EECE 571F: Deep Learning with Structures

Lecture 5: Expressiveness & Generalization of Graph Neural Networks

Renjie Liao

University of British Columbia

Winter, Term 1, 2022

Course Scope

- Brief Intro to Deep Learning

- Geometric Deep Learning

    - Deep Learning Models for Sets and Sequences: Deep Sets & Transformers
    - Deep Learning Models for Graphs: Graph Convolution & Message Passing GNNs
    - Expressiveness & Generalizations of GNNs
    - Unsupervised/Self-supervised Graph Representation Learning

- Probabilistic Deep Learning

    - Deep Generative Models:
      Auto-regressive models, GANs, VAEs, Diffusion/Score based models
    - Discrete/Hybrid Latent Variable Models: RBMs, Latent Graph Models
    - Stochastic Gradient Estimation

Course Scope

- Brief Intro to Deep Learning

- Geometric Deep Learning

    - Deep Learning Models for Sets and Sequences: Deep Sets & Transformers
    - Deep Learning Models for Graphs: Graph Convolution & Message Passing GNNs
    - **Expressiveness & Generalizations of GNNs**
    - Unsupervised/Self-supervised Graph Representation Learning

- Probabilistic Deep Learning

    - Deep Generative Models:
      Auto-regressive models, GANs, VAEs, Diffusion/Score based models
    - Discrete/Hybrid Latent Variable Models: RBMs, Latent Graph Models
    - Stochastic Gradient Estimation

Theoretical Aspects of GNNs

- Expressiveness / Capacity

Theoretical Aspects of GNNs

- Expressiveness / Capacity

  What functions on graphs can be represented by GNNs?

  Can GNNs distinguish isomorphic graphs?

  Are GNNs universal approximators?

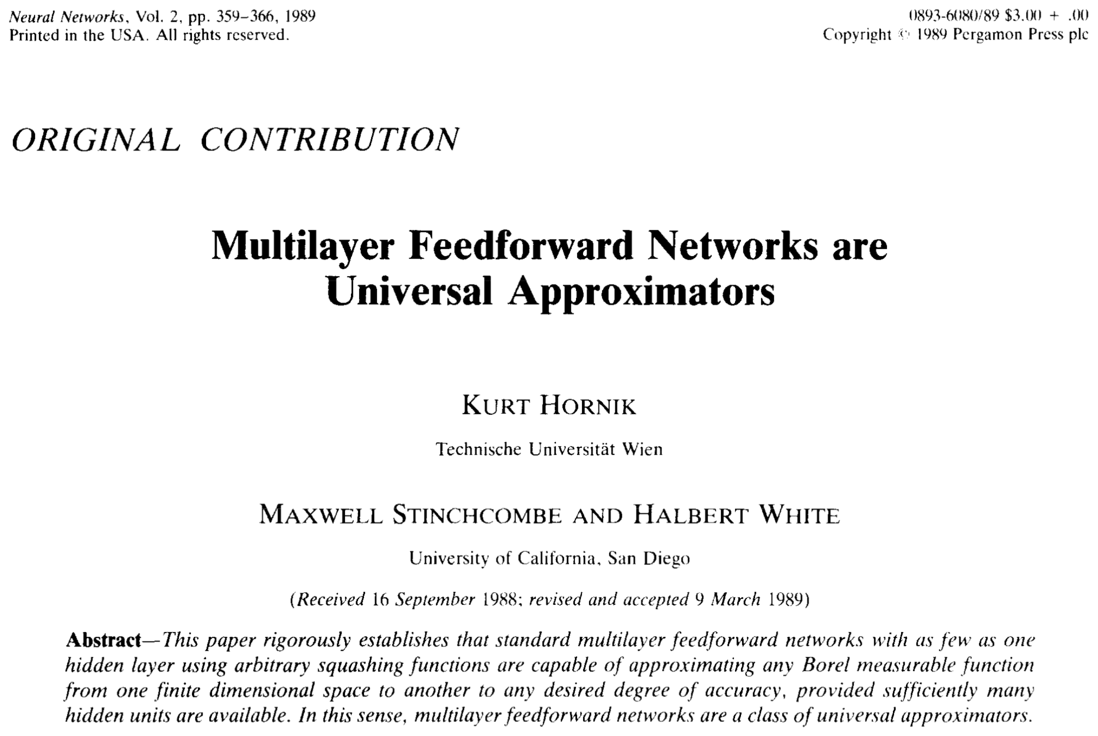
Theoretical Aspects of GNNs

- Expressiveness / Capacity

What functions on graphs can be represented by GNNs?

Can GNNs distinguish isomorphic graphs?

Are GNNs universal approximators?



Neural Networks, Vol. 2, pp. 359–366, 1989
Printed in the USA. All rights reserved.

0893-6080/89 $3.00 + .00
Copyright © 1989 Pergamon Press plc

ORIGINAL CONTRIBUTION

Multilayer Feedforward Networks are Universal Approximators

KURT HORNIK

Technische Universität Wien

MAXWELL STINCHCOMBE AND HALBERT WHITE

University of California, San Diego

(*Received* 16 *September* 1988; *revised and accepted* 9 *March* 1989)

Abstract—*This paper rigorously establishes that standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are a class of universal approximators.*

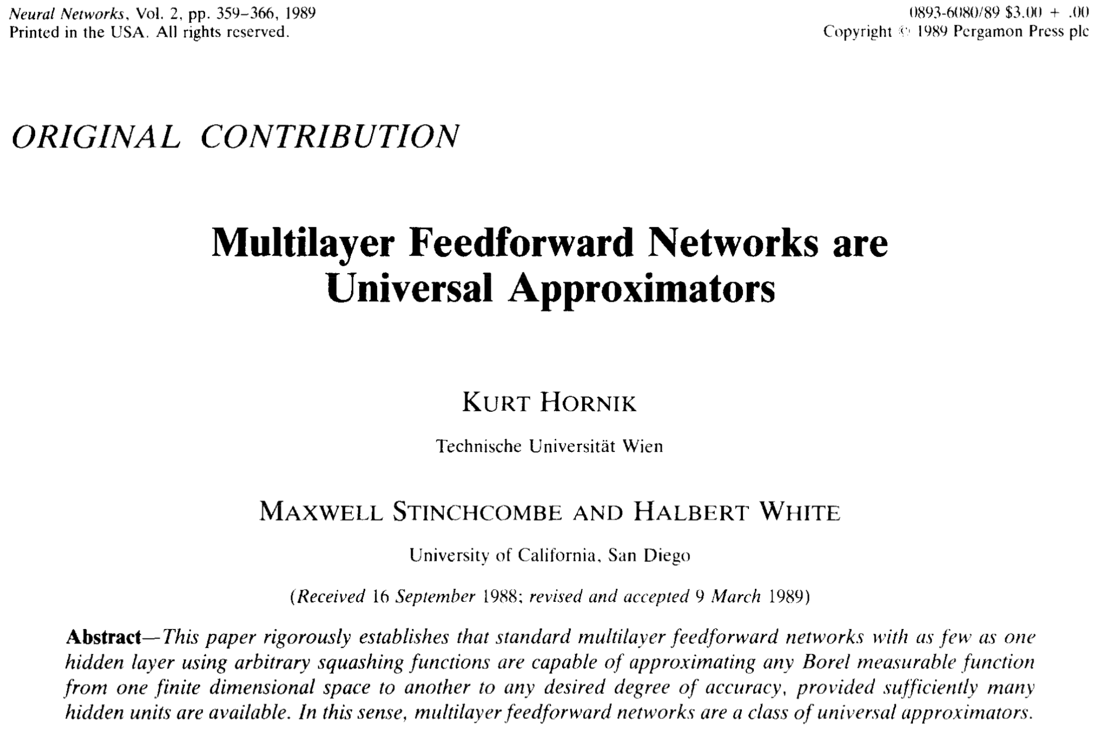
Theoretical Aspects of GNNs

- Expressiveness / Capacity Model/Hypothesis Class

What functions on graphs can be represented by GNNs?

Can GNNs distinguish isomorphic graphs?

Are GNNs universal approximators?



Neural Networks, Vol. 2, pp. 359–366, 1989
Printed in the USA. All rights reserved.

0893-6080/89 $3.00 + .00

ORIGINAL CONTRIBUTION

Multilayer Feedforward Networks are Universal Approximators

KURT HORNIK

Technische Universität Wien

MAXWELL STINCHCOMBE AND HALBERT WHITE

University of California, San Diego

(*Received* 16 *September* 1988; *revised and accepted* 9 *March* 1989)

Abstract—*This paper rigorously establishes that standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are a class of universal approximators.*

Theoretical Aspects of GNNs

- Expressiveness / Capacity Model/Hypothesis Class

  What functions on graphs can be represented by GNNs?

  Can GNNs distinguish isomorphic graphs?

  Are GNNs universal approximators?

- Generalization

Theoretical Aspects of GNNs

- Expressiveness / Capacity Model/Hypothesis Class

  What functions on graphs can be represented by GNNs?

  Can GNNs distinguish isomorphic graphs?

  Are GNNs universal approximators?

- Generalization

  How well do GNNs generalize to unseen graphs?

  What are factors that affect the generalization of GNNs?

Theoretical Aspects of GNNs

- Expressiveness / Capacity   Model/Hypothesis Class

  What functions on graphs can be represented by GNNs?

  Can GNNs distinguish isomorphic graphs?

  Are GNNs universal approximators?

- Generalization   Model/Hypothesis Class + Learning Process

  How well do GNNs generalize to unseen graphs?

  What are factors that affect the generalization of GNNs?

Theoretical Aspects of GNNs

- **Expressiveness / Capacity**

  What functions on graphs can be represented by GNNs?

  Can GNNs distinguish isomorphic graphs?

  Are GNNs universal approximators?

- Generalization

  How well do GNNs generalize to unseen graphs?

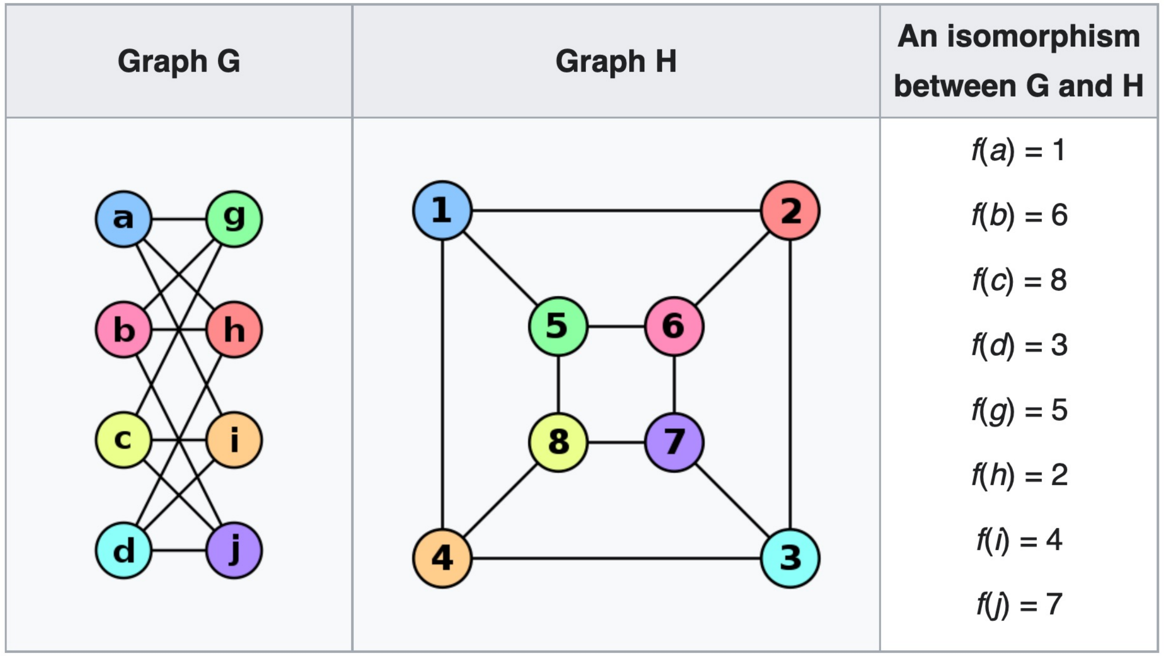
  What are factors that affect the generalization of GNNs?

Graph Isomorphism Problem

Given two graphs, are they isomorphic?

Graph Isomorphism Problem

Given two graphs, are they isomorphic?

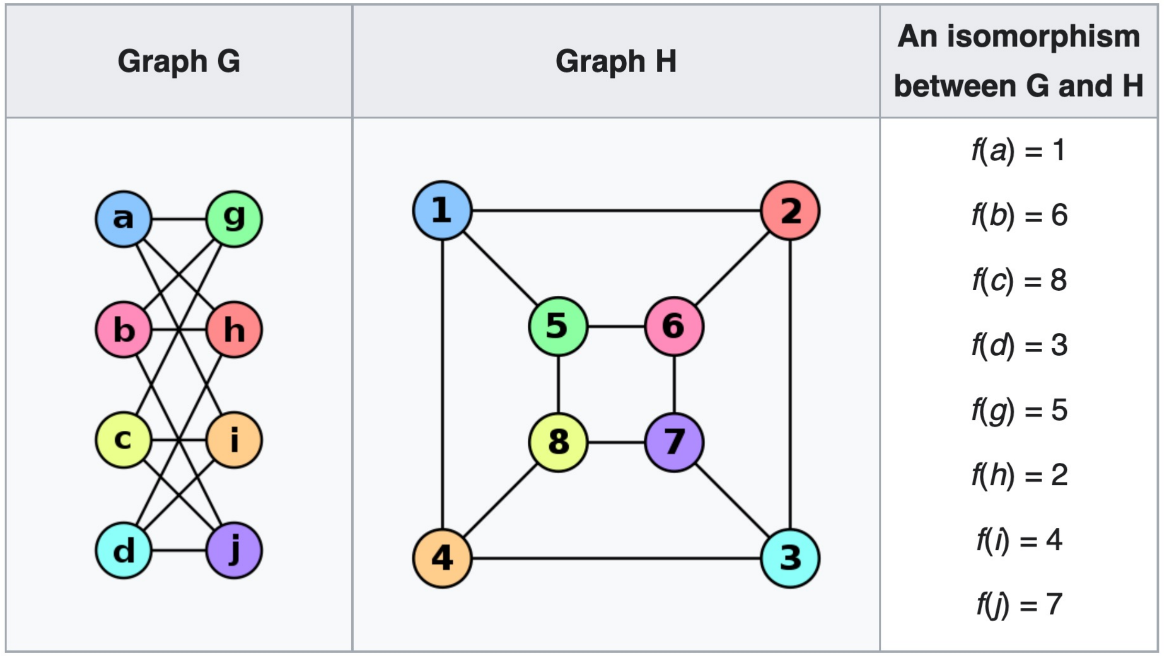


Graph G	Graph H	An isomorphism between G and H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

Image Credit: https://en.wikipedia.org/wiki/Graph_isomorphism

Graph Isomorphism Problem

Given two graphs, are they isomorphic?



Graph G	Graph H	An isomorphism between G and H
		$f(a) = 1$ $f(b) = 6$ $f(c) = 8$ $f(d) = 3$ $f(g) = 5$ $f(h) = 2$ $f(i) = 4$ $f(j) = 7$

a
g
b
h
c
i
d
j
1
2
5
6
8
7
4
3

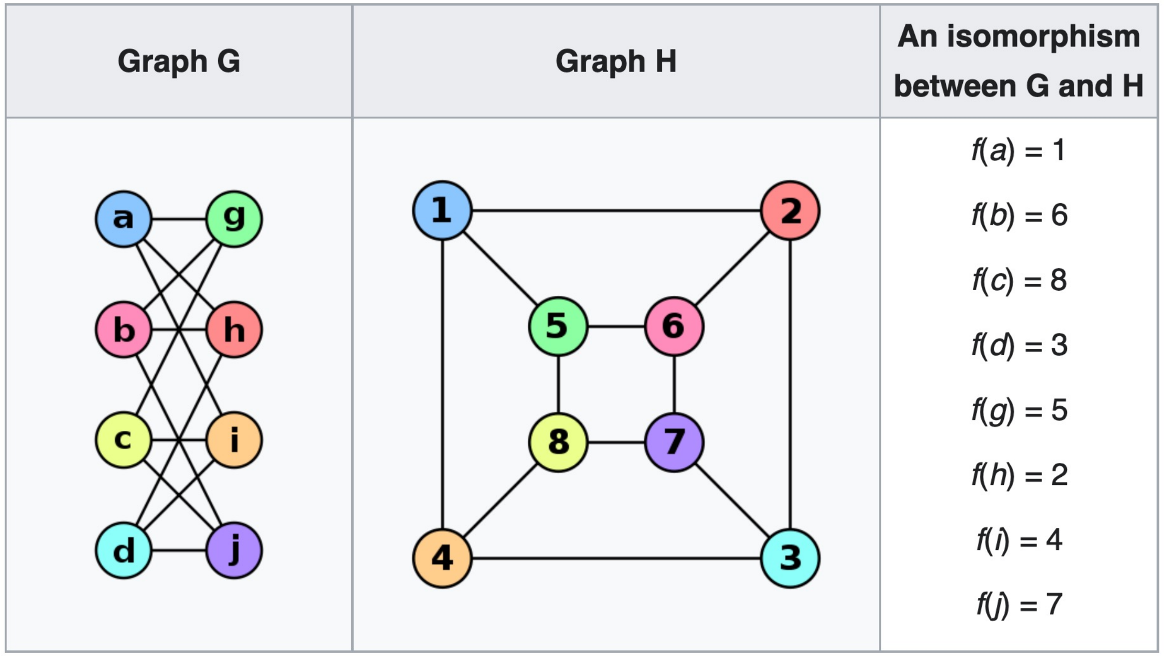
Given two adjacency matrices A_1, A_2, is there a permutation matrix P s.t. $PA_1P^\top = A_2$?

