GNN 理论分析

刘闯
chuangliu@whu.edu.cn
2019.03.08

Reference

1. HOW POWERFUL ARE GRAPH NEURAL NETWORKS? , Jure Leskovec, ICLR 2019 (P3 - P13)
2. GNNExplainer: Generating Explanations for Graph Neural Networks, Jure Leskovec, 2019 NIPS (P14 - P18)
3. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. (P19 - P20)

GNN 理论分析

HOW POWERFUL ARE GRAPH NEURAL NETWORKS?

典型的 GNN

$$\mathbf{h}_{\mathcal{N}(v)}^{k} \leftarrow \text{AGGREGATE}_k\left(\left\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\right\}\right)$$

$$\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}\left(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k\right)\right)$$

Aggregate 和 Concat 一般是启发式的设计 (empirical intuition, heuristics, and experimental trial-and- error)

1. 如何得到最好的表征能力(representational capacity)

2. 表征能力的上限是什么

表征能力

表征能力

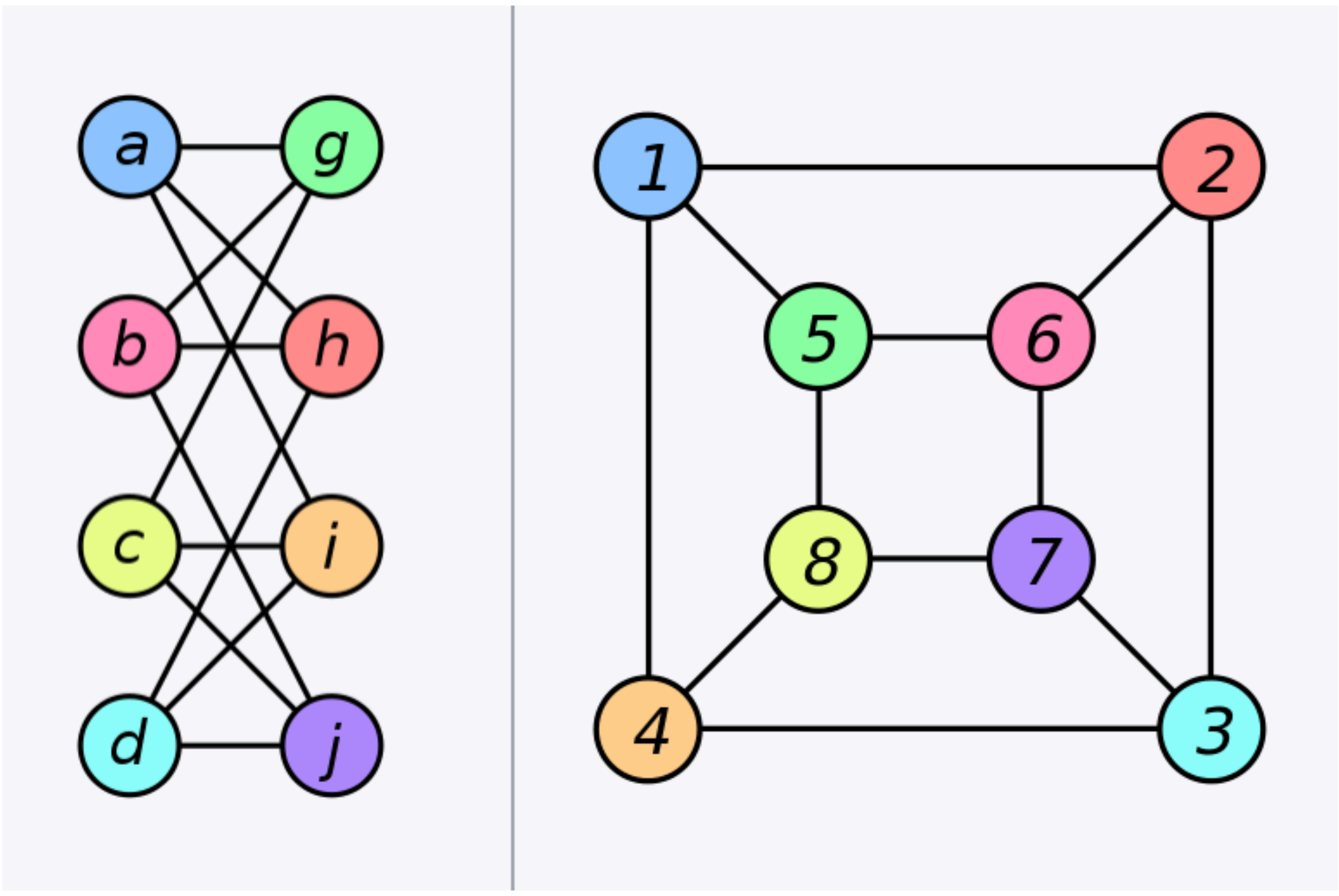
不同局部结构的节点， 嵌入的位置不同
