How Powerful are Graph Neural Networks?

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka

ICLR 2019 (oral)

Yizhu Jiao

2019/06/04

Introduction

- Tasks on graph
  - Graph classification
  - Node classification
  - Link prediction

- Currently used framework in GNN
  - neighborhood aggregation (message passing)

Graph Neural Networks (GNN)

- Iteratively update the representation of nodes

$$a_v^{(k)} = \text{AGGREGATE}^{(k)} \left(\left\{ h_u^{(k-1)} : u \in \mathcal{N}(v) \right\} \right)$$

$$h_v^{(k)} = \text{COMBINE}^{(k)} \left(h_v^{(k-1)}, a_v^{(k)} \right)$$

- Get the entire graph's representation

$$h_G = \text{READOUT}\big(\{h_v^{(K)} \mid v \in G\}\big)$$

GNN variants

- GraphSAGE (Hamilton et al., 2017)
  - AGGREGATE

$$a_v^{(k)} = \mathrm{MAX}\left(\left\{\mathrm{ReLU}\left(W \cdot h_u^{(k-1)}\right),\ \forall u \in \mathcal{N}(v)\right\}\right)$$

  - COMBINE

$$W \cdot \left[h_v^{(k-1)} \middle| a_v^{(k)}\right]$$

GNN variants

- Graph Convolutional Networks (GCN)

  - AGGREGATE and COMBINE are integrated

$$h_v^{(k)} = \mathrm{ReLU}\left(W \cdot \mathrm{MEAN}\left\{h_u^{(k-1)},\ \forall u \in \mathcal{N}(v) \cup \{v\}\right\}\right)$$

Motivations

- Design of new GNNs
  - mostly based on empirical intuition and experimental trial-and-error

- How to analyze the representational power of GNNs ?
- How to design a more powerful GNN ?

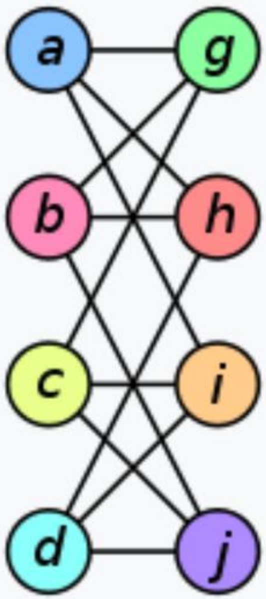
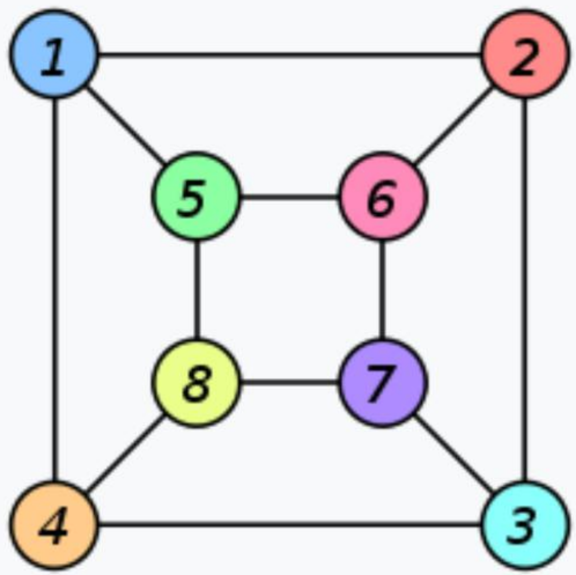
Theoretical Framework

- Base on neighborhood aggregation framework

- Base on graph classification
  - distinguish different graph structures
    - imply solving graph isomorphism
  - capture different graph's structural similarity

