Graph Representation Learning

Jonas De Schouwer

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) Define K_2 as a graph with only two nodes ($V_{K_2} = \{x, y\}$) and a single edge between them ($E_{K_2} = \{(x, y)\}$). Further, let the node features of K_2 be $\boldsymbol{x}_x = \boldsymbol{x}_y = \mathbf{1}^d$.

Let f be any function that associates the following mapping (1) to the graph K_2. We will prove that f has the required property. Note that the mapping of f associated with other graphs is not important for this question, any extension to other graphs suffices.

$$f(K_2)(u) = \begin{cases} 1 & \text{if } u = x \\ 0 & \text{if } u = y \end{cases} \tag{1}$$

The following result holds:

Theorem 1.1. *Let f be any function such that $f(K_2)$ is given by (1). Then there is no parametrization $\tilde{\mathcal{S}}$ of $\mathcal{S}$ satisfying $f(K_2)(u) = \tilde{S}(K_2)(u)$ for all $u \in V_{K_2}$.*

Proof. Consider any parametrization $\tilde{\mathcal{S}}$ of $\mathcal{S}$. Then we will prove by induction that, for all $0 \leq t \leq T$, the representations computed by $\tilde{\mathcal{S}}(K_2)$ satisfy the following equation:

$$\boldsymbol{h}_x^{(t)} = \boldsymbol{h}_y^{(t)} \tag{2}$$

For the base case $t = 0$, it is given in the problem statement that $\boldsymbol{h}_x^{(0)} = \mathbf{1}^d = \boldsymbol{h}_y^{(0)}$. Now assume that the induction hypothesis (2) holds for $t - 1$. Then

$$\begin{aligned} \boldsymbol{h}_x^{(t)} &= \mathrm{MLP}^{(t)}\left(W_s^{(t)}\boldsymbol{h}_x^{(t-1)} + W_n^{(t)}\boldsymbol{h}_y^{(t-1)} + \boldsymbol{b}^{(t)}\right) \\ &= \mathrm{MLP}^{(t)}\left(W_s^{(t)}\boldsymbol{h}_y^{(t-1)} + W_n^{(t)}\boldsymbol{h}_x^{(t-1)} + \boldsymbol{b}^{(t)}\right) \\ &= \boldsymbol{h}_y^{(t)} \end{aligned} \tag{3}$$

which finishes the induction.

Consequently, the final representations $\boldsymbol{z}_x, \boldsymbol{z}_y$ of x and y will be equal. Hence, it is impossible that $\tilde{\mathcal{S}}(K_2)(x) = f(K_2)(x)$ and $\tilde{\mathcal{S}}(K_2)(y) = f(K_2)(y)$, since $f(K_2)(x) \neq f(K_2)(y)$. □

Because there is no parametrization $\tilde{\mathcal{S}}$ that equals f on K_2, there is evidently no $\tilde{\mathcal{S}}$ that equals f on all graphs G.

Remark 1.1.1. *Theorem 1.1 can also be proven by noticing that the 1-WL algorithm on K_2 generates the same representations for x and y and that $\mathcal{S}$ is a subclass of MPNNs. However, bottom-up inductive proofs are preferred to references to 1-WL throughout this project. That allows us to obtain self-contained proofs and avoid the need to check off every assumption of existing results.*

(b) For this question, we will construct a parametrization $\tilde{\mathcal{S}}$ of S that equals a boolean function f on all bounded-degree graphs. Afterwards, we will prove that no parametrization of $\mathcal{M}$ can equal f on this domain by considering two graphs, C_4 and K_4, that $\mathcal{M}$ cannot distinguish while $\tilde{\mathcal{S}}$ can. This disproves the statement in the project assignment.

Lemma 1.2. *For any $d \in \mathbb{Z}^+$, there exists a 3-layer MLP $\mathscr{L}_d : \mathbb{R}^d \to \mathbb{R}^d$ that applies the clipped ReLU (CReLU) function (4) element-wise.*

$$CReLU(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leq x < 1 \\ 1 & \text{if } 1 \leq x \end{cases} \tag{4}$$

Proof. The only property of the MLP specified in the assignment is the ReLU activation. Therefore, we assume a standard MLP consisting of a sequence of linear layers $(\mathcal{L}_1, \ldots, \mathcal{L}_n)$ with ReLU nonlinearities in between. Each layer $\mathcal{L}_k$ is of the form

$$\mathcal{L}_k : \mathbb{R}^{d_{k-1}} \to \mathbb{R}^{d_k} : \boldsymbol{x} \mapsto W_k \boldsymbol{x} + \boldsymbol{b}_k \tag{5}$$

Construct $\mathscr{L}$ by taking three layers with the following weights and biases:

$$W_1 = W_2 = -I_d \tag{6}$$
$$W_3 = I_d \tag{7}$$
$$\boldsymbol{b}_1 = \boldsymbol{b}_2 = \mathbf{1}^d \tag{8}$$
$$\boldsymbol{b}_3 = \mathbf{0}^d \tag{9}$$

Then $\mathscr{L}_d(\boldsymbol{x})$ applies the following operation on every element of $\boldsymbol{x}$:

$$\mathscr{L}_d(\boldsymbol{x})_j = 1 \cdot \mathrm{ReLU}(1 - \mathrm{ReLU}(1 - x_j)) = \mathrm{CReLU}(x_j) \tag{10}$$

□

Consider the 2-layer parametrization $\tilde{\mathcal{S}}$ of $\mathcal{S}$ with the following parameters (where we drop the vector notation for all 1×1-matrices and all vectors of dimension 1). We will prove that $\tilde{\mathcal{S}}$ implements a boolean function f on all bounded-degree graphs.

$$W_s^{(1)} = W_n^{(1)} = \mathbf{0}_{1 \times d} \tag{11}$$
$$b^{(1)} = 1 \tag{12}$$
$$\mathrm{MLP}^{(1)} = \mathrm{ReLU} \tag{13}$$
$$W_s^{(2)} = 0 \tag{14}$$
$$W_n^{(2)} = 1 \tag{15}$$
$$b^{(2)} = 2 \tag{16}$$
$$\mathrm{MLP}^{(2)} = \mathscr{L}_1 \tag{17}$$

Lemma 1.3. *For any bounded-degree graph G with initial features $\boldsymbol{x}_u \in [0,1]^d$ for all $u \in V_G$,*

$$\tilde{\mathcal{S}}(G) = \begin{cases} 1 & \text{if } \exists u \in V_G : |\mathcal{N}(u)| \geq 3 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Proof. Consider any bounded-degree graph G with initial features $\boldsymbol{x}_u \in [0,1]^d$ for all $u \in V_G$. Then, for any $u \in V_G$:

$$h_u^{(1)} = \mathrm{ReLU}\left(0 + 0 + 1\right) = 1 \tag{19}$$

$$z_u = h_u^{(2)} = \mathscr{L}_1 \left(0 + \sum_{v \in \mathcal{N}(u)} h_v^{(1)} - 2 \right) = \mathrm{CReLU}\left(|\mathcal{N}(u)| - 2 \right) \tag{20}$$

where the fact that G has bounded degree ensures that the sum converges.

