Topologically Informed Graph Neural Networks

Ben Cullen

Università di Pisa

Ingredients for Graph Neural Networks

A brief introduction

UNIVERSITÀ DI PISA

Ingredients for Graph Neural Networks

- Traditional ML techniques cannot process native graph structured data.

  $\Rightarrow$ **Graph neural nets (GNNs) can!**

- Three core concepts that we require:
  1. **Permutation invariance;**
  2. **Permutation equivariance;**
  3. **Locality.**
- With the above, can define the general class GNNs.

Permutation Invariance

Want $f: G \mapsto \mathbb{R}^{n \times d}$ indifferent to 'representation' of G:

$$G_1 \cong G_2 \Rightarrow f(G_1) = f(G_2)$$

- Consider $f: \{x_1, \cdots, x_n\} \mapsto \mathbb{R}^d$.
- Must construct feature matrix $\boldsymbol{X}$:

  construction ⇒ ordering of $\{x_1, \cdots, x_n\}$!
- f **must** be indifferent to labelling, i.e.:

  $f(\boldsymbol{PX}) = f(\boldsymbol{X})$ for $\boldsymbol{P} \in S_n$

  ⇒ **permutation invariance**

Courtesy: www.math.cmu.edu

Permutation Equivariance

- Now, consider $f: G = (\boldsymbol{X}, \boldsymbol{A}) \mapsto \mathbb{R}^{n \times d}$ i.e. output over every node.
- Shuffling labels should **at most** shuffle outputs of f, i.e.:

$$f(\boldsymbol{PX}, \boldsymbol{PAP}^T) = \boldsymbol{P}f(\boldsymbol{X}, \boldsymbol{A}) \text{ for } \boldsymbol{P} \in S_n$$

$\Rightarrow$ **permutation equivariance**

Courtesy: www.math.cmu.edu

Locality

- Graphs have natural notion of **locality.**
- For every v, define its **(1-hop) neighbourhood:**

$$N_v := \{u \in V : (v,u) \in E\}$$

- Define multiset $N := \{\{N_v : v \in V\}\}$.
- Want permutation equivariant f that exploit locality of G

$\Rightarrow$ **define f over the multiset N appropriately!**

A general framework for GNNs

Putting ingredients together, construct $f: (\boldsymbol{X}, \boldsymbol{A}) \mapsto \mathbb{R}^{n \times d}$:

$$f(\boldsymbol{X}, \boldsymbol{A}) := \begin{bmatrix} h_{v_1} \\ \vdots \\ h_{v_n} \end{bmatrix} = \begin{bmatrix} g(N_{v_1}, \boldsymbol{A}) \\ \vdots \\ g(N_{v_n}, \boldsymbol{A}) \end{bmatrix},$$

Where $g: N_v \mapsto \mathbb{R}^d$ is:

1. **permutation invariant;**
2. **local.**

$\Rightarrow f$ **is permutation equivariant.**

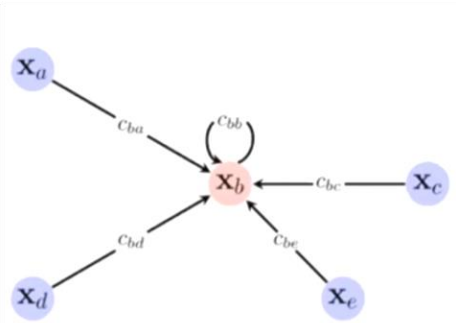
- GNN can be applied across three main tasks:
  - **Node focused;**
  - **Graph focused;**
  - **Edge focused.**

Flavours of GNNs

- Local functions g determine behaviour of overall model.
- Three main flavours of g:
  - **Convolutional;**
  - **Attentional;**
  - **Message Passing.**

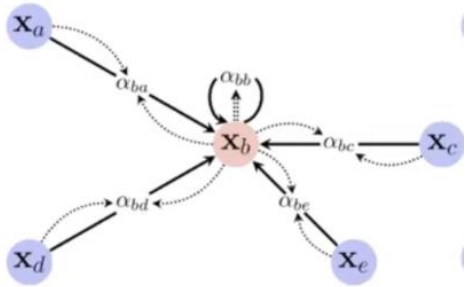
Message passing most general

⇒ **most expressive!**



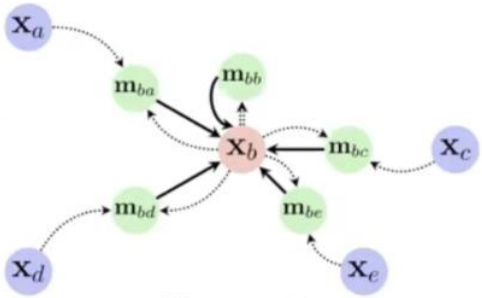
Convolutional

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} c_{ij}\psi(\mathbf{x}_j)\right)$$