Image Credit: https://en.wikipedia.org/wiki/Graph_isomorphism

Graph Isomorphism Problem

Given two graphs, are they isomorphic?

It is unknow if this problem is in P or NP-Complete!



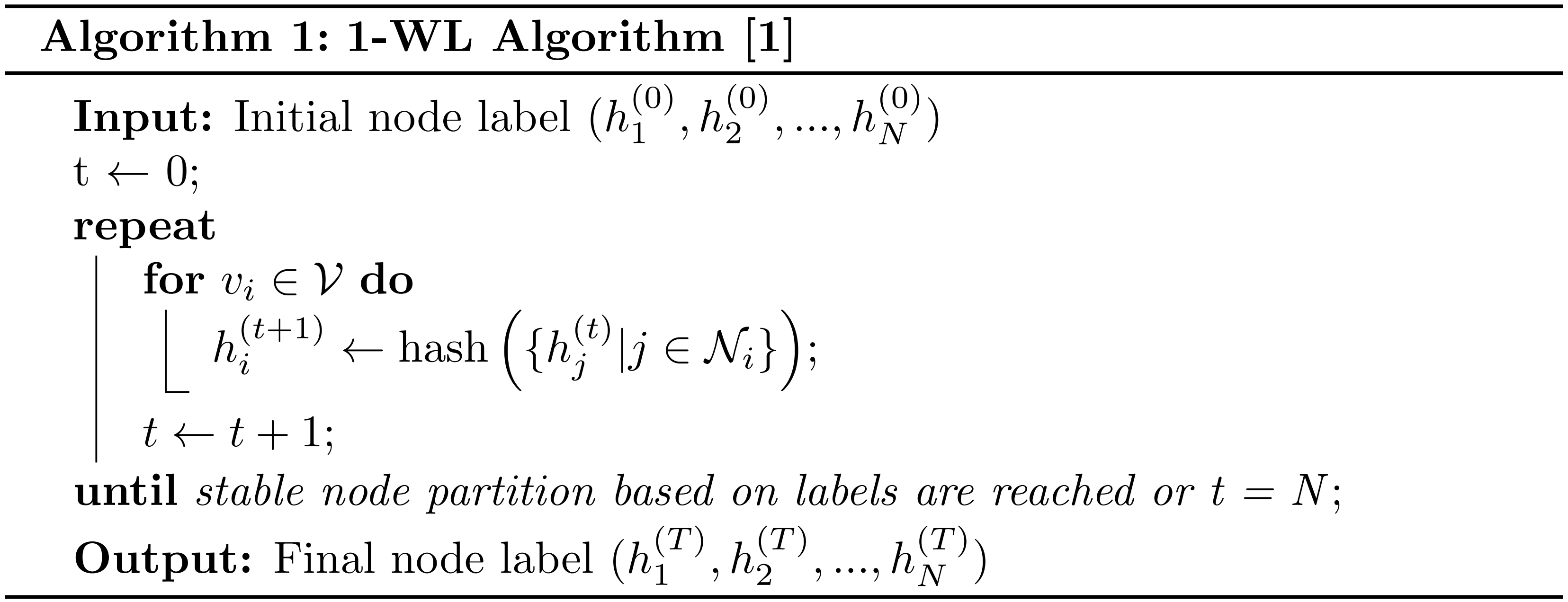
Graph G	Graph H	An isomorphism between G and H
		f(*a*) = 1 *f*(*b*) = 6 *f*(*c*) = 8 *f*(*d*) = 3 *f*(*g*) = 5 *f*(*h*) = 2 *f*(*i*) = 4 *f*(*j*) = 7

a
g
b
h
c
i
d
j
1
2
5
6
8
7
4
3

Given two adjacency matrices A_1, A_2 , is there a permutation matrix P s.t. $PA_1P^\top = A_2$?

Image Credit: https://en.wikipedia.org/wiki/Graph_isomorphism

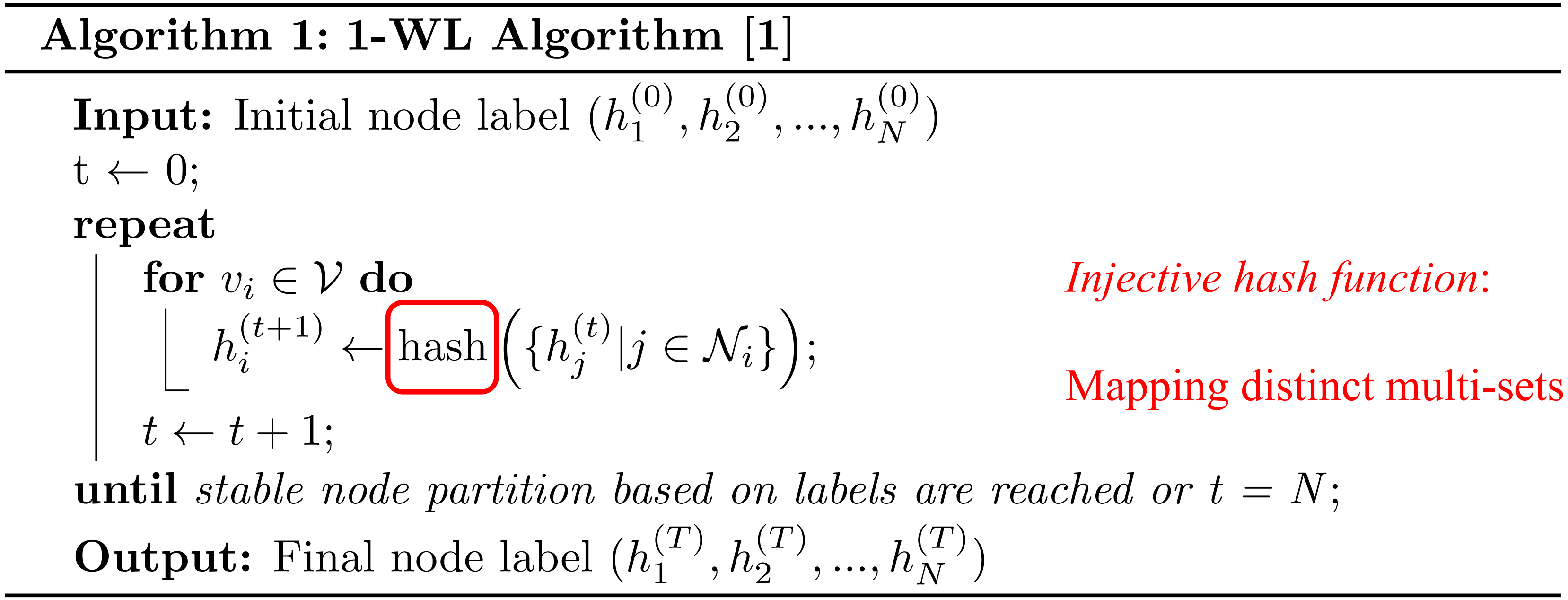
The Weisfeiler-Lehman Isomorphism Test



```
Algorithm 1: 1-WL Algorithm [1]
Input: Initial node label $(h_1^{(0)}, h_2^{(0)}, ..., h_N^{(0)})$
t ← 0;
repeat
    for $v_i \in \mathcal{V}$ do
        $h_i^{(t+1)} \leftarrow \mathrm{hash}\left(\{h_j^{(t)} | j \in \mathcal{N}_i\}\right)$;
    $t \leftarrow t + 1$;
until stable node partition based on labels are reached or $t = N$;
Output: Final node label $(h_1^{(T)}, h_2^{(T)}, ..., h_N^{(T)})$
```

[1]

The Weisfeiler-Lehman Isomorphism Test



Algorithm 1: 1-WL Algorithm [1]

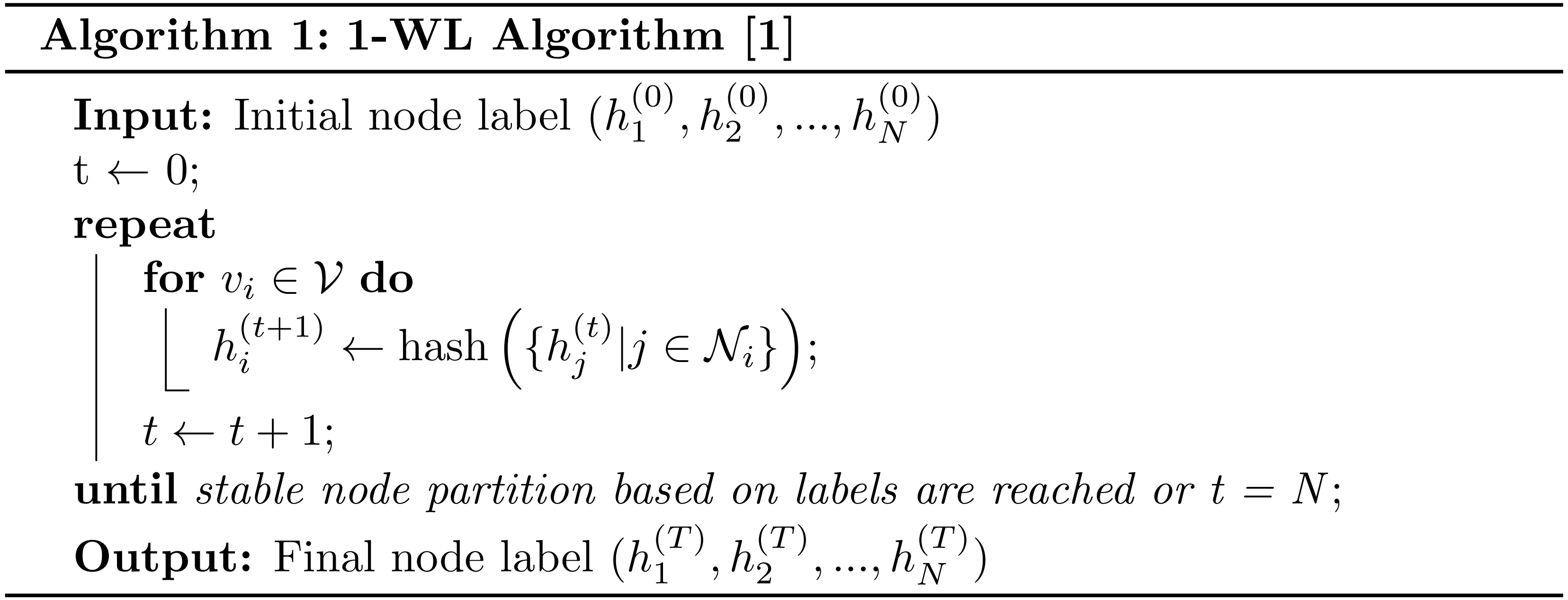
```
Input: Initial node label $(h_1^{(0)}, h_2^{(0)}, ..., h_N^{(0)})$
t ← 0;
repeat
    for $v_i \in \mathcal{V}$ do
        $h_i^{(t+1)} \leftarrow \boxed{\text{hash}}\Big(\{h_j^{(t)} | j \in \mathcal{N}_i\}\Big)$;
    $t \leftarrow t + 1$;
until stable node partition based on labels are reached or $t = N$;
Output: Final node label $(h_1^{(T)}, h_2^{(T)}, ..., h_N^{(T)})$
```

Injective hash function:

Mapping distinct multi-sets to distinct labels

[1]

The Weisfeiler-Lehman Isomorphism Test



```
Algorithm 1: 1-WL Algorithm [1]
Input: Initial node label $(h_1^{(0)}, h_2^{(0)}, ..., h_N^{(0)})$
t ← 0;
repeat
    for $v_i \in \mathcal{V}$ do
        $h_i^{(t+1)} \leftarrow \text{hash}\left(\{h_j^{(t)} | j \in \mathcal{N}_i\}\right)$;
    $t \leftarrow t + 1$;
until stable node partition based on labels are reached or $t = N$;
Output: Final node label $(h_1^{(T)}, h_2^{(T)}, ..., h_N^{(T)})$
```

If the partition of nodes by labels are different, then two graphs are non-isomorphic!

If the partition of nodes by labels are same, we can not decide!

[1]