If there exists a node u with degree at least 3, then $h_u^{(2)} = 1$. Since all other node representations are also in $[0,1]$, $\tilde{\mathcal{S}}(G) = \max_{u \in V_G} z_u = 1$. Otherwise, all nodes have degree at most 2, in which case $z_u = 0$ for all $u \in V_G$ and hence $\tilde{\mathcal{S}}(G) = \max_{u \in V_G} z_u = 0$. □

Corollary 1.3.1. *$\tilde{\mathcal{S}}(G) \in \mathbb{B}$ for every bounded-degree graph G with initial features $\boldsymbol{x}_u \in [0,1]^d$ for all $u \in V_G$.*

Now we will construct two graphs that $\tilde{\mathcal{S}}$ can distinguish while no parametrization of $\mathcal{M}$ can. Denote by C_4 the cycle graph with 4 nodes and input features $\boldsymbol{x}_u = \mathbf{1}^d$ for all $u \in V_{C_4}$. Similarly, denote by K_4 the complete graph with 4 nodes and input features $\boldsymbol{x}_u = \mathbf{1}^d$ for all $u \in V_{K_4}$.

Lemma 1.4. *Consider the graphs C_4 and K_4. Then any parametrization $\tilde{\mathcal{M}}$ of $\mathcal{M}$ satisfies.*

$$\tilde{\mathcal{M}}(C_4) = \tilde{\mathcal{M}}(K_4) \tag{21}$$

Proof. Consider any parametrization $\tilde{\mathcal{M}}$ of $\mathcal{M}$ with parameters $\mathrm{MLP}^{(t)}, W_s^{(t)}, W_n^{(t)}, \boldsymbol{b}^{(t)}$ for every $0 < t \leq T$. We will prove by induction that, for all $0 \leq t \leq T$, there exists a $\boldsymbol{h}^{(t)} \in \mathbb{R}^{d^{(t)}}$ that all node representations in both C_4 and K_4 are equal to.

The base case $t = 0$ is satisfied, because the input features for all nodes in both graphs are $\mathbf{1}^d$. Now assume that the induction hypothesis holds for $t-1$. Then, for every $u \in V_{C_4} \cup V_{K_4}$:

$$\begin{aligned} \boldsymbol{h}_u^{(t)} &= \mathrm{MLP}^{(t)} \left(W_s^{(t)} \boldsymbol{h}_u^{(t-1)} + W_n^{(t)} \sum_{v \in \mathcal{N}(u)} \frac{\boldsymbol{h}_v^{(t-1)}}{|\mathcal{N}(u)|} + \boldsymbol{b}^{(t)} \right) \\ &= \mathrm{MLP}^{(t)} \left(W_s^{(t)} \boldsymbol{h}^{(t-1)} + W_n^{(t)} \, \frac{|\mathcal{N}(u)| \, \boldsymbol{h}^{(t-1)}}{|\mathcal{N}(u)|} + \boldsymbol{b}^{(t)} \right) \\ &= \mathrm{MLP}^{(t)} \left(W_s^{(t)} \boldsymbol{h}^{(t-1)} + W_n^{(t)} \boldsymbol{h}^{(t-1)} + \boldsymbol{b}^{(t)} \right) \end{aligned} \tag{22}$$

which does not depend on u. So all $\boldsymbol{h}_u^{(t)}$ are equal to each other, and hence to some constant vector $\boldsymbol{h}^{(t)} \in \mathbb{R}^{d^{(t)}}$, which finishes the induction step.

Because all final representations are equal, it is clear that

$$\tilde{\mathcal{M}}(C_4) = \boldsymbol{h}^{(T)} = \tilde{\mathcal{M}}(K_4) \tag{23}$$

□

Theorem 1.5. *There exists a boolean function f on bounded-degree graphs with initial features in $[0,1]^d$ that can be captured by $\mathcal{S}$ but not by $\mathcal{M}$.*

Proof. Define the function $f \coloneqq \tilde{\mathcal{S}}$, where $\tilde{\mathcal{S}}$ is the parametrization of $\mathcal{S}$ with the parameters given in (11-17). Corollary 1.3.1 ensures that $f(G) \in \mathbb{B}$ for all bounded-degree graphs. Clearly f can be captured by $\mathcal{S}$.

On the other hand, $f(C_4) = \tilde{\mathcal{S}}(C_4) = 0$ and $f(K_4) = \tilde{\mathcal{S}}(K_4) = 1$ by Lemma 1.3. So it is impossible that f is captured by $\mathcal{M}$, as Lemma 1.4 shows that $\tilde{\mathcal{M}}(C_4) = \tilde{\mathcal{M}}(K_4)$ for any parametrization $\tilde{\mathcal{M}}$ of $\mathcal{M}$. □

(c) In this question, we will prove the statement given in the project assignment (Theorem 1.7). The key insight is that the set of possible pairs $(\boldsymbol{x}_u, \sum_{v \in \mathcal{N}(u)} \boldsymbol{x}_u)$ over all nodes u of all bounded-degree graphs is finite, so we can carefully construct the MLP of $\tilde{\mathcal{S}}$ to output the exact same representations as $\tilde{\mathcal{M}}$. We will start with a preliminary lemma.

Lemma 1.6. *Given a finite set $W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m\} \subset \mathbb{R}^k$ where all $\boldsymbol{w}_i$ are distinct, there is a 5-layer MLP $\mathscr{W} : \mathbb{R}^k \to \mathbb{R}^{|W|}$ that transforms every $\boldsymbol{w}_i$ into the one-hot encoded vector $\vec{\boldsymbol{e_i}} \in \mathbb{R}^{|W|}$.*

Proof. Choose any $\epsilon > 0$ and any function $f : \mathbb{R}^k \to \mathbb{R}^{|W|}$ that is continuous on the compact set $\mathbb{K} = \{\boldsymbol{x} \in \mathbb{R}^k \mid \|\boldsymbol{x}\| \leq \max_{\boldsymbol{w} \in W} \|\boldsymbol{w}\| + 1\}$ and satisfies, for every $1 \leq i \leq |W|$:

$$\begin{aligned} f(\boldsymbol{w}_i) &= (1+2\epsilon)\vec{\boldsymbol{e_i}} - \epsilon \mathbf{1}^{|W|} \\ &= \begin{pmatrix} -\epsilon & -\epsilon & \ldots & -\epsilon & 1+\epsilon & -\epsilon & \ldots & -\epsilon \end{pmatrix}^T \end{aligned} \tag{24}$$

Then by the universal approximation theorem [Leshno et al., 1993], there exists a 2-layer MLP $\mathscr{K}$ with ReLU activations such that

$$\sup_{\boldsymbol{x} \in \mathbb{K}} \|f(\boldsymbol{x}) - \mathscr{K}(\boldsymbol{x})\| < \epsilon \tag{25}$$