Attentional

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} a(\mathbf{x}_i, \mathbf{x}_j)\psi(\mathbf{x}_j)\right)$$



Message-passing

$$\mathbf{h}_i = \phi\left(\mathbf{x}_i, \bigoplus_{j \in \mathcal{N}_i} \psi(\mathbf{x}_i, \mathbf{x}_j)\right)$$

Courtesy: P. Veličković

Expressivity:

Analysing the power of GNNs

UNIVERSITÀ DI PISA

Expressivity

- The study of:
  - Computational capabilities, and;
  - Behaviour of GNNs.
- Canonical framework relies on **graph isomorphism problem (GIP)**:

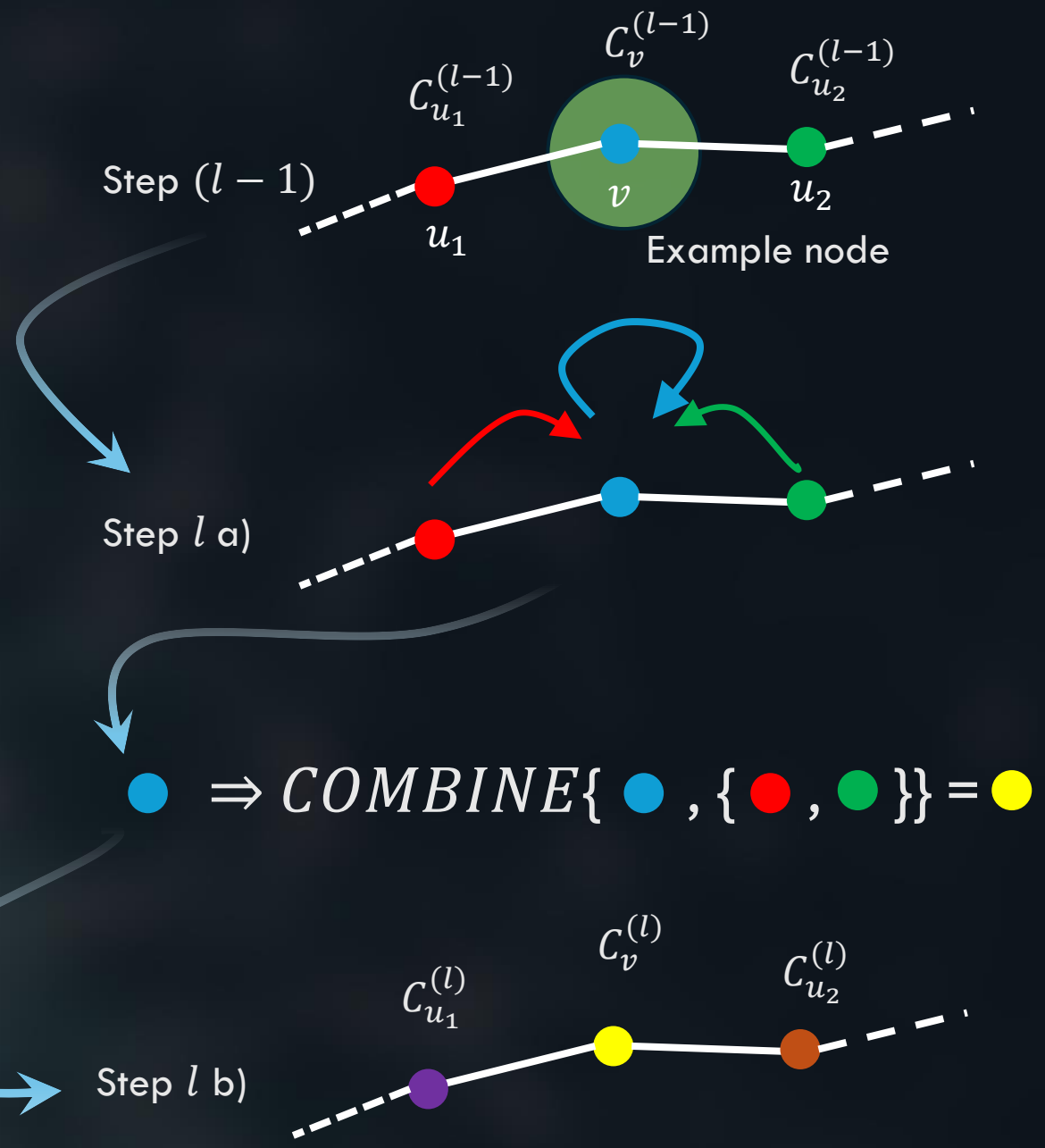
  "Given two graphs G_1 and G_2, can we decide if they are isomorphic or not?"
- Gold standard for heuristics:

  $\Rightarrow$ **Weisfeiler-Lehman** graph isomorphism test.

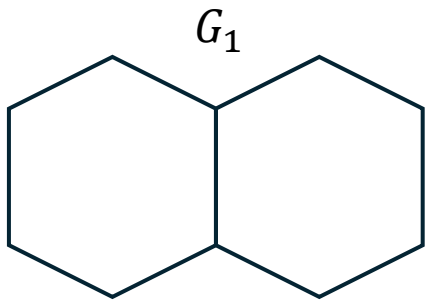
Weisfeiler-Lehman test

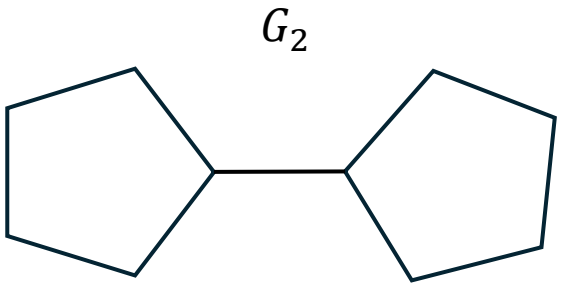
- Represent G_1 and G_2 as multisets of node colourings $C^{(l,1)}$ and $C^{(l,2)}$.
- Iteratively refine node colourings.
- Terminate algorithm if:

1. No bijection between $C^{(l,1)}$ and $C^{(l,2)} \Rightarrow G_1 \ncong G_2$
2. Colours at step l 'same' as at step $(l-1)$