The Weisfeiler-Lehman Isomorphism Test

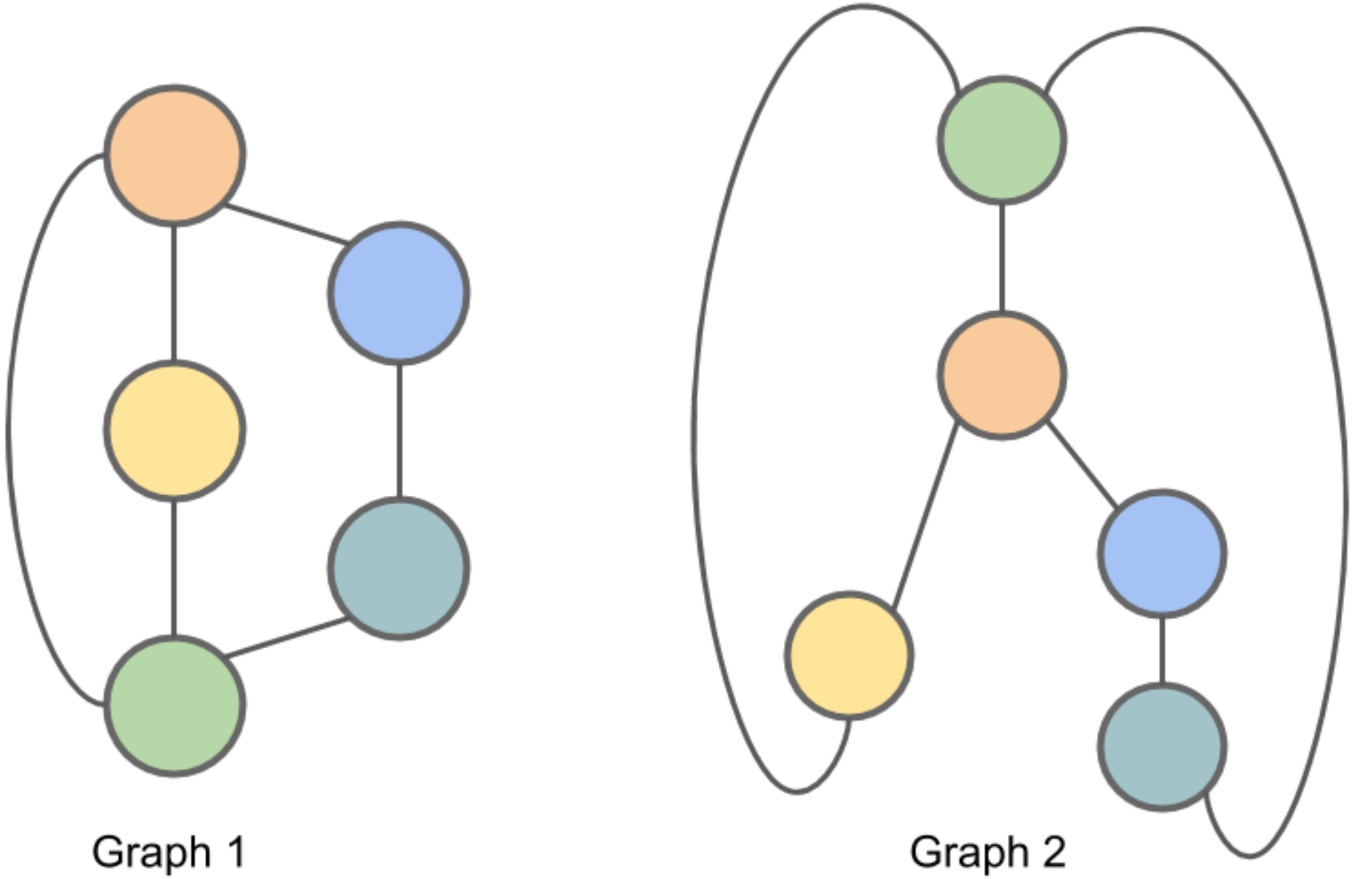
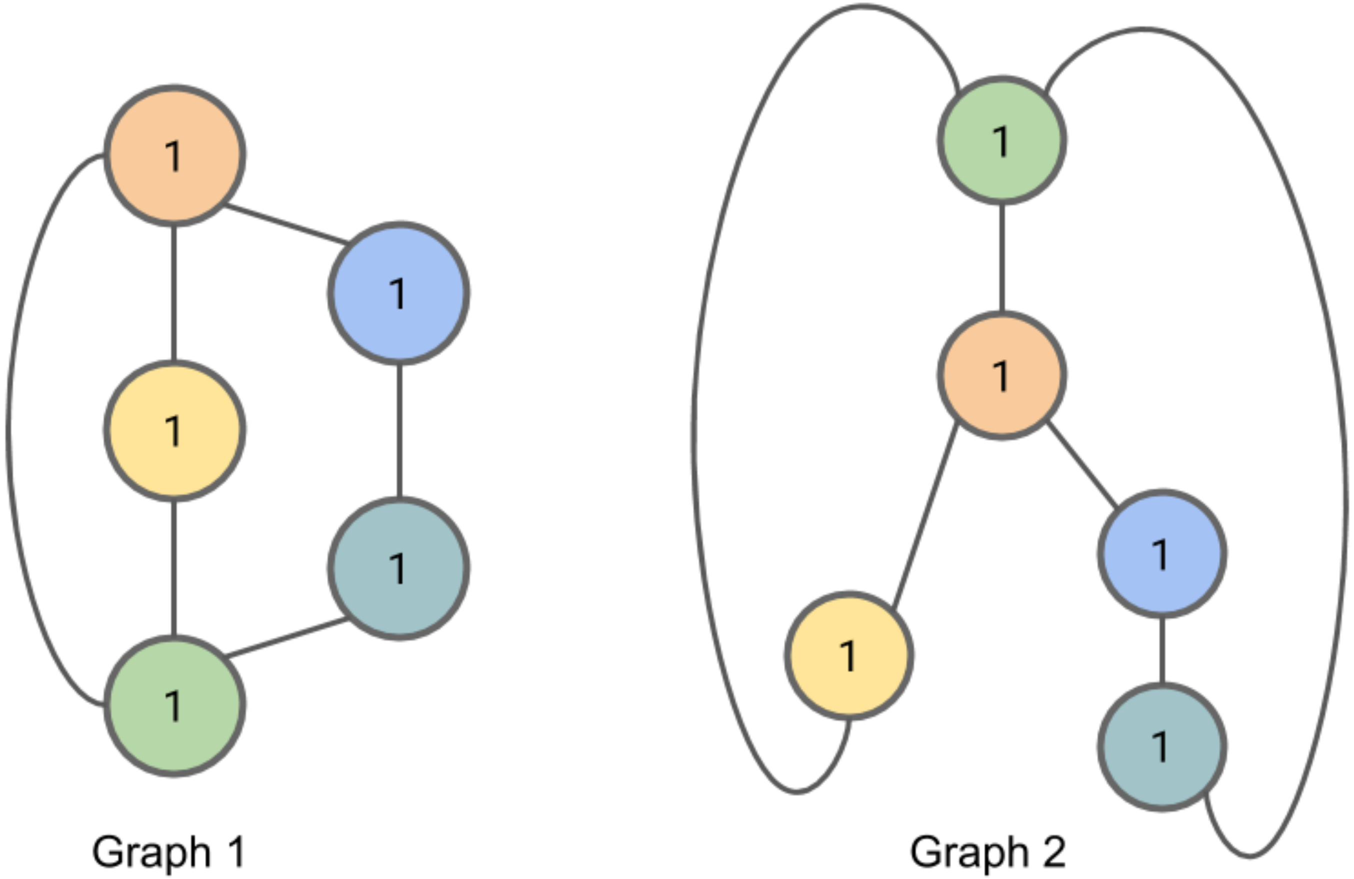


Image Credit: https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

The Weisfeiler-Lehman Isomorphism Test

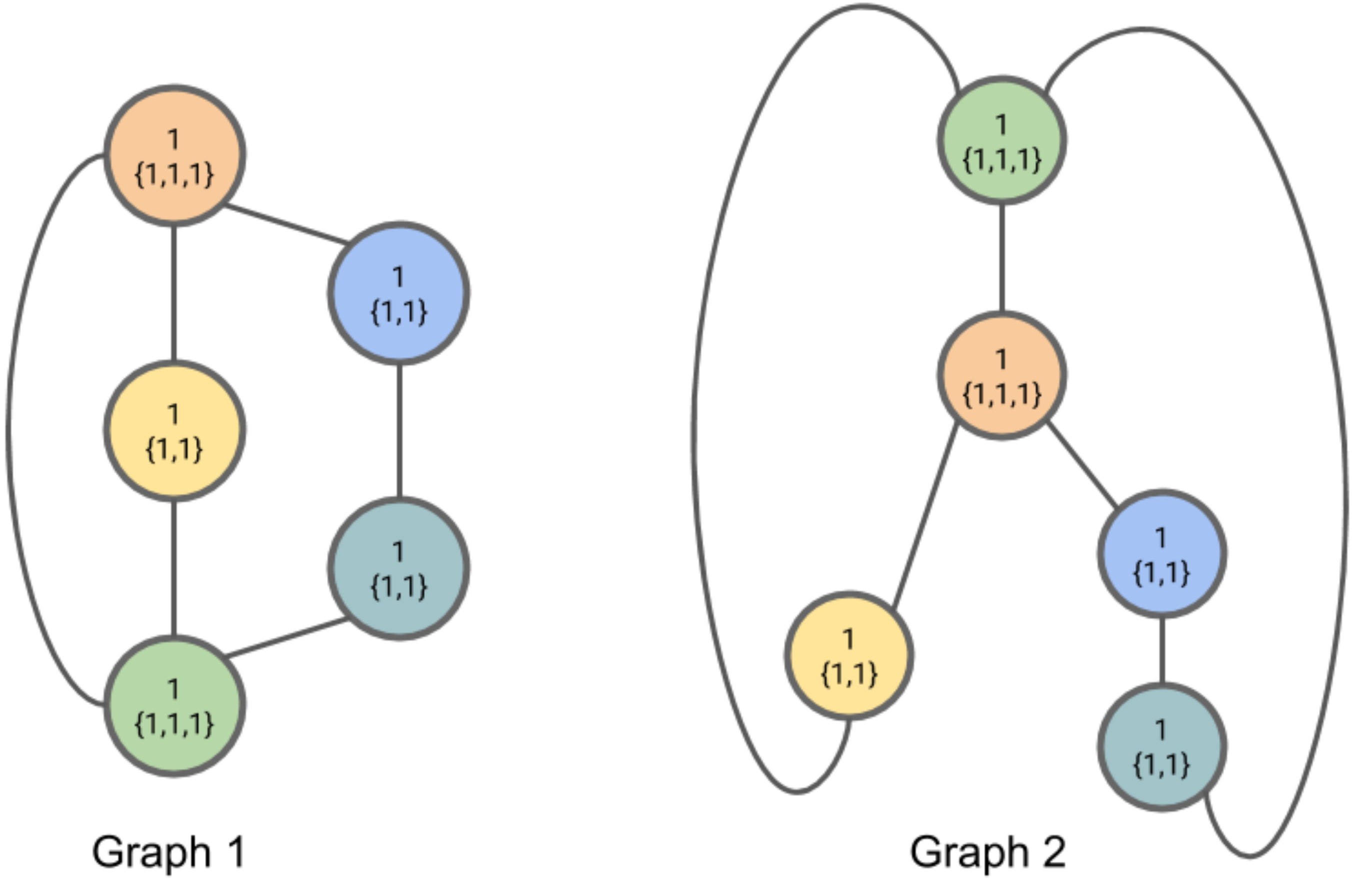


{1,1,1,1,1}

{1,1,1,1,1}

Image Credit: https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

The Weisfeiler-Lehman Isomorphism Test



Graph 1

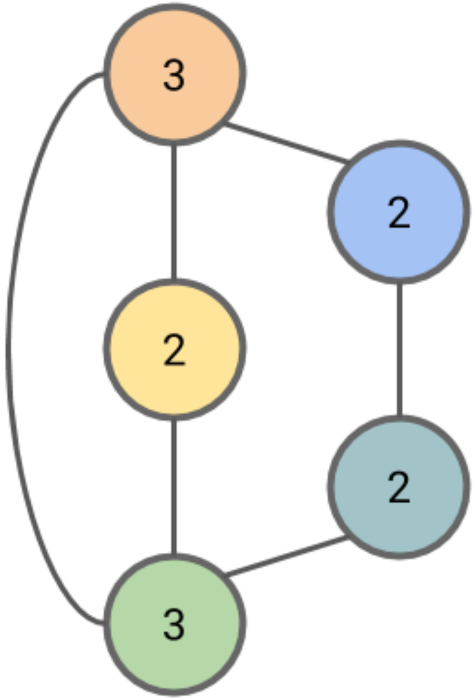
Graph 2

Hash Table:

{1,1} -> 2
{1,1,1} -> 3
{2,3} -> 4
{3,3} -> 5
{2,2,3} -> 6
{4,6} -> 7
{6,6} -> 8
{4,5,6} -> 9

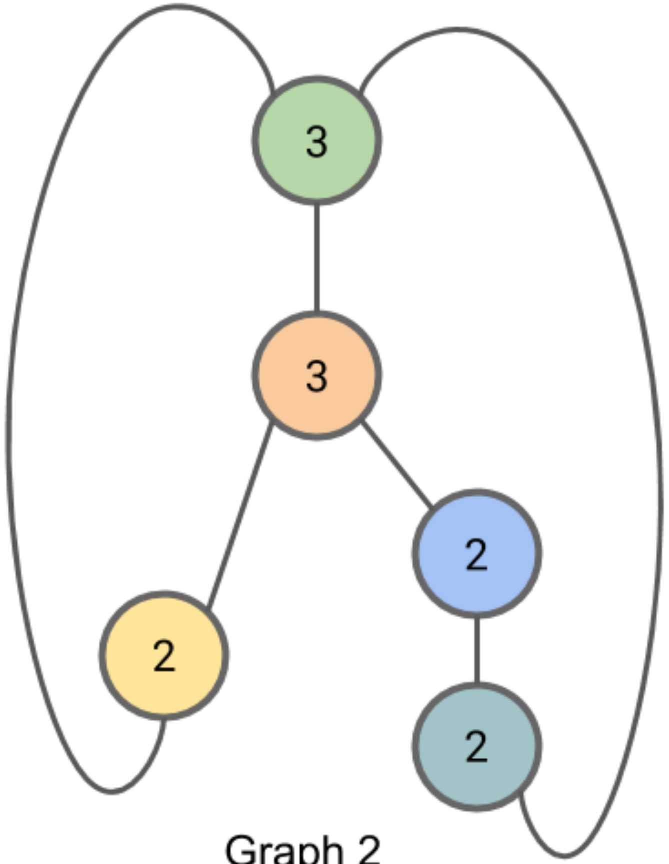
Image Credit: https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

The Weisfeiler-Lehman Isomorphism Test



Graph 1

$\{2,2,2\},\{3,3\}$

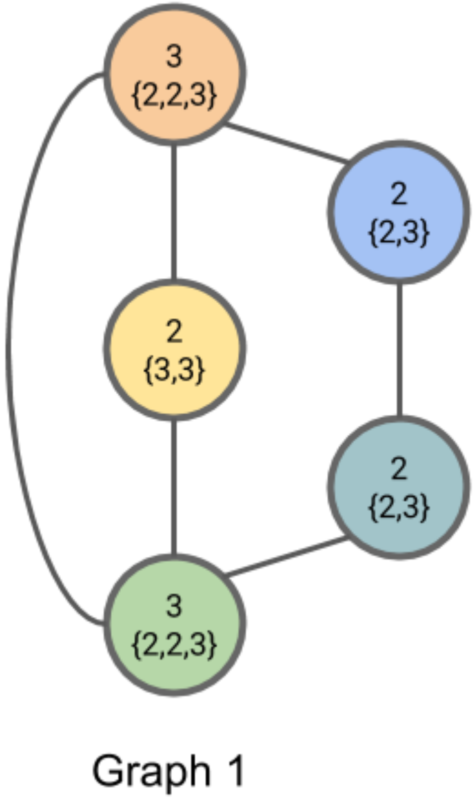


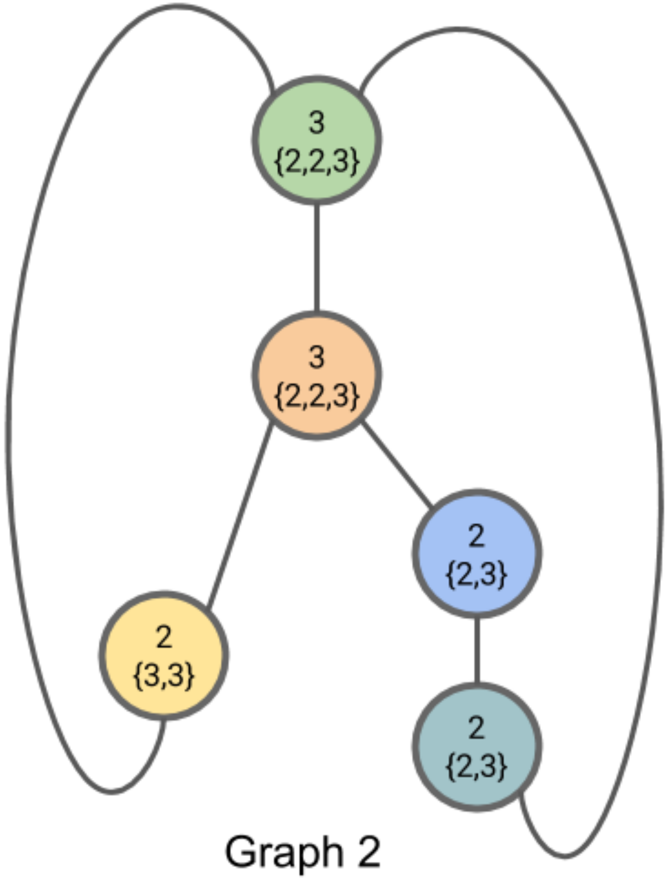
Graph 2

$\{2,2,2\},\{3,3\}$

Image Credit: https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

The Weisfeiler-Lehman Isomorphism Test



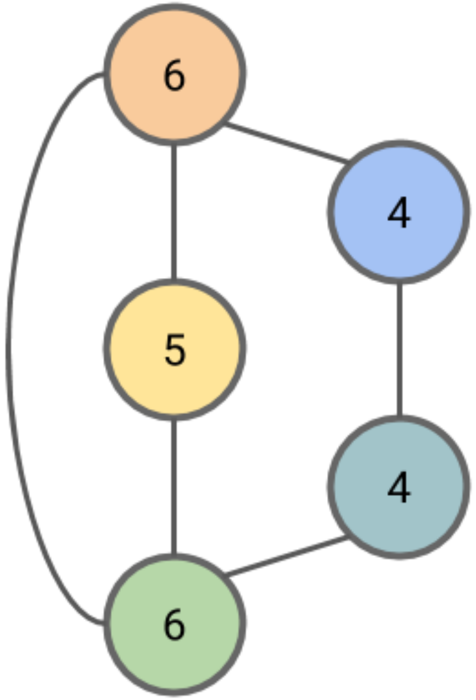


Hash Table:

{1,1}	-> 2
{1,1,1}	-> 3
{2,3}	-> 4
{3,3}	-> 5
{2,2,3}	-> 6
{4,6}	-> 7
{6,6}	-> 8
{4,5,6}	-> 9

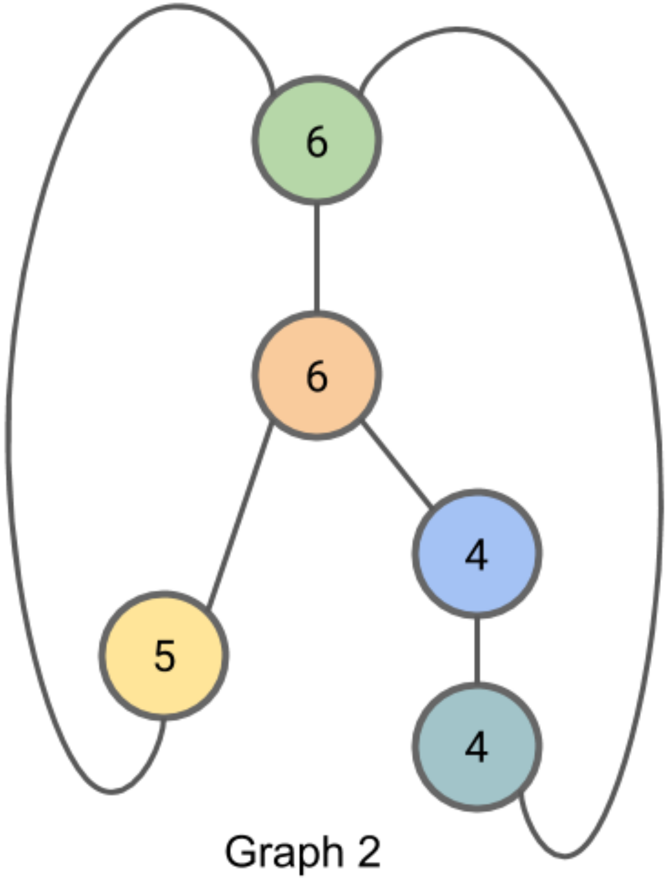
Image Credit: https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

