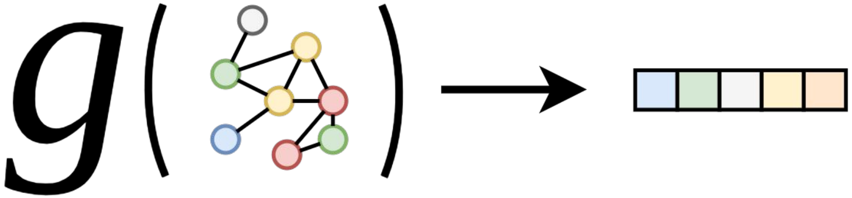
Graph Regression & Classification

- Real-world Problems

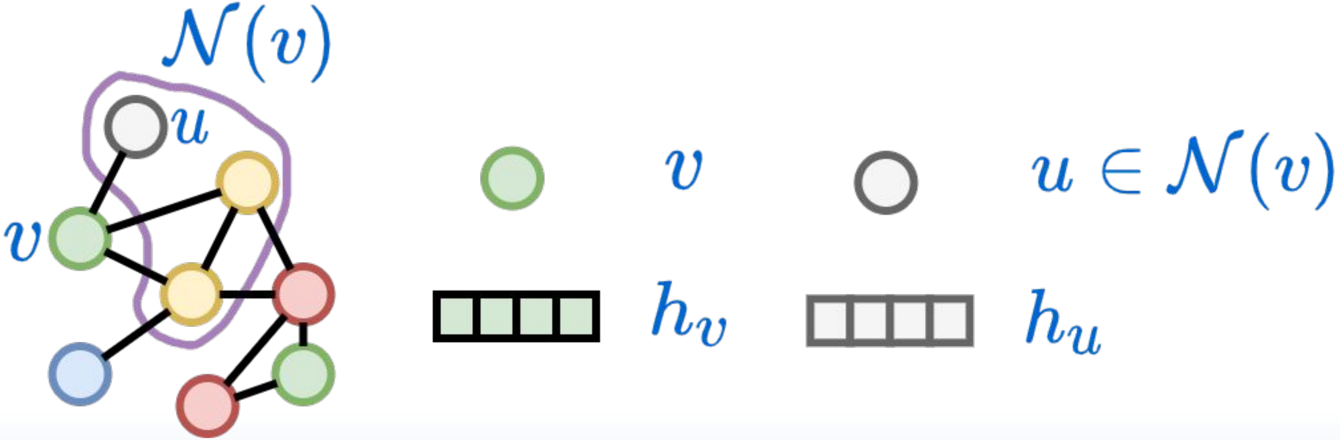


Cheminformatics

Graph Neural Network

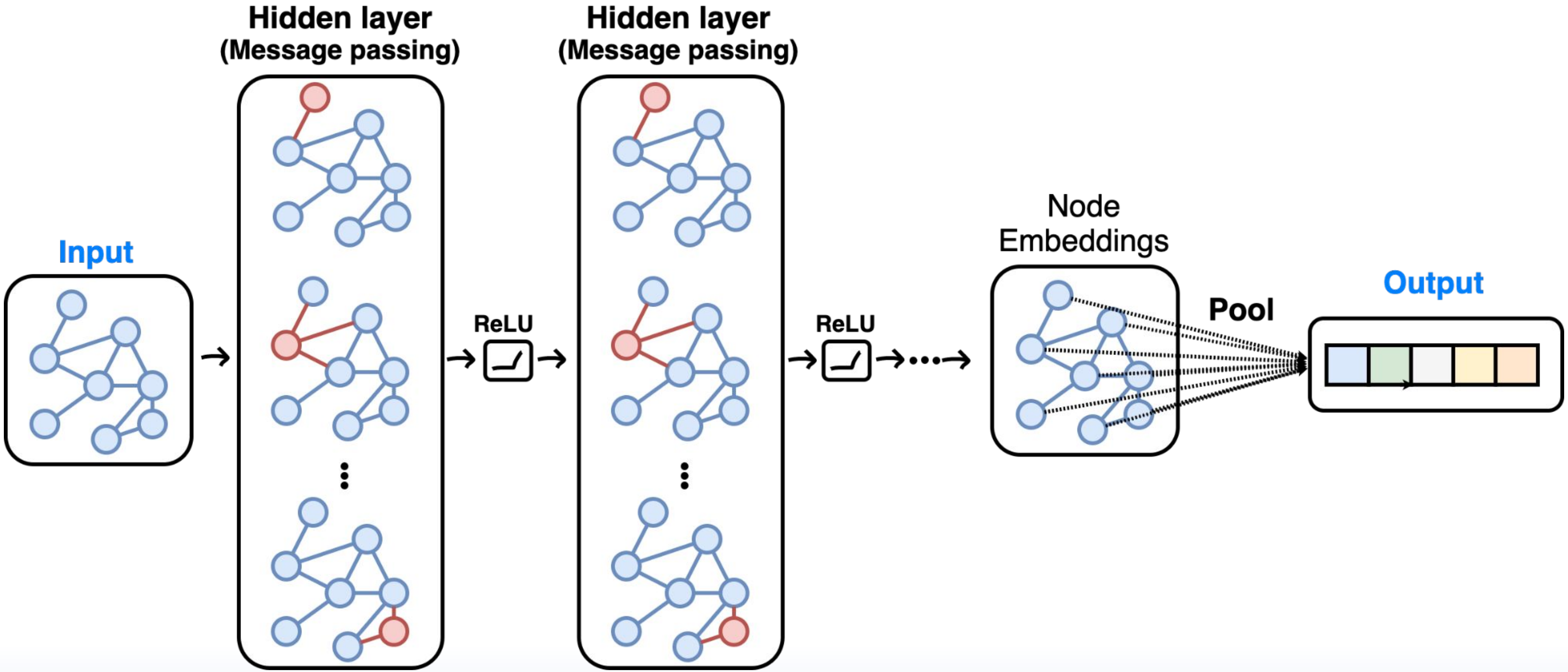


- Input Graph



Graph Neural Network

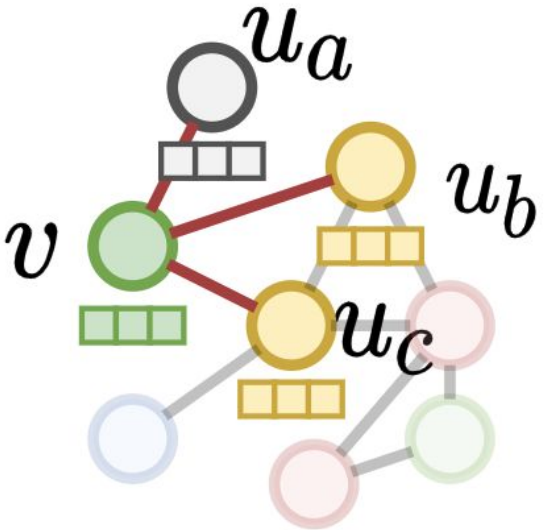
- Architecture: stacking message passing layers



Graph Neural Network

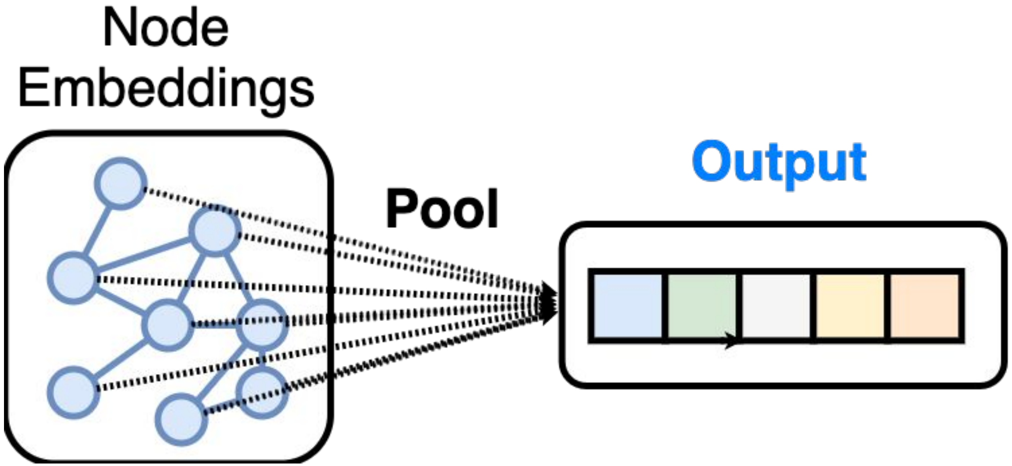
- (t-th) Message Passing Layer