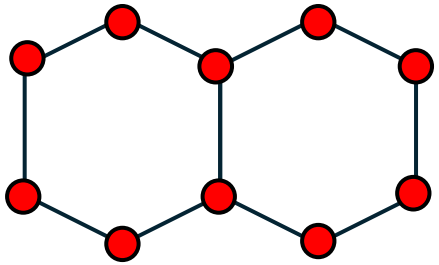


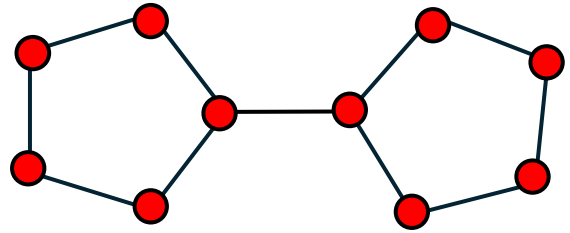
Example



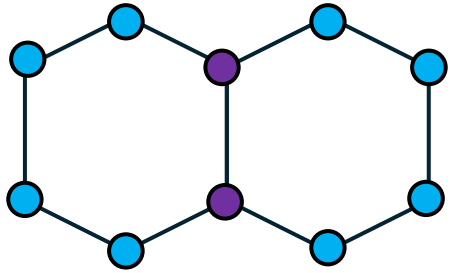


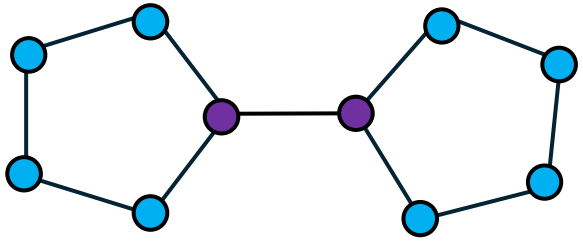
Step 1



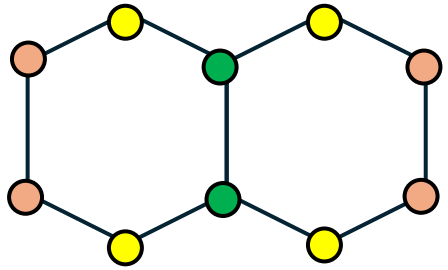


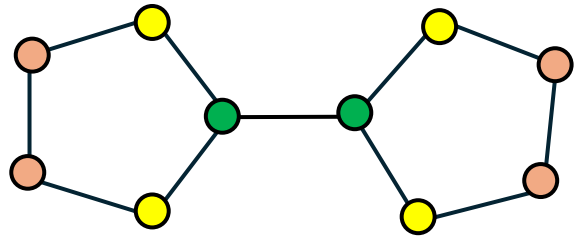
Step 2



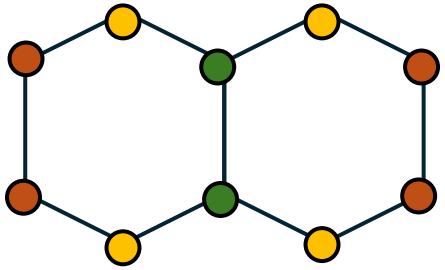


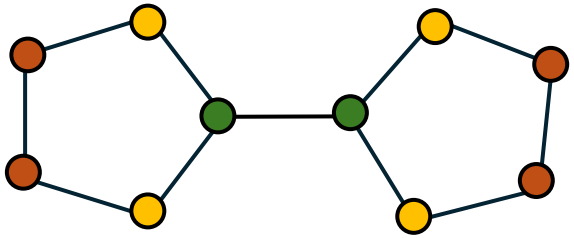
Step 3





Step 4





TERMINATE

1-WL & GNN Equivalence I

- 1-WL is reminiscent of the message passing mechanism!

"A standard message passing GNN is at most as expressive as 1-WL"

Theorem (Xu et al., Morris et al.): equivalence holds if:

1. *Composition of MSG, AGG and UPT constructs injective map from $\left(h_v^{(l-1)}, \left\{h_u^{(l-1)} : u \in N_v\right\}\right) \rightarrow h_v^{(l)}$ and;*
2. *$f_{readout} : \left\{h_v^{(L)} : v \in V\right\} \mapsto \mathbb{R}^d$ is injective.*

- Conditions are sufficient but not necessary:

⇒ **Can we find necessary conditions?**

Keyulu Xu et al. "How powerful are graph neural networks?" In: arXiv preprint arXiv:1810.00826 (2018)
Christopher Morris et al. "Weisfeiler and leman go neural: Higherorder graph neural networks". In: Proceedings of the AAAI conference on artificial intelligence. Vol. 33. 01. 2019, pp. 4602–4609

1-WL & GNN Equivalence II

- Expressivity frameworks ⇒ can understand a lot about GNNs!
- Some known classes of graphs impervious to 1-WL, e.g. k-regular:

'Optimal message passing GNN architectures

cannot distinguish k-regular graphs.'

- **What is the complete characterisation of the classes of graphs impervious to 1-WL or higher k-WL tests?**
  - Important: could help us design more expressive graph-based models!

Developments & Drawbacks

- Higher order hierarchical heuristics, e.g. k-WL (citation)
  - k-tuples of adjacent nodes used to construct new colourings:

    ⇒ information content in each colouring is greater;

    ⇒ mechanism becomes more 'non-local' for greater k, can distinguish more substructures in graphs;

    ⇒ computationally expensive!
- Inspired hierarchical models: k-GNNs, Message Passing Simplicial Networks (MPSNs), Cell Complex Networks (CWNs) etc.:
  - All much more powerful than 1-WL but very **computationally expensive;**
  - Message passing mechanism becomes more **'non-local'** ⇒ generalisation issues?

Christopher Morris et al. "Weisfeiler and leman go neural: Higherorder graph neural networks". In: Proceedings of the AAAI conference on artificial intelligence. Vol. 33. 01. 2019, pp. 4602–4609.
Cristian Bodnar et al. "Weisfeiler and lehman go topological: Message passing simplicial networks". In: International Conference on Machine Learning. PMLR. 2021, pp. 1026–1037.
Bodnar, Cristian, et al. "Weisfeiler and lehman go cellular: Cw networks." *Advances in Neural Information Processing Systems* 34 (2021): 2625-2640.

Developments & Drawbacks

- A more complete notion of expressivity?