The Weisfeiler-Lehman Isomorphism Test



Graph 1

{4,4},{5},{6,6}

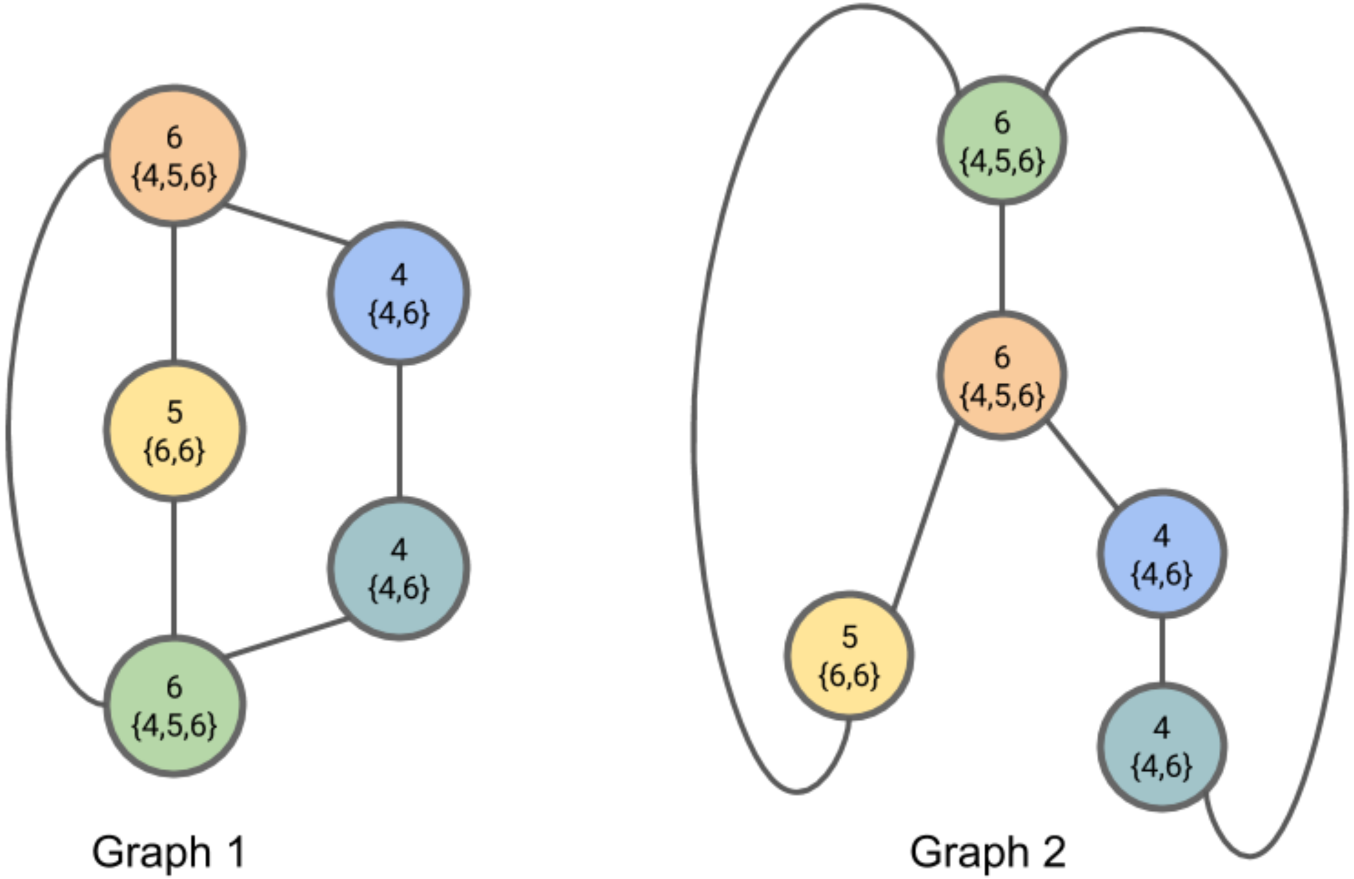


Graph 2

{4,4},{5},{6,6}

Image Credit: https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

The Weisfeiler-Lehman Isomorphism Test



Hash Table:

{1,1} -> 2

{1,1,1} -> 3

{2,3} -> 4

{3,3} -> 5

{2,2,3} -> 6

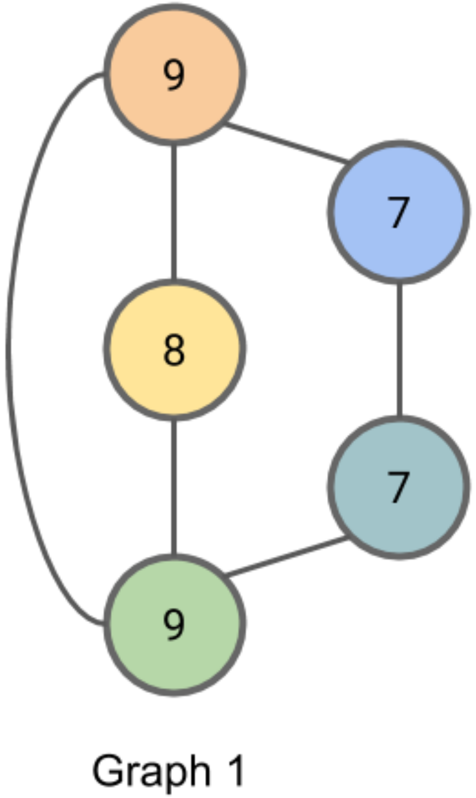
{4,6} -> 7

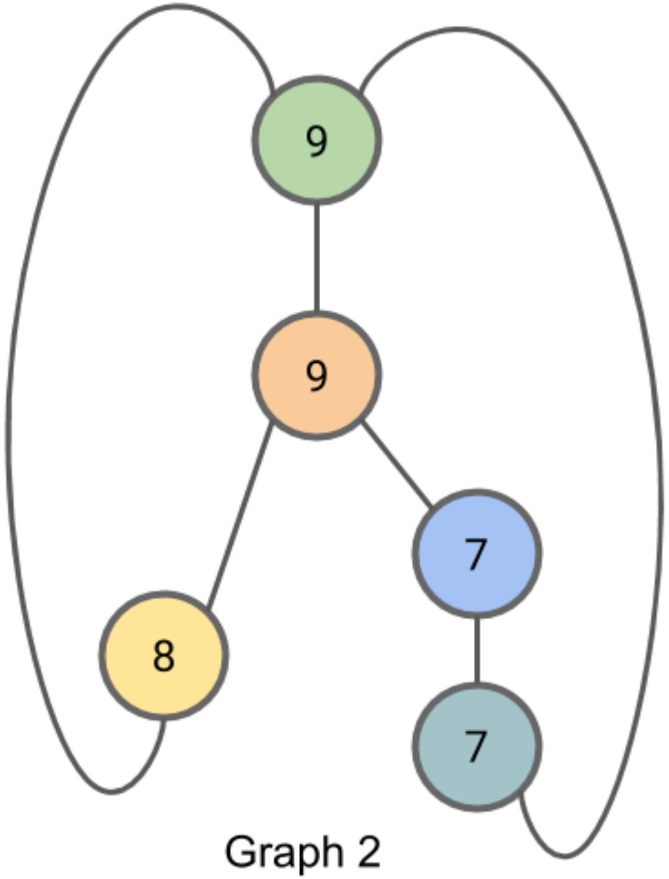
{6,6} -> 8

{4,5,6} -> 9

Image Credit: https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

The Weisfeiler-Lehman Isomorphism Test





{7,7},{8},{9,9}

{7,7},{8},{9,9}

Image Credit: https://davidbieber.com/post/2019-05-10-weisfeiler-lehman-isomorphism-test/

A Recap on GNNs

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

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

Image Credit: [2]

A Recap on GNNs

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

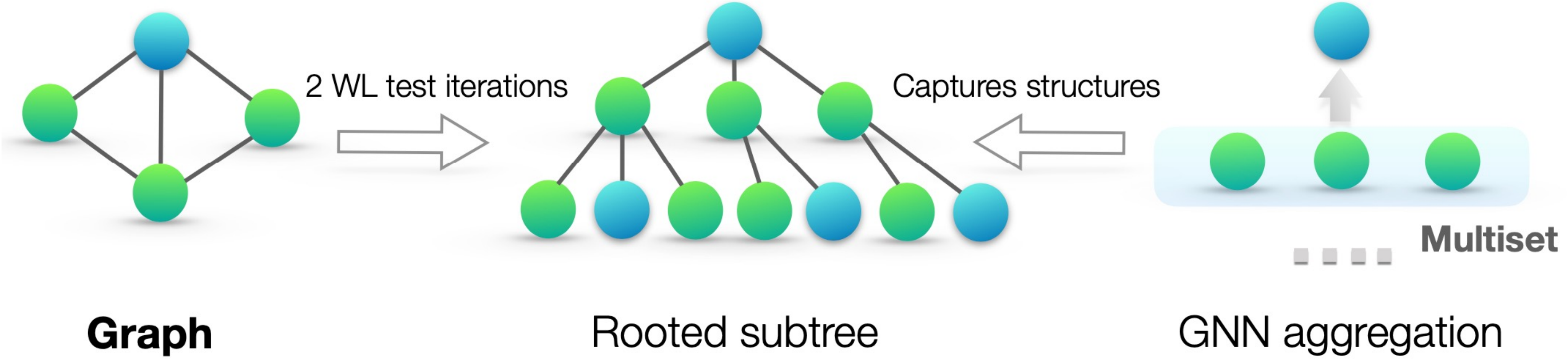
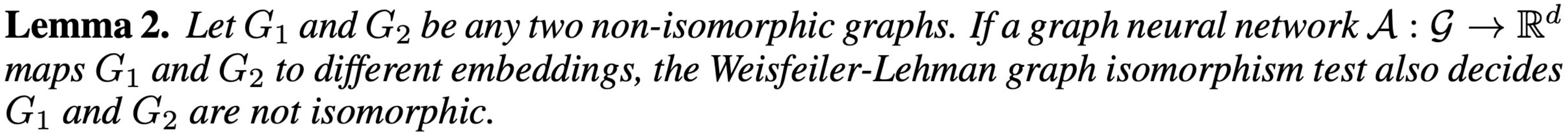


Image Credit: [2]

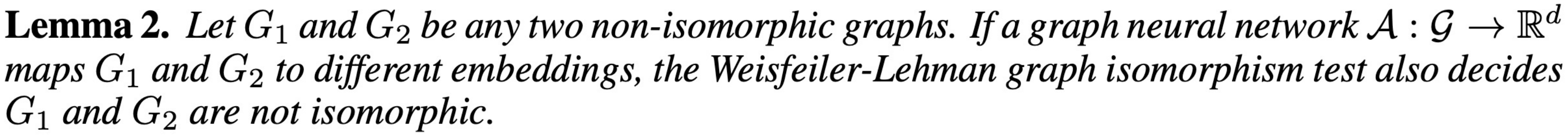
GNNs Are as Powerful as 1-WL



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

[2]

GNNs Are as Powerful as 1-WL



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.*

Proof Sketch: Proof by contradiction.

Suppose the GNN has $\mathcal{A}(G_1) \neq \mathcal{A}(G_2)$, but WL test outputs the same node partition based on labels

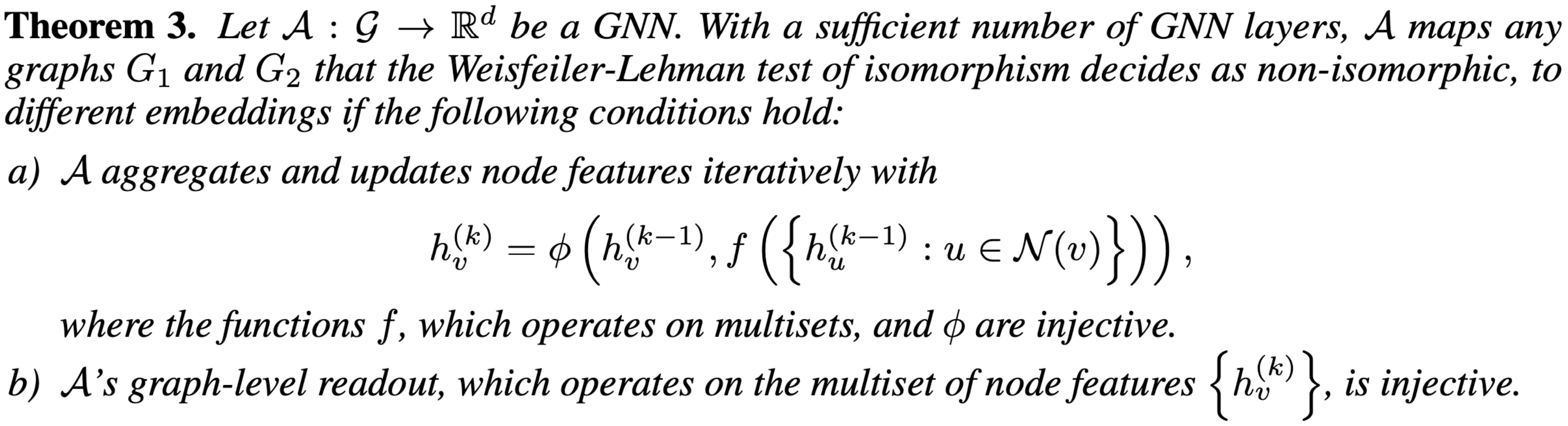
By induction, prove that there exists a valid mapping from node label (WL) to node feature (GNN)

Readout of GNN is permutation invariant => Same node partition generates same graph representation

Therefore, we have $\mathcal{A}(G_1) = \mathcal{A}(G_2)$

[2]

GNNs Are as Powerful as 1-WL



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

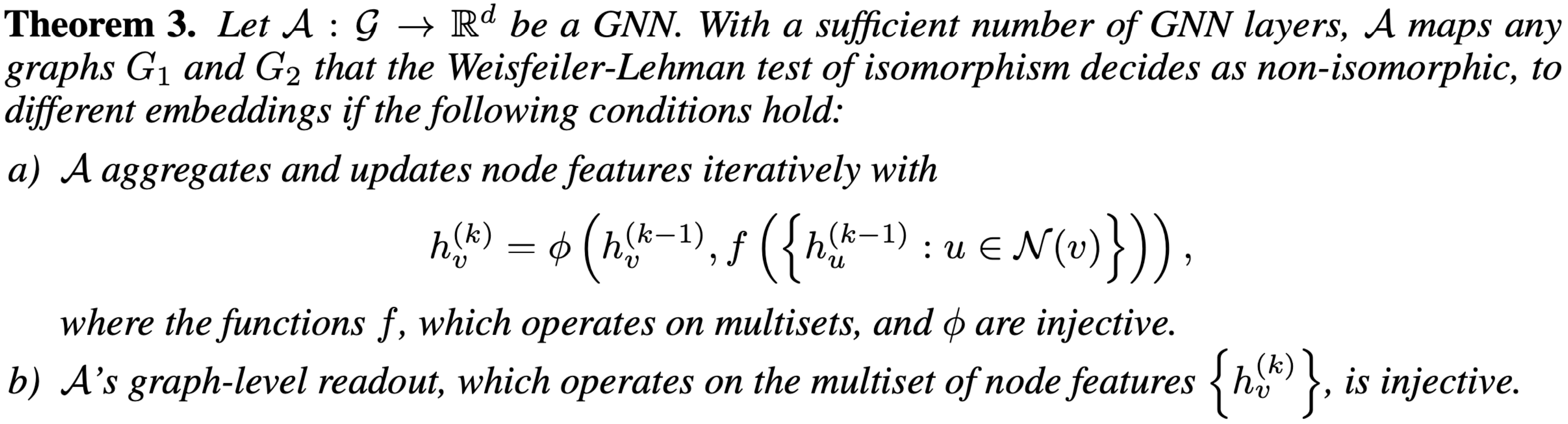
$$h_v^{(k)} = \phi\left(h_v^{(k-1)}, f\left(\left\{h_u^{(k-1)} : u \in \mathcal{N}(v)\right\}\right)\right),$$

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.

