Graph Representation Learning

Please use the official LaTeX template to type your answers available in Moodle. Please respect the notation from Table 1 in your answers whenever applicable:

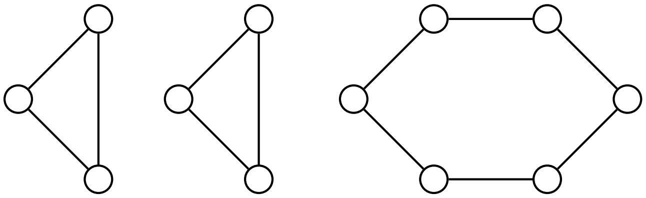
Table 1: Notation.

σ	An element-wise non-linearity.
t	Iteration, or layer t.
$d^{(t)}$	The dimension of a vector at iteration t.
d	The dimension of a vector, and an abbreviation for $d^{(0)}$.
$\mathbf{1}^d \in \mathbb{R}^d$	A d-dimensional vector of all 1's.
$\mathbb{I}^d \subseteq \mathbb{R}^d$	The set of d-dimensional one-hot vectors.
$\mathbb{B}$	Boolean domain $\{0, 1\}$.
$\mathbf{b}^{(t)} \in \mathbb{R}^d$	A bias vector.
$\mathbf{x}_u \in \mathbb{R}^d$	The feature of a node $u \in V$.
$\mathbf{h}_u^{(t)} \in \mathbb{R}^{d^{(t)}}$	The representation of a node $u \in V$ at layer t.
$\mathbf{z}_u = \mathbf{h}_u^{(T)} \in \mathbb{R}^{d^{(T)}}$	The final representation of a node $u \in V$ after T layers/iterations.
$\mathbf{W}_x^{(t)} \in \mathbb{R}^{d^{(t+1)} \times d^{(t)}}$	Learnable parameter matrix at layer t.
MLP	A multilayer perceptron with ReLU as nonlinearity.

Question 1

(a) Let us define $f : V_G \to \mathbb{B}$ the boolean function which returns 1 **if the node is part of a subgraph strictly larger than** 3 **and** 0 **otherwise**.

Consider the following two graphs:



(a) G_1 (b) G_2

Figure 1: Graphs G_1 and G_2. All nodes have the same initial features.

These graphs cannot be distinguished by the 1-dimensional Weisfeiler Lehman Algorithm: since the model architecture $\mathcal{S}$ is no more expressive thant the 1-dimensional Weisfeiler Lehman Algorithm, there is no parameterisation of $\mathcal{S}$ that can distinguish these nodes.

$$\boxed{\text{There is no parameterisation of } \mathcal{S} \text{ that can represent the function } f \text{ for all graphs.}}$$

(b) Let d be the degree limit such that we consider graphs whose nodes all have a degree lower than or equal to d.

- If $d \leq 1$, then models $\mathcal{M}$ and $\mathcal{S}$ are equivalent since the sum aggregation and mean aggregation are equivalent in that case. Then the result holds.
- We now suppose that $d \geq 2$.

In the following, we will disprove the claim that for any function f such that $f(G) \in \mathbb{B}$, if there is a parameterisation of $\mathcal{S}$ satisfying $f(G) = \tilde{\mathcal{S}}(G)$ for all d-bounded-degree graphs G then there is a parameterisation of $\mathcal{M}$ satisfying $f(G) = \tilde{\mathcal{M}}(G)$ for all d-bounded-degree graphs G.

Intuition: The mean aggregation loses information about the number of neighbours in comparison to the sum aggregation.

Suppose that there exists a parameterisation of $\mathcal{S}$ satisfying $f(G) = \tilde{\mathcal{S}}(G)$ for all bounded-degree graphs G. Denote $T_{\mathcal{S}} \in \mathbb{Z}^+$ the number of layers associated with the parameterisation. Then:

$$\forall 0 < t \leq T_{\mathcal{S}}, h_{u,\mathcal{S}}^{(t)} = \mathrm{MLP}_{\mathcal{S}}^{(t)}\left(W_{s,\mathcal{S}}^{(t)} h_{u,\mathcal{S}}^{(t-1)} + W_{n,\mathcal{S}}^{(t)} \sum_{v \in \mathcal{N}(u)} h_{v,\mathcal{S}}^{(t-1)} + b_{\mathcal{S}}^{(t)}\right)$$

Suppose that there exists a parameterisation of $\mathcal{M}$ satisfying $f(G) = \tilde{\mathcal{M}}$ for all bounded-degree graphs G. Denote $T_{\mathcal{M}} \in \mathbb{Z}^+$ the number of layers associated with the parameterisation. Then:

$$\forall 0 < t \leq T_{\mathcal{M}}, h_{u,\mathcal{M}}^{(t)} = \text{MLP}_{\mathcal{M}}^{(t)}\Big(W_{s,\mathcal{M}}^{(t)} h_{u,\mathcal{M}}^{(t-1)} + W_{n,\mathcal{M}}^{(t)} \sum_{v \in \mathcal{N}(u)} \frac{h_{v,\mathcal{M}}^{(t-1)}}{|\mathcal{N}(u)|} + b_{\mathcal{M}}^{(t)}\Big)$$

Let G be a graph such that there is no isolated node and such that all nodes have the same node features:

$$\forall u \in V_G, x_u = x_0 \in [0,1]^d, x_0 \neq 0_{\mathbb{R}^d}$$

Then:

$$\forall t \in \{1, ..., T_{\mathcal{S}}\}, h_{u,\mathcal{S}}^{(t)} = \text{MLP}_{\mathcal{S}}^{(t)}\Big(\big(W_{s,\mathcal{S}}^{(t)} + |\mathcal{N}(u)| W_{n,\mathcal{S}}^{(t)}\big) h_{u,\mathcal{S}}^{(t-1)} + b_{\mathcal{S}}^{(t)}\Big)$$
$$\forall t \in \{1, ..., T_{\mathcal{M}}\}, h_{u,\mathcal{M}}^{(t)} = \text{MLP}_{\mathcal{M}}^{(t)}\Big(\big(W_{s,\mathcal{M}}^{(t)} + W_{n,\mathcal{M}}^{(t)}\big) h_{u,\mathcal{M}}^{(t-1)} + b_{\mathcal{M}}^{(t)}\Big)$$

Note that **the embeddings in $\tilde{\mathcal{S}}$ depend only on the initial feature x_u and on the degree of a node**.

Also note that the parameterized model $\tilde{\mathcal{M}}$ returns the same embedding for all nodes and that this embedding is a function of the initial feature x_u: **the embeddings do not depend on degree of the node for the model $\tilde{\mathcal{M}}$**. This means that a node will have the same final representation whatever its degree.