不同拓扑结构的图， 嵌入的位置不同

表征能力评价

图同构 graph isomorphism ： GNN 能将不同结构的图嵌入到不同的位置

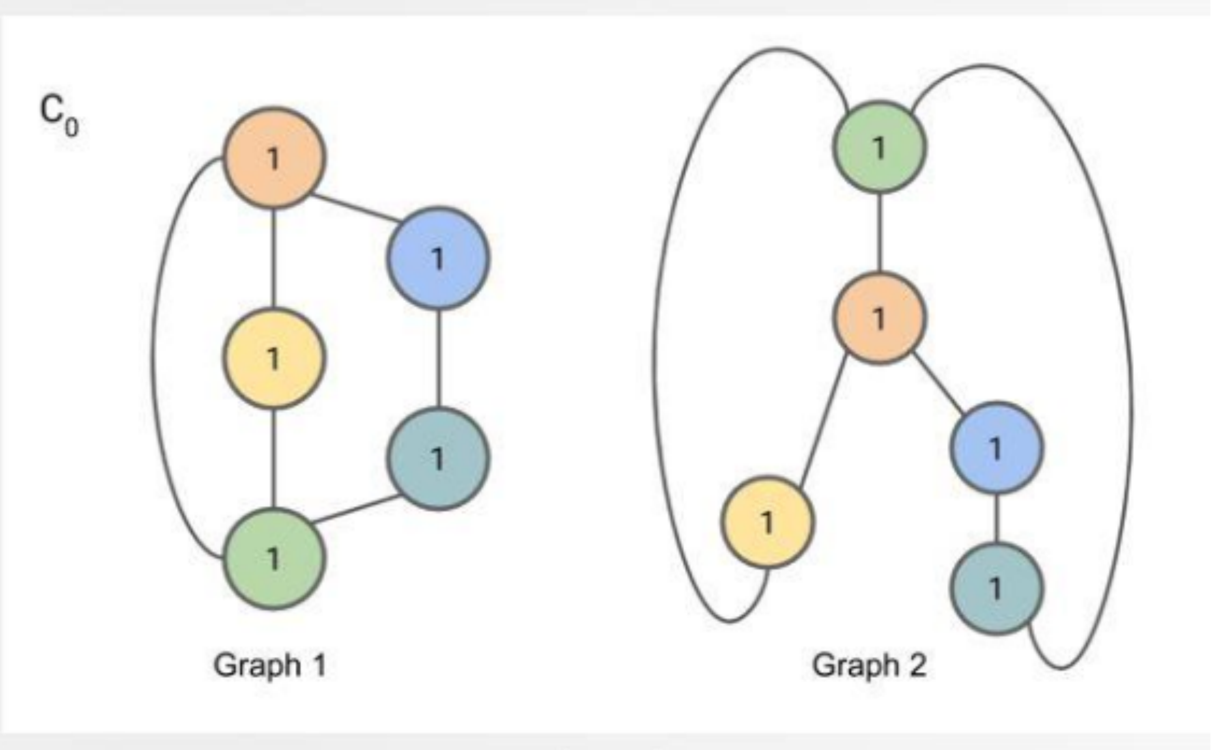
图同构的评价标准： **WL test**

相同节点数的图， 节点之间
一一映射，邻接矩阵相同

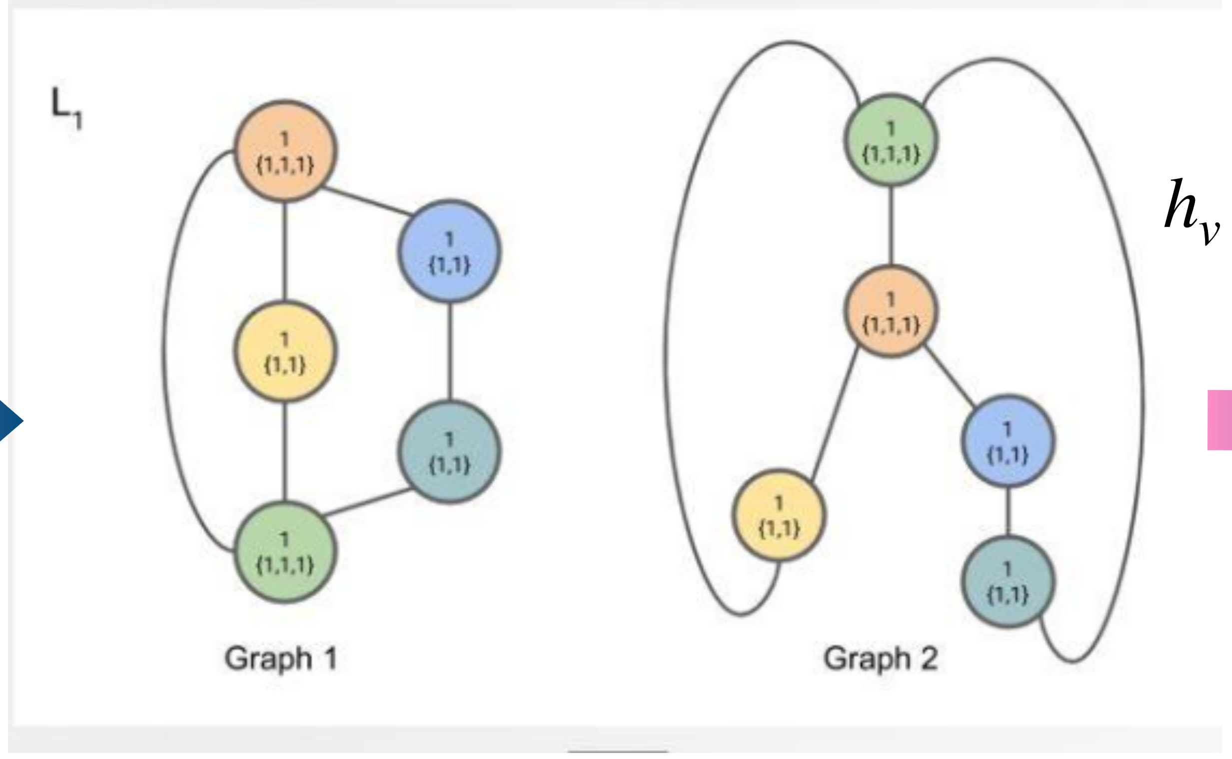


GNN 能否达到 WL test 的表征能力

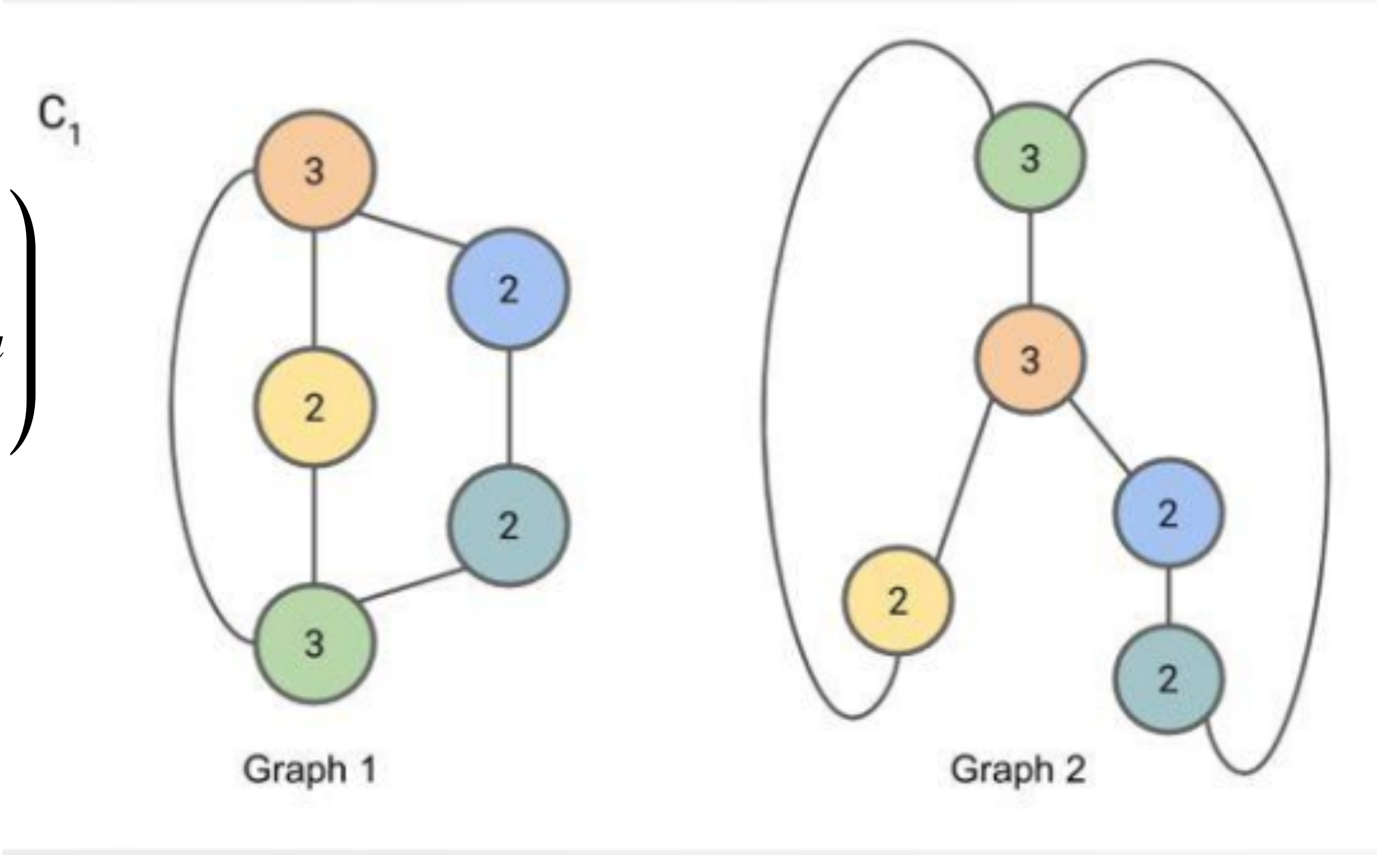
WL test



聚合



$$h_v \leftarrow \mathrm{hash}\left(\sum_{u \in \mathcal{N}(v)} h_u\right)$$



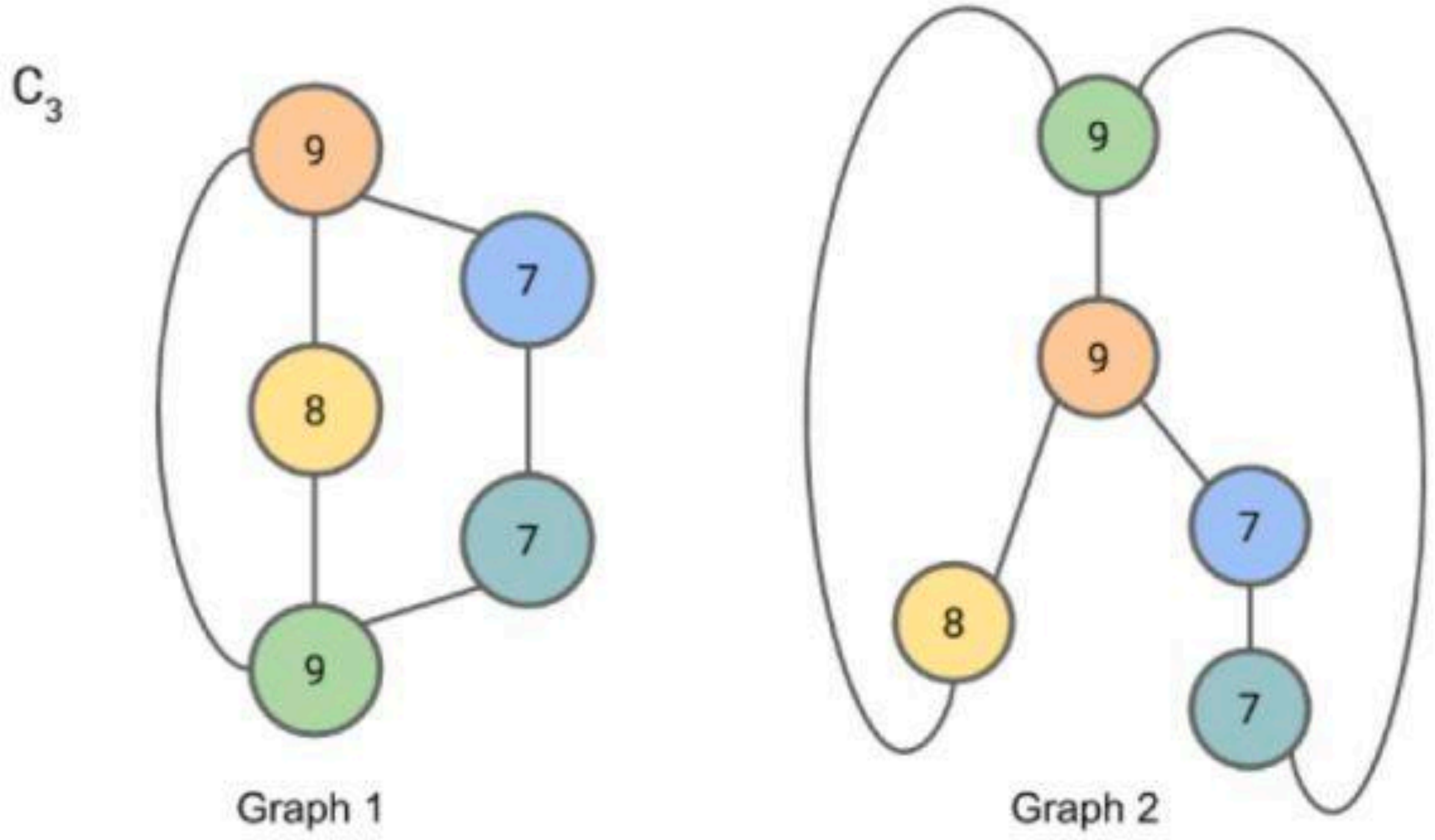
迭代

稳定时：统计各个label的分布

图1： 1个8，2个7，2个9

图2： 1个8，2个7，2个9

则，我们不排除其同构的可能性



GNN 理论分析

结论1：GNN 的表征能力上界是 WL test

Lemma 2. *Let G_1 and G_2 be any two non-isomorphic graphs. If a graph neural network $\mathcal{A} : \mathcal{G} \rightarrow \mathbb{R}^d$ 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.*

结论2：GNN 聚合函数是单射函数， 那么 GNN 的表征能力和 WL 相同

Theorem 3. *Let $\mathcal{A} : \mathcal{G} \rightarrow \mathbb{R}^d$ 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.