[2]

GNNs Are as Powerful as 1-WL



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.

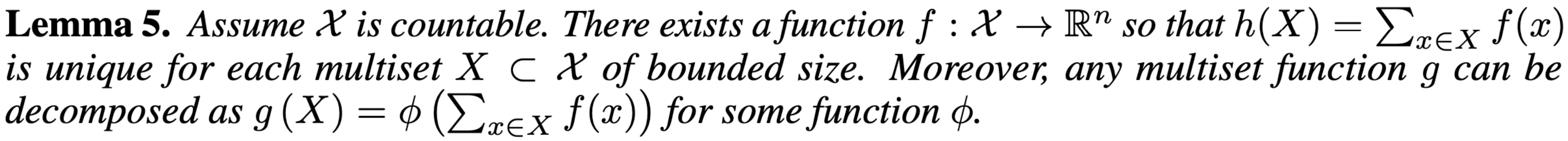
Proof Sketch:

By induction, prove that there exists an injective mapping from node label (WL) to node feature (GNN)

Since readout is injective, different multi-sets will be mapped to different graph embeddings

[2]

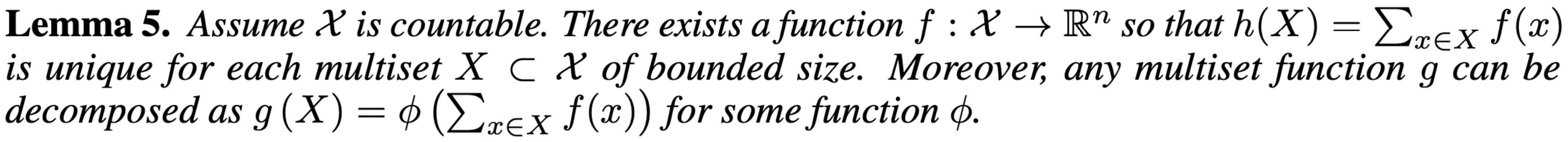
Graph Isomorphism Networks (GINs)



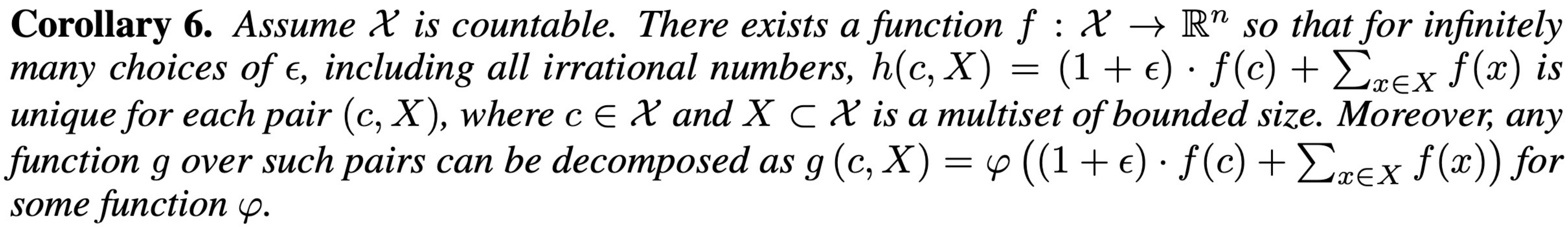
Lemma 5. *Assume $\mathcal{X}$ is countable. There exists a function $f : \mathcal{X} \to \mathbb{R}^n$ so that $h(X) = \sum_{x \in X} f(x)$ is unique for each multiset $X \subset \mathcal{X}$ of bounded size. Moreover, any multiset function g can be decomposed as $g(X) = \phi\left(\sum_{x \in X} f(x)\right)$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

[2]

Graph Isomorphism Networks (GINs)



Lemma 5. *Assume $\mathcal{X}$ is countable. There exists a function $f : \mathcal{X} \rightarrow \mathbb{R}^n$ 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 $g\left(X\right) = \phi\left(\sum_{x \in X} f(x)\right)$ for some function ϕ.*



Corollary 6. *Assume $\mathcal{X}$ is countable. There exists a function $f : \mathcal{X} \rightarrow \mathbb{R}^n$ 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 $g\left(c, X\right) = \varphi\left((1 + \epsilon) \cdot f(c) + \sum_{x \in X} f(x)\right)$ for some function φ.*

[2]

Graph Isomorphism Networks (GINs)

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 $g\left(X\right) = \phi\left(\sum_{x \in X} f(x)\right)$ 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 $g\left(c, X\right) = \varphi\left((1 + \epsilon) \cdot f(c) + \sum_{x \in X} f(x)\right)$ for some function φ.*

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

[2]

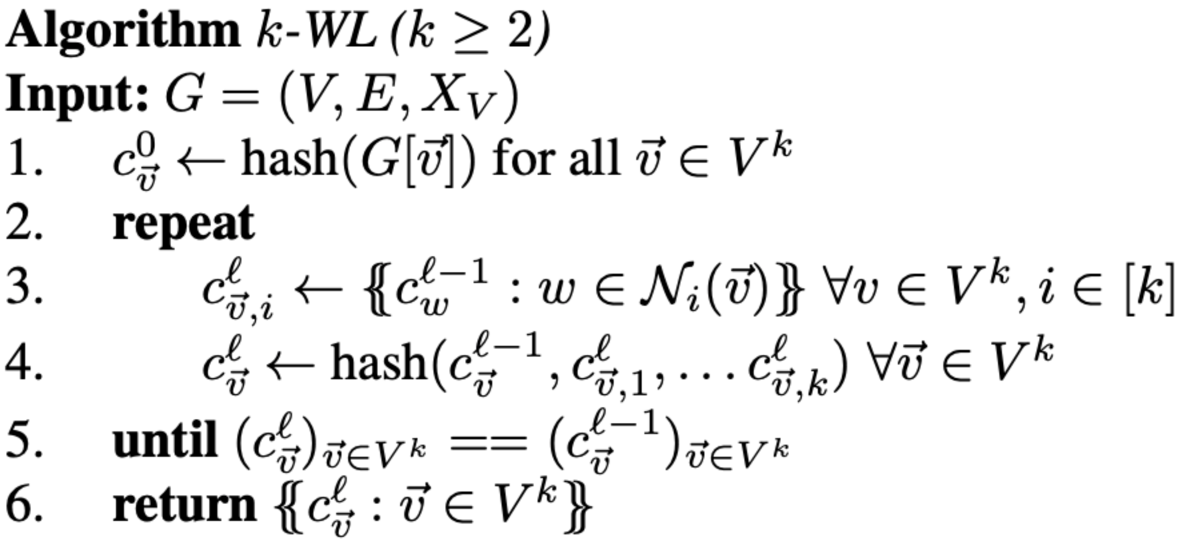
Other Results & Open Questions?

- GNNs are universal approximators for functions on graphs [5,6]
- GNNs are Turing universal [7]
- Expressiveness of GNNs using *communication capacity* [8]
- Expressiveness of GNNs in terms of *counting substructures* [9]
- Can we design GNNs that go beyond 1-WL? [3,17,18,19]

[2]

Other Results & Open Questions?

- GNNs are universal approximators for functions on graphs [5,6]
- GNNs are Turing universal [7]
- Expressiveness of GNNs using *communication capacity* [8]
- Expressiveness of GNNs in terms of *counting substructures* [9]
- Can we design GNNs that go beyond 1-WL? [3,17,18,19]



```
Algorithm k-WL ($k \geq 2$)
Input: $G = (V, E, X_V)$
    $c^0_{\vec{v}} \leftarrow \mathrm{hash}(G[\vec{v}])$ for all $\vec{v} \in V^k$
    repeat
        $c^\ell_{\vec{v},i} \leftarrow \{\!\{c^{\ell-1}_w : w \in \mathcal{N}_i(\vec{v})\}\!\} \; \forall v \in V^k, i \in [k]$
        $c^\ell_{\vec{v}} \leftarrow \mathrm{hash}(c^{\ell-1}_{\vec{v}}, c^\ell_{\vec{v},1}, \ldots c^\ell_{\vec{v},k}) \; \forall \vec{v} \in V^k$
    until $(c^\ell_{\vec{v}})_{\vec{v} \in V^k} == (c^{\ell-1}_{\vec{v}})_{\vec{v} \in V^k}$
    return $\{\!\{c^\ell_{\vec{v}} : \vec{v} \in V^k\}\!\}$
```

[2]

Theoretical Aspects of GNNs

- Expressiveness / Capacity

  What functions on graphs can be represented by GNNs?

  Can GNNs distinguish isomorphic graphs?

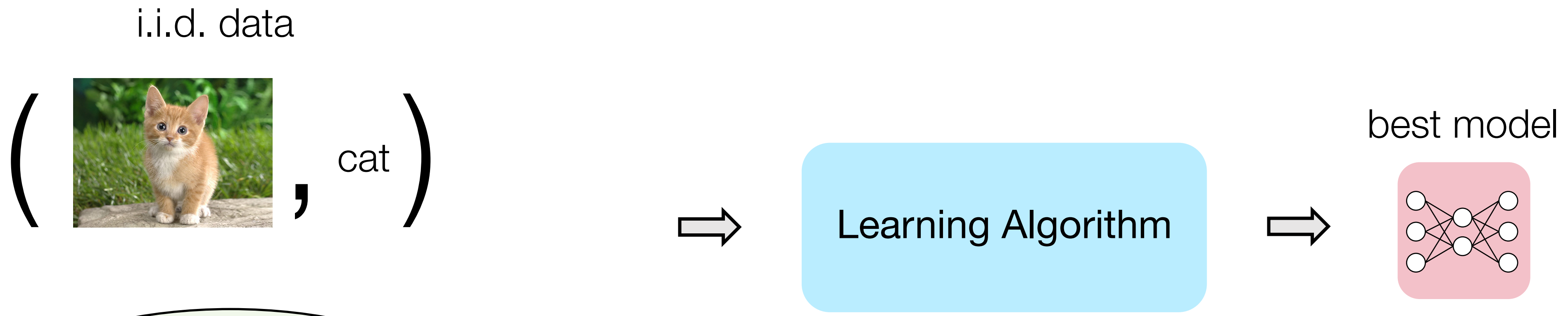
  Are GNNs universal approximators?

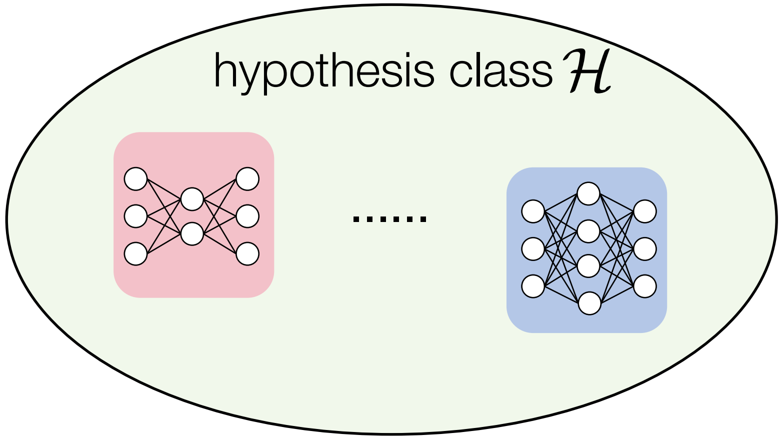
- **Generalization**

  How well do GNNs generalize to unseen graphs?

  What are factors that affect the generalization of GNNs?

Statistical Learning Theory





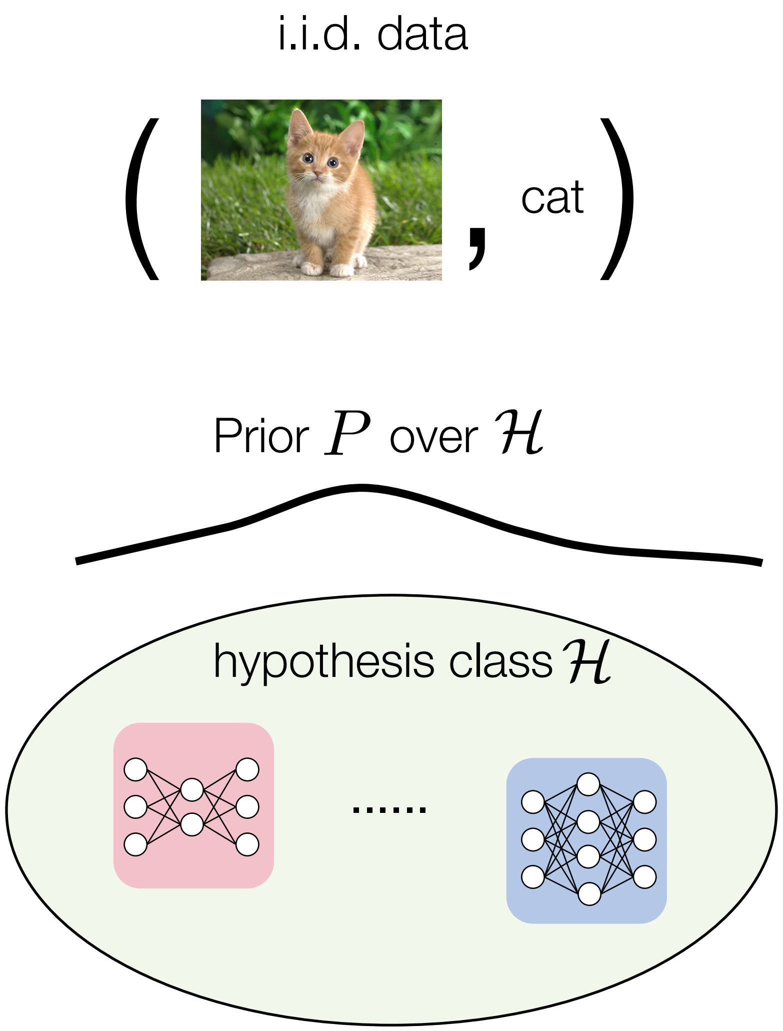
(Uniform Convergence) Generalization Bound:

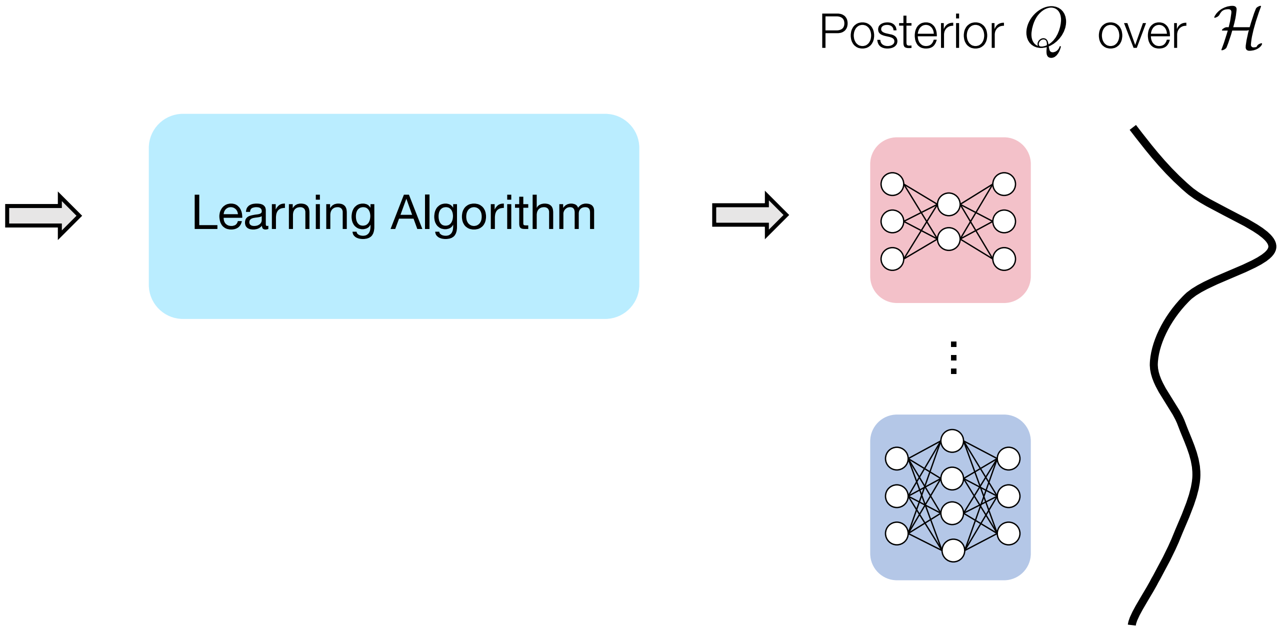
For **any hypothesis** in $\mathcal{H}$, any data distribution, any $\delta \in (0, 1)$,

with probability $1 - \delta$,

True error <= Train error + f(complexity of $\mathcal{H}$, sample size, ...)