So the result can only stand if for all pairs of graphs $G_1 = (V_{G_1}, E_{G_1}, X_{G_1}), G_2 = (V_{G_2}, E_{G_2}, X_{G_2})$:

$$\begin{cases} V_{G_1} = V_{G_2} \\ X_{G_1} = X_{G_2} \\ \{u | \mathcal{N}_{G_1}(u) = \emptyset\} = \{u | \mathcal{N}_{G_2}(u) = \emptyset\} \end{cases} \implies f(G_1) = f(G_2)$$

As a counter-example, we introduce the function $\boxed{f_k : (V_G, E_G, X_G) \mapsto \mathbb{I}\big(\max_{u \in V_G} \deg(u) = k\big)}$ for any $k \in \mathbb{N}$. Clearly, f_k does not satisfy the condition above.

For all $2 \leq k \leq d$, there is no parameterisation of $\mathcal{M}$ satisfying $f_k(G) = \tilde{\mathcal{M}}(G)$.

We will conclude by showing that there exists a parameterisation of $\mathcal{S}$ that can represent f_k for any $2 \leq k \leq d$.

The 1-dimensional Weisfeiler-Lehman algorithm can distinguish any two graphs that differ from a node degree, so there exists a parameterisation of $\mathcal{S}$ that distinguish such two graphs. We can adapt such a parameterisation so that $f_k(G) = \tilde{\mathcal{S}}(G)$ for any graph G such that G has a degree lower than d (we can multiply/add a factor to weights, and use the sigmoid activation function).

(c) Let d be the degree limit such that we consider graphs whose nodes all have a degree lower than or equal to d.

- If $d \leq 1$, then models $\mathcal{M}$ and $\mathcal{S}$ are equivalent since the sum aggregation and mean aggregation are equivalent in that case. Then the result holdS.
- We now suppose that $d \geq 2$.

Intuition: Because mean aggregation loses information about the number of neighbours, $\mathcal{M}$ should behave similarly whatever the number of identical neighbours, whereas $\mathcal{S}$ should take it into account.

Let us fix single-layer parameterisation of $\mathcal{M}$ as follows:

$$\forall u \in V_G, h^{(0)}_{\mathcal{M},u} = x_u \text{ and } h^{(1)}_{\mathcal{M},u} = \text{MLP}\Big(W_s h^{(0)}_{\mathcal{M},u} + W_n \sum_{v \in \mathcal{N}(u)} \frac{h^{(0)}_{\mathcal{M},u}}{|\mathcal{N}(u)|} + b\Big)$$

Let us denote W and c the parameters of the multilayer perceptron. The parameterisation can be rewritten as follows:

$$\forall u \in V_G, h^{(0)}_{\mathcal{M},u} = x_u \text{ and } h^{(1)}_{\mathcal{M},u} = \text{ReLU}\Big(WW_s x_u + WW_n \sum_{v \in \mathcal{N}(u)} \frac{x_u}{|\mathcal{N}(u)|} + Wb + c\Big)$$

In the following, we will disprove the claim that for any single-layer parameterisation of $\mathcal{M}$, there exists a single-layer parameterisation of $\mathcal{S}$ such that the node representations computed by $\mathcal{M}$ and $\mathcal{S}$ are identical for all bounded-degree graphs G, and for all nodes $u \in V_G$.

Let us take the special case where **all parameters for $\mathcal{M}$ are nonnegative and** $WW_n \neq 0$. Then we can omit the ReLU activation function since it will behave as the identity function (as the x_u are one-hot encoding vectors). Now:

$$\forall u \in V_G, h^{(0)}_{\mathcal{M},u} = x_u \text{ and } h^{(1)}_{\mathcal{M},u} = WW_s x_u + WW_n \sum_{v \in \mathcal{N}(u)} \frac{x_u}{|\mathcal{N}(u)|} + Wb + c$$

Let us fix a single-layer parameterisation of $\mathcal{S}$ as follows:

$$\begin{aligned}\forall u \in V_G, h^{(0)}_{\mathcal{S},u} = x_u \text{ and } h^{(1)}_{\mathcal{S},u} &= \text{MLP}'\Big(W'_s h^{(0)}_{\mathcal{S},u} + W'_n \sum_{v \in \mathcal{N}(u)} h^{(0)}_{\mathcal{S},u} + b'\Big) \\ h^{(1)}_{\mathcal{S},u} &= \text{ReLU}\Big(W'W'_s x_u + W'W'_n \sum_{v \in \mathcal{N}(u)} x_u + W'b' + c'\Big)\end{aligned}$$

Assume that the node representations computed by $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{S}}$ are identical for all bounded-degree graphs G and for all nodes $u \in V_G$.

Since the activation function for $\tilde{\mathcal{M}}$ behaves as the identity function for all input, then so does it for $\tilde{\mathcal{S}}$. We can therefore write:

$$\forall u \in V_G, h^{(0)}_{\mathcal{S},u} = x_u \text{ and } h^{(1)}_{\mathcal{S},u} = W'W'_s x_u + W'W'_n \sum_{v \in \mathcal{N}(u)} x_u + W'b' + c'$$

First, clearly the bias terms must be the same: $\boxed{Wb + c = W'b' + c'}$.

Now take a graph G with only isolated nodes, ie $E_G = \emptyset$. Then:

$$\begin{aligned}
&\forall u \in V_G, WW_s x_u + Wb + c = W'W_s' x_u + Wb + c \\
&\text{ie } \forall u \in V_G, WW_s x_u = W'W_s' x_u \\
&\text{Hence } \boxed{WW_s = W'W_s'}
\end{aligned}$$

This last equation can formally be derived by using all one-hot encoding vectors that have all elements but 1 set to 0.

We now have that for all bounded-degree graphs G, for all nodes $u \in V_G$:

$$WW_n \sum_{v \in \mathcal{N}(u)} \frac{x_u}{|\mathcal{N}(u)|} = W'W_n' \sum_{v \in \mathcal{N}(u)} x_u$$

Similarly, taking the case of a graph G with all nodes of degree 1, we can show that:

$$\boxed{WW_n = W'W_n}$$

Now taking the case of a graph G with node $u_0 \in V_G$ such that u_0 has exactly two neighbours v_0 and v_1. Then:

$$\begin{aligned}
&\forall x_{v_0}, x_{v_1} \in \mathbb{I}^d, WW_n\left(\frac{x_{v_0}}{2} + \frac{x_{v_1}}{2}\right) = WW_n(x_{v_0} + x_{v_1}) \\
&\forall x_{v_0}, x_{v_1} \in \mathbb{I}^d, \frac{x_{v_0}}{2} + \frac{x_{v_1}}{2} = 0 \text{ because } WW_n \neq 0
\end{aligned}$$

Which cannot hold because of the one-hot encoding.