  'Similarity' more useful than 'sameness'

- Can we develop an 'approximate' version of the Weisfeiler-Lehman test?
- Models with 1-WL expressivity (very good) perform poorly on **substructure identification (Chen et. al)**.

Zhengdao Chen et al. "Can graph neural networks count substructures?" In: Advances in neural information processing systems 33 (2020), pp. 10383–10395.

Algebraic Topology:

A new approach to GNNs

UNIVERSITÀ DI PISA

General Idea

- Want GNN models that can exploit:
  1. Relational information among nodes, and;
  2. Structural information of the larger topology.
- Algebraic topology: encode topological structure of G in algebraic objects.
- Use algebraic objects as means for improving substructure identification while preserving relational information.

  $\Rightarrow$ **Our idea: use graph polynomials!**

Graph Polynomials

- Active area of combinatorics/algebraic graph theory.
- Graph polynomial: polynomial representation of G.
- Example: from adjacency matrix A of G, the characteristic polynomial $p_A(x, \lambda) := \det(A - \lambda I)$.
- Many graph polynomials exist
  - We consider the **Tutte-Whitney Polynomial**.

Tutte-Whitney Polynomial

$$T(G;x,y) \coloneqq \sum_{A \subseteq E} (x-1)^{r_G - r_G(A)} (y-1)^{n_G}$$
$$= \sum_{i,j \geq 0} b_{i,j} x^i y^j \,, \quad b_{i,j} \in \mathbb{Z}.$$

- Generalisation of the chromatic polynomial $P(G;\lambda)$.
- Encodes **many** interesting structures in:
  1. The evaluations of $T(G;x,y)$, and;
  2. The coefficients $b_{i,j}$.
- Proving above theorems (combinatorically) is **difficult.**
- **Can ML models learn to interpret $T(G;x,y)$ even in absence of theorems?**

Our Research I

- Integrating topological information $T(G; x, y)$ in a GNN:
  1. **GNNconcat;**
  2. **GNNhybrid.**
- Testing models on toy data sets:
  - Can they identify structures such as triangles, squares etc.?
  - Do they balance locality with substructure identification well?
- Test best models on real world data where substructure identification is important but not principal task.

Our Research II

- GNNconcat: redundant data augmentation?
- GNNhybrid: better model design?
  - Introduce specialised unit for polynomial interpretation?
- Easy to define distances on the space of finite bivariate polynomials:
  - Similarity metric?
- Biggest stumbling block:
  - Polynomials neglect node information;
  - Computational complexity of $T(G; x, y)$;
  - $T(G; x, y)$ has nice properties (e.g. multiplicativity): **can we exploit them?**
  - More information than needed?
    1. Calculate T only over subgraphs of interest?
    2. Better choices of graph polynomials that are less expensive?

Addenda:

Additional slides

UNIVERSITÀ DI PISA

Permutation Invariance

- If $G_1 \cong G_2$ then $f(G_1) = f(G_2)$!
- Consider just the set of feature vectors:

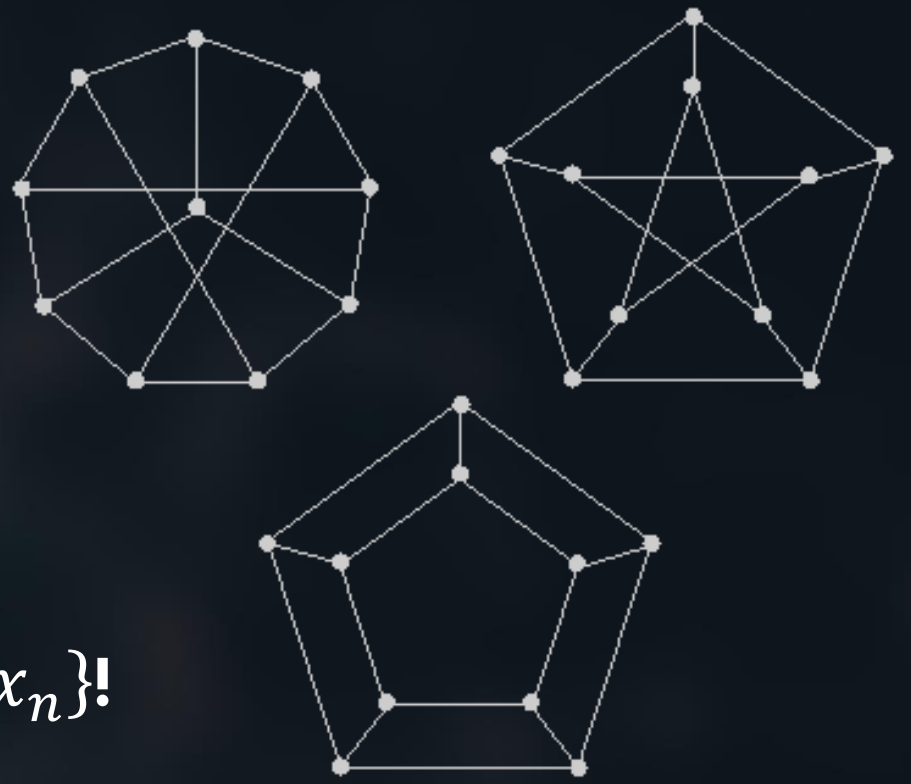
$$X = \{x_1, \cdots, x_n\} \subseteq \chi, x_i \in \mathbb{R}^k.$$