$$h_v^{(t)} = \mathrm{AGG}^{(t)}\left(h_v^{(t-1)}, \left\{\mathrm{MSG}^{(t)}(h_u^{(t-1)}) | u \in \mathcal{N}(v)\right\}\right)$$



Graph Neural Network

- Pool Layer



$$h_G = \text{Pool}\Big(\{h_u^{(T)} | u \in V_G\}\Big)$$

- Sum or Mean

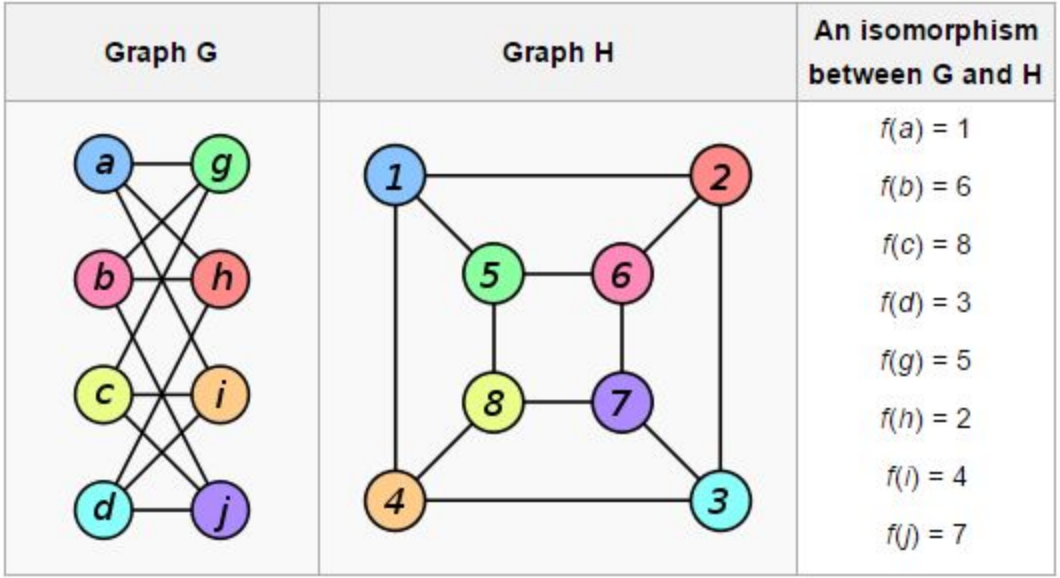
Expressivity & University

- MLP is universal function approximator
  - Function space: Functions over <u>Euclidean</u> space
  - Given enough neurons.