Theoretical Framework

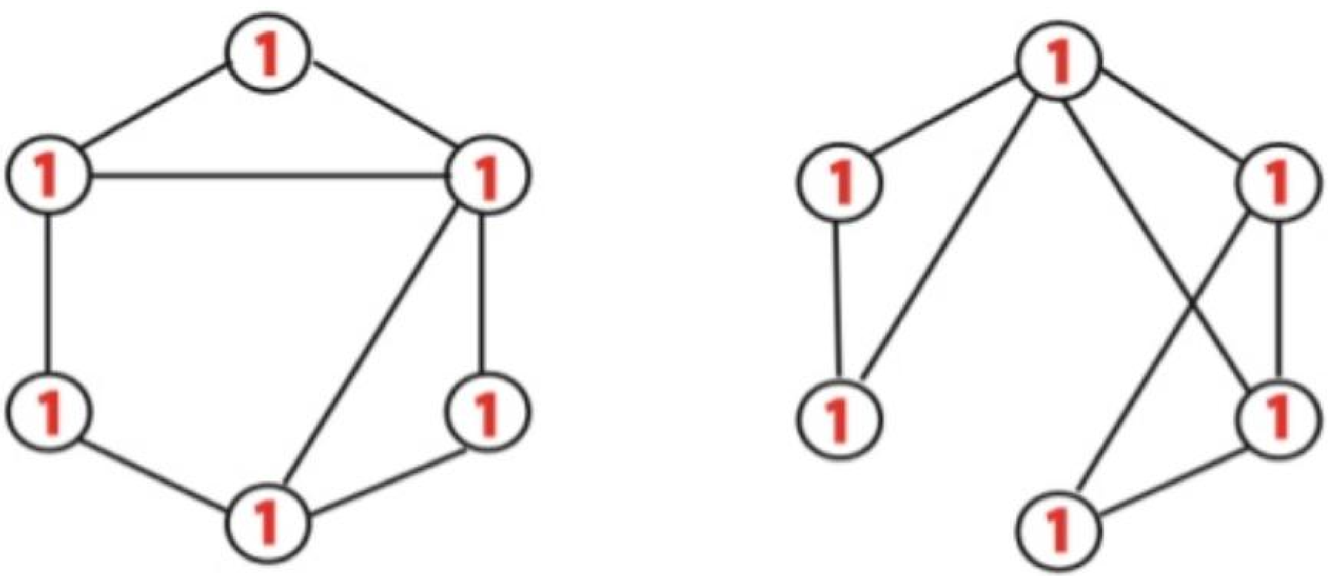
- Graph isomorphism

图G	图H	从图G到图H的同构映射σ
		$\sigma(a)=1$ $\sigma(b)=6$ $\sigma(c)=8$ $\sigma(d)=3$ $\sigma(g)=5$ $\sigma(h)=2$ $\sigma(i)=4$ $\sigma(j)=7$

Theoretical Framework

- weaker criterion: Weisfeiler-Lehman (WL) test

Given two graphs G and G'