- Let $f: \chi \to \mathbb{R}$.
- To apply f to X, must construct a feature matrix $\boldsymbol{X}$:

construction ⇒ ordering of $\{x_1, \cdots, x_n\}$!

- f must be **permutation invariant:**

$$f(\boldsymbol{PX}) = f(\boldsymbol{X}) \text{ for } \boldsymbol{P} \in S_n$$



Courtesy: www.math.cmu.edu

Permutation Equivariance I

- Now, suppose $f: A \to \mathbb{R}^n$.
- f **still** must be agnostic to ordering of $\{x_1, \cdots, x_n\}$!
- f must be **permutation equivariant:**

$$\Rightarrow f(\boldsymbol{PX}) = \boldsymbol{P}f(\boldsymbol{X}) \text{ for } \boldsymbol{P} \in S_n.$$

Courtesy: www.math.cmu.edu

Permutation Equivariance II

- Now, define $f: G \rightarrow \mathbb{R}^n, G = (V, E)$.
- Represent G via adjacency matrix A

$$a_{ij} = \begin{cases} 1, if\ (i,j) \in E, \\ 0, otherwise. \end{cases}$$

- P must now be applied to both rows and columns in A such that $\boldsymbol{PAP^T = A}$.
- Hence, **permutation equivariance becomes:**

  $\Rightarrow f(\boldsymbol{PX}, \boldsymbol{PAP^T}) = f(\boldsymbol{X}, \boldsymbol{A})$ for $\boldsymbol{P} \in S_n$

Courtesy: www.math.cmu.edu

Message Passing GNNs

Let G be an attributed graph. Then a message passing GNN builds latent vector representations h_v at each node v in the following iterative fashion:

1. Initialise: $h_v^{(0)} \leftarrow x_v, \forall v \in V$;
2. For $0 < l \leq L$, update the latent vectors $h_v^{(l)}$:
   I. Message: $m_{vu}^{(l)} \leftarrow MSG^{(l-1)}\left(h_v^{(l-1)}, h_u^{(l-1)}\right)$ for all $u \in N_v$;
   II. Aggregate: $a_v^{(l)} \leftarrow AGG^{(l-1)}\left(\left\{m_{vu}^{(l-1)} : u \in N_v\right\}\right)$;
   III. Update: $h_v^{(l)} \leftarrow UPT^{(l-1)}\left(h_v^{(l-1)}, a_v^{(l-1)}\right)$.

1-D Weisfeiler-Lehman Test

Let G_1 and G_2 be attributed graphs. Then:

1. Initialise each node $v \in V$ with colour $C_v^{(i,0)} \leftarrow X_v^{(i)}$ for $i \in \{1,2\}$;
2. For $l = 1, 2, \cdots, \max\{|V_1|, |V_2|\}$:
   a) Update node colours: $C_v^{(i,l)} \leftarrow HASH\left(C_v^{(i,l-1)}, \left\{\left\{C_u^{(i,l-1)} : u \in N_v\right\}\right\}\right)$ for all $v \in V$ and $i \in \{1,2\}$;
   b) Test: If $\left\{\left\{C_v^{(1,l)} : v \in V\right\}\right\} \neq \left\{\left\{C_v^{(2,l)} : v \in V\right\}\right\}$ then $G_1 \ncong G_2$.

- If colours in step l 'same' as in step $(l-1)$, **terminate.**
- $HASH$ is injective.