- How about GNN?
  - Function space: Functions over <u>graph</u> space
  - GNN is **NOT** universal approximator!
  - Universal approximator over graph ⇔ Solving **graph isomorphism test** problem [Chen et al. 19]

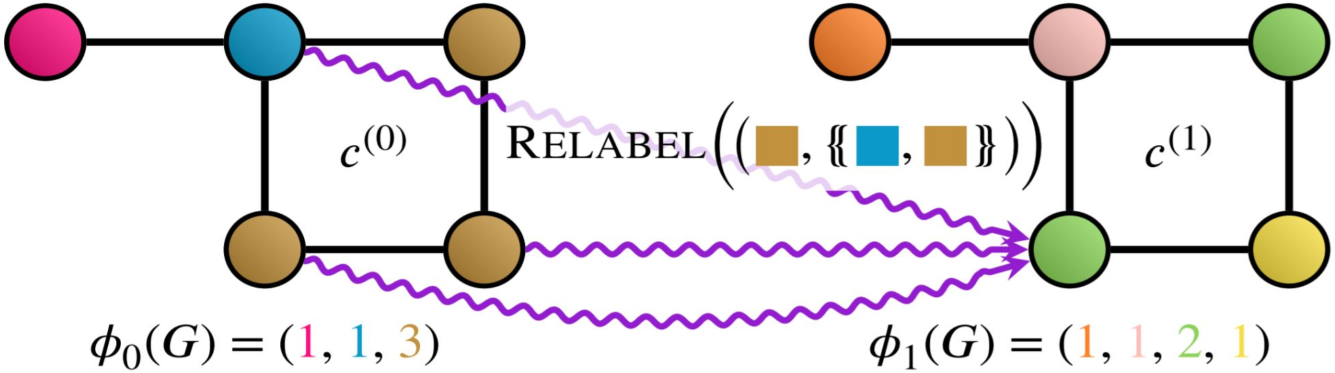
Expressivity & University

- Graph isomorphism test
  - NP-intermediate Problem (if P!= NP)



Expressiveness & Universality

- Weisfeiler-Lehman Isomorphism Test (1-WL)



  - t-th iteration

$$c^{(t)}(v) = \mathrm{HASH}\Big(c^{(t-1)}(v), \{c^{(t-1)}(u) | u \in \mathcal{N}_v\}\Big)$$

  - Output histogram of colors after T iterations.

Expressivity & University

- The expressivity of GNN
  - Upper bounded by 1-WL test [Xu et al. 19]
  - **Cannot**
    - Find cycles
    - Find triangles
    - Calculate diameter
    - Distinguish regular graphs
    - ...

- How to improve **expressivity** efficiently?

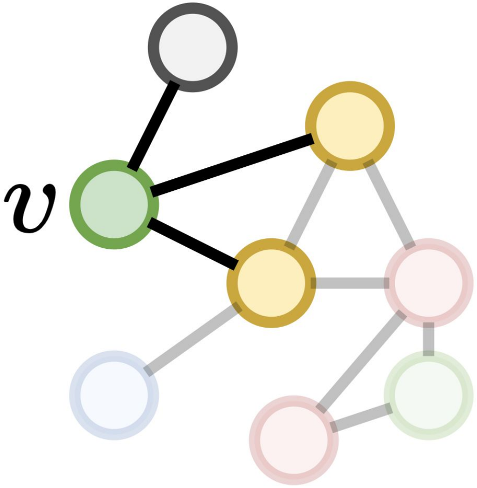
Agenda

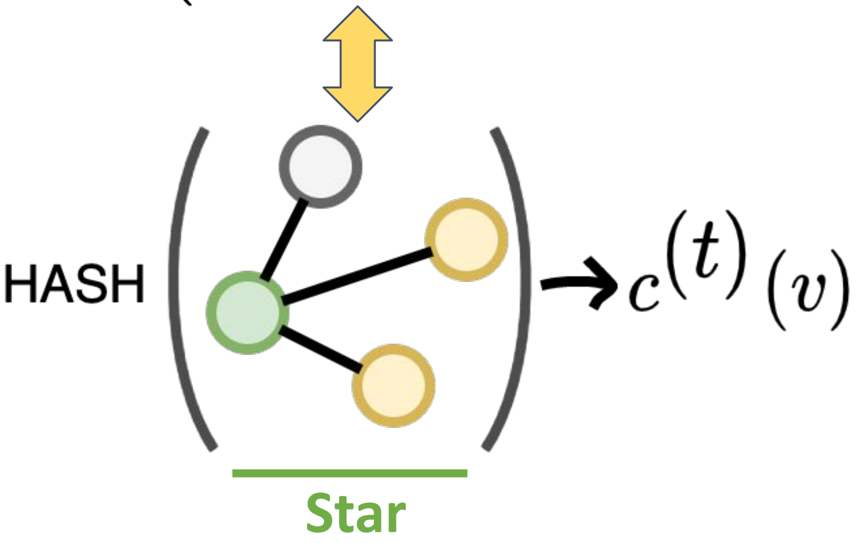
- Background
- From Stars to Subgraphs
  - Subgraph-1-WL
  - Subgraph-1-WL*
  - Theoretical Analysis
- GNN-AK: Boosting Expressivity for Any GNN
- SubgraphDrop: Improving Scalability
- Experiments