From (25) follows the following property of the jth component of $\mathscr{K}(\boldsymbol{w}_i)$. For any $1 \leq i, j \leq |W|$:

$$\begin{cases} \mathscr{K}(\boldsymbol{w}_i)_j < 0 & \text{if } i \neq j \\ \mathscr{K}(\boldsymbol{w}_i)_j > 1 & \text{if } i = j \end{cases} \tag{26}$$

Recall the 3-layer MLP $\mathscr{L}_{|W|}$ we constructed in Lemma 1.2 that applies an element-wise clipped ReLU function. Let $\mathscr{W} = \mathscr{L}_{|W|} \circ \text{ReLU} \circ \mathscr{K}$. Then $\mathscr{W}$ is a 5-layer MLP such that, for all $1 \leq i, j \leq |W|$:

$$\begin{cases} \mathscr{W}(\boldsymbol{w}_i)_j = 0 & \text{if } i \neq j \\ \mathscr{W}(\boldsymbol{w}_i)_j = 1 & \text{if } i = j \end{cases} \tag{27}$$

Hence, for all $1 \leq i \leq |W|$:

$$\mathscr{W}(\boldsymbol{w}_i) = \vec{\boldsymbol{e_i}} \tag{28}$$

□

Theorem 1.7. *Let D be any integer. For any single-layer parametrization of $\mathcal{M}$, there exists a single-layer parametrization of $\mathcal{S}$ such that the node representations computed by $\tilde{\mathcal{S}}$ and $\tilde{\mathcal{M}}$ are identical for all nodes of all graphs with maximum degree at most D and initial features in $\mathbb{I}^d$.*

Proof. Consider any single-layer parametrization $\tilde{\mathcal{M}}$ with parameters $\tilde{\text{MLP}}, \tilde{W}_s, \tilde{W}_n, \tilde{\boldsymbol{b}}$, where we omitted the superscript $^{(1)}$ for the sake of readability. Then we will prove that the parametrization $\tilde{\mathcal{S}}$ with the following parameters, supplemented with a suitable MLP to be defined later, satisfies the theorem:

$$\hat{W}_s = \begin{pmatrix} 1 & (D+1) & \ldots & (D+1)^{d-1} \\ & \mathbf{0}_{(d^{(1)}-1)\times d} & & \end{pmatrix} \tag{29}$$

$$\hat{W}_n = \begin{pmatrix} (D+1)^d & (D+1)^{d+1} & \ldots & (D+1)^{2d-1} \\ & \mathbf{0}_{(d^{(1)}-1)\times d} & & \end{pmatrix} \tag{30}$$

$$\hat{\boldsymbol{b}} = \mathbf{0}^{d^{(1)}} \tag{31}$$

Let $Z = \{\boldsymbol{h} \in \{0, 1, \ldots, D\}^d \mid \boldsymbol{h}^T \mathbf{1}^d \leq D\}$. Then Z is finite and contains all sums of the form $\sum_{v \in \mathcal{N}(u)} \boldsymbol{x}_v$ where $|\mathcal{N}(u)| \leq D$ and $\boldsymbol{x}_v \in \mathbb{I}^d$ for all $v \in \mathcal{N}(u)$. Now $Y = \mathbb{I}^d \times Z$ is also finite, so denote its cardinality by $n \in \mathbb{Z}^+$ and number its elements $Y = \{y_1, \ldots, y_n\}$.

We will now define desired input and output vectors to the MLP of $\tilde{\mathcal{S}}$. For every $1 \leq i \leq n$, let $y_i = (\boldsymbol{x}_i, \boldsymbol{s}_i)$ and define the following two vectors, both in $\mathbb{R}^{d^{(1)}}$:

$$\boldsymbol{p}_i = \hat{W}_s \boldsymbol{x}_i + \hat{W}_n \boldsymbol{s}_i \tag{32}$$

$$\boldsymbol{q}_i = \tilde{\text{MLP}} \left(\tilde{W}_s \boldsymbol{x}_i + \frac{\tilde{W}_n \boldsymbol{s}_i}{\boldsymbol{s}_i^T \mathbf{1}^d} + \tilde{\boldsymbol{b}} \right) \tag{33}$$

Note that the definitions of $\hat{W}_s, \hat{W}_n$ and the domains of $\boldsymbol{x}_i, \boldsymbol{s}_i$ ensure that all $\boldsymbol{p}_i$ are distinct. Let $P = \{p_1, \ldots, p_n\}$. Then by Lemma 1.6, there exists a 5-layer MLP $\mathscr{W} : \mathbb{R}^{d^{(1)}} \to \mathbb{R}^n$ that transforms every $\boldsymbol{p}_i$ into the one-hot encoded vector $\vec{\boldsymbol{e_i}} \in \mathbb{R}^n$. Now consider the one-layer MLP $\mathcal{L} : \mathbb{R}^n \to \mathbb{R}^{d^{(1)}}$ with parameters

$$\boldsymbol{b}_{\mathcal{L}} = \mathbf{0}^{d^{(1)}} \tag{34}$$

$$W_{\mathcal{L}} = \begin{pmatrix} \boldsymbol{q}_1 & \boldsymbol{q}_2 & \ldots & \boldsymbol{q}_n \end{pmatrix} \tag{35}$$

and let

$$\hat{\text{MLP}} = \mathcal{L} \circ \text{ReLU} \circ \mathscr{W} \tag{36}$$

Then for every $1 \leq i \leq n$:

$$\hat{\text{MLP}}(\boldsymbol{p}_i) = \mathcal{L} \circ \text{ReLU} \circ \mathscr{W}(\boldsymbol{p}_i) = \mathcal{L} \circ \text{ReLU}(\vec{\boldsymbol{e_i}}) = \mathcal{L}(\vec{\boldsymbol{e_i}}) = \boldsymbol{q}_i \tag{37}$$