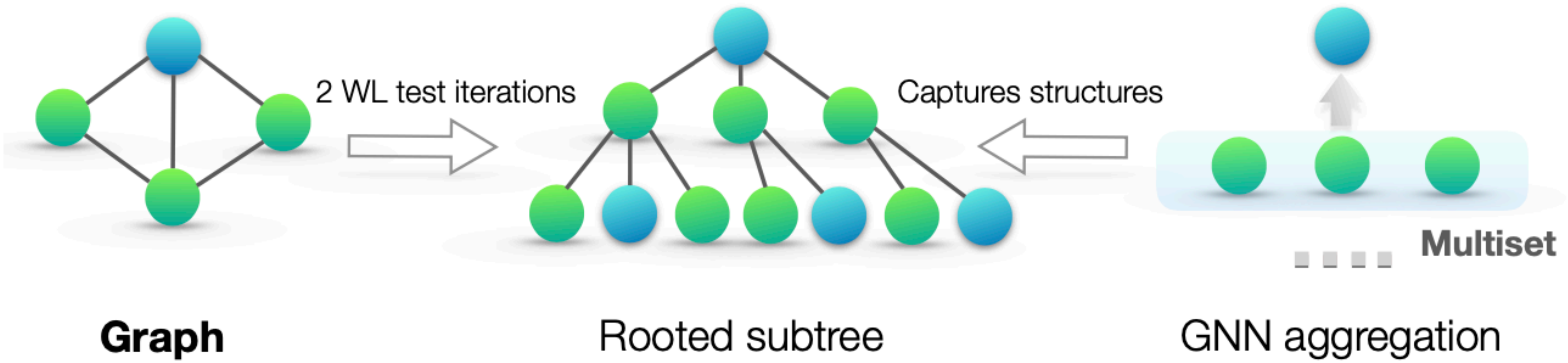
b) *$\mathcal{A}$'s graph-level readout, which operates on the multiset of node features $\left\{h_v^{(k)}\right\}$, is injective.*

Injective

Injective

1. **GNN 聚合框架看作是 multiset 函数**
2. **GNN 聚合函数应该是 单射(不同的结构，映射不同的结果）**

multiset 包含重复元素的集合



$$h_v \leftarrow \text{hash}\left(\sum_{u \in \mathcal{N}(v)} h_u\right)$$

单射

$$\mathbf{h}^k_{\mathcal{N}(v)} \leftarrow \text{AGGREGATE}_k\left(\left\{\mathbf{h}^{k-1}_u, \forall u \in \mathcal{N}(v)\right\}\right)$$

不一定单射

Injective Function

✱ 推论： 如何设计 injective 函数

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.*

$f()$ *and* $\varphi()$ 都只是证明其存在性，无具体形式

根据 universal approximation theorem，我们可以使用 MLP 近似 $f()$ *and* $\varphi()$

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

PS:

MLP 拟合的是 composition of functions, $f()$ *and* $\varphi()$

ϵ 可以取常数(实验中取 0)，或者是可学习的参数　　为什么是 无理数

GIN VS GNNs

GIN

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

GCN

$$Z = f(X, A) = \mathrm{softmax}\left(\hat{A}\,\mathrm{ReLU}\left(\hat{A}XW^{(0)}\right)W^{(1)}\right)$$

GraphSAGE

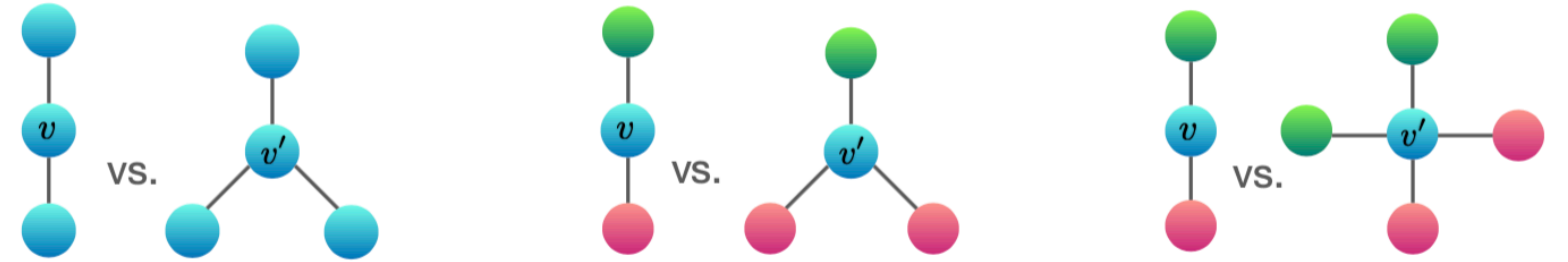
$$\mathbf{h}^k_{\mathcal{N}(v)} \leftarrow \mathrm{AGGREGATE}_k\left(\left\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\right\}\right)$$
$$\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \mathrm{CONCAT}\left(\mathbf{h}_v^{k-1}, \mathbf{h}^k_{\mathcal{N}(v)}\right)\right)$$

多层感知机 VS 单层感知机

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

单层感知机近似于线性映射，难于区分

GIN VS GNNs



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

Figure 3: **Examples of graph structures that mean and max aggregators fail to distinguish.**