Bottleneck of 1-WL

- The star pattern

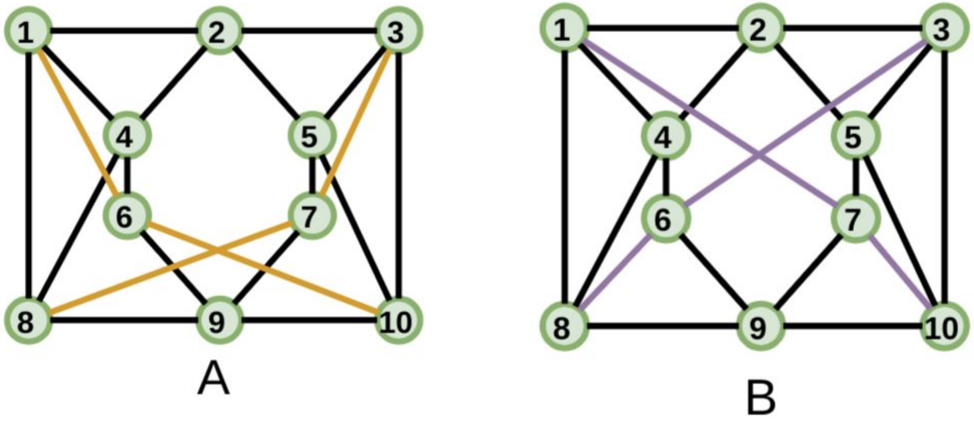
$$c^{(t)}(v) = \mathrm{HASH}\Big(c^{(t-1)}(v), \{c^{(t-1)}(u) | u \in \mathcal{N}_v\}\Big)$$



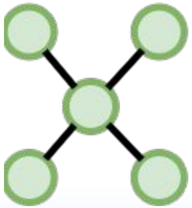


Bottleneck of 1-WL

- Example: two non-isomorphic **regular** graphs

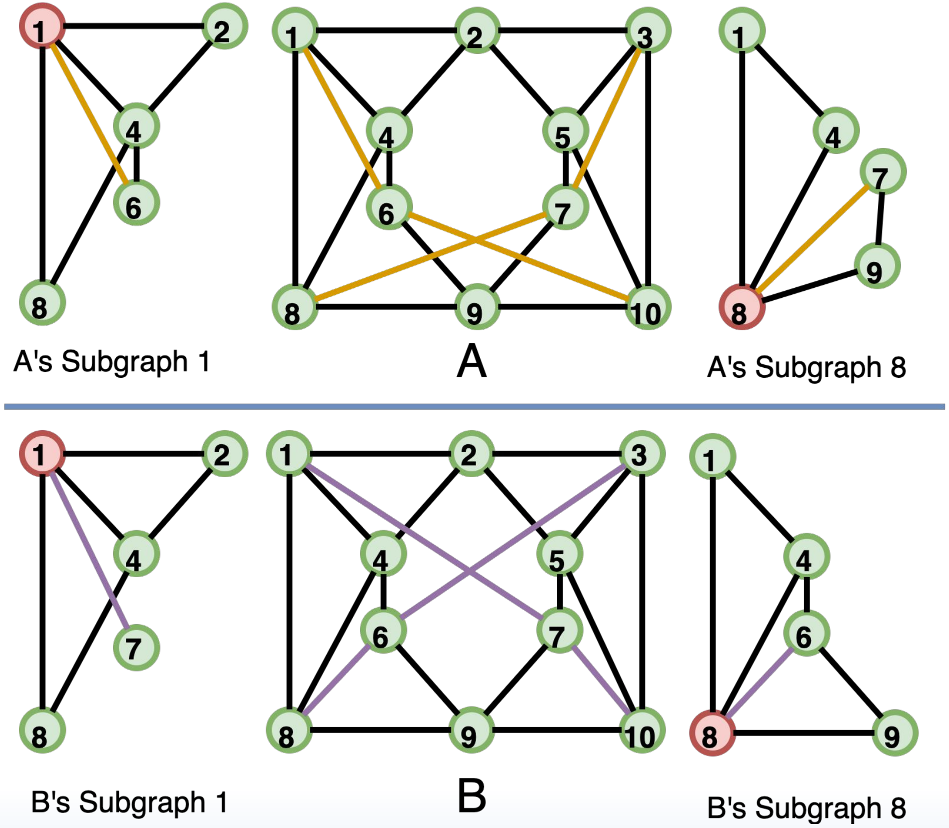


- All stars are the **same**:



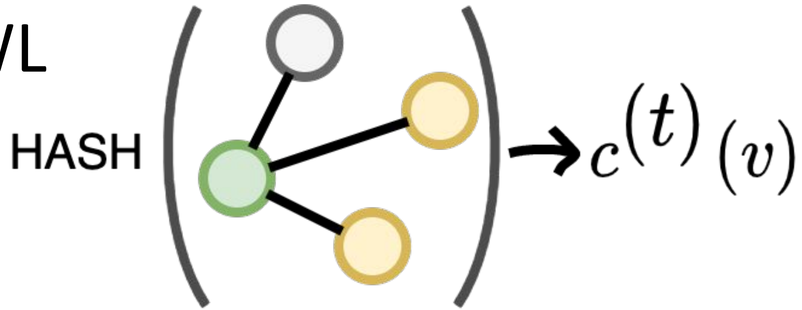
From Stars to Subgraphs

- Go beyond star
- Subgraphs are **NOT** same!