Now we can prove that the node representations computed by $\tilde{\mathcal{M}}$ and $\tilde{\mathcal{S}}$ are the same. Consider any graph G with maximum degree at most D and initial features $\boldsymbol{x}_u \in \mathbb{I}^d$ for all $u \in V_G$. Furthermore, pick any node u of this graph. Letting $\boldsymbol{s}_u = \sum_{v \in \mathcal{N}(u)} \boldsymbol{x}_v$, then $\boldsymbol{s}_u \in Z$ so $(\boldsymbol{x}_u, \boldsymbol{s}_u) \in Y$. Consequently, $(\boldsymbol{x}_u, \boldsymbol{s}_u) = (\boldsymbol{x}_i, \boldsymbol{s}_i)$ for some $1 \leq i \leq n$, so

$$\begin{aligned}
\tilde{\mathcal{S}}(G)(u) &= \hat{\text{MLP}} \left(\hat{W}_s \boldsymbol{x}_u + \hat{W}_n \boldsymbol{s}_u + \hat{\boldsymbol{b}} \right) \\
&= \hat{\text{MLP}} \left(\hat{W}_s \boldsymbol{x}_i + \hat{W}_n \boldsymbol{s}_i \right) \\
&= \hat{\text{MLP}} \left(\boldsymbol{p}_i \right) \\
&= \boldsymbol{q}_i \\
&= \tilde{\text{MLP}} \left(\tilde{W}_s \boldsymbol{x}_i + \frac{\tilde{W}_n \boldsymbol{s}_i}{\boldsymbol{s}_i^T \mathbf{1}^d} + \tilde{\boldsymbol{b}} \right) \\
&= \tilde{\text{MLP}} \left(\tilde{W}_s \boldsymbol{x}_u + \frac{\tilde{W}_n \sum_{v \in \mathcal{N}(u)} \boldsymbol{x}_v}{|\mathcal{N}(u)|} + \tilde{\boldsymbol{b}} \right) \\
&= \tilde{\mathcal{M}}(G)(u)
\end{aligned} \tag{38}$$

Here we used the fact $\boldsymbol{s}_i^T \mathbf{1}^d = |\mathcal{N}(u)|$. This follows from $\boldsymbol{s}_i = \sum_{v \in \mathcal{N}(u)} \boldsymbol{x}_v$ and $\boldsymbol{x}_v \in \mathbb{I}^d$ for all $v \in \mathcal{N}(u)$.

□

Question 2

(a) Define the iterative test κ as follows:

1. Start with the initial coloring for every node $u \in V_G$.
$$\forall u \in V_G : \kappa^{(t)}(G)(u) = c(G)(u) \tag{39}$$
2. Iteratively assign a new label to each node $u \in V_G$.
$$\kappa^{(t)}(G)(u) = \text{HASH}\left(\kappa^{(t-1)}(G)(u), \left\{\!\left\{\kappa^{(t-1)}(G)(v) \mid v \in \mathcal{N}_R(u)\right\}\!\right\}, \left\{\!\left\{\kappa^{(t-1)}(G)(v) \mid v \in \mathcal{N}_P(u)\right\}\!\right\}\right) \tag{40}$$
Where HASH is any injective function from $\mathcal{C} \times \mathbb{N}^{\mathcal{C}} \times \mathbb{N}^{\mathcal{C}}$ to $\mathcal{C}$. Here, $\mathcal{C}$ is the set of possible colours and $\mathbb{N}^{\mathcal{C}}$ is the set of multisets over these colours.
3. For the purpose of this question, one can repeat Step 2 for all $0 < t \leq T$. For the purpose of isomorphism testing on graphs, one may instead repeat Step 2 until the equality classes of the node labels converge.

(b) We will prove the contrapositive (Theorem 2.1) by induction on t, from which the statement to be proven follows immediately.

Theorem 2.1. *For all graphs G with initial node features $\{\boldsymbol{x}_u = c(G)(u) \mid u \in V_G\}$, for all T layer models of $\mathcal{A}$, and for all $0 \leq t \leq T$:*
$$\forall u, v \in V_G : \kappa^{(t)}(G)(u) = \kappa^{(t)}(G)(v) \Rightarrow \boldsymbol{h}_u^{(t)} = \boldsymbol{h}_v^{(t)} \tag{41}$$

Proof. Fix any graph G with initial node features $\{\boldsymbol{x}_u = c(G)(u) \mid u \in V_G\}$ and any T layer model of $\mathcal{A}$. Then we prove (41) by induction on t.

For the base case $t = 0$, $\kappa^{(0)}(G)(u) = c(G)(u) = \boldsymbol{h}_u^{(0)}$ for any $u \in V_G$. So for any $u, v \in V_G$, the following implications hold:
$$\kappa^{(0)}(G)(u) = \kappa^{(0)}(G)(v) \Rightarrow c(G)(u) = c(G)(v) \Rightarrow \boldsymbol{h}_u^{(0)} = \boldsymbol{h}_v^{(0)} \tag{42}$$

Now assume the induction hypothesis holds for $t-1$.

Then consider any $u, v \in V_G$ for which $\kappa^{(t)}(G)(u) = \kappa^{(t)}(G)(v)$. The following implications hold:
$$\begin{aligned}
&\kappa^{(t)}(G)(u) = \kappa^{(t)}(G)(v) \\
&\Rightarrow \begin{cases} \kappa^{(t-1)}(G)(u) = \kappa^{(t-1)}(G)(v) \\ \left\{\!\left\{\kappa^{(t-1)}(G)(x) \mid x \in \mathcal{N}_R(u)\right\}\!\right\} = \left\{\!\left\{\kappa^{(t-1)}(G)(x) \mid x \in \mathcal{N}_R(v)\right\}\!\right\} \\ \left\{\!\left\{\kappa^{(t-1)}(G)(x) \mid x \in \mathcal{N}_P(u)\right\}\!\right\} = \left\{\!\left\{\kappa^{(t-1)}(G)(x) \mid x \in \mathcal{N}_P(v)\right\}\!\right\} \end{cases} \\
&\Rightarrow \begin{cases} \boldsymbol{h}_u^{(t-1)} = \boldsymbol{h}_v^{(t-1)} \\ \left\{\!\left\{\boldsymbol{h}_x^{(t-1)} \mid x \in \mathcal{N}_R(u)\right\}\!\right\} = \left\{\!\left\{\boldsymbol{h}_x^{(t-1)} \mid x \in \mathcal{N}_R(v)\right\}\!\right\} \\ \left\{\!\left\{\boldsymbol{h}_x^{(t-1)} \mid x \in \mathcal{N}_P(u)\right\}\!\right\} = \left\{\!\left\{\boldsymbol{h}_x^{(t-1)} \mid x \in \mathcal{N}_P(v)\right\}\!\right\} \end{cases} \\
&\Rightarrow \boldsymbol{h}_u^{(t)} = \boldsymbol{h}_v^{(t)}
\end{aligned} \tag{43}$$

The first implication holds because HASH is injective, the second due to the induction hypothesis and the third because the only node-dependent components in the recursive definition of $\boldsymbol{h}_i^{(t)}$ are $\boldsymbol{h}_i^{(t-1)}$, $\sum_{x\in\mathcal{N}_R(i)} \boldsymbol{h}_x^{(t-1)}$ and $\sum_{x\in\mathcal{N}_R(i)} \boldsymbol{h}_x^{(t-1)}$. These three expressions are (by the previous statement) all independent of our choice of $i \in \{u, v\}$. □

(c) Over the course of this question, we will prove Theorem 2.8, which is equivalent to the statement to be proven (Corollary 2.8.1). For this, we will supply a self-contained proof very similar to the one given in [Morris et al., 2021] and adhere to the notation and terminology used in that paper, the most important of which is summarized in Notations 2.2 and Definitions 2.3 and 2.4.