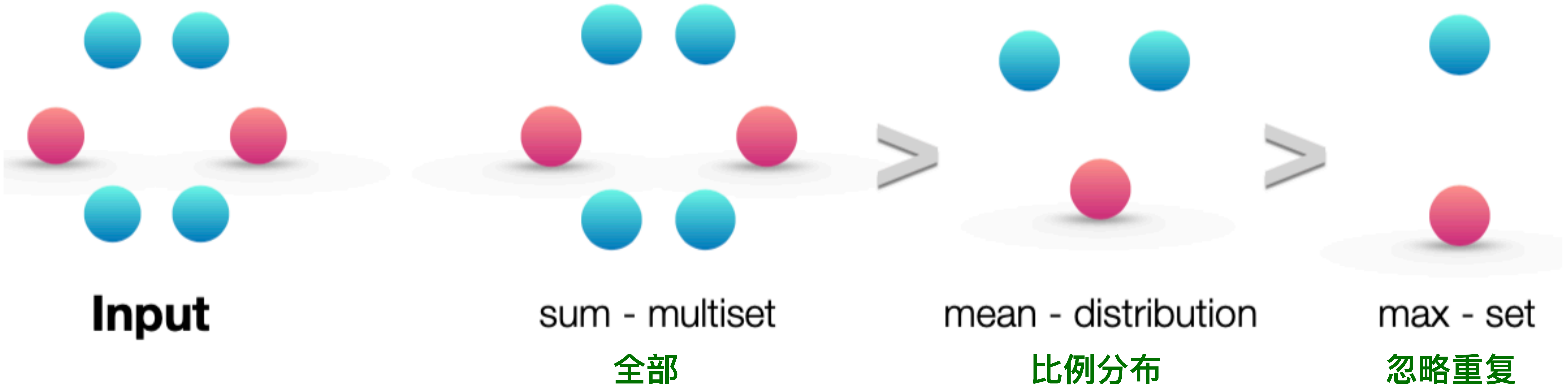


Figure 2: **Ranking by expressive power for sum, mean and max aggregators over a multiset.**

补充

MEAN Learns distribution

- the statistical and distributional information in the graph is more important than the exact structure.
- the node features are diverse and rarely repeat

所以节点分类任务较好，因为特征难重复

MAX-POOLING learns sets with distinct elements

- suitable for tasks where it is important to identify representative elements or the “skeleton”

Results

	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	Sum–MLP (**GIN-0**)	$\mathbf{75.1 \pm 5.1}$	$\mathbf{52.3 \pm 2.8}$	$\mathbf{92.4 \pm 2.5}$	$\mathbf{57.5 \pm 1.5}$	$\mathbf{80.2 \pm 1.9}$	$\mathbf{89.4 \pm 5.6}$	$\mathbf{76.2 \pm 2.8}$	$\mathbf{64.6 \pm 7.0}$	$\mathbf{82.7 \pm 1.7}$
	Sum–MLP (**GIN-ϵ**)	$\mathbf{74.3 \pm 5.1}$	$\mathbf{52.1 \pm 3.6}$	$\mathbf{92.2 \pm 2.3}$	$\mathbf{57.0 \pm 1.7}$	$\mathbf{80.1 \pm 1.9}$	$\mathbf{89.0 \pm 6.0}$	$\mathbf{75.9 \pm 3.8}$	63.7 ± 8.2	$\mathbf{82.7 \pm 1.6}$
	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	20.0 ± 0.0	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 (GCN)	74.0 ± 3.4	51.9 ± 3.8	50.0 ± 0.0	20.0 ± 0.0	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 (GraphSAGE)	72.3 ± 5.3	50.9 ± 2.2	–	–	–	85.1 ± 7.6	75.9 ± 3.2	63.9 ± 7.7	77.7 ± 1.5

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

分析讨论

- [x] 本文给出基于 Aggregation 的 GNNs 的上界
- [ ] 所以难以考虑网络连边的方向和权重
- [ ] 基于 LSTM 和 Attention 的框架没有考虑