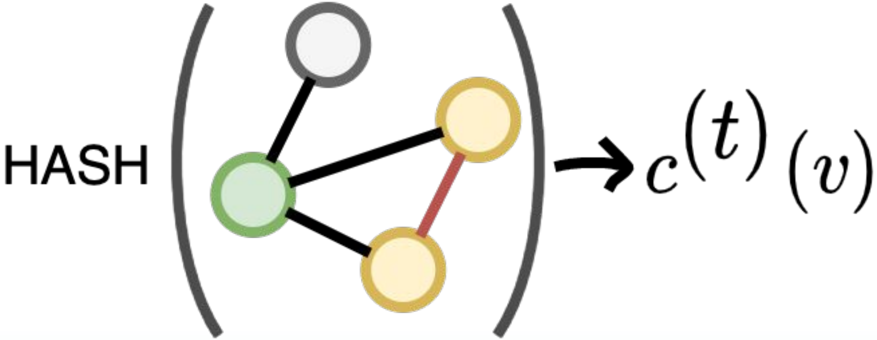


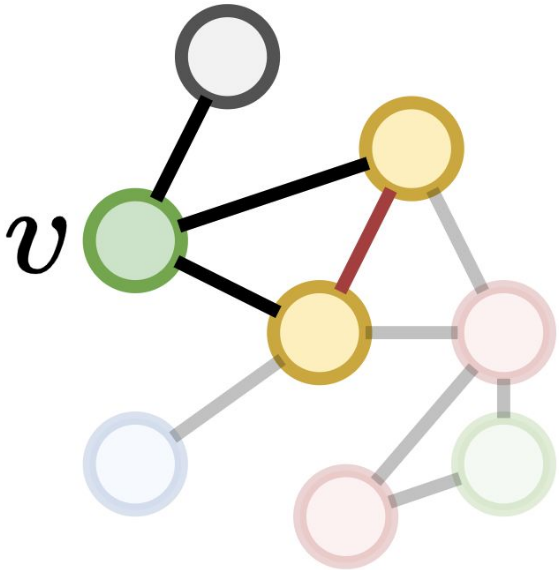
Subgraph-1-WL

- 1-WL



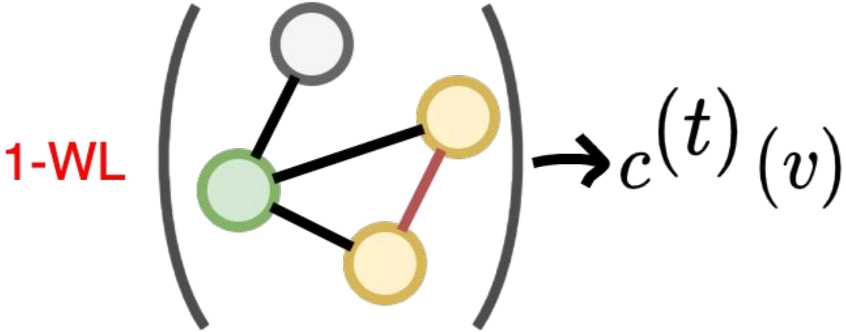
- Subgraph-1-WL





Subgraph-1-WL*

- HASH a graph is not trivial!
- Solution: a weaker version - Subgraph-1-WL*



Theoretical Analysis

- How expressive is Subgraph-1-WL?
  - Strictly more powerful than 1&2-WL
  - No less powerful than 3-WL

- Expressivity upper bound:
  *For any **k**>=3, exist a pair of **k**-WL-failed graphs that cannot be distinguished by Subgraph-1-WL*

- Established sufficient conditions for successful isomorphism test using Subgraph-1-WL

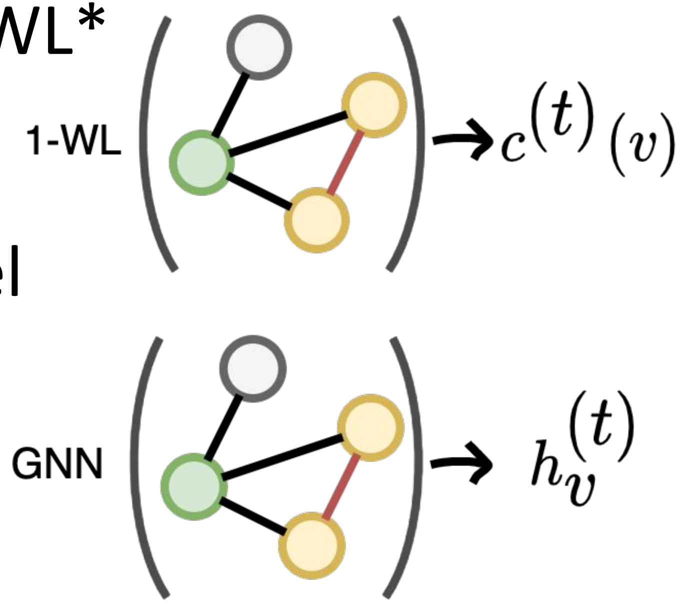
See proofs in paper.