Theoretical Framework

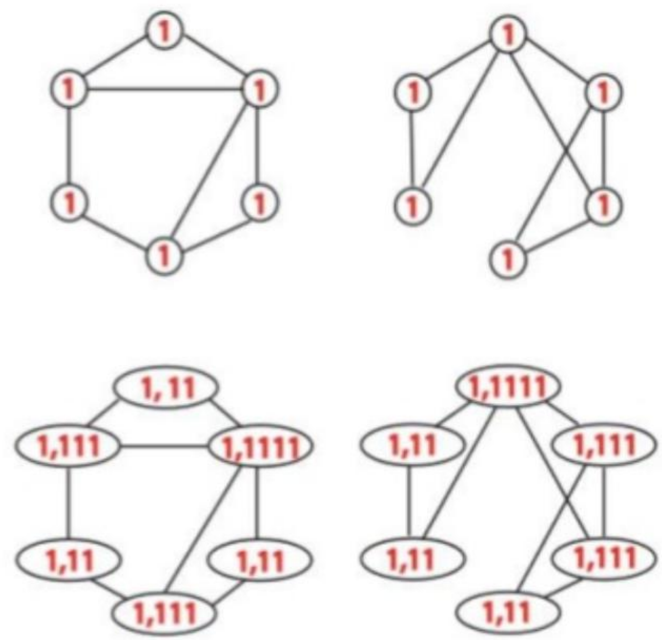
- WL graph isomorphism test

WL Algorithm :iteration 1

Each Iteration of WL test comprises of following steps:-

1.Multiset label determination and sorting

- O(m) via Bucket Sort



Theoretical Framework

- WL graph isomorphism test

WL Algorithm :iteration 1

Each Iteration of WL test comprises of following steps:-

1.Multiset label determination and sorting

- O(m) via Bucket Sort

2.label compression

- O(m) via Radix Sort

1, 11	1, 11	1,111
1, 11	1, 11	1,111
1, 11	1,111	1,1111
1, 11	1,111	1,1111

1, 11 $\longrightarrow$ 2

1,111 $\longrightarrow$ 3

1,1111 $\longrightarrow$ 4

Theoretical Framework

- WL graph isomorphism test

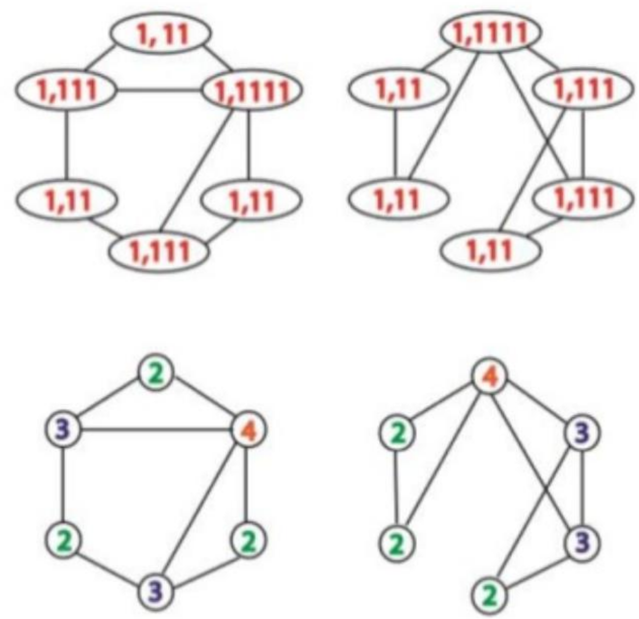
WL Algorithm :iteration 1

Each Iteration of WL test comprises of following steps:-

1.Multiset label determination and sorting
  - O(m) via Bucket Sort

2.label compression
  - O(m) via Radix Sort

3. Relabeling
  - O(n)



Theoretical Framework

- WL graph isomorphism test

WL Algorithm :iteration 1

Each Iteration of WL test comprises of following steps:-

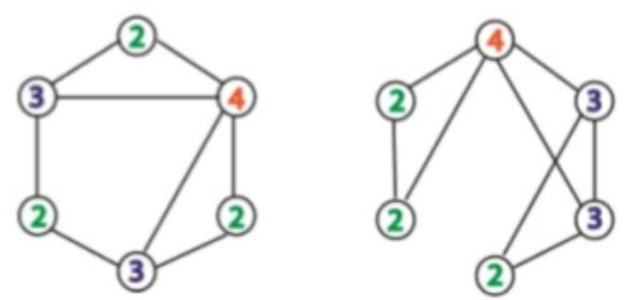
1.Multiset label determination and sorting
- O(m) via Bucket Sort

2.label compression
- O(m) via Radix Sort

3. Relabeling
- O(n)

4.Are the labels of G and G' identical?

Yes, continue.



Theoretical Framework

- WL graph isomorphism test

WL Algorithm :iteration 2

Each Iteration of WL test comprises of following steps:-

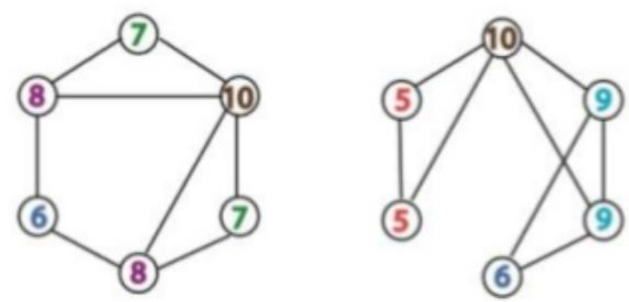
1.Multiset label determination and sorting
- O(m) via Bucket Sort

2.label compression
- O(m) via Radix Sort

3. Relabeling
- O(n)

4.Are the labels of G and G' identical?
NO, output YES.

5.complexity O(hm) for h iteration



Theoretical Framework

- Multiset
  - the feature vectors of a node’s neighbors

$$X = (S, m)$$

Theoretical Framework

Lemma 2. *Let G_1 and G_2 be any non-isomorphic graphs. If a graph neural network $\mathcal{A} : \mathcal{G} \to \mathbb{R}^d$ following the neighborhood aggregation scheme maps G_1 and G_2 to different embeddings, the Weisfeiler-Lehman graph isomorphism test also decides G_1 and G_2 are not isomorphic.*

- GNNs are at most as powerful as the WL test in distinguishing graph structures

Theoretical Framework

Theorem 3. *Let $\mathcal{A} : \mathcal{G} \rightarrow \mathbb{R}^d$ be a GNN following the neighborhood aggregation scheme. With sufficient iterations, $\mathcal{A}$ 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.