- [x] 本文的理论基础： 节点特征是离散的，有限的
- [ ] 特征空间如果连续，难于分析

- [x] 主要适用于网络结构相关的任务
- [ ] 节点分类任务不具有优势

Model Explain

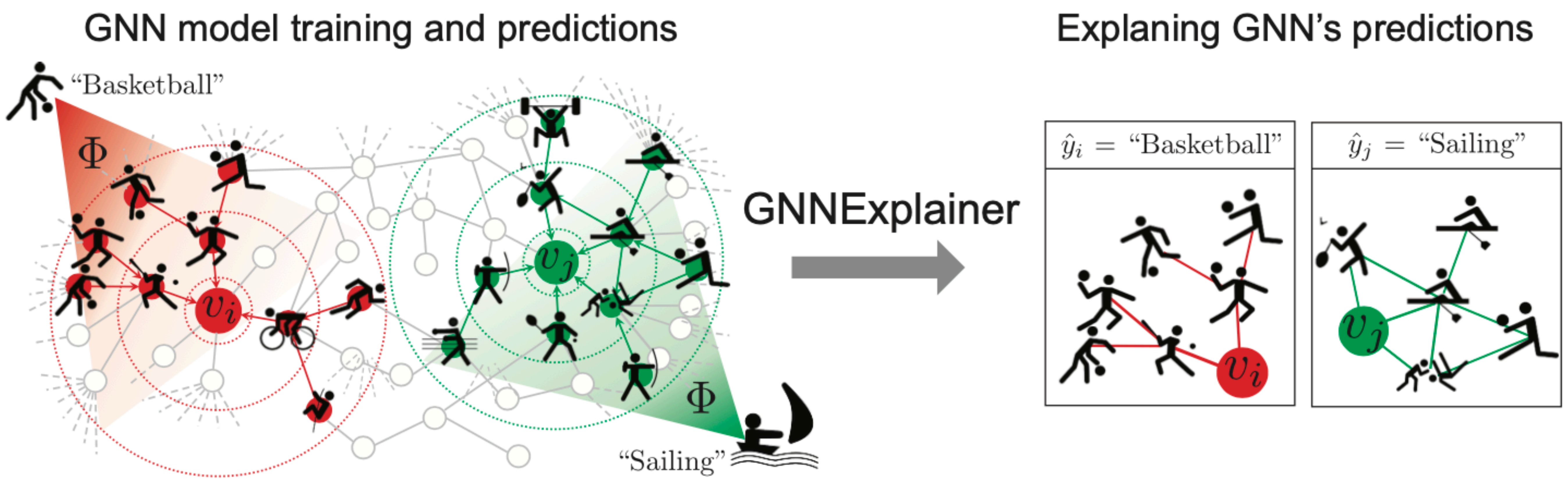
GNNExplainer: Generating Explanations for Graph Neural Networks

an optimization task that maximizes the mutual information between a GNN's prediction and distribution of possible subgraph structures

模型解释路线

1. **surrogate models 进行逼近**
2. **检查模型的相关特性： 高阶特征的定性解释或者有影响的输入**

GNNExplainer 考虑到图的特殊性， 节点之间有强依赖关系



找出最影响预测的节点 (xgboost)

Problem formulation

GNN 模型分解为三部分：

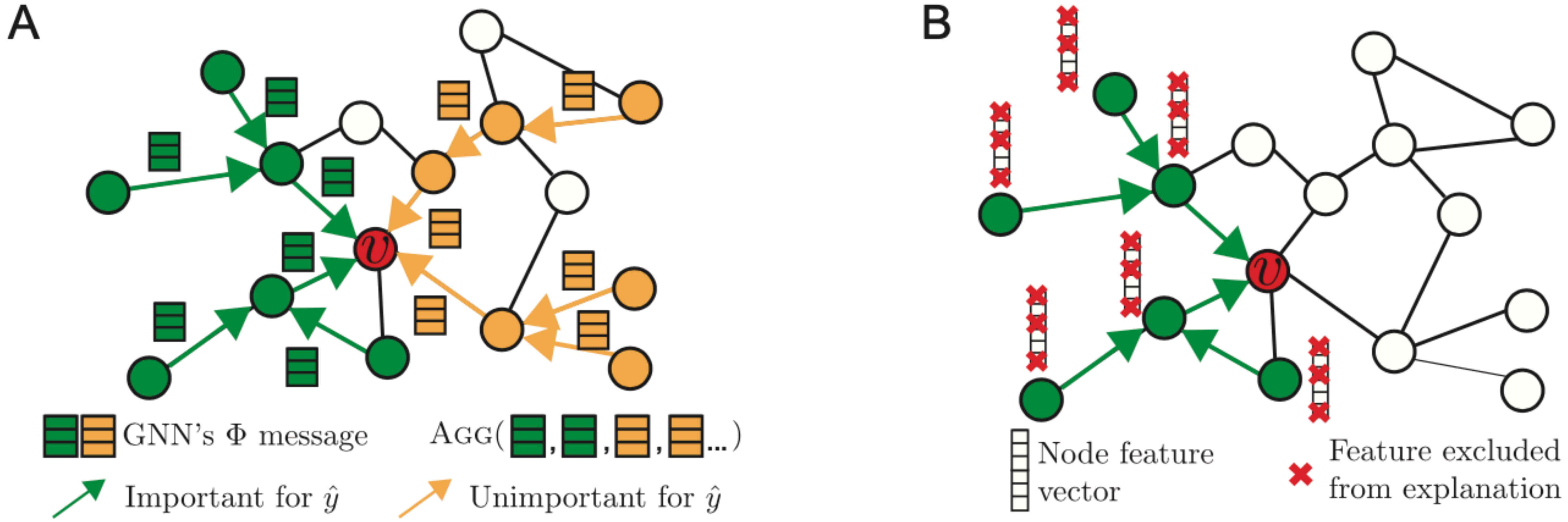
- MSG : neural messages between every pair of nodes.
- AGG : Aggregate messages from neighborhood
- UPDATE : non-linearly transforms to obtain final representation