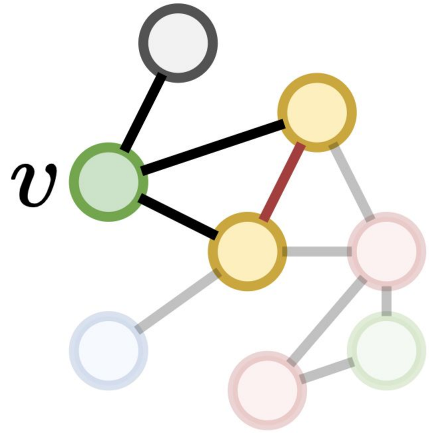
Agenda

- Background
- From Stars to Subgraphs
- ➢ GNN-AK: Boosting Expressivity for Any GNN
  - General formulation
  - GNN-AK
  - GNN-AK+
- SubgraphDrop: Improving Scalability
- Experiments

GNN-AK: the neural version Subgraph-1-WL*

- Under sufficient condition, GNN is as expressive as 1-WL [Chen et al. 19]
- Subgraph-1-WL*
- **GNNAsKernel**





GNN-AK

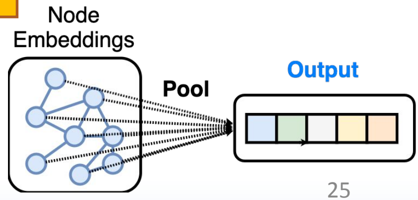
- t-th layer of GNN

$$h_v^{(t)} = \underbrace{\mathrm{AGG}^{(t)}\left(h_v^{(t-1)}, \left\{\mathrm{MSG}^{(t)}(h_u^{(t-1)}) | u \in \mathcal{N}(v)\right\}\right)}_{\text{Encoding of Star[v]}}$$

- t-th layer of GNN-AK

$$h_v^{(t)} = \underbrace{\mathrm{GNN}^{(t)}\left(\mathrm{Subgraph}^{(t-1)}[v]\right)}_{\text{Encoding of Subgraph[v] via a GNN}}$$

$$= \mathrm{Pool}\left(\left\{\underbrace{\mathrm{Emb}(i|v)}_{\text{Subgraph node embeddings pre pool}} \mid i \in \mathrm{Subgraph}[v]\right\}\right)$$

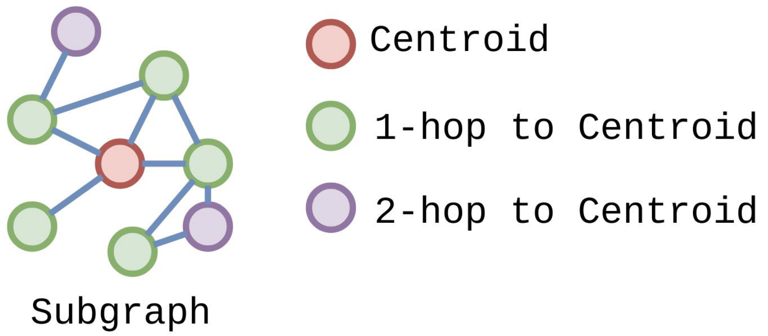


GNN-AK+: more powerful than GNN-AK

- Additional one type of encoding: context encoding

$$h_v^{(t)|\text{context}} = \text{Pool}\Big(\{\text{Emb}(v|j) \mid \forall j \text{ s.t. } v \in \text{Subgraph}[j]\}\Big)$$

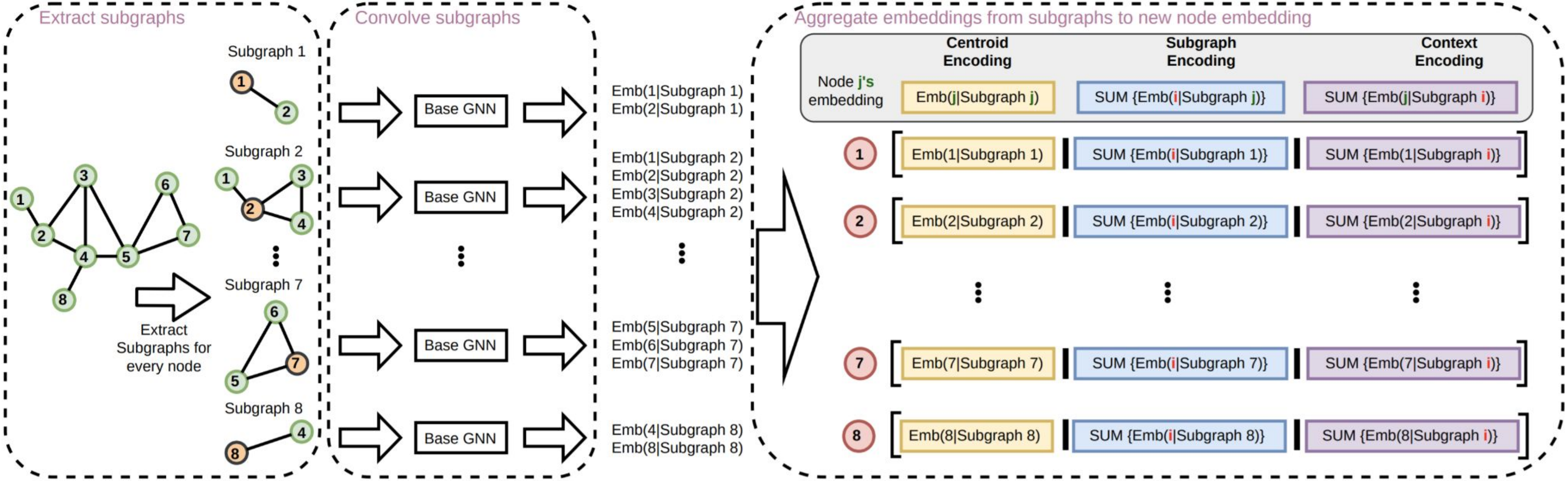
- Using Distance-To-Centroid feature
  - Free feature calculated during subgraph extraction
  - Help expressivity [Li et al. 2020]*



* [Li et al. 2020] shows pairwise shortest path distance can help improving expressivity.