- GNN maps different graph structures to different embedding if AGGREGATE, COMBINE, READOUT are injective

Graph Isomorphism Network (GIN)

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 finite multiset. Moreover, any function g over such pairs can be decomposed as $g\,(c, X) = \phi\left((1 + \epsilon) \cdot f(c) + \sum_{x \in X} f(x)\right)$ for some function ϕ.*

- $g\,(c, X) = \phi\left((1 + \epsilon) \cdot f(c) + \sum_{x \in X} f(x)\right)$
- model $f^{(k+1)} \circ \phi^{(k)}$ with one MLP

Graph Isomorphism Network (GIN)

- AGGREGATE and COMBINE

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

- READOUT

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

- GIN provably generalizes the WL test

Analysis on less powerful GNNs

- Ablation on the aggregator in

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

- Two aspects
  - 1-layer perceptrons instead of MLPs
  - Mean or Max-pooling instead of the sum
    - Mean -> GCN
    - Max -> GraphSAGE

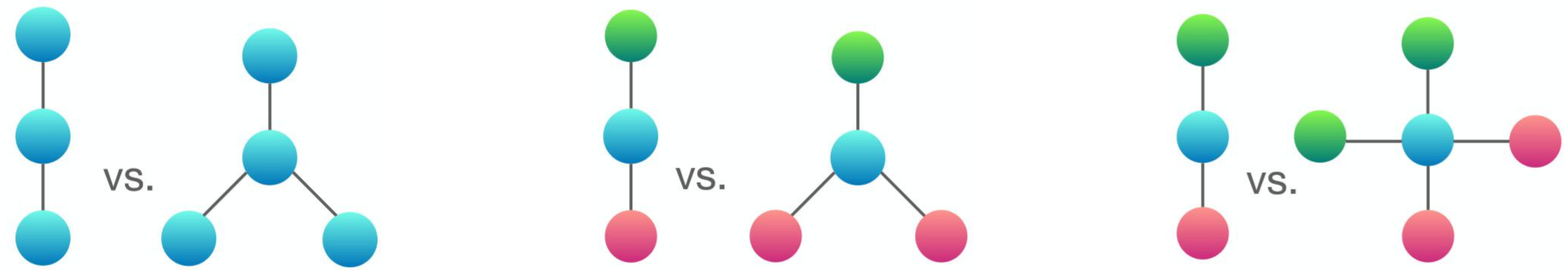
Analysis on less powerful GNNs

Lemma 7. *There exist finite multisets* $X_1 \neq X_2$ *so that for any linear mapping* W, $\sum_{x \in X_1} \mathrm{ReLU}(Wx) = \sum_{x \in X_2} \mathrm{ReLU}(Wx)$.

- Linear mapping + ReLU is not sufficient
- Linear mapping + bias + ReLU can distinguish to some degree
    - May not adequately capture structural similarity
    - Difficult for simple classifiers to fit

Analysis on less powerful GNNs

- Structures that confuse mean and max-pooling



(a) Mean and Max both fail (b) Max fails (c) Mean and Max both fail

- $f(a)$ and $2 \cdot f(a)$ are different

Analysis on less powerful GNNs

- Mean-pooling learns distributions

$$X_1 = (S, m) \qquad X_2 = (S, k \cdot m)$$

- Mean aggregator is as powerful as the sum aggregator if node features are diverse and rarely repeat