This **disproves the statement** that for any single-layer parameterisation of $\mathcal{M}$, there exists a single-layer parameterisation of $\mathcal{S}$ such that the node representations computed by $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{S}}$ are identical for all bounded-degree graphs G and for all nodes $u \in V_G$.

Question 2

(a) We define the variation of the 1-dimensional Weisfeiler Lehman algorithm as follows:

Let $G = (V_G, E_G, c)$ be a graph with an initial coloring $c(G) : V_G \to \mathbb{I}^d$.

1. **Initialization**: $\forall u \in V_G, \kappa^{(0)}(G)(u) = c(G)(u)$.
2. **Refinement**: Let t be a positive integer and assume that $\kappa^{(t-1)}(G)$ has already been defiend. All nodes $u \in V_G$ are then recursively recolored at iteration t as follows:

$$\forall u \in V_G, \kappa^{(t)}(u) = \tau\Big(\kappa^{(t-1)}(u), \{\{\kappa^{(t-1)}(v) | v \in \mathcal{N}_R(u)\}\}, \{\{\kappa^{(t-1)}(v) | v \in \mathcal{N}_P(u)\}\}\Big)$$

   Where τ bijectively maps any triplet (composed of a color and two multisets of colors) to a unique color.
3. **Stop**: The algorithm terminates at iteration j, where j is the minimal integer satisfying:

$$\forall u, v \in V_G, \kappa^{(j+1)}(u) = \kappa^{(j+1)}(v) \iff \kappa^{(j)}(u) = \kappa^{(j)}(v)$$

Let us demontrate that the stopping condition is always achieved.

For a given integer j, the set $\{\kappa^{(j)}(u) | u \in V_G\}$ is non-empty, of finite cardinality and upper-bounded by the number of nodes in the graph $\mathrm{Card}(V_G)$.

Let j be an integer, $u, v \in V_G$ two nodes of a graph G. Suppose that $\kappa^{(j)}(u) \neq \kappa^{(j)}(v)$. Then the triplet that is given to the bijection map τ is different for node u and node v, so τ returns a different color for both these nodes and $\kappa^{(j+1)}(u) \neq \kappa^{(j+1)}(v)$. This proves that:

$$\forall u, v \in V_G, \forall j \in \mathbb{N}, \kappa^{(j+1)}(u) = \kappa^{(j+1)}(v) \implies \kappa^{(j)}(u) = \kappa^{(j)}(v)$$

This proves that the series which takes for i-th value the cardinality of the set $\{\kappa^{(i)}(u) | u \in V_G\}$ is a non-decreasing upper-bounded series of integer values. Thus it is a stationary sequence and its stationary value corresponds to the maximum cardinality.

There exists a minimal integer j such that $\forall k \geq j, \mathrm{Card}\big(\{\kappa^{(j)}(u) | u \in V_G\}\big) = \mathrm{Card}\big(\{\kappa^{(k)}(u) | u \in V_G\}\big)$. This integer j is exactly the number of iterations after which the stopping criterion is satisfied. **The algorithm terminates.**

(b) Let G be a graph with initial node features $\{\mathbf{x}_u = c(G)(u) | u \in V_G\}$. Let us fix a parameterisation of $\mathcal{A}$ with T layers.

Proof. Let us proceed by induction to show that the iterative test κ satisfies the following property $\mathcal{P}$:

$$\boxed{\forall 0 \leq t \leq T, \forall u, v \in V_G, \kappa^{(t)}(G)(u) = \kappa^{(t)}(G)(v) \implies h_u^{(t)} = h_v^{(t)}}$$

- For $t = 0$: Let u, v be two nodes in V_G. Then since the initial coloring c is injective, $x_u \neq x_v \implies c(G)(u) \neq c(G)(v)$. This proves the property $\mathcal{P}$ for $t = 0$.
- Let us fix $1 \leq t \leq T$ and assume that the property $\mathcal{P}$ is true for $t-1$. Let us fix $u, v \in V_G$ such that $\kappa^{(t)}(G)(u) = \kappa^{(t)}(G)(v)$. Then, since τ is injective:

$$\begin{cases} \kappa^{(t-1)}(G)(u) = \kappa^{(t-1)}(G)(v) \\ \{\{\kappa^{(t-1)}(G)(w) | w \in N_R(u)\}\} = \{\{\kappa^{(t-1)}(G)(w) | w \in N_R(v)\}\} \\ \{\{\kappa^{(t-1)}(G)(w) | w \in N_P(u)\}\} = \{\{\kappa^{(t-1)}(G)(w) | w \in N_P(v)\}\} \end{cases} \tag{1}$$

  Since the property $\mathcal{P}$ is true for $t-1$, the set of equations can be rewritten as:

$$\begin{cases} h_u^{(t-1)} = h_v^{(t-1)} \\ \{\{h_w^{(t-1)} | w \in N_R(u)\}\} = \{h_w^{(t-1)} | w \in N_R(v)\}\} \\ \{\{h_w^{(t-1)} | w \in N_P(u)\}\} = \{\{h_w^{(t-1)} | w \in N_P(v)\}\} \end{cases} \tag{2}$$

  Therefore,

$$\begin{cases} h_u^{(t-1)} = h_v^{(t-1)} \\ W_R^{(t)} \sum\limits_{w \in \mathcal{N}_R(u)} h_w^{(t-1)} = W_R^{(t)} \sum\limits_{w \in \mathcal{N}_R(v)} h_w^{(t-1)} \\ W_P^{(t)} \sum\limits_{w \in \mathcal{N}_P(u)} h_w^{(t-1)} = W_P^{(t)} \sum\limits_{w \in \mathcal{N}_P(v)} h_w^{(t-1)} \end{cases} \tag{3}$$

  According to the definition of the model architecture $\mathcal{A}$, it is clear from the set of equations that the node representation at iteration t will be the same for nodes u and v:

$$h_u^{(t)} = h_v^{(t)}$$

  The property $\mathcal{P}$ is satisfied for t.

□

We just proved that **the model architecture $\mathcal{A}$ is at most as powerful as the iterative test κ.**

(c) In this paragraph, we aim to prove that there exists a parameterisation of $\mathcal{A}$ such that the model is as powerful as κ.