Visualization

- 1 layer of GNN-AK(+)



Agenda

- Background
- From Stars to Subgraphs
- GNN-AK: Boosting Expressivity for Any GNN
- ➢ SubgraphDrop: Improving Scalability
  - Complexity Analysis
  - Drop Subgraphs in Training
- Experiments

SubgraphDrop: Improving Scalability

- Complexity Overheard Analysis
  - t-th layer of GNN: $O\big(|V_G| + |E_G|\big)$
  - t-th layer of GNN-AK(+):

$$O\Big(\sum_{v \in V_G} |V_{\mathrm{Subgraph}[v]}| + \sum_{v \in V_G} |E_{\mathrm{Subgraph}[v]}|\Big)$$

- How to reduce this overhead?
  - Reduce subgraph size
  - Reduce **number of subgraphs**

Reduce Number of Subgraphs

- Subsampling some subgraphs?
  ✖ Still throw out informations
- Solution:
  - Only <u>randomly</u> drop subgraphs during **training**!
    - Every subgraph is used across epochs
  - Still use **ALL** subgraphs during evaluation
- Difficulty:
  - Handling misalignment between train and evaluation

Handling misalignment

- Context Encoding
  - Scale is smaller during training
  - Solution: scale it up
- Centroid & Subgraph Encoding
  - Some nodes don’t have encoding
  - Solution: estimate them by propagation

Context Encoding

SUM {Emb(j|Subgraph i)}

Centroid Encoding

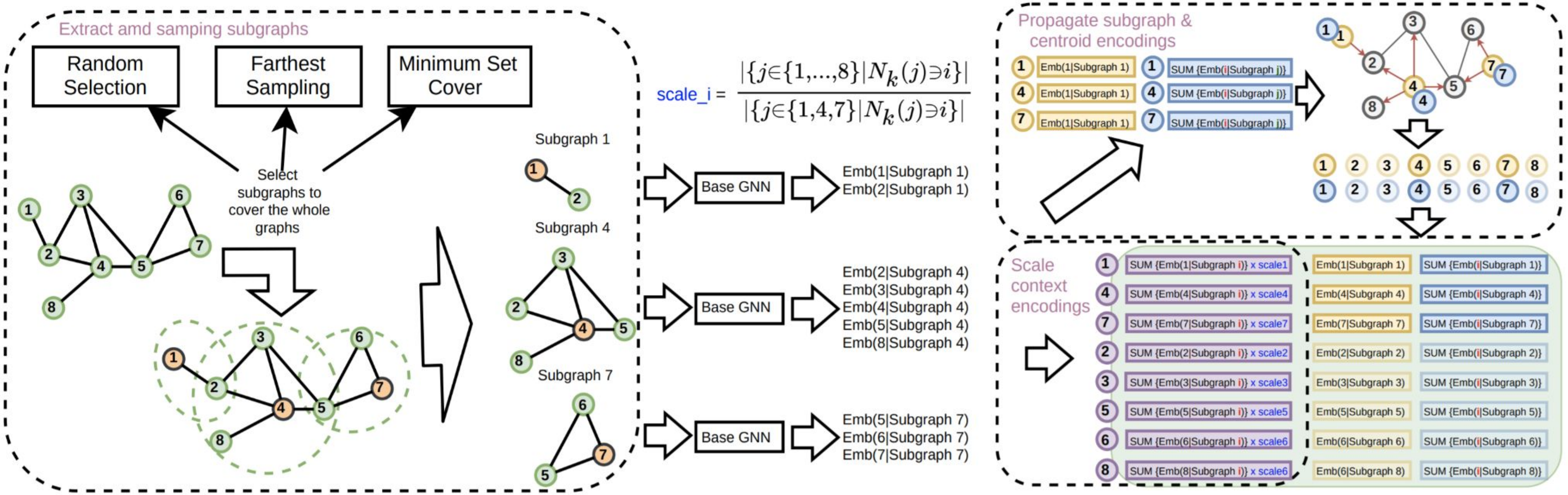
Emb(j|Subgraph j)

Subgraph Encoding

SUM {Emb(i|Subgraph j)}

SubgraphDrop: Improving Scalability

- Each node is covered **R** times by sampled subgraphs.



Agenda

- Background
- From Stars to Subgraphs
- GNN-AK: Boosting Expressivity for Any GNN
- SubgraphDrop: Improving Scalability
- ➢ Experiments
    - Expressivity Verification
    - The Ability in Counting Substructures
    - The Ability in Regressing Graph Properties
    - Real-World Performance

Experiment: Expressivity Verification

- EXP: 600 pairs of 1-WL-failed graphs, split into 2 classes.
- SR25: 15 3-WL-failed graphs, 15-class classification.
- Base GNNs
  - GCN [Kipf & Welling, 2017]
  - GIN [Xu et al., 2018]
  - PNA [Corso et al., 2020]
  - PPGN [Maron et al., 2019a] (more powerful than others)

Method	EXP (ACC)	SR25 (ACC)
GCN	50%	6.67%
GCN-AK+	**100%**	6.67%
GIN	50%	6.67%
GIN-AK+	**100%**	6.67%
PNA*	50%	6.67%
PNA*-AK+	**100%**	6.67%
PPGN	100%	6.67%
PPGN-AK+	100%	**100%**