Notations 2.2.

- *Denote the number of nodes of the graph G by N_G. We will omit the subscript when it is clear which graph we are talking about.*
- *Denote the i'th row of a matrix M by M_i. Moreover, if the rows of $M \in \mathbb{R}^{N\times d}$ correspond to nodes of a graph, we may write M_u to refer to the row corresponding to node u.*
- *let $J_{m\times n}$ be the all-1 matrix of dimension $m \times n$.*
- *For any graph G, let A_R be the $N \times N$-matrix derived from the adjacency matrix A where all columns corresponding to a node not in R are set to zero. Define A_P analogously.*
- *Rewrite the update rule of the $\mathcal{A}$ architecture in matrix form:*

$$\begin{aligned} H^{(t)} &= \Lambda_{W_s^{(t)},W_R^{(t)},W_P^{(t)},B^{(t)}}(H^{(t-1)}) \\ &= ReLU(H^{(t-1)}W_s^{(t)} + A_R H^{(t-1)} W_R^{(t)} + A_P H^{(t-1)} W_P^{(t)} + B^{(t)}) \end{aligned} \tag{44}$$

  Where $H^{(t)}, B^{(t)} \in \mathbb{R}^{N\times d^{(t)}}$ and $W_s^{(t)}, W_R^{(t)}, W_P^{(t)} \in \mathbb{R}^{d^{(t-1)}\times d^{(t)}}$. Further, B has the restriction that it must have identical rows, as the bias term is the same for all nodes. In this form, the feature vector $\boldsymbol{h}_u^{(t)}$ is replaced by the row $H_u^{(t)}$. Note that we restricted ourselves to MLPs consisting of only a single ReLU-activation (with two identity layers around it), but this is all we will need to construct an $\mathcal{A}$ model that is as powerful as the κ test.
- *Given a matrix $F \in \mathbb{R}^{N\times d}$, let $\Gamma_R(F)$ be the graph colouring that assigns the same colour to two nodes u, v iff $\{\!\{F_x \mid x \in \mathcal{N}_R(u)\}\!\} = \{\!\{F_x \mid x \in \mathcal{N}_R(v)\}\!\}$. Define $\Gamma_P(F)$ analogously.*

Definition 2.3. *Two matrices $A \in \mathbb{R}^{k\times l}$ and $B \in \mathbb{R}^{k\times m}$ are equivalent if their row equivalency classes are the same. I.e. if for every $1 \le i, j \le k$, $A_i = A_j \Leftrightarrow B_i = B_j$. In this case, we write $A \equiv B$.*

In the same way, we say that a matrix $A \in \mathbb{R}^{N\times d}$ is equivalent to a colouring $\kappa(G)$ if, for all $u, v \in V_G$, $A_u = A_v \Leftrightarrow \kappa(G)(u) = \kappa(G)(v)$.

Definition 2.4. *A matrix is row-independent modulo equality (RIME) if the set of all rows appearing in the matrix is linearly independent.*

Lemma 2.5. *Let $M \in \mathbb{Z}^{s\times t}$ be a matrix where all entries are in $\{0, 1, \ldots, B-1\}$ for some bound $B \in \mathbb{Z}^+$ and the rows of M are pairwise distinct. Then there is a matrix $X \in \mathbb{R}^{t\times s}$ such that $ReLU(MX - J_{s,s})$ is non-singular.*

Proof. Let $\boldsymbol{z} = (1, B, B^2, \ldots, B^{t-1})^T \in \mathbb{R}^t$ and $\boldsymbol{m} = M\boldsymbol{z} \in \mathbb{R}^s$. Then the entries of $\boldsymbol{m}$ are nonnegative and pairwise distinct. Without loss of generality, we assume that $\boldsymbol{m} = (m_1, \ldots, m_s)^T$ such that $m_1 > m_2 > \cdots > m_s \geq 0$. Now we choose numbers $x_1, \ldots, x_s \in \mathbb{R}$ such that

$$\begin{cases} m_i \cdot x_j < 1 & \text{if } i > j \\ m_i \cdot x_j > 1 & \text{if } i \leq j \end{cases} \tag{45}$$

Let $\boldsymbol{x} = (x_1, \ldots, x_s) \in \mathbb{R}^{1\times s}$, $C = \boldsymbol{m} \cdot \boldsymbol{x} \in \mathbb{R}^{s\times s}$ and $\hat{C} = \mathrm{ReLU}(C - J_{s\times s})$. Then C has entries $C_{ij} = m_i \cdot x_j$, so

$$\begin{cases} \hat{C}_{ij} = 0 & \text{if } i > j \\ \hat{C}_{ij} > 0 & \text{if } i \leq j \end{cases} \tag{46}$$

Thus $\hat{C}$ is an upper triangular matrix with nonzero elements on the main diagonal, hence it is non-singular. Now letting $X = \boldsymbol{z} \cdot \boldsymbol{x}$ yields $MX = C$, so $\mathrm{ReLU}(MX - J_{s,s}) = \hat{C}$. □

Lemma 2.6. *Given a matrix $F \in \mathbb{R}^{N\times d}$ that is RIME, there exists a matrix $W_R \in \mathbb{R}^{d\times N}$ such that $ReLU(A_RFW_R - J_{N\times N})$ is both RIME and equivalent to $\Gamma_R(F)$.*

Proof. Let $Q_1, \ldots, Q_r$ be the row equivalency classes of F. I.e., for all $u, v \in V_G$, it holds that $F_u = F_v \Leftrightarrow \exists j \in \{1, \ldots, r\} : u, v \in Q_j$. Let $\tilde{F} \in \mathbb{R}^{r\times d}$ with rows $\tilde{F}_j = F_v$ for all $j \in \{1, \ldots, r\}, v \in Q_j$. Then $\tilde{F}$ has linearly independent rows, so there is a matrix $M \in \mathbb{R}^{d\times r}$ such that $\tilde{F}M = I_r$. Moreover, by the definition of $\tilde{F}$, $F = L\tilde{F}$ where L is the matrix with entries

$$L_{vj} = \begin{cases} 1 & \text{if } v \in Q_j \\ 0 & \text{otherwise} \end{cases} \tag{47}$$

Let $D \in \mathbb{Z}^{N\times r}$ be the matrix with entries $D_{vj} = |N_R(v) \cap Q_j|$. Note that

$$A_RFM = A_RL = D \tag{48}$$

because for all $v \in V$ and $j \in [t]$ we have

$$(A_RL)_{vj} = \sum_{v' \in V_G} A_{Rvv'}L_{v'j} = \sum_{v' \in Q_j} A_{Rvv'} = D_{vj} \tag{49}$$

Moreover, D_v contains the number of occurrences of each equivalence class in $R(v)$, so $D_u = D_v$ iff $\{\!\{F_x \mid x \in \mathcal{N}_R(u)\}\!\} = \{\!\{F_x \mid x \in \mathcal{N}_R(v)\}\!\}$. Hence,

$$D \equiv \Gamma_R(F) \tag{50}$$

Let $P_1, \ldots, P_s$ be the row equivalency classes of D and let $\tilde{D} \in \mathbb{Z}^{s\times r}$ be the matrix with rows $\tilde{D}_j = D_v$ for all $j \in \{1, \ldots, s\}, v \in P_j$. Then $0 \leq \tilde{D}_{ij} \leq N - 1$ for all i, j and the rows of $\tilde{D}$ are pairwise distinct. By Lemma 2.5, there is a matrix $X \in \mathbb{R}^{r\times s}$ such that $\mathrm{ReLU}(\tilde{D}X - J_{s\times s})$ is non-singular, so all of its rows are linearly independent. This implies that $\mathrm{ReLU}(A_RFMX - J_{N\times s}) = \mathrm{ReLU}(DX - J_{N\times s})$ is both RIME and equivalent to D, which is by (50) equivalent to $\Gamma_R(F)$.