Intuition: At each iteration the graph of the neighbourhood of a node can be seen as the superposition of one graph with the neighbourhood of type R and one graph with the neighbourhood of type P. There exists a parameterization of a basic GNN that can distinguish the R-neighbourhood graphs, and another one that can distinguish the P-neighbourhood graphs. Combining the 2 set of parameters will give the result.

Let us denote MLP' as a MultiLayer Perceptron without nonlinearity.

Let G be a graph with initial node features $\{x_u = c(G)(u) | u \in V_G\}$. Let us fix $u, v \in V_G$ and $T \in \mathbb{Z}^+$.

Proof. For each $0 \leq t \leq T$, let us find a parameterisation of layer T such that

$$\forall u, v \in V_G, \kappa^{(t)}(G)(u) \neq \kappa^{(t)}(G)(v) \iff h_u^{(t)} \neq h_v^{(t)}$$

We proceed by induction.

For $t = 0$, the result always holds since the iterative test κ is initalized with the initial node features.

Fix $0 \leq t \leq T$. Assume that we have already determined the first $t-1$ layers.

Using the property proven in the previous question, we just need to ensure that:

$$\forall u, v \in V_G, \kappa^{(t)}(G)(u) \neq \kappa^{(t)}(G)(v) \implies h_u^{(t)} \neq h_v^{(t)}$$

Note that for all u, v nodes in V_G:

$$\begin{aligned}
\kappa^{(t)}(G)(u) \neq \kappa^{(t)}(G)(v) &\implies \begin{cases} \kappa^{(t-1)}(G)(u) \neq \kappa^{(t-1)}(G)(v) \\ \text{or} \\ \{\{\kappa^{(t-1)}(G)(w) | w \in \mathcal{N}_R(u)\}\} \neq \{\{\kappa^{(t-1)}(G)(w) | w \in \mathcal{N}_R(v)\}\} \\ \text{or} \\ \{\{\kappa^{(t-1)}(G)(w) | w \in \mathcal{N}_P(u)\}\} \neq \{\{\kappa^{(t-1)}(G)(w) | w \in \mathcal{N}_P(v)\}\} \end{cases} \\
&\implies \begin{cases} h_u^{(t-1)} \neq h_v^{(t-1)} \\ \text{or} \\ \{\{h_w^{(t-1)} | w \in \mathcal{N}_R(u)\}\} \neq \{\{h_w^{(t-1)} | w \in \mathcal{N}_R(v)\}\} \\ \text{or} \\ \{\{h_w^{(t-1)} | w \in \mathcal{N}_P(u)\}\} \neq \{\{h_w^{(t-1)} | w \in \mathcal{N}_P(v)\}\} \end{cases} \\
&\quad \text{because the } t-1\text{-th layer verifies the desired property}
\end{aligned}$$

Message Passing Neural Networks with sum aggregation can be as powerful as the 1-dimensional Weisfeiler Lehman algorithm, which can distinguish all pairs of the form $\left(h_u^{(t-1)}, \{\{h_v^{(t-1)} | v \in \mathcal{N}_R(u)\}\}\right)$. Therefore there are parameters $\text{MLP'}_1^{(t)}, W_{s,R}^{(t)}, W_R^{(t)}$ such that:

$$\forall u \in V_G, \tilde{h}_{R,u}^{(t)} = \text{MLP'}_1^{(t)}\left(W_{s,R}^{(t)} h_u^{(t-1)} + \sum_{v \in \mathcal{N}_R(u)} W_R^{(t)} h_v^{(t-1)}\right)$$

and

$$\forall u, v \in V_G, \begin{cases} h_u^{(t-1)} \neq h_v^{(t-1)} \\ \text{or} \\ \{\{h_w^{(t-1)} | w \in \mathcal{N}_R(u)\}\} \neq \{\{h_w^{(t-1)} | w \in \mathcal{N}_R(v)\}\} \end{cases} \implies \tilde{h}_{R,u}^{(t)} \neq \tilde{h}_{R,v}^{(t)}$$

Note that the ReLU non-linearity is more limiting since it squashes all negative values to 0.

Similarly, there are parameters $\text{MLP'}_2^{(t)}, W_{s,P}^{(t)}, W_P^{(t)}$ such that:

$$\forall u, v \in V_G, \begin{cases} h_u^{(t-1)} \neq h_v^{(t-1)} \\ \text{or} \\ \{\{h_w^{(t-1)} | w \in \mathcal{N}_P(u)\}\} \neq \{\{h_w^{(t-1)} | w \in \mathcal{N}_P(v)\}\} \end{cases} \implies \tilde{h}_{P,u}^{(t)} \neq \tilde{h}_{P,v}^{(t)}$$

Where $\forall u \in V_G, \tilde{h}_{P,u}^{(t)} = \text{MLP'}_2^{(t)}\Big(W_{s,P}^{(t)} h_u^{(t-1)} + \sum\limits_{v \in \mathcal{N}_P(u)} W_P^{(t)} h_v^{(t-1)}\Big)$.

Now let us fix a final MultiLayer Perceptron $\text{MLP}_F^{(t)}$ such that from two embeddings $\tilde{h}_{R,u}^{(t)}, \tilde{h}_{P,u}^{(t)}$, $\text{MLP}_F^{(t)}$ uniquely maps two embeddings $(\tilde{h}_{R,u}^{(t)}, \tilde{h}_{P,u}^{(t)})$ to a final embedding $h_u^{(t)}$.

Explicitly, we define the t-th layer as:

$$h_u^{(t)} = \text{MLP}_F^{(t)}\Big(\big(W_1^{(t)} W_{s,R}^{(t)} + W_2^{(t)} W_{s,P}^{(t)}\big) h_u^{(t-1)} + \sum_{v \in \mathcal{N}_R(u)} W_1^{(t)} W_R^{(t)} h_v^{(t-1)} + \sum_{v \in \mathcal{N}_P(u)} W_2^{(t)} W_P^{(t)} h_v^{(t-1)} + b_1^{(t)} + b_2^{(t)}\Big)$$

Where $W_1^{(t)}, b_1^{(t)}$ are the parameters of the layer $\text{MLP'}_1^{(t)}$ and $W_2^{(t)}, b_2^{(t)}$ are the parameters of the layer $\text{MLP'}_2^{(t)}$.