Experiment: Counting Substructures

Method	Counting Substructures (MAE)			
	Triangle	Tailed Tri.	Star	4-Cycle
GCN	0.4186	0.3248	0.1798	0.2822
GCN-AK$^+$	0.0137	0.0134	0.0174	0.0183
GIN	0.3569	0.2373	0.0224	0.2185
GIN-AK$^+$	0.0123	0.0112	0.0150	0.0126
PNA*	0.3532	0.2648	0.1278	0.2430
PNA*-AK$^+$	0.0118	0.0138	0.0166	0.0132
PPGN	0.0089	0.0096	0.0148	0.0090
PPGN-AK$^+$	OOM	OOM	OOM	OOM

Experiment: Regress Graph Properties

Method	Graph Properties ($\log_{10}$(MAE))		
	IsConnected	Diameter	Radius
GCN	-1.7057	-2.4705	-3.9316
GCN-AK$^+$	-2.6705	-3.9102	-5.1136
GIN	-1.9239	-3.3079	-4.7584
GIN-AK$^+$	-2.7513	-3.9687	-5.1846
PNA*	-1.9395	-3.4382	-4.9470
PNA*-AK$^+$	-2.6189	-3.9011	-5.2026
PPGN	-1.9804	-3.6147	-5.0878
PPGN-AK$^+$	OOM	OOM	OOM

Experiment: Real-World Performance

- Datasets:
  - 3 from Benchmarking GNN [Dwivedi et al., 2020]
  - 2 from Open Graph Benchmark [Hu et al., 2020]

Dataset	Task Semantic	# Cls./Tasks	# Graphs	Ave. # Nodes	Ave. # Edges
ZINC-12K	Regress molecular property	1	10000 / 1000 / 1000	23.1	49.8
CIFAR10	10-class classification	10	45000 / 5000 / 10000	117.6	1129.8
PATTERN	Recognize certain subgraphs	2	10000 / 2000 / 2000	118.9	6079.8
MolHIV	1-way binary classification	1	32901 / 4113 / 4113	25.5	54.1
MolPCBA	128-way binary classification	128	350343 / 43793 / 43793	25.6	55.4

Experiment: Real-World Performance

Method	ZINC-12K (MAE)	CIFAR10 (ACC)	PATTERN (ACC)	MolHIV (ROC)	MolPCBA (AP)
GatedGCN	0.363 ± 0.009	69.37 ± 0.48	84.480 ± 0.122	–	–
HIMP	0.151 ± 0.006	–	–	0.7880 ± 0.0082	–
PNA	0.188 ± 0.004	70.47 ± 0.72	86.567 ± 0.075	0.7905 ± 0.0132	0.2838 ± 0.0035
DGN	0.168 ± 0.003	72.84 ± 0.42	86.680 ± 0.034	0.7970 ± 0.0097	0.2885 ± 0.0030
GSN	0.115 ± 0.012	–	–	0.7799 ± 0.0100	–
CIN	$\mathbf{0.079} \pm \mathbf{0.006}$	–	–	$\mathbf{0.8094} \pm \mathbf{0.0057}$	–
GCN	0.321 ± 0.009	58.39 ± 0.73	85.602 ± 0.046	0.7422 ± 0.0175	0.2385 ± 0.0019
GCN-AK$^+$	0.127 ± 0.004	72.70 ± 0.29	$\mathbf{86.887} \pm \mathbf{0.009}$	0.7928 ± 0.0101	0.2846 ± 0.0002
GIN	0.163 ± 0.004	59.82 ± 0.33	85.732 ± 0.023	0.7881 ± 0.0119	0.2682 ± 0.0006
GIN-AK$^+$	$\mathbf{0.080} \pm \mathbf{0.001}$	72.19 ± 0.13	86.850 ± 0.057	0.7961 ± 0.0119	$\mathbf{0.2930} \pm \mathbf{0.0044}$
PNA*	0.140 ± 0.006	73.11 ± 0.11	85.441 ± 0.009	0.7905 ± 0.0102	0.2737 ± 0.0009
PNA*-AK$^+$	0.085 ± 0.003	OOM	OOM	0.7880 ± 0.0153	0.2885 ± 0.0006
GCN-AK$^+$-S	0.127 ± 0.001	71.93 ± 0.47	86.805 ± 0.046	0.7825 ± 0.0098	0.2840 ± 0.0036
GIN-AK$^+$-S	0.083 ± 0.001	72.39 ± 0.38	86.811 ± 0.013	0.7822 ± 0.0075	0.2916 ± 0.0029
PNA*-AK$^+$-S	0.082 ± 0.000	$\mathbf{74.79} \pm \mathbf{0.18}$	86.676 ± 0.022	0.7821 ± 0.0143	0.2880 ± 0.0012

Experiment: Scalability via SubgraphDrop

- Each node is covered **R** times by sampled subgraphs.
- R = 1: similar resource usage as base GNN

Table 4: Resource analysis of SubgraphDrop.

Dataset		GIN-AK$^+$-S R=1	R=2	R=3	R=4	R=5	GIN-AK$^+$	GIN
ZINC-12K	MAE	0.1216	0.0929	0.0846	0.0852	0.0854	0.0806	0.1630
	Runtime (S/Epoch)	10.8	11.2	12.0	12.4	12.5	9.4	6.0
	Memory (MB)	392	811	1392	1722	1861	1911	124
CIFAR10	ACC	71.68	72.07	72.39	72.20	72.32	72.19	59.82
	Runtime (S/Epoch)	80.7	89.1	100.5	110.9	119.7	241.1	55.0
	Memory (MB)	2576	4578	6359	8716	10805	30296	801

Summary

- GNN-AK: The first **general** framework of uplifting GNN with **theoretical support**, great **scalability**, and remarkable **practical** performance.
- Code: https://github.com/GNNAsKernel/GNNAsKernel

Thank you!