PAC-Bayes Theory



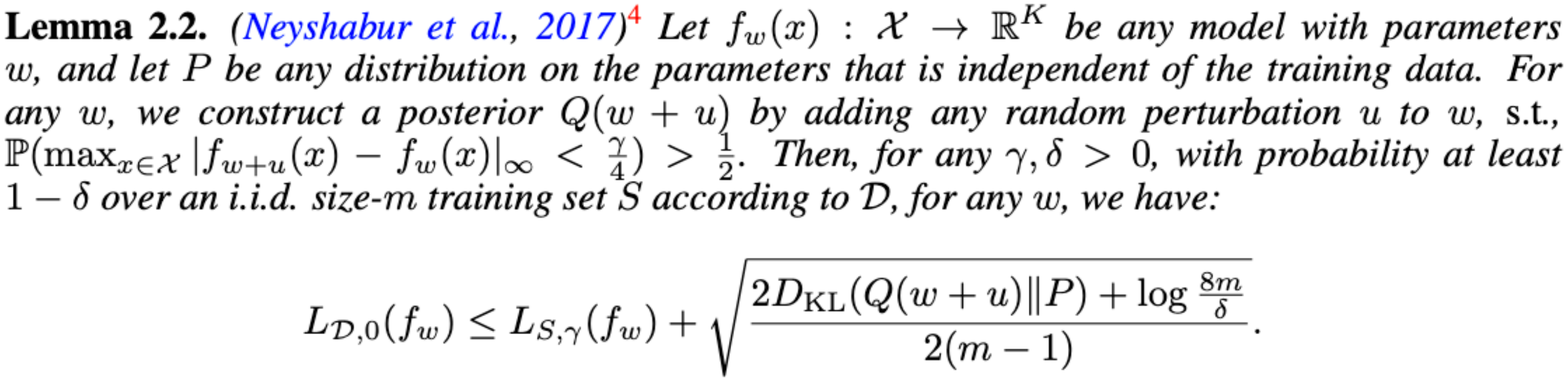


True $\mathbb{E}_Q$[error] <= Train $\mathbb{E}_Q$[error] + f(P , Q, sample size, ...)

$$L_{\mathcal{D},\gamma}(Q) \le L_{S,\gamma}(Q) + \sqrt{\frac{D_{\mathrm{KL}}(Q\|P) + \ln\frac{2m}{\delta}}{2(m-1)}}$$

[10]

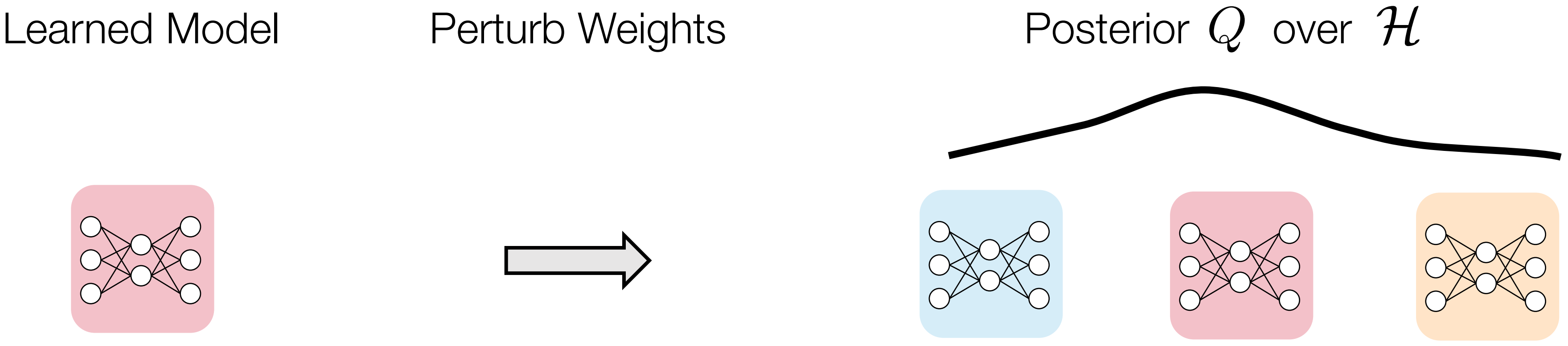
PAC-Bayes Theory



Lemma 2.2. *(Neyshabur et al., 2017)*[4] *Let $f_w(x) : \mathcal{X} \to \mathbb{R}^K$ be any model with parameters w, and let P be any distribution on the parameters that is independent of the training data. For any w, we construct a posterior $Q(w+u)$ by adding any random perturbation u to w,* s.t., $\mathbb{P}(\max_{x\in\mathcal{X}} |f_{w+u}(x) - f_w(x)|_\infty < \frac{\gamma}{4}) > \frac{1}{2}$. *Then, for any $\gamma, \delta > 0$, with probability at least $1-\delta$ over an i.i.d. size-m training set S according to $\mathcal{D}$, for any w, we have:*

$$L_{\mathcal{D},0}(f_w) \le L_{S,\gamma}(f_w) + \sqrt{\frac{2D_{\mathrm{KL}}(Q(w+u)\|P) + \log\frac{8m}{\delta}}{2(m-1)}}.$$

True error <= Train error + f(P, Q, sample size, ...)



Learned Model

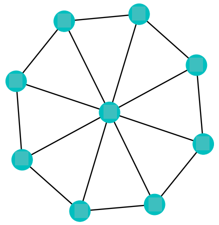
Perturb Weights

Posterior Q over $\mathcal{H}$

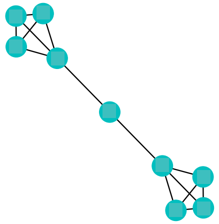
[11]

Problems & Assumptions

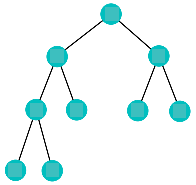
Graph Classification



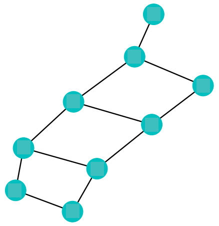
Wheel



Barbell



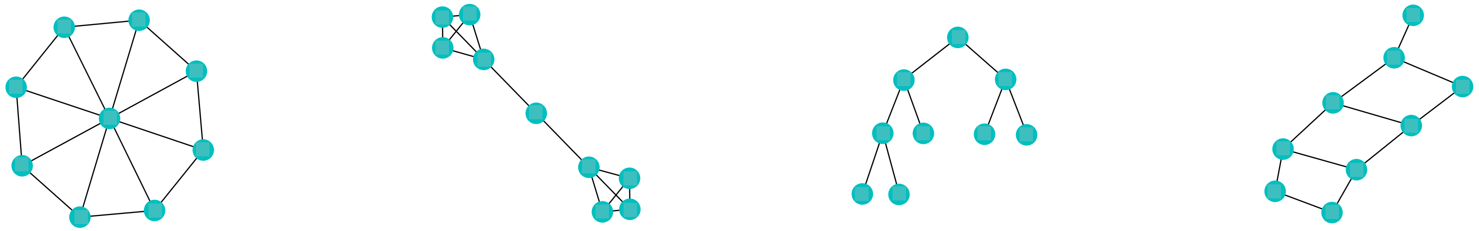
Binary Tree



Ladder

Problems & Assumptions

Graph Classification



Wheel

Barbell

Binary Tree

Ladder

A1 Data, *i.e.*, triplets (A, X, y), are i.i.d. samples drawn from some unknown distribution $\mathcal{D}$.

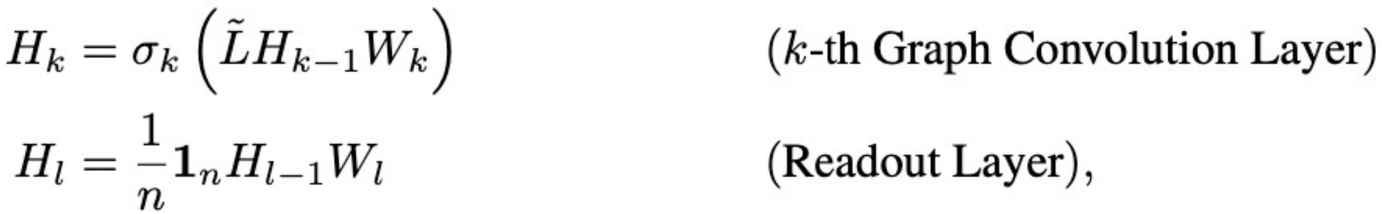
A2 The maximum hidden dimension across all layers is h.

A3 Node feature of any graph is contained in a ℓ_2-ball with radius B. Specifically, we have $\forall i \in \mathbb{N}_n^+$, the i-th node feature $X[i,:] \in \mathcal{X}_{B,h_0} = \{x \in \mathbb{R}^{h_0} | \sum_{j=1}^{h_0} x_j^2 \leq B^2\}$.

A4 We only consider simple graphs (*i.e.*, undirected, no loops[1], and no multi-edges) with maximum node degree as $d - 1$.

Models

Graph Convolutional Networks (GCNs) [12]



$$H_k = \sigma_k \left(\tilde{L} H_{k-1} W_k \right) \qquad \text{(k-th Graph Convolution Layer)}$$

$$H_l = \frac{1}{n} \mathbf{1}_n H_{l-1} W_l \qquad \text{(Readout Layer)},$$

Models

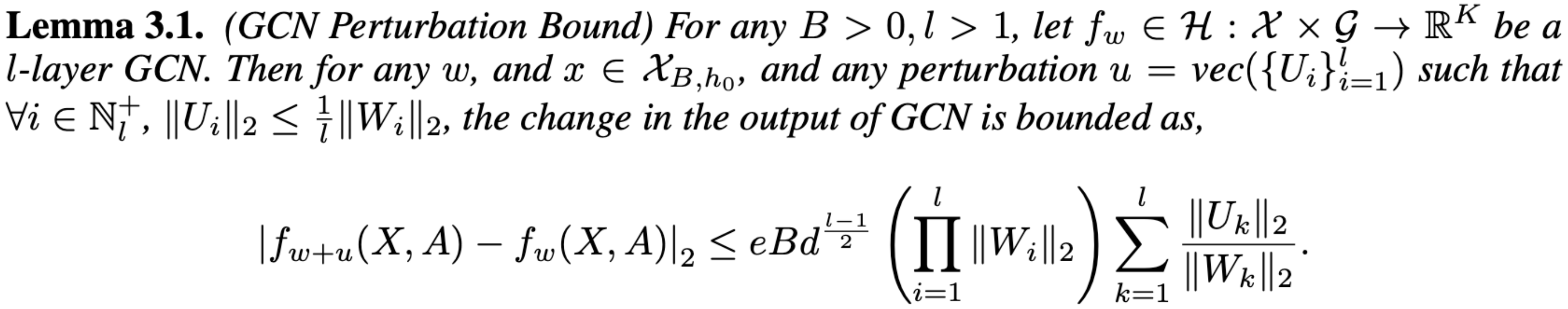
Graph Convolutional Networks (GCNs) [12]

$$\begin{aligned} H_k &= \sigma_k\left(\tilde{L} H_{k-1} W_k\right) && \text{(k-th Graph Convolution Layer)} \\ H_l &= \frac{1}{n} \mathbf{1}_n H_{l-1} W_l && \text{(Readout Layer)}, \end{aligned}$$

Message Passing Graph Neural Networks (MPGNNs) [13]

$$\begin{aligned} M_k &= g(C_{\text{out}}^{\top} H_{k-1}) && \text{(k-th step Message Computation)} \\ \bar{M}_k &= C_{\text{in}} M_k && \text{(k-th step Message Aggregation)} \\ H_k &= \phi\left(X W_1 + \rho\left(\bar{M}_k\right) W_2\right) && \text{(k-th step Node State Update)} \\ H_l &= \frac{1}{n} \mathbf{1}_n H_{l-1} W_l && \text{(Readout Layer)}, \end{aligned}$$

Perturbation Bounds

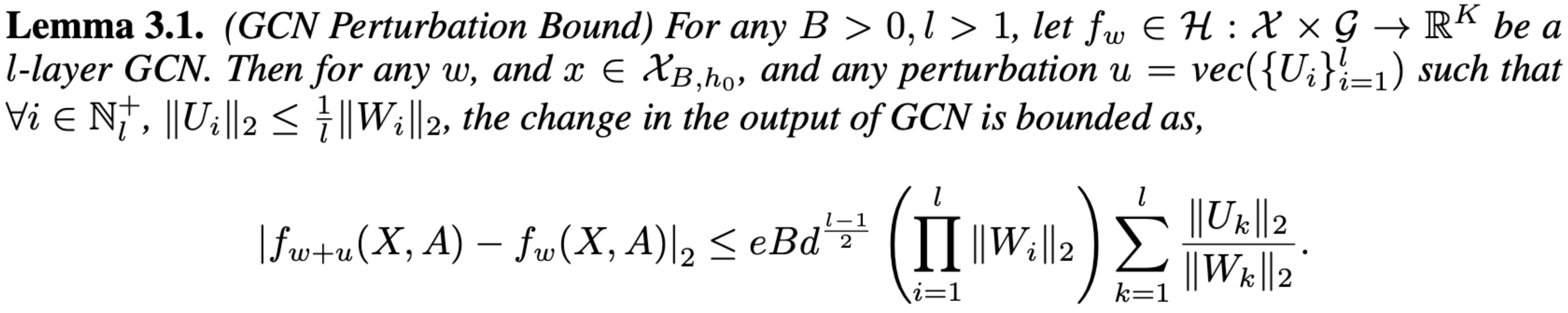


Lemma 3.1. *(GCN Perturbation Bound) For any* $B > 0, l > 1$*, let* $f_w \in \mathcal{H} : \mathcal{X} \times \mathcal{G} \to \mathbb{R}^K$ *be a* l*-layer GCN. Then for any* w*, and* $x \in \mathcal{X}_{B,h_0}$*, and any perturbation* $u = vec(\{U_i\}_{i=1}^{l})$ *such that* $\forall i \in \mathbb{N}_l^+$*,* $\|U_i\|_2 \leq \frac{1}{l}\|W_i\|_2$*, the change in the output of GCN is bounded as,*

$$|f_{w+u}(X, A) - f_w(X, A)|_2 \leq eBd^{\frac{l-1}{2}} \left(\prod_{i=1}^{l} \|W_i\|_2 \right) \sum_{k=1}^{l} \frac{\|U_k\|_2}{\|W_k\|_2}.$$

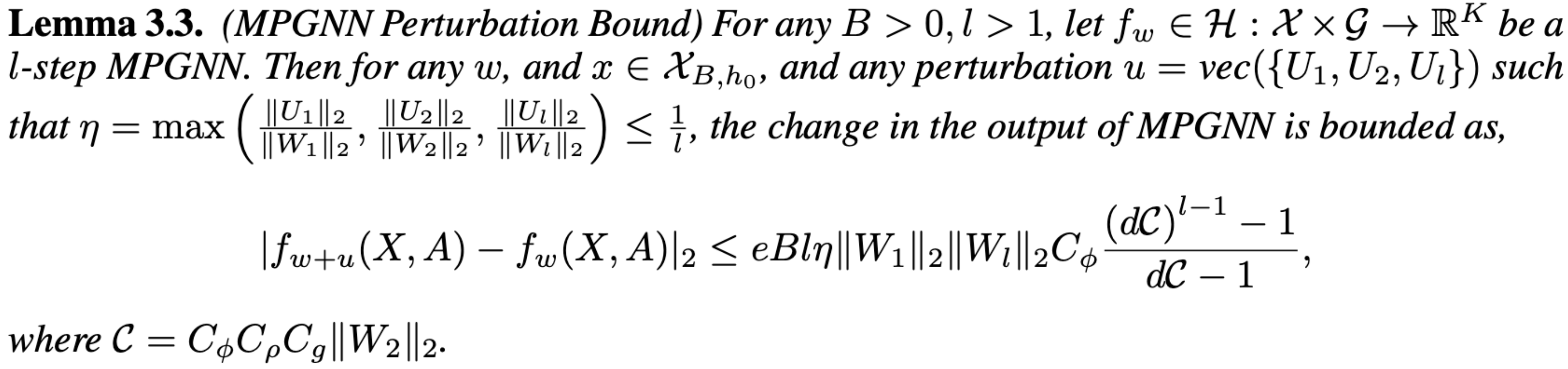
[14]