The combination of all these parameters gives:

$$\boxed{\kappa^{(t)}(G)(u) \neq \kappa^{(t)}(G)(v) \iff h_u^{(t)} \neq h_v^{(t)}}$$

□

(d) Intuition: The model architecture $\mathcal{A}$ capture local properties but distinguishes node types. It will show the same limitations as the basic Graph Neural Network model for graph-level properties. The model architecture $\mathcal{B}$ is a Message Passing Neural Network with Global Readout. It has the ability to capture graph-level properties but does not capture information about the node types and will not be able to distinguish two nodes with different node types.

We will show that architectures $\mathcal{A}$ and $\mathcal{B}$ cannot be ranked in terms of expressiveness, as one architecture can distinguish graphs that the other cannot.

- Architecture $\mathcal{B}$ can be more expressive than architecture $\mathcal{A}$.

This is due to the fact that $\mathcal{A}$ show the same limitations as a Message Passing Neural Network without global readout: information cannot travel across disconnected components, and the information of a node can only reach node that are within a L-hop of this node, L being the number of layers of the model.

As a simple example, take a pair of two graphs H and G that cannot be distinguished by the basic Graph Neural Network but can be distinguished by the Graph Neural Network with global readout (we know such a pair exists, see an example as [include example]). Define $R_H = V_H, P_H = \emptyset$ and $R_G = V_G, P_G = \emptyset$. (R_H, P_H) and (R_G, P_G) verify the conditions: $R \cup P = V$ and $R \cap P = \emptyset$. In this case, the model architecture $\mathcal{A}$ can be rewritten as a basic Graph Neural Network:

$$h_u^{(t)} = \mathrm{MLP}^{(t)}\Big(W_s^{(t-1)} h_u^{(t-1)} + \sum_{v \in \mathcal{N}_R} W_R h_v^{(t-1)}\Big)$$

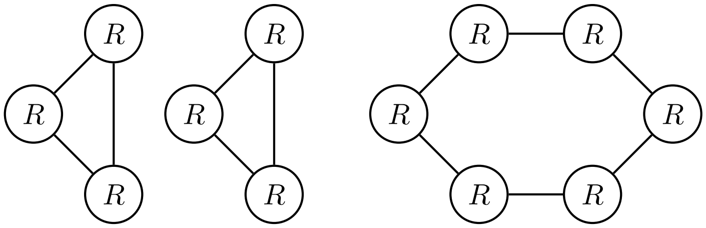
So a model of the form $\mathcal{A}$ is not able to distinguish H and G with the associated (R_H, P_H) and (R_G, P_G) but a model of the form $\mathcal{B}$ is.

Using logical propositions, $\mathcal{B}$ is able to distinguish the following formula:

$$Red(x) \wedge \exists y : Blue(y)$$

But $\mathcal{A}$ is not.

Using graphs, $\mathcal{B}$ is able to distinguish the following, whereas $\mathcal{A}$ is not:

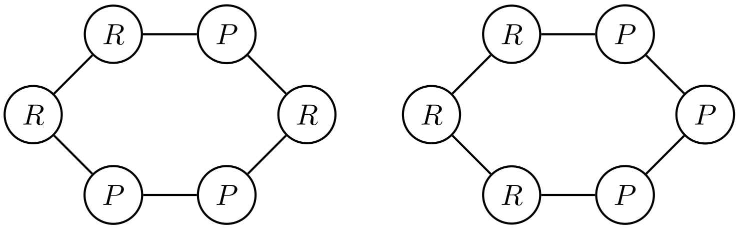


(a) G_1 (b) G_2

Figure 2: Graphs G_1 and G_2

- Architecture $\mathcal{A}$ can be more expressive than architecture $\mathcal{B}$.

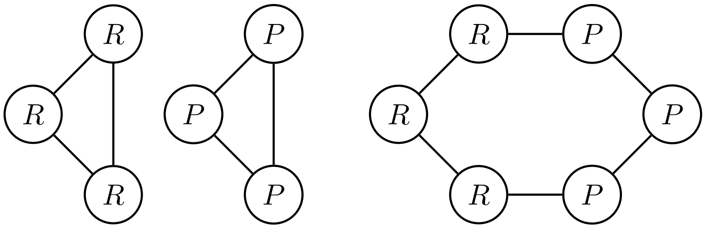
Since $\mathcal{A}$ does not consider R and P in a graph, it is clear that $\mathcal{B}$ is able to distinguish graphs that $\mathcal{A}$ cannot. Here is an example: Here, $\mathcal{A}$ is able to distinguish G_3 and G_4 but $\mathcal{B}$ is not.



(a) G_3 (b) G_4

Figure 3: Graphs G_3 and G_4

Note that $\mathcal{A}$ is strictly more expressive than a Message Passing Neural Network without global readout:



(a) G_5

(b) G_6

Figure 4: Graphs G_5 and G_6

Both $\mathcal{A}$ and $\mathcal{B}$ are able to distinguish G_5 and G_6. However a Message Passing Neural Network without global readout is not able to distinguish them.

$\boxed{\mathcal{A} \text{ and } \mathcal{B} \text{ can distinguish different classes of graphs.}}$ We cannot say that one is more expressive than the other.

Details:
In the following, we formally compare the expressiveness of the 1-Weisfeiler Lehman test with the κ test. Since we know that the 1-Weisfeiler Lehman can capture graded formal-logic, which makes them strictly less expressive than the Message Passing Neural Networks with readout which can capture C^2, we will be able to deduce comparisons of expressiveness between the architectures $\mathcal{A}$ and $\mathcal{B}$. This is additional formal study but does not change the results stated above.

We introduce the following notations:

- For two matrices X_1, X_2, we write $X_1 \simeq X_2$ when X_2 can be obtained by permuting the lines of X_1.
- wl is the 1-dimensional Weisfeiler-Lehman test.
- For a graph G, we define the graph-level colour as follows:

$$\forall t \in \mathbb{N}, \kappa^{(t)}(G) = \tau\Big(\{\{\kappa^{(t)}(u) | u \in V_G\}\}\Big)$$

  where τ bijectively maps any multisubset of colours to a unique colour.
- Given a graph G of node features X_G, X_G^R and X_G^P are the matrices of node features for nodes in R and P respectively.

Let $G_1 = (V_{G_1}, E_{G_1}, X_{G_1})$ and $G_2 = (V_{G_2}, E_{G_2}, X_{G_2})$ be two distinct graphs. For $i \in \{1, 2\}$, denote (R_i, P_i) the set of nodes of types R and P of graph G_i: $V_{G_i} = R_i \cup P_i$ and $R_i \cap P_i = \emptyset$.

Result: $\boxed{wl \text{ is at most as powerful as } \kappa}$.