问题关键：　the computation graph of node v ：

(the information the GNN uses to generate prediction)

Computation graph $G_c(v)$

Node features $X_c(v)$



GNNExplainer: Subgraph

$$G_S \subseteq G_c \qquad X_S = \left\{ x_j \mid v_j \in G_S \right\}$$

1. 利用互信息 mutual information(MI)

$$\max_{G_S} MI\left(Y, (G_S, X_S)\right) = H(Y) - H\left(Y \mid G = G_S, X = X_S\right)$$

2. MI 量化预测概率的变化　　例如： 移除一个节点， 对预测概率有较大的影响，那么这个节点是属于预测的 subgraph

3. 其中 熵 H(Y) 是固定不变

$$minH\left(Y \mid G = G_S, X = X_S\right)$$

$$H\left(Y \mid G = G_S, X = X_S\right) = -\mathbb{E}_{Y|G_S,X_S}\left[\log P_\Phi\left(Y \mid G = G_S, X = X_S\right)\right]$$

$$\min_{\mathcal{G}} H\left(Y \mid G = \mathbb{E}_{\mathcal{G}}\left[G_S\right], X = X_S\right)$$

$$\mathbb{E}_{\mathcal{G}}\left[G_S\right] = A_c \odot \sigma(M)$$

M 是待学习的 mask

GNNExplainer: Feature

定义一个特征选择 F $$F \in \{0,1\}^d$$

$$X_S^F = \left\{ x_j^F \mid v_j \in G_S \right\}, \quad x_j^F = \left[x_{j,t_1}, \ldots, x_{j,t_k} \right] \text{ for } F_{t_i} = 1$$

$$\max_{GS,F} MI\left(Y, \left(G_S, F\right)\right) = H(Y) - H\left(Y \mid G = G_S, \boxed{X = X_S^F}\right)$$

$$X_S^F = X_S \odot F$$

GNNExplainer

- Any machine learning task on graphs.
- Any GNN model.

- train a single GNN
- use GNNExplainer to explain the predictions made by the GNN

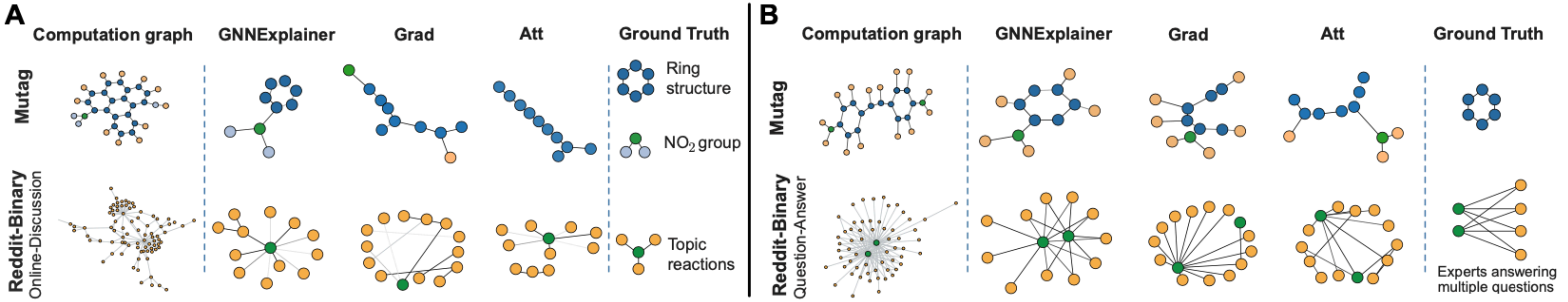


Figure 4: Evaluation of single-instance explanations. **A-B.** Shown are exemplar explanation subgraphs for graph classification task on two datasets, MUTAG and REDDIT-BINARY.

GCN : Dynamic System

Graph Neural Networks Exponentially Lose Expressive Power for Node Classification

1. Investigate the expressive power of GNNs by analyzing their **asymptotic behaviors** as the layer size goes to infinity.
2. To generalize the forward propagation of a Graph Convolutional Network as a **specific dynamical system**

Conclusion

Our theory gives new theoretical conditions under which neither **layer stacking nor non-linearity** contributes to improving expressive power

When a graph is dense, graph convolution operations mix signals on nodes and move them closer to each other quickly.
*From this theorem, we can hypothesize that deep graph NNs perform poorly due to **information loss** via signal mixing by graph convolutions.*

Theorem

有限的离散的马尔可夫过程(irreducible and aperiodic)， 将指数性的收敛到一个唯一的一个平衡态，收敛的速率和转移概率矩阵的特征值有关

GCN 区别于 马尔可夫过程， GCN 中存在着非线性函数

Theorem 2. *For any initial value* $X^{(0)}$*, the output of l-th layer* $X^{(l)}$ *satisfies* $d_{\mathcal{M}}(X^{(l)}) \leq (s\lambda)^l d_{\mathcal{M}}(X^{(0)})$*. In particular,* $d_{\mathcal{M}}(X^{(l)})$ *exponentially converges to* 0 *when* $s\lambda < 1$.