Perturbation Bounds



Lemma 3.1. *(GCN Perturbation Bound) For any $B > 0, l > 1$, let $f_w \in \mathcal{H} : \mathcal{X} \times \mathcal{G} \to \mathbb{R}^K$ be a l-layer GCN. Then for any w, and $x \in \mathcal{X}_{B,h_0}$, and any perturbation $u = vec(\{U_i\}_{i=1}^{l})$ such that $\forall i \in \mathbb{N}_l^+$, $\|U_i\|_2 \le \frac{1}{l}\|W_i\|_2$, the change in the output of GCN is bounded as,*

$$|f_{w+u}(X,A) - f_w(X,A)|_2 \le eBd^{\frac{l-1}{2}} \left(\prod_{i=1}^{l} \|W_i\|_2\right) \sum_{k=1}^{l} \frac{\|U_k\|_2}{\|W_k\|_2}.$$



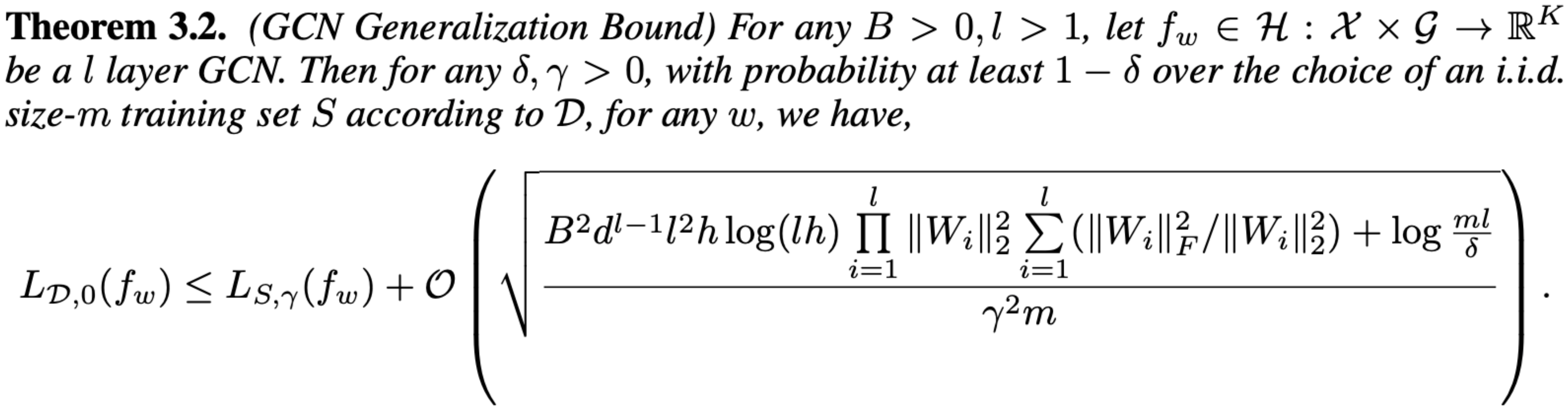
Lemma 3.3. *(MPGNN Perturbation Bound) For any $B > 0, l > 1$, let $f_w \in \mathcal{H} : \mathcal{X} \times \mathcal{G} \to \mathbb{R}^K$ be a l-step MPGNN. Then for any w, and $x \in \mathcal{X}_{B,h_0}$, and any perturbation $u = vec(\{U_1, U_2, U_l\})$ such that $\eta = \max\left(\frac{\|U_1\|_2}{\|W_1\|_2}, \frac{\|U_2\|_2}{\|W_2\|_2}, \frac{\|U_l\|_2}{\|W_l\|_2}\right) \le \frac{1}{l}$, the change in the output of MPGNN is bounded as,*

$$|f_{w+u}(X,A) - f_w(X,A)|_2 \le eBl\eta\|W_1\|_2\|W_l\|_2 C_\phi \frac{(d\mathcal{C})^{l-1} - 1}{d\mathcal{C} - 1},$$

where $\mathcal{C} = C_\phi C_\rho C_g \|W_2\|_2$.

[14]

Generalization Bounds for GCNs

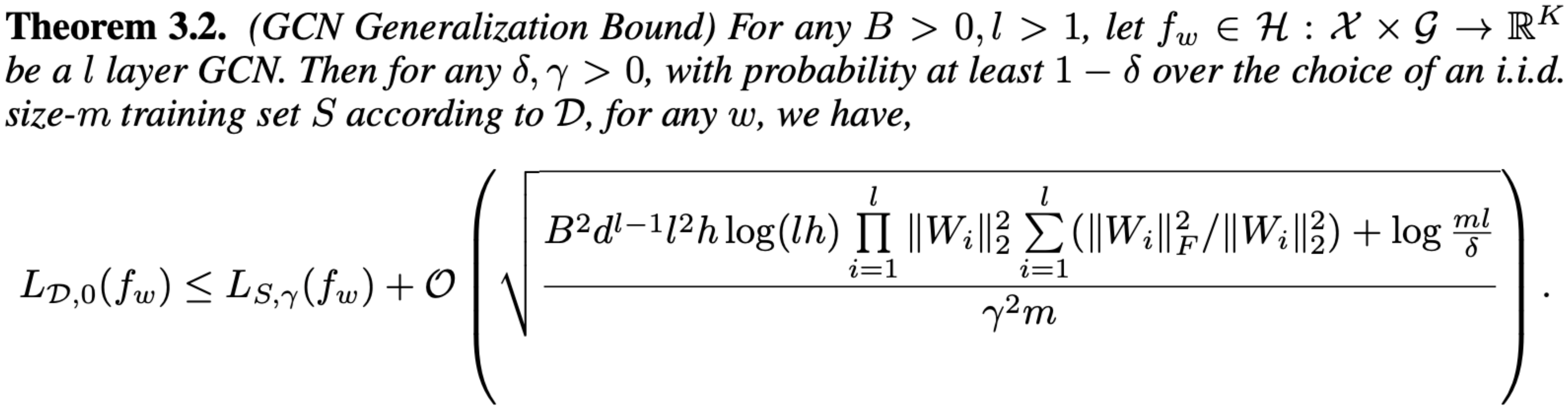


Theorem 3.2. *(GCN Generalization Bound) For any $B > 0, l > 1$, let $f_w \in \mathcal{H} : \mathcal{X} \times \mathcal{G} \to \mathbb{R}^K$ be a l layer GCN. Then for any $\delta, \gamma > 0$, with probability at least $1 - \delta$ over the choice of an i.i.d. size-m training set S according to $\mathcal{D}$, for any w, we have,*

$$L_{\mathcal{D},0}(f_w) \leq L_{S,\gamma}(f_w) + \mathcal{O}\left(\sqrt{\frac{B^2 d^{l-1} l^2 h \log(lh) \prod_{i=1}^{l} \|W_i\|_2^2 \sum_{i=1}^{l} (\|W_i\|_F^2 / \|W_i\|_2^2) + \log \frac{ml}{\delta}}{\gamma^2 m}}\right).$$

[14]

Generalization Bounds for GCNs



Theorem 3.2. *(GCN Generalization Bound) For any $B > 0, l > 1$, let $f_w \in \mathcal{H} : \mathcal{X} \times \mathcal{G} \to \mathbb{R}^K$ be a l layer GCN. Then for any $\delta, \gamma > 0$, with probability at least $1 - \delta$ over the choice of an i.i.d. size-m training set S according to $\mathcal{D}$, for any w, we have,*

$$L_{\mathcal{D},0}(f_w) \leq L_{S,\gamma}(f_w) + \mathcal{O}\left(\sqrt{\frac{B^2 d^{l-1} l^2 h \log(lh) \prod_{i=1}^{l} \|W_i\|_2^2 \sum_{i=1}^{l} (\|W_i\|_F^2 / \|W_i\|_2^2) + \log \frac{ml}{\delta}}{\gamma^2 m}}\right).$$

Proof Sketch:

Perturbation bound + Measure Concentration Inequalities => A model with certain learned weights could generalize

Similar bounds hold for a finite covering set of all possible weights and then use union bound to derive the final result

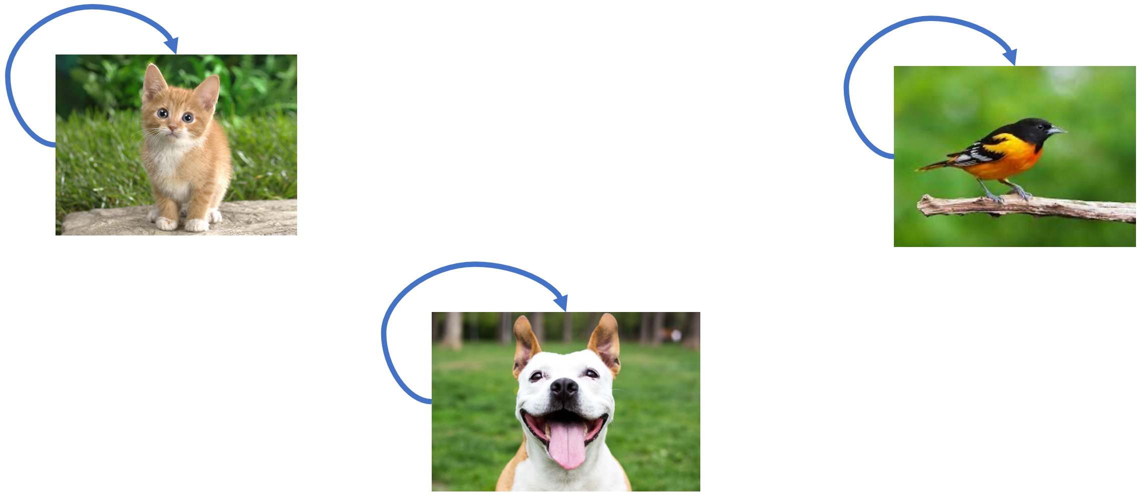
[14]

Generalization Bounds for GCNs

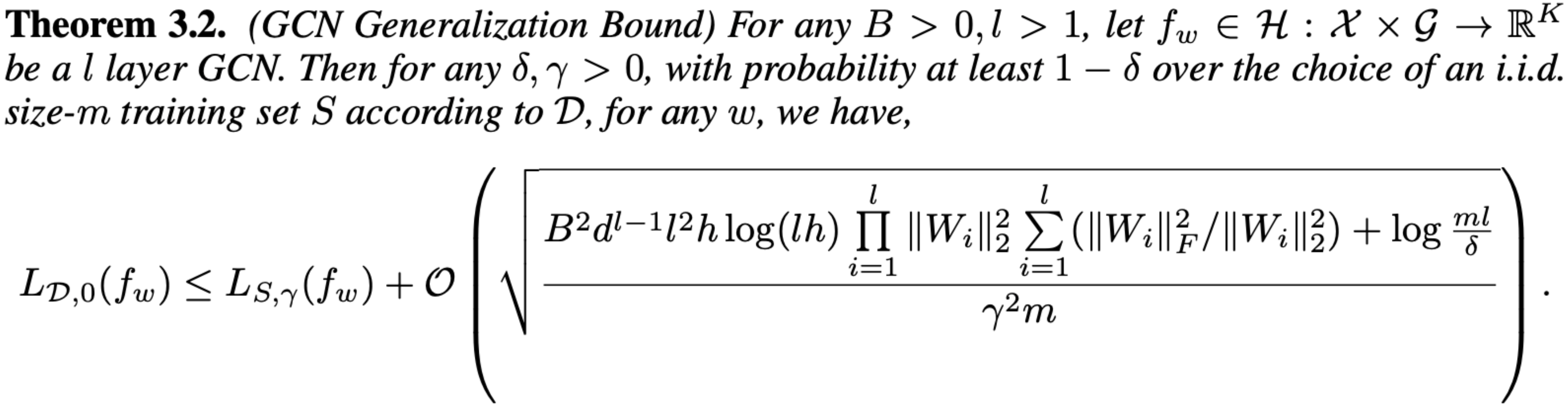
MLPs/CNNs are special GCNs if we construct the following graph:

- Each image is a node
- No edges exist except self-loops

(Convolution is matrix multiplication as well)



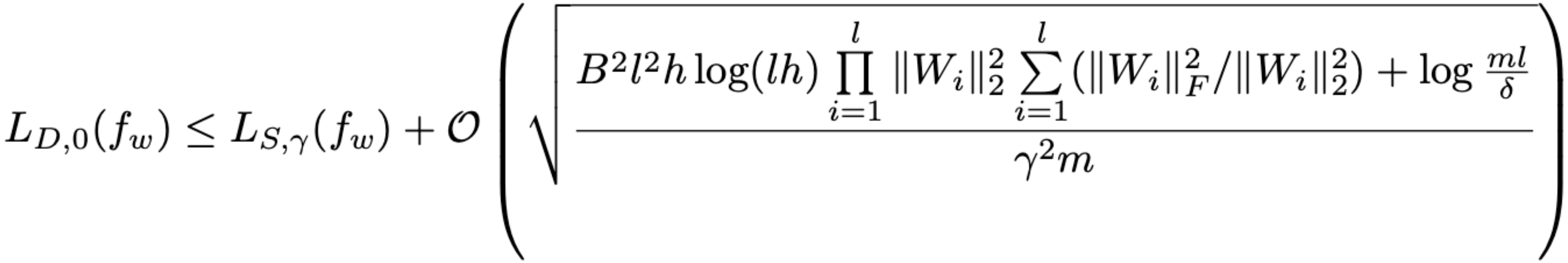
Generalization Bounds for GCNs



Theorem 3.2. *(GCN Generalization Bound) For any $B > 0, l > 1$, let $f_w \in \mathcal{H} : \mathcal{X} \times \mathcal{G} \to \mathbb{R}^K$ be a l layer GCN. Then for any $\delta, \gamma > 0$, with probability at least $1 - \delta$ over the choice of an i.i.d. size-m training set S according to $\mathcal{D}$, for any w, we have,*

$$L_{\mathcal{D},0}(f_w) \leq L_{S,\gamma}(f_w) + \mathcal{O}\left(\sqrt{\frac{B^2 d^{l-1} l^2 h \log(lh) \prod_{i=1}^{l} \|W_i\|_2^2 \sum_{i=1}^{l} (\|W_i\|_F^2 / \|W_i\|_2^2) + \log \frac{ml}{\delta}}{\gamma^2 m}}\right).$$

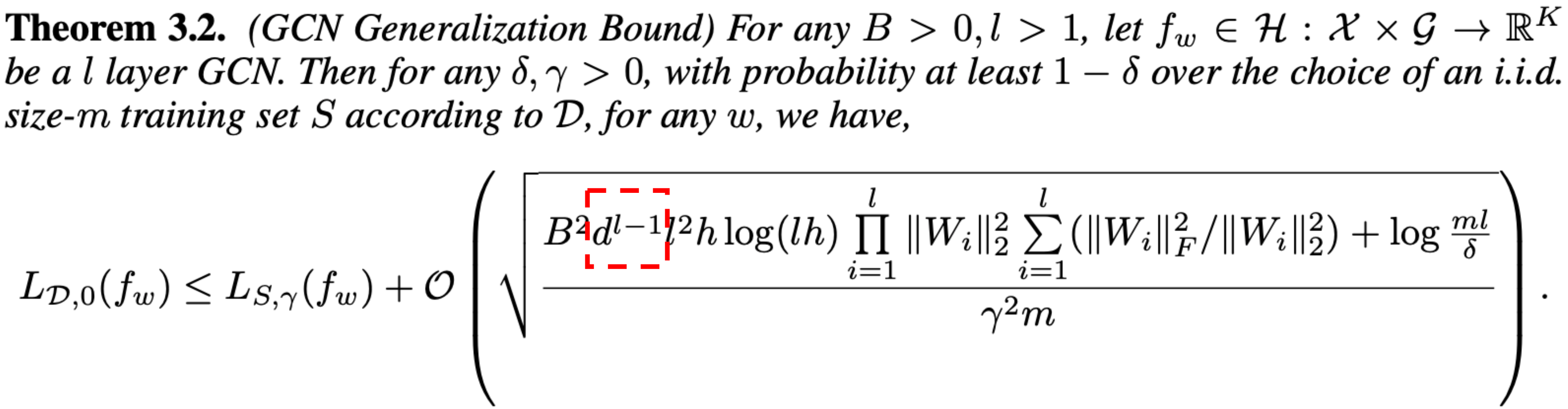
PAC-Bayes bounds for MLPs/CNNs (w/ ReLU)



$$L_{D,0}(f_w) \leq L_{S,\gamma}(f_w) + \mathcal{O}\left(\sqrt{\frac{B^2 l^2 h \log(lh) \prod_{i=1}^{l} \|W_i\|_2^2 \sum_{i=1}^{l} (\|W_i\|_F^2 / \|W_i\|_2^2) + \log \frac{ml}{\delta}}{\gamma^2 m}}\right)$$

[14]