By letting $W_R \in \mathbb{R}^{d\times N}$ be the matrix obtained from $MX \in \mathbb{R}^{d\times s}$ by adding $N - s \geq 0$ all-zero columns, it follows that $\mathrm{ReLU}(A_RFW_R - J_{N\times N})$ is also RIME and equivalent to $\Gamma_R(F)$. □

Lemma 2.7. *Given a matrix $F \in \mathbb{R}^{N\times d}$ that is RIME, there exists a matrix $W_P \in \mathbb{R}^{d\times N}$ such that $ReLU(A_P F W_P - J_{N\times N})$ is both RIME and equivalent to $\Gamma_P(F)$.*

Proof. Analogous to the proof of Lemma 2.6. □

Theorem 2.8. *Let G be a graph with initial node features $\{x_u = c(G)(u) \mid u \in V_G\}$ and T be any positive integer. Then there exists a model of $\mathcal{A}$ such that for all $0 \le t \le T$:*

$$H^{(t)} \equiv \kappa^{(t)}(G) \tag{51}$$

Proof of Theorem 2.8. We will prove by induction on t that there exists a model of $\mathcal{A}$ that satisfies:

$$\begin{cases} H^{(t)} \text{ is RIME} \\ H^{(t)} \equiv \kappa^{(t)}(G) \end{cases} \tag{52}$$

For the base case $t = 0$, clearly $H^{(0)}$ is RIME, as all initial node features are in $\mathbb{I}^d$. Furthermore, $H^{(0)}_u = c(G)(u) = \kappa^{(0)}(G)(u)$ for any $u \in V_G$. So for any $u, v \in V_G$, the following implications hold:

$$H^{(0)}_u = H^{(0)}_v \Rightarrow c(G)(u) = c(G)(v) \Rightarrow \kappa^{(0)}(G)(u) = \kappa^{(0)}(G)(v) \tag{53}$$

Now assume that there exists a model of $\mathcal{A}$ such that the induction hypothesis (52) holds for $t-1$. For this model, denote the dimensions of $H^{(t-1)}$ by $N \times d$.

Then we will construct matrices $W^{(t)}_s, W^{(t)}_R, W^{(t)}_P \in \mathbb{R}^{d\times(d+2N)}$ and $B^{(t)} \in \mathbb{R}^{N\times(d+2N)}$ such that $H^{(t)}$ is equivalent to $\kappa^{(t)}(G)$, thereby extending the model of $\mathcal{A}$ to a model with t layers for which the induction hypothesis holds.

First, let

$$W^{(t)}_s = \begin{pmatrix} I_d & \mathbf{0}_{d\times N} & \mathbf{0}_{d\times N} \end{pmatrix} \tag{54}$$

$$B^{(t)} = \begin{pmatrix} \mathbf{0}_{N\times d} & -J_{N\times N} & -J_{N\times N} \end{pmatrix} \tag{55}$$

Secondly, let $W^*_R \in \mathbb{R}^{d\times N}$ be a matrix such that $\mathrm{ReLU}(A_R H^{(t-1)} W^*_R - J_{N\times N})$ is RIME and equivalent to $\Gamma_R(H^{(t-1)})$. The existence of such a matrix is guaranteed by Lemma 2.6, where we note that $H^{(t-1)}$ is RIME by the induction hypothesis. Let

$$W^{(t)}_R = \begin{pmatrix} \mathbf{0}_{d\times d} & W^*_R & \mathbf{0}_{d\times N} \end{pmatrix} \tag{56}$$

Analogously, let $W^*_P \in \mathbb{R}^{d\times N}$ be a matrix such that $\mathrm{ReLU}(A_P H^{(t-1)} W^*_P - J_{N\times N})$ is RIME and equivalent to $\Gamma_P(H^{(t-1)})$, and let

$$W^{(t)}_P = \begin{pmatrix} \mathbf{0}_{d\times d} & \mathbf{0}_{d\times N} & W^*_P \end{pmatrix} \tag{57}$$

Note the following:

$$\begin{aligned} H^{(t)} &= \mathrm{ReLU}(H^{(t-1)} W^{(t)}_s + A_R H^{(t-1)} W^{(t)}_R + A_P H^{(t-1)} W^{(t)}_P + B^{(t)}) \\ &= \begin{pmatrix} \mathrm{ReLU}(H^{(t-1)}) & \mathrm{ReLU}(A_R H^{(t-1)} W^*_R - J_{N\times N}) & \mathrm{ReLU}(A_P H^{(t-1)} W^*_P - J_{N\times N}) \end{pmatrix} \\ &= \begin{pmatrix} H^{(t-1)} & \mathrm{ReLU}(A_R H^{(t-1)} W^*_R - J_{N\times N}) & \mathrm{ReLU}(A_P H^{(t-1)} W^*_P - J_{N\times N}) \end{pmatrix} \end{aligned} \tag{58}$$

Because the three constituent blocks in the right hand side of (58) are all RIME, so is the whole matrix $H^{(t)}$.