Proof.

$$\begin{aligned}
\kappa^{(t)}(G_1)(u) = \kappa^{(t)}(G_2)(u) &\implies \begin{cases} \kappa^{(t-1)}(G_1)(u) = \kappa^{(t-1)}(G_2)(u) \\ \{\{\kappa^{(t-1)}(G_1)(v) | v \in \mathcal{N}_R(u)\}\} = \{\{\kappa^{(t-1)}(G_2)(v) | v \in \mathcal{N}_R(u)\}\} \\ \{\{\kappa^{(t-1)}(G_1)(v) | v \in \mathcal{N}_P(u)\}\} = \{\{\kappa^{(t-1)}(G_2)(v) | v \in \mathcal{N}_P(u)\}\} \end{cases} \\
&\implies \begin{cases} \kappa^{(t-1)}(G_1)(u) = \kappa^{(t-1)}(G_2)(u) \\ \{\{\kappa^{(t-1)}(G_1)(v) | v \in \mathcal{Q}(u)\}\} = \{\{\kappa^{(t-1)}(G_2)(v) | v \in \mathcal{Q}(u)\}\} \end{cases} \\
&\implies \begin{cases} wl^{(t-1)}(G_1)(u) = wl^{(t-1)}(G_2)(u) \\ \{\{wl^{(t-1)}(G_1)(v) | v \in \mathcal{Q}(u)\}\} = \{\{wl^{(t-1)}(G_2)(v) | v \in \mathcal{Q}(u)\}\} \end{cases} \\
&\implies wl^{(t)}(G_1)(u) = wl^{(t)}(G_2)(u)
\end{aligned}$$

□

Result: Let $u \in V_G$. Let us order the neighbours of u: $\mathcal{Q}(u) = \{v_1, ..., v_n\}$. Then for any $t \in \mathbb{N}$, if there is a permutation σ such that $(wl^{(t-1)}(G_1)(v_i))_{1 \leq i \leq n} = (wl^{(t-1)}(G_2)(\sigma(v_i)))_{1 \leq i \leq n}$, then the permutation is also true at the initialisation step: $(wl^{(0)}(G_1)(v_i))_{1 \leq i \leq n} = (wl^{(0)}(G_2)(\sigma(v_i)))_{1 \leq i \leq n}$.

Proof. By induction:

- The result holds for $t = 0$.
- Let $t \in \mathbb{N}^*$ and assume that the result holds for $t - 1$. Assume that such a permutation σ exists. Then $\forall i \in \{1, ..., n\}$, $wl^{(t)}(G_1)(v_i) = wl^{(t)}(G_2)(\sigma(v_i))$ so in particular, since the mapping is bijective, $wl^{(t-1)}(G_1)(v_i) = wl^{(t-1)}(G_2)(\sigma(v_i))$. Since the result holds for $t - 1$, it also holds for t.

□

We now formally compare $\mathcal{A}$ and $\mathcal{B}$ through the 1-WL algorithm.

- **Suppose that** $V_{G_1} \neq V_{G_2}$**.** Clearly, both $\mathcal{A}$ and $\mathcal{B}$ are sensitive to the number of nodes and will be able to distinguish G_1 and G_2.
- **Suppose that** $V_{G_1} = V_{G_2} = V_G$**. Suppose that** $X^R_{G_1} \simeq X^R_{G_2}$ **and** $X^P_{G_1} \simeq X^P_{G_2}$ (ie the node features for nodes of type R in G_1 are a permutation of node features for nodes of type R in G_1, and similarly for type P).

Let $u \in V_G$. Then $\mathcal{N}^{G_1}_R(u) = \mathcal{N}^{G_2}_R(u)$, otherwise there would be a node v such that $v \in \mathcal{N}^{G_1}_R(u)$ and $v \in \mathcal{N}^{G_2}_P(u)$ and $X^R_{G_1} \neq X^R_{G_2}$. Similarly, $\mathcal{N}^{G_1}_P(u) = \mathcal{N}^{G_2}_P(u)$. (Note that here the node are defined by their features.)

Result: $\boxed{\text{In this case, the 1-WL algorithm is as powerful as } \kappa}$.

$$\forall t \in \mathbb{N}, \forall u \in V_G, wl^{(t)}(G_1)(u) = wl^{(t)}(G_2)(u) \iff \kappa^{(t)}(G_1)(u) = \kappa^{(t)}(G_2)(u) (\mathcal{P}_t)$$

Proof. By induction (note that one way is already proved).

For $t = 0$, we have for all node $u \in V_G$, $wl^{(0)}(G_1)(u) = wl^{(0)}(G_2)(u) \iff \kappa^{(0)}(G_1)(u) = \kappa^{(0)}(G_2)(u)$.

Let us fix $t \in \mathbb{N}^*$ and assume that $\mathcal{P}_{t-1}$ is true. Let $u \in V_{G_1}$.

$$wl^{(t)}(G_1)(u) = wl^{(t)}(G_2)(u) \implies \begin{cases} wl^{(t-1)}(G_1)(u) = wl^{(t-1)}(G_2)(u) \\ \{\{wl^{(t-1)}(G_1)(v)|v \in \mathcal{Q}(u)\}\} = \{\{wl^{(t-1)}(G_2)(v)|v \in \mathcal{Q}(u)\}\} \end{cases}$$

If $\{\{wl^{(t-1)}(G_1)(v)|v \in \mathcal{Q}(u)\}\} = \{\{wl^{(t-1)}(G_2)(v)|v \in \mathcal{Q}(u)\}\}$ then there is a permutation σ that maps the elements $wl^{(t-1)}(G_1)(v)$ for $v \in \mathcal{Q}(u)$ to the elements $wl^{(t-1)}(G_2)(\sigma(v))$. Then this permutation also maps the initial node features of G_1 to those of G_2. Since we assumed that $X^R_{G_1} \simeq X^R_{G_2}$ and $X^P_{G_1} \simeq X^P_{G_2}$, σ can be decomposed into two cycles, one that maps R_1 to R_2 and the other one mapping P_1 to P_2. Thus if $\{\{wl^{(t-1)}(G_1)(v)|v \in \mathcal{Q}(u)\}\} = \{\{wl^{(t-1)}(G_2)(v)|v \in \mathcal{Q}(u)\}\}$, then $\{\{wl^{(t-1)}(G_1)(v)|v \in \mathcal{N}_P(u)\}\} = \{\{wl^{(t-1)}(G_2)(v)|v \in \mathcal{N}_P(u)\}\}$ and $\{\{wl^{(t-1)}(G_1)(v)|v \in \mathcal{N}_P(u)\}\} = \{\{wl^{(t-1)}(G_2)(v)|v \in \mathcal{N}_P(u)\}\}$. Hence:

$$\begin{aligned} wl^{(t)}(G_1)(u) = wl^{(t)}(G_2)(u) &\implies \begin{cases} wl^{(t-1)}(G_1)(u) = wl^{(t-1)}(G_2)(u) \\ \{\{wl^{(t-1)}(G_1)(v)|v \in \mathcal{N}_R(u)\}\} = \{\{wl^{(t-1)}(G_2)(v)|v \in \mathcal{N}_R(u)\}\} \\ \{\{wl^{(t-1)}(G_1)(v)|v \in \mathcal{N}_P(u)\}\} = \{\{wl^{(t-1)}(G_2)(v)|v \in \mathcal{N}_P(u)\}\} \end{cases} \\ &\implies \begin{cases} \kappa^{(t-1)}(G_1)(u) = \kappa^{(t-1)}(G_2)(u) \\ \{\{\kappa^{(t-1)}(G_1)(v)|v \in \mathcal{N}_R(u)\}\} = \{\{\kappa^{(t-1)}(G_2)(v)|v \in \mathcal{N}_R(u)\}\} \\ \{\{\kappa^{(t-1)}(G_1)(v)|v \in \mathcal{N}_P(u)\}\} = \{\{\kappa^{(t-1)}(G_2)(v)|v \in \mathcal{N}_P(u)\}\} \end{cases} \\ &\implies \kappa^{(t)}(G_1)(u) = \kappa^{(t)}(G_2)(u) \end{aligned}$$

This proves that in that case, the 1-WL iterative test is as powerful as κ. □

Since the model architecture $\mathcal{B}$ is strictly more expressive than the 1-WL test, $\mathcal{B}$ is strictly more expressive than $\mathcal{A}$ for graphs such that $V_{G_1} = V_{G_2}$, $X^R_{G_1} \simeq X^R_{G_2}$ and $X^P_{G_1} \simeq X^P_{G_2}$.

- **Suppose that $V_{G_1} = V_{G_2}$, $X_{G_1} \simeq X_{G_2}$ but $X^R_{G_1} \not\simeq X^R_{G_2}$.**

Here $\mathcal{A}$ can distinguish graphs that $\mathcal{B}$ cannot (see figure 3).

- **If we both have $X^R_{G_1} \not\simeq X^R_{G_2}$ and G_1 and G_2 have a different number of subgraphs, then both models are able to distinguish G_1 and G_2 (see figure 4).**

Question 3

In order to train expressive Graph Neural Networks (GNNs), model architectures need to be able to distinguish graphs even at initialisation time. In their recent work on *Zero-One Laws of Graph Neural Networks*[1], Adam-Day et al. formally and experimentally proved the negative result that, under mild assumptions, binary classification models, such as Graph Convolutional Networks (GCNs), obey a 0-1 Law on Erdös-Rényi graphs. The 0-1 Law asserts that, as the graphs size increases, the model output collapses to a fixed binary prediction, regardless of its input graph. In practice, the authors' approach relies on proving that the model converges to a fixed average graph embedding vector, which is then a sufficient condition for the 0-1 Law to hold true. We name this condition the convergence hypothesis $\mathcal{H}$, and, in this report, we ask the following question: *In concrete experimental settings, does the 0-1 Law only hold true when the convergence hypothesis $\mathcal{H}$ is satisfied, or does it generalize to additional settings?*
The objectives of this study include:

1. Highlighting the convergence hypothesis $\mathcal{H}$ for the GCN and Mean GNN models, since $\mathcal{H}$ was theoretically proven for the general theorem but not empirically shown in Adam-Day et al[1].

2. Examining the convergence hypothesis $\mathcal{H}$ on data that was not generated from the Erdös-Rényi random graphs model.

3. Examining $\mathcal{H}$ on a set of models beyond those explored in Adam-Day et al[1].

The code for the experiments is available online at github.com/grl-student/grl-mini-project. Our implementation closely aligns with the choices outlined in Adam-Day et al.[1]. Random graphs are generated once and for all for sizes ranging from 10 to 3000 nodes, with 30 graphs per graph size. Experiments encompass the following hyperparameters, ensuring that the necessary conditions for the 0-1 Law theorem are met:

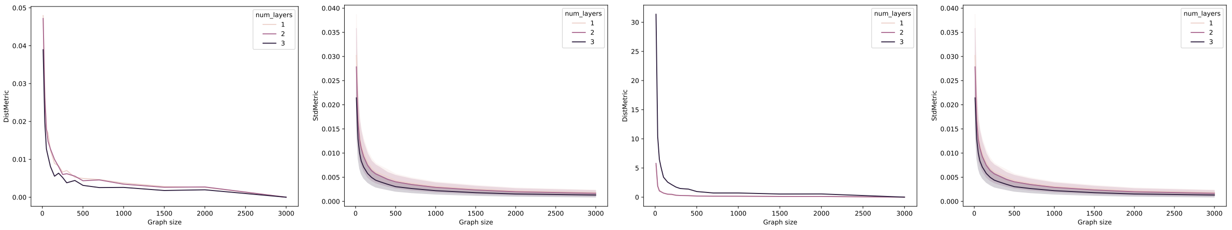
- The core architecture is followed by an average pooling and by a 2-layer MLP, with a ReLU activation function. The final activation function is the sigmoid function. Layers are initialized according to the Xavier-Glorot initialization. We also use the ReLU activation function for the graph architecture, as it is Lipschitz continuous.

- 10 models are evaluated on 30 generated graphs for each graph dimension. The uniform (sub-Gaussian) distribution is used to generate node features of dimension 128. The edge rate is set to 0.5. Models are randomly initialised and not trained.

To experimentally prove hypothesis $\mathcal{H}$, we use two metrics to measure the convergence of the average embedding vector. The first one, termed DistMetric, computes the L2-norm between the average graph embedding at a fixed graph dimension and the limit vector for an infinitely large graph. Since the limit vector is not known, we approximate it by its average value at the highest graph dimension, namely 3000. For the second metric, we measure whether the vector is converging by computing the standard deviation across all graphs of a given size, and checking whether this statistic converges to zero across all coordinates as the size of the graph grows to infinity. This is referred to as the StdMetric in the rest of this report.

Experiment 1: GCN and Mean GNN models.