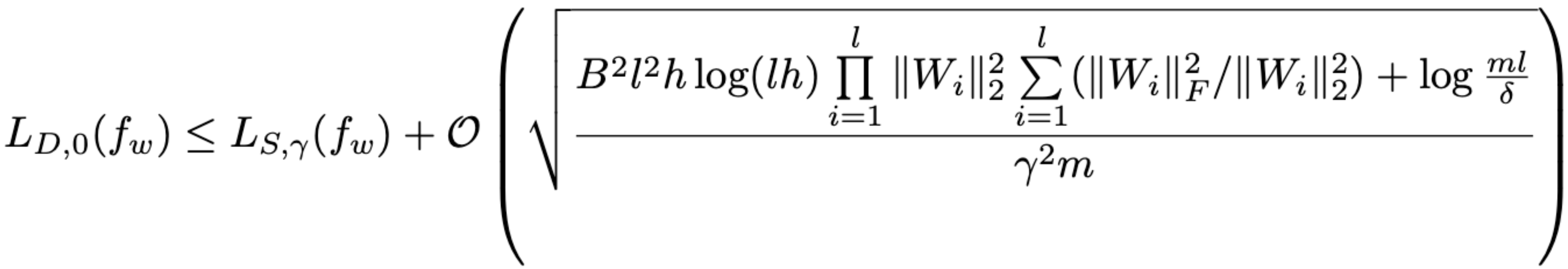
Generalization Bounds for GCNs



Theorem 3.2. *(GCN Generalization Bound) For any $B > 0, l > 1$, let $f_w \in \mathcal{H} : \mathcal{X} \times \mathcal{G} \to \mathbb{R}^K$ be a l layer GCN. Then for any $\delta, \gamma > 0$, with probability at least $1 - \delta$ over the choice of an i.i.d. size-m training set S according to $\mathcal{D}$, for any w, we have,*

$$L_{\mathcal{D},0}(f_w) \leq L_{S,\gamma}(f_w) + \mathcal{O}\left(\sqrt{\frac{B^2 \boxed{d^{l-1}} l^2 h \log(lh) \prod_{i=1}^{l} \|W_i\|_2^2 \sum_{i=1}^{l} (\|W_i\|_F^2 / \|W_i\|_2^2) + \log \frac{ml}{\delta}}{\gamma^2 m}}\right).$$

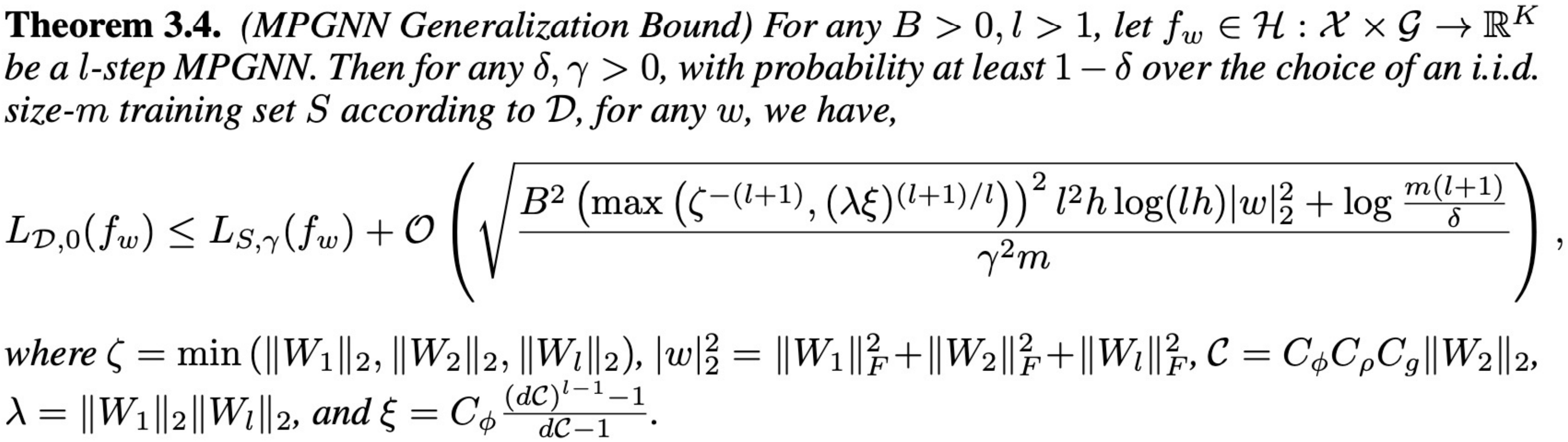
PAC-Bayes bounds for MLPs/CNNs (w/ ReLU)



$$L_{D,0}(f_w) \leq L_{S,\gamma}(f_w) + \mathcal{O}\left(\sqrt{\frac{B^2 l^2 h \log(lh) \prod_{i=1}^{l} \|W_i\|_2^2 \sum_{i=1}^{l} (\|W_i\|_F^2 / \|W_i\|_2^2) + \log \frac{ml}{\delta}}{\gamma^2 m}}\right)$$

[14]

Generalization Bounds for Message Passing GNNs



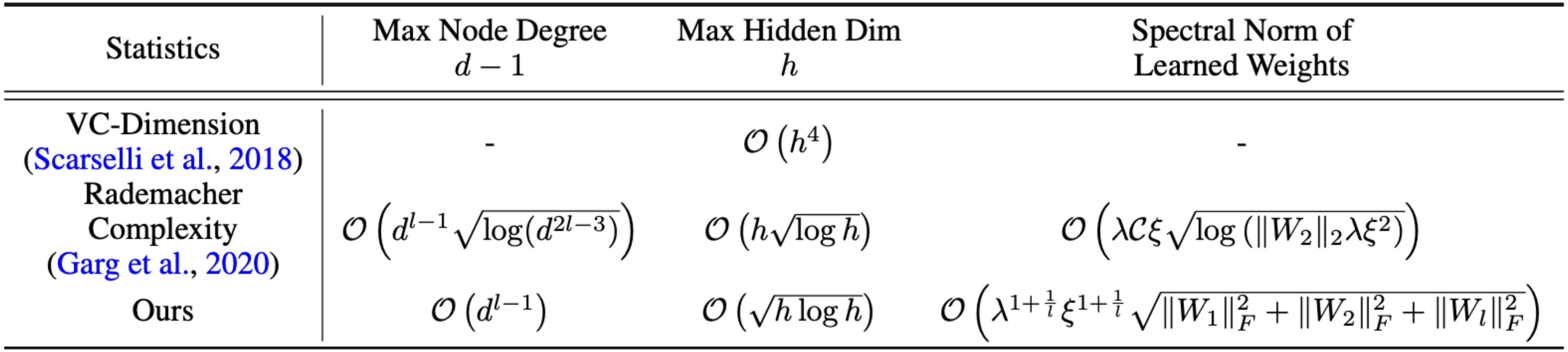
Theorem 3.4. *(MPGNN Generalization Bound) For any $B > 0, l > 1$, let $f_w \in \mathcal{H} : \mathcal{X} \times \mathcal{G} \to \mathbb{R}^K$ be a l-step MPGNN. Then for any $\delta, \gamma > 0$, with probability at least $1 - \delta$ over the choice of an i.i.d. size-m training set S according to $\mathcal{D}$, for any w, we have,*

$$L_{\mathcal{D},0}(f_w) \leq L_{S,\gamma}(f_w) + \mathcal{O}\left(\sqrt{\frac{B^2\left(\max\left(\zeta^{-(l+1)}, (\lambda\xi)^{(l+1)/l}\right)\right)^2 l^2 h \log(lh)|w|_2^2 + \log\frac{m(l+1)}{\delta}}{\gamma^2 m}}\right),$$

where $\zeta = \min\left(\|W_1\|_2, \|W_2\|_2, \|W_l\|_2\right)$, $|w|_2^2 = \|W_1\|_F^2 + \|W_2\|_F^2 + \|W_l\|_F^2$, $\mathcal{C} = C_\phi C_\rho C_g \|W_2\|_2$, $\lambda = \|W_1\|_2 \|W_l\|_2$, and $\xi = C_\phi \frac{(d\mathcal{C})^{l-1} - 1}{d\mathcal{C} - 1}$.

[14]

Generalization Bounds for Message Passing GNNs

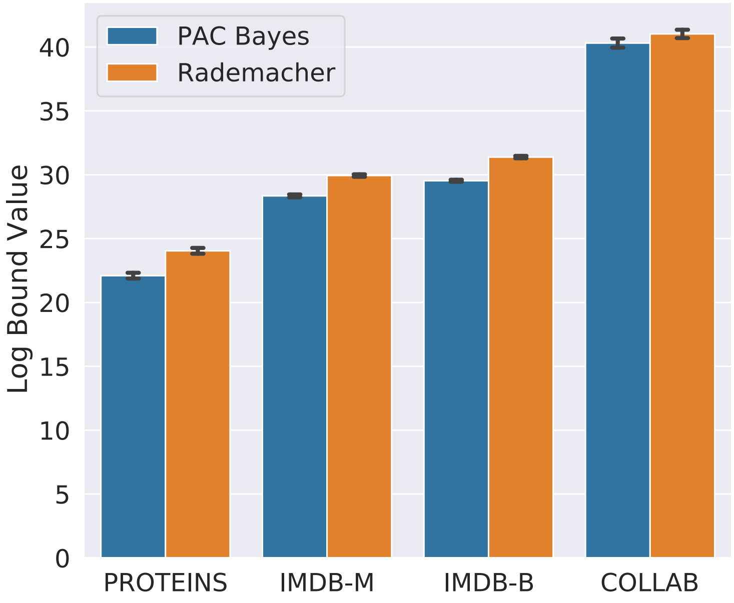


Statistics	Max Node Degree $d-1$	Max Hidden Dim h	Spectral Norm of Learned Weights
VC-Dimension (Scarselli et al., 2018)	-	$\mathcal{O}\left(h^4\right)$	-
Rademacher Complexity (Garg et al., 2020)	$\mathcal{O}\left(d^{l-1}\sqrt{\log(d^{2l-3})}\right)$	$\mathcal{O}\left(h\sqrt{\log h}\right)$	$\mathcal{O}\left(\lambda\mathcal{C}\xi\sqrt{\log\left(\\|W_2\\|_2\lambda\xi^2\right)}\right)$
Ours	$\mathcal{O}\left(d^{l-1}\right)$	$\mathcal{O}\left(\sqrt{h\log h}\right)$	$\mathcal{O}\left(\lambda^{1+\frac{1}{l}}\xi^{1+\frac{1}{l}}\sqrt{\\|W_1\\|_F^2+\\|W_2\\|_F^2+\\|W_l\\|_F^2}\right)$

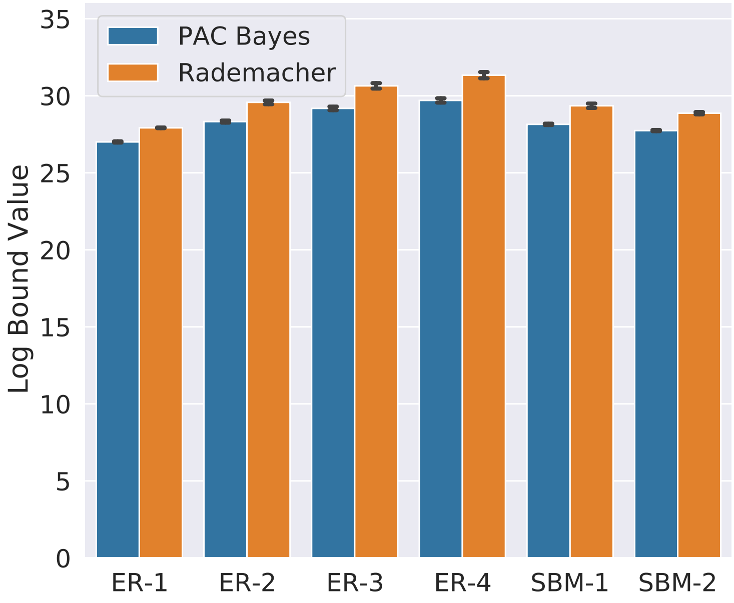
[14, 15]

Generalization Bounds for Message Passing GNNs

Statistics	Max Node Degree $d-1$	Max Hidden Dim h	Spectral Norm of Learned Weights
VC-Dimension (Scarselli et al., 2018)	-	$\mathcal{O}\left(h^4\right)$	-
Rademacher Complexity (Garg et al., 2020)	$\mathcal{O}\left(d^{l-1}\sqrt{\log(d^{2l-3})}\right)$	$\mathcal{O}\left(h\sqrt{\log h}\right)$	$\mathcal{O}\left(\lambda \mathcal{C} \xi \sqrt{\log\left(\|W_2\|_2 \lambda \xi^2\right)}\right)$
Ours	$\mathcal{O}\left(d^{l-1}\right)$	$\mathcal{O}\left(\sqrt{h \log h}\right)$	$\mathcal{O}\left(\lambda^{1+\frac{1}{l}} \xi^{1+\frac{1}{l}} \sqrt{\|W_1\|_F^2 + \|W_2\|_F^2 + \|W_l\|_F^2}\right)$



real-world graphs



synthetic graphs

[14]

Take Home Messages & Other Results

- Spectral norm of weights and max node degree play key roles in the generalization of GNNs
- Bounds for MLPs/CNNs (w/ ReLU) are special cases of bounds for GCNs, which reconciles with the observation that MLPs/CNNs (w/ ReLU) can be viewed as special GCNs
- PAC-Bayes bounds are empirically tighter than the recent Rademacher complexity based bounds on several synthetic and real-world graph datasets
- Rademacher complexity based generalization bounds [15]
- VC-Dimension based generalization bounds [16]

Open Questions

- Is the maximum node degree the only graph statistic that has an impact on the generalization ability of GNNs?

- Would the analysis still work for other interesting problem setups like out-of-distribution generalizations?

- What is the impact of the optimization algorithms like SGD on the generalization of GNNs?

References

[1] Weisfeiler, B. and Leman, A., 1968. The reduction of a graph to canonical form and the algebra which appears therein. NTI, Series, 2(9), pp.12-16.

[2] Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks?. arXiv preprint arXiv:1810.00826.

[3] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., & Grohe, M. (2019, July). Weisfeiler and leman go neural: Higher-order graph neural networks. In Proceedings of the AAAI conference on artificial intelligence (Vol. 33, No. 01, pp. 4602-4609).

[4] Shervashidze, N., Schweitzer, P., Van Leeuwen, E.J., Mehlhorn, K. and Borgwardt, K.M., 2011. Weisfeiler-lehman graph kernels. Journal of Machine Learning Research, 12(9).

[5] Chen, Z., Villar, S., Chen, L. and Bruna, J., 2019. On the equivalence between graph isomorphism testing and function approximation with gnns. Advances in neural information processing systems, 32.

[6] Brüel Gabrielsson, R., 2020. Universal Function Approximation on Graphs. Advances in Neural Information Processing Systems, 33, pp.19762-19772.

[7] Loukas, A., 2019. What graph neural networks cannot learn: depth vs width. arXiv preprint arXiv:1907.03199.

[8] Loukas, A., 2020. How hard is to distinguish graphs with graph neural networks?. Advances in neural information processing systems, 33, pp.3465-3476.

[9] Chen, Z., Chen, L., Villar, S. and Bruna, J., 2020. Can graph neural networks count substructures?. Advances in neural information processing systems, 33, pp.10383-10395.

[10] McAllester, D., 2003. Simplified PAC-Bayesian margin bounds. In Learning theory and Kernel machines (pp. 203-215). Springer, Berlin, Heidelberg.

References

[11] Neyshabur, B., Bhojanapalli, S. and Srebro, N., 2017. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. arXiv preprint arXiv:1707.09564.

[12] Kipf, T.N. and Welling, M., 2016. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[13] Dai, H., Dai, B. and Song, L., 2016, June. Discriminative embeddings of latent variable models for structured data. In International conference on machine learning (pp. 2702-2711). PMLR.

[14] Liao, R., Urtasun, R. and Zemel, R., 2020. A pac-bayesian approach to generalization bounds for graph neural networks. arXiv preprint arXiv:2012.07690.

[15] Garg, V., Jegelka, S. and Jaakkola, T., 2020, November. Generalization and representational limits of graph neural networks. In International Conference on Machine Learning (pp. 3419-3430). PMLR.

[16] Scarselli, F., Tsoi, A.C. and Hagenbuchner, M., 2018. The vapnik–chervonenkis dimension of graph and recursive neural networks. Neural Networks, 108, pp.248-259.

[17] Maron, H., Ben-Hamu, H., Shamir, N. and Lipman, Y., 2018. Invariant and equivariant graph networks. arXiv preprint arXiv:1812.09902.

[18] Maron, H., Ben-Hamu, H., Serviansky, H. and Lipman, Y., 2019. Provably powerful graph networks. Advances in neural information processing systems, 32.

[19] Maron, H., Ben-Hamu, H. and Lipman, Y., 2019. Open problems: Approximation power of invariant graph networks. In NeurIPS 2019 Graph Representation Learning Workshop, 2019a. URL https://grlearning. github. io/papers/31. pdf (Vol. 2, No. 8, p. 31).

Questions?