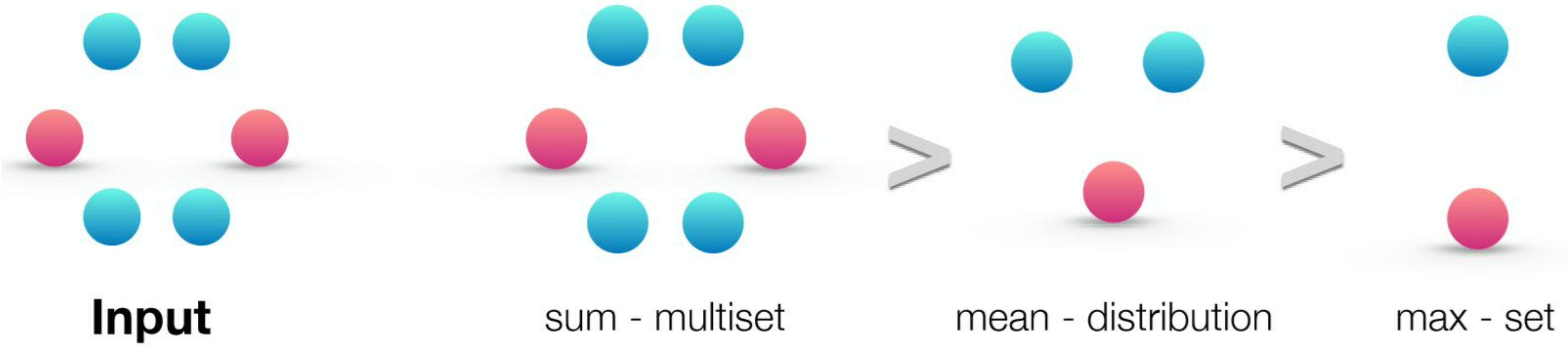
Analysis on less powerful GNNs

- Max-pooling learns sets with distinct elements
  - Treat a multiset as a set
  - Capture graph skeleton

- Max aggregator may be suitable for tasks where it is important to identify representative elements

Analysis on less powerful GNNs

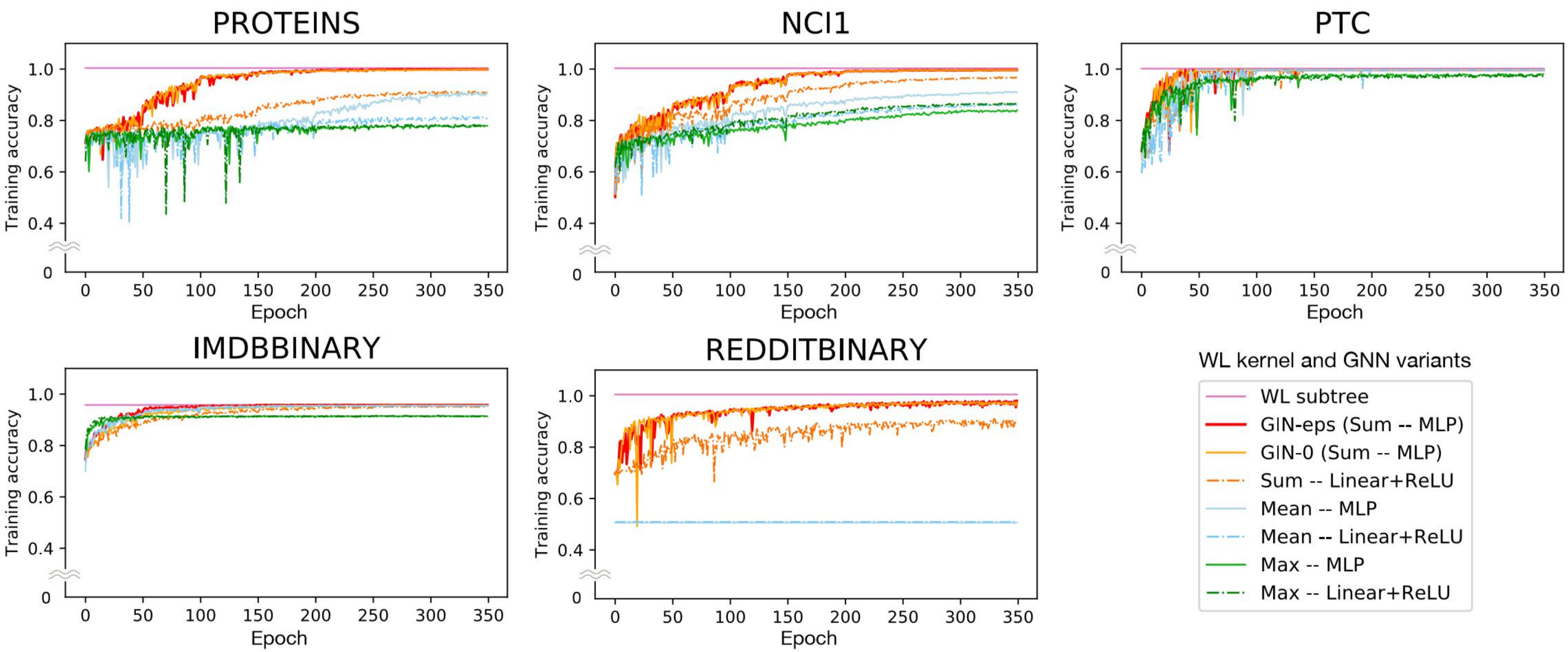
- Ranking by expressive power



Analysis on less powerful GNNs

- Other neighbor aggregations (not message-passing-based)
  - Graph Attention Networks (GAT)
    - Self-attention
  - LSTM pooling (Hamilton et al., 2017a; Murphy et al., 2018)