[1] *Zero-One Laws of Graph Neural Networks*, Adam-Day, Iliant and Ceylan. NeurIPS 2023

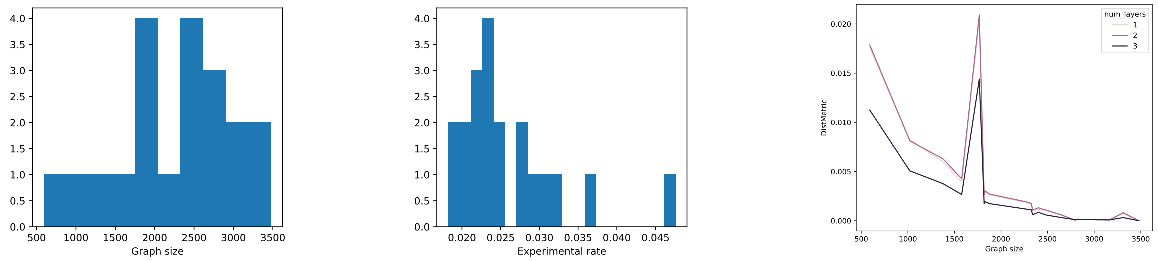


(a) (b) (c) (d)

Figure 5: DistMetric for GCN models and MeanGNN models in figures 5a and 5c resp., and StdMetric for GCN and MeanGNN models in figures 5b and 5d resp., as functions of graph size.

For both models, the embedding vector tends to a constant vector as graphs become larger. The result is independent of the number of layers in the model and remains true even for 1 layer (although the convergence rate depends on the number of layers). This result was implicitly covered by the proof of the convergence hypothesis $\mathcal{H}$ in the paper[1], but we confirm here that we can observe it empirically, even for a graph size smaller than 3000. This motivates the use of sizes up to 3000 for the rest of this report.

Experiment 2: GCN model on the PPI dataset. Random graphs are often not representative of real-world data, as they often exhibit specific structural properties that Erdös-Rényi random graphs may not capture (such as cycles or clusters). Having that in mind, we selected a real-world dataset whose graph sizes are of the order of that chosen for the previous experiments: the Protein-Protein Interaction (PPI) dataset[3]. Experiments are similar to the previous ones, using 10 GCN models with average pooling and from 1 to 3 layers.



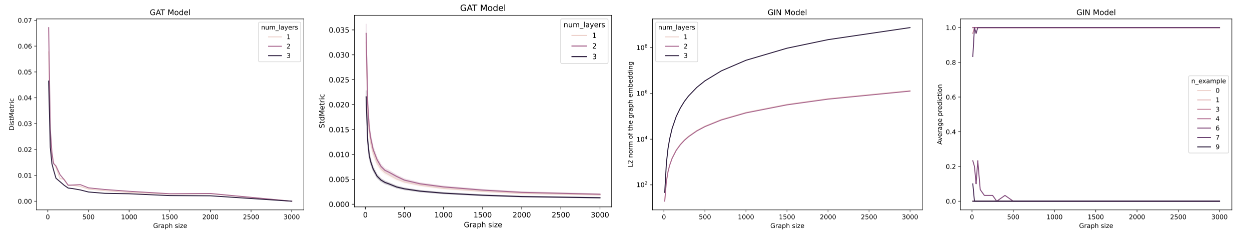
(a) (b) (c)

Figure 6: Histogram of graph sizes in the PPI dataset in figure 6a. Histogram of the experimental rate in the PPI dataset in figure 6b. The experimental rate is the number of edges divided by the maximum number of edges ($\frac{n(n-1)}{2}$). DistMetric as a function of the graph size in figure 6c.

Apart from the anomaly at around 1700 nodes, the graph embedding converges to a constant vector. All models exhibit a 0-1 Law, showing that this behaviour can be extended to other graph distributions. Figure 6b shows that the rate is not constant across all graphs in the dataset. This proves that the assumption of having graphs generated with Erdös-Rényi model is not necessary for the convergence hypothesis $\mathcal{H}$ to hold true. Therefore this collapsing behaviour may also occur in real-world datasets, although a more thorough investigation would be required for any conclusive statement on the empirical effect of the data distribution.

[3] https://paperswithcode.com/dataset/ppi

Experiment 3: GAT and GIN models. We now attempt to extend the 0-1 Law and to evaluate the convergence hypothesis $\mathcal{H}$ on the GAT and GIN models with $1, 2$ and 3 layers. Parameters are similar as previously, using 10 models each time.



(a) (b) (c) (d)

Figure 7: DistMetric and StdMetric for GAT models as functions of graph size, in figures 7a and 7b respectively. Norm of the graph embedding vector for GIN models as a function of the graph size, in figure 7c. Average prediction per graph size in figure 7d for models with 3 layers.

The GAT model follows the convergence hypothesis $\mathcal{H}$ as shown in figures $7a$ and $7b$. However, in the case of the GIN model, the norm of the graph embedding vector grows exponentially as the graph size increases, showing that the average embedding vector does not converge to finite values, at least for the range of graph sizes studied here. This observation potentially invalidates the convergence hypothesis $\mathcal{H}$ in this setting. Nevertheless, the GIN model still appears to follow the 0-1 Law as shown in figure 7d. This is a potential example of a setting in which the convergence hypothesis $\mathcal{H}$ is not necessary for the 0-1 Law to hold true. The proof scheme developed in Adam-Day et al[1] would not be applicable here, calling for the exploration of alternative theoretical proofs that could extend and generalize the theorem introduced in their paper.

Discussion While the work of Adam-Day et al[1] focuses on the 0-1 Law, this report investigates the convergence hypothesis $\mathcal{H}$, which is a sufficient condition for the 0-1 Law to hold true. In particular, our preliminary experiments show that (1) the convergence hypothesis can hold true outside of the data distribution generated by the Erdös-Rényi random graphs model, and (2) the hypothesis empirically holds true for the GCN, Mean GNN and GAT models but not for the GIN model (at least in the regime considered here).

More interestingly, we observe that the 0-1 Law still holds approximately true for the GIN model, despite the lack of convergence of the average embedding vector. As a result, we hypothesize that the 0-1 Law may hold more generally than shown in Adam-Day, e.g. due to saturating effects of the activations rather than the convergence of the average embedding vector.

Conclusion These results show that randomly-initialized models are already limited in their expressiveness, and unable to distinguish graphs with a high number of nodes even before the training starts. It seems that the 0-1 Law is quite common, for various models and datasets, and many conditions stated in the original theorem could be relaxed. Experiment 2 showed similar behaviour could be observed with real-world data, impacting current attempts to develop GNNs.

In order to strengthen our experimental findings, we would need to use a greater variety of datasets, as well as exploring larger graph sizes to truly approximate the asymptotical behaviour. Further, in order to obtain statistically significant results, we would need to run a larger number of repeated experiments.