Experiments



Training set performance

Experiments

	Datasets	IMDB-B	IMDB-M	RDT-B	RDT-M5K	COLLAB	MUTAG	PROTEINS	PTC	NCI1
Datasets	# graphs	1000	1500	2000	5000	5000	188	1113	344	4110
	# classes	2	3	2	5	3	2	2	2	2
	Avg # nodes	19.8	13.0	429.6	508.5	74.5	17.9	39.1	25.5	29.8
Baselines	WL subtree	73.8 ± 3.9	50.9 ± 3.8	81.0 ± 3.1	52.5 ± 2.1	78.9 ± 1.9	90.4 ± 5.7	75.0 ± 3.1	59.9 ± 4.3	$\mathbf{86.0 \pm 1.8}$ *
	DCNN	49.1	33.5	–	–	52.1	67.0	61.3	56.6	62.6
	PATCHYSAN	71.0 ± 2.2	45.2 ± 2.8	86.3 ± 1.6	49.1 ± 0.7	72.6 ± 2.2	$\mathbf{92.6 \pm 4.2}$ *	75.9 ± 2.8	60.0 ± 4.8	78.6 ± 1.9
	DGCNN	70.0	47.8	–	–	73.7	85.8	75.5	58.6	74.4
	AWL	74.5 ± 5.9	51.5 ± 3.6	87.9 ± 2.5	54.7 ± 2.9	73.9 ± 1.9	87.9 ± 9.8	–	–	–
GNN variants	GIN-ϵ (SUM–MLP)	$\mathbf{74.3 \pm 5.1}$	$\mathbf{52.1 \pm 3.6}$	$\mathbf{92.2 \pm 2.3}$	$\mathbf{57.0 \pm 1.7}$	$\mathbf{80.1 \pm 1.9}$	$\mathbf{89.0 \pm 6.0}$	$\mathbf{75.9 \pm 3.8}$	63.7 ± 8.2	$\mathbf{82.7 \pm 1.6}$
	GIN-0 (SUM–MLP)	$\mathbf{75.1 \pm 5.1}$	$\mathbf{52.3 \pm 2.8}$	$\mathbf{92.4 \pm 2.5}$	$\mathbf{57.5 \pm 1.5}$	$\mathbf{80.2 \pm 1.9}$	$\mathbf{89.4 \pm 5.6}$	$\mathbf{76.2 \pm 2.8}$	$\mathbf{64.6 \pm 7.0}$	$\mathbf{82.7 \pm 1.7}$
	SUM–1-LAYER	74.1 ± 5.0	$\mathbf{52.2 \pm 2.4}$	90.0 ± 2.7	55.1 ± 1.6	$\mathbf{80.6 \pm 1.9}$	$\mathbf{90.0 \pm 8.8}$	$\mathbf{76.2 \pm 2.6}$	63.1 ± 5.7	82.0 ± 1.5
	MEAN–MLP	73.7 ± 3.7	$\mathbf{52.3 \pm 3.1}$	50.0 ± 0.0 † (71.2 ± 4.6)	20.0 ± 0.0 † (41.3 ± 2.1)	79.2 ± 2.3	83.5 ± 6.3	75.5 ± 3.4	$\mathbf{66.6 \pm 6.9}$	80.9 ± 1.8
	MEAN–1-LAYER	74.0 ± 3.4	51.9 ± 3.8	50.0 ± 0.0 † (69.7 ± 3.2)	20.0 ± 0.0 † (39.7 ± 2.4)	79.0 ± 1.8	85.6 ± 5.8	76.0 ± 3.2	64.2 ± 4.3	80.2 ± 2.0
	MAX–MLP	73.2 ± 5.8	51.1 ± 3.6	–	–	–	84.0 ± 6.1	76.0 ± 3.2	64.6 ± 10.2	77.8 ± 1.3
	MAX–1-LAYER	72.3 ± 5.3	50.9 ± 2.2	–	–	–	85.1 ± 7.6	75.9 ± 3.2	63.9 ± 7.7	77.7 ± 1.5

Classification accuracies

Conclusion

- Develop theoretical foundations for reasoning about expressive power of GNNs
- Prove tight bounds on the representational capacity of popular GNN variants
- Design a provably most powerful GNN under the message passing framework

Q & A