Now consider any $u, v \in V_G$ for which $H_u^{(t)} = H_v^{(t)}$. Then the following equivalences hold:

$$
\begin{aligned}
& H_u^{(t)} = H_v^{(t)} \\
\Leftrightarrow & \begin{cases} H_u^{(t-1)} = H_v^{(t-1)} \\ \Gamma_R(H^{(t-1)})(u) = \Gamma_R(H^{(t-1)})(v) \\ \Gamma_P(H^{(t-1)})(u) = \Gamma_P(H^{(t-1)})(v) \end{cases} \\
\Leftrightarrow & \begin{cases} H_u^{(t-1)} = H_v^{(t-1)} \\ \left\{\!\!\left\{ H_x^{(t-1)} \mid x \in \mathcal{N}_R(u) \right\}\!\!\right\} = \left\{\!\!\left\{ H_x^{(t-1)} \mid x \in \mathcal{N}_R(v) \right\}\!\!\right\} \\ \left\{\!\!\left\{ H_x^{(t-1)} \mid x \in \mathcal{N}_P(u) \right\}\!\!\right\} = \left\{\!\!\left\{ H_x^{(t-1)} \mid x \in \mathcal{N}_P(v) \right\}\!\!\right\} \end{cases} \\
\Leftrightarrow & \begin{cases} \kappa^{(t-1)}(G)(u) = \kappa^{(t-1)}(G)(v) \\ \left\{\!\!\left\{ \kappa^{(t-1)}(G)(x) \mid x \in \mathcal{N}_R(u) \right\}\!\!\right\} = \left\{\!\!\left\{ \kappa^{(t-1)}(G)(x) \mid x \in \mathcal{N}_R(v) \right\}\!\!\right\} \\ \left\{\!\!\left\{ \kappa^{(t-1)}(G)(x) \mid x \in \mathcal{N}_P(u) \right\}\!\!\right\} = \left\{\!\!\left\{ \kappa^{(t-1)}(G)(x) \mid x \in \mathcal{N}_P(v) \right\}\!\!\right\} \end{cases} \\
\Leftrightarrow & \ \kappa^{(t)}(G)(u) = \kappa^{(t)}(G)(v)
\end{aligned}
\tag{59}
$$

The first equivalence holds due to (58) and the definitions of W_R^* and W_P^*. The second holds due to the definitions of $\Gamma_R(\cdot)$ and $\Gamma_P(\cdot)$, the third thanks to the inductive hypothesis and the last because the HASH-function in the recursive definition of $\kappa^{(t)}(G)$ is injective. We conclude that

$$H^{(t)} \equiv \kappa^{(t)}(G) \tag{60}$$

Which, combined with the observation that $H^{(t)}$ is RIME, finishes the induction step.

□

Corollary 2.8.1. *For all graphs G with initial node features $\{x_u = c(G)(u) \mid u \in V_G\}$, for all nodes $u, v \in V_G$, and for all choices of $T \in \mathbb{Z}^+$, there exists a model of $\mathcal{A}$ such that for all $0 \leq t \leq T$:*

$$\kappa^{(t)}(G)(u) \neq \kappa^{(t)}(G)(v) \Leftrightarrow \boldsymbol{h}_u^{(t)} \neq \boldsymbol{h}_v^{(t)} \tag{61}$$

(d) This question asks to compare the expressive power of model architectures $\mathcal{A}$ and $\mathcal{B}$ in terms of their *power to distinguish node representations*. This statement was found ambiguous by some students, including myself, and was later clarified to refer to *expressive power in terms of graph distinguishability in the sense discussed in the lectures*. So, we will compare $\mathcal{A}$ and $\mathcal{B}$ in terms of their ability to obtain distinct *sets* of node representations for non-isomorphic graphs G_1 and G_2.

With this sense of expressive power, we will prove that $\mathcal{A}$ is strictly more expressive than $\mathcal{B}$. This statement will be proven in two theorems, Theorem 2.9 and Theorem 2.10.

Theorem 2.9. *For any graphs G_1, G_2 that can be distinguished by a parametrization of $\mathcal{B}$, there exists a parametrization of $\mathcal{A}$ that distinguishes G_1, G_2.*

Proof. Consider a parametrization $\tilde{\mathcal{B}}$ that distinguishes G_1 and G_2. Suppose that $\tilde{\mathcal{B}}$ has $\tilde{T}$ layers and parameters $\tilde{\mathrm{MLP}}^{(t)}, \tilde{W}_s^{(t)}, \tilde{W}_Q^{(t)}, \tilde{W}_V^{(t)}, \tilde{\boldsymbol{b}}^{(t)}$ for $0 < t \leq \tilde{T}$. Further, call the intermediate node representations $\boldsymbol{q}_u^{(t)}$ for $0 \leq t \leq \tilde{T}$ and $u \in V_{G_1} \cup V_{G_2}$, where the representations from G_1 and G_2

can be told apart by the node index. As clarified in the beginning of this question, the fact that $\tilde{\mathcal{B}}$ can distinguish G_1 and G_2 means

$$\left\{\!\!\left\{ \boldsymbol{q}_v^{(\tilde{T})} \mid v \in V_{G_1} \right\}\!\!\right\} \neq \left\{\!\!\left\{ \boldsymbol{q}_v^{(\tilde{T})} \mid v \in V_{G_2} \right\}\!\!\right\} \tag{62}$$

There are two cases. For each case, we will construct a parametrization $\tilde{\mathcal{A}}$ that can distinguish G_1 and G_2.

Case 1. *For every* $0 \leq t < \tilde{T}$*:*

$$\sum_{v \in V_{G_1}} \boldsymbol{q}_v^{(t)} = \sum_{v \in V_{G_2}} \boldsymbol{q}_v^{(t)} \tag{63}$$

In this case, construct the parametrization $\tilde{\mathcal{A}}$ with the following parameters for every $0 < t \leq T = \tilde{T}$:

$$\begin{aligned}
\mathrm{MLP}^{(t)} &= \tilde{\mathrm{MLP}}^{(t)} && (64)\\
W_s^{(t)} &= \tilde{W}_s^{(t)} && (65)\\
W_R^{(t)} &= \tilde{W}_Q^{(t)} && (66)\\
W_P^{(t)} &= \tilde{W}_Q^{(t)} && (67)\\
\boldsymbol{b}^{(t)} &= \sum_{v \in V_{G_1}} \tilde{W}_V^{(t)} \boldsymbol{q}_v^{(t-1)} + \tilde{\boldsymbol{b}}^{(t)} \\
&= \sum_{v \in V_{G_2}} \tilde{W}_V^{(t)} \boldsymbol{q}_v^{(t-1)} + \tilde{\boldsymbol{b}}^{(t)} && (68)
\end{aligned}$$

Call the intermediate representations generated by $\tilde{\mathcal{A}}$ $\boldsymbol{p}_u^{(t)}$ for all $0 \leq t \leq T$ and $u \in V_{G_1} \cup V_{G_2}$. Now we will prove by induction that, for all $u \in V_{G_1} \cup V_{G_2}$ and $0 \leq t \leq T$:

$$\boldsymbol{p}_u^{(t)} = \boldsymbol{q}_u^{(t)} \tag{69}$$

The base case $t = 0$ follows from the fact that $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ start with the same initial representations for both G_1 and G_2. Now assume that (69) holds for $t-1$. Then, for all $G \in \{G_1, G_2\}$ and $u \in V_G$:

$$\begin{aligned}
\boldsymbol{p}_u^{(t)} &= \mathrm{MLP}^{(t)} \left(W_s^{(t)} \boldsymbol{p}_u^{(t-1)} + \sum_{v \in \mathcal{N}_R(u)} W_R^{(t)} \boldsymbol{p}_v^{(t-1)} + \sum_{v \in \mathcal{N}_P(u)} W_P^{(t)} \boldsymbol{p}_v^{(t-1)} + \boldsymbol{b}^{(t)} \right) \\
&= \tilde{\mathrm{MLP}}^{(t)} \left(\tilde{W}_s^{(t)} \boldsymbol{p}_u^{(t-1)} + \sum_{v \in \mathcal{Q}(u)} \tilde{W}_Q^{(t)} \boldsymbol{p}_v^{(t-1)} + \sum_{v \in V_G} \tilde{W}_V^{(t)} \boldsymbol{q}_v^{(t-1)} + \tilde{\boldsymbol{b}}^{(t)} \right) \\
&= \tilde{\mathrm{MLP}}^{(t)} \left(\tilde{W}_s^{(t)} \boldsymbol{q}_u^{(t-1)} + \sum_{v \in \mathcal{Q}(u)} \tilde{W}_Q^{(t)} \boldsymbol{q}_v^{(t-1)} + \sum_{v \in V_G} \tilde{W}_V^{(t)} \boldsymbol{q}_v^{(t-1)} + \tilde{\boldsymbol{b}}^{(t)} \right) \\
&= \boldsymbol{q}_u^{(t)}
\end{aligned} \tag{70}$$

Here, the first and last equalities are the definitions of $\boldsymbol{p}_u^{(t)}$ and $\boldsymbol{q}_u^{(t)}$. The second equality follows from the parameter definitions (64-68) and the fact that $\mathcal{Q}(u) = \mathcal{N}_R(u) \cup \mathcal{N}_P(u)$. The third follows from the induction hypothesis (69). This concludes the induction step.

From (62) and (69) now follows that

$$\left\{\!\!\left\{ \boldsymbol{p}_v^{(T)} \mid v \in V_{G_1} \right\}\!\!\right\} = \left\{\!\!\left\{ \boldsymbol{q}_v^{(\tilde{T})} \mid v \in V_{G_1} \right\}\!\!\right\} \neq \left\{\!\!\left\{ \boldsymbol{q}_v^{(\tilde{T})} \mid v \in V_{G_2} \right\}\!\!\right\} = \left\{\!\!\left\{ \boldsymbol{p}_v^{(T)} \mid v \in V_{G_2} \right\}\!\!\right\} \tag{71}$$

So $\tilde{\mathcal{A}}$ can distinguish G_1 and G_2.

Case 2. *There are some* $0 \leq t < \tilde{T}$ *such that*

$$\sum_{v \in V_{G_1}} \boldsymbol{q}_v^{(t)} \neq \sum_{v \in V_{G_2}} \boldsymbol{q}_v^{(t)} \tag{72}$$

Let t^* be the smallest t for which (72) holds. Then construct $\tilde{\mathcal{A}}$ with $T = t^*$ and parameters defined by (64-68) for $0 < t \leq t^*$. The inductive proof from Case 1 is still valid for $0 \leq t \leq t^*$, because $t = t^*$ is the first value for which the assumption (63) breaks. Hence, for all $u \in V_{G_1} \cup V_{G_2}$:

$$\boldsymbol{p}_u^{(t^*)} = \boldsymbol{q}_u^{(t^*)} \tag{73}$$

From (72) follows that

$$\left\{\!\!\left\{ \boldsymbol{q}_v^{(t^*)} \mid v \in V_{G_1} \right\}\!\!\right\} \neq \left\{\!\!\left\{ \boldsymbol{q}_v^{(t^*)} \mid v \in V_{G_2} \right\}\!\!\right\} \tag{74}$$

Combined with (73), this gives

$$\left\{\!\!\left\{ \boldsymbol{p}_v^{(t^*)} \mid v \in V_{G_1} \right\}\!\!\right\} = \left\{\!\!\left\{ \boldsymbol{q}_v^{(t^*)} \mid v \in V_{G_1} \right\}\!\!\right\} \neq \left\{\!\!\left\{ \boldsymbol{q}_v^{(t^*)} \mid v \in V_{G_2} \right\}\!\!\right\} = \left\{\!\!\left\{ \boldsymbol{p}_v^{(t^*)} \mid v \in V_{G_2} \right\}\!\!\right\} \tag{75}$$

Which means that $\tilde{\mathcal{A}}$ can distinguish G_1 and G_2. □

Theorem 2.10. *There exist graphs* G_1, G_2 *that can be distinguished by a parametrization of* $\mathcal{A}$*, but not by any parametrization of* $\mathcal{B}$*.*

Proof. It is not entirely clear from the assignment whether the division $V_G = R \cup P$ should be considered part of the graph or not. If so, this proof becomes trivial and one can take as an example the graphs $G_1 = C_3, R_1 = \{v_1\}$ and $G_2 = C_3, R_2 = \emptyset$ with the same input features for every node in both graphs. These graphs can trivially not be distinguished by any parametrization of $\mathcal{B}$, as $\mathcal{B}$ has no information about R at all. However, I chose not to risk making the assignment easier due to a misinterpretation, and therefore construct two *non-isomorphic* graphs that can be distinguished by $\mathcal{A}$ but not by $\mathcal{B}$.

Let $G_1 = C_3 \cup C_3$ and $G_2 = C_6$, where C_n is the cycle graph with n nodes and the initial node features are $x_u = 1 \in \mathbb{I}^1$ for all $u \in V_{G_1} \cup V_{G_2}$ (dropping the vector notation for vectors of dimension 1 and matrices of size 1×1). Further, let R be any set of two neighbouring nodes, in both graphs. Then the parametrization $\tilde{\mathcal{A}}$ with 1 layer and the following parameters distinguishes G_1 and G_2:

$$W_s^{(1)} = 0 \tag{76}$$

$$W_R^{(1)} = 1 \tag{77}$$

$$W_P^{(1)} = 0 \tag{78}$$

$$b^{(1)} = 0 \tag{79}$$

$$\mathrm{MLP}^{(1)} = \mathrm{Id} \tag{80}$$

Because the sets of generated node representations are not equal for both graphs:

$$\left\{\!\!\left\{ \tilde{\mathcal{A}}(G_1)(u) \mid u \in V_{G_1} \right\}\!\!\right\} = \{\!\{2, 1, 1, 0, 0, 0\}\!\} \tag{81}$$

$$\left\{\!\!\left\{ \tilde{\mathcal{A}}(G_2)(u) \mid u \in V_{G_2} \right\}\!\!\right\} = \{\!\{1, 1, 1, 1, 0, 0\}\!\} \tag{82}$$

In contrast, no parametrization of $\mathcal{B}$ can distinguish G_1 and G_2. Consider any parametrization $\tilde{\mathcal{B}}$ with parameters $\mathrm{MLP}^{(t)}, W_s^{(t)}, W_Q^{(t)}, W_V^{(t)}, \boldsymbol{b}^{(t)}$ for every $0 < t \leq T$. We will prove by induction that, for all $0 \leq t \leq T$, there exists a $\boldsymbol{h}^{(t)} \in \mathbb{R}^{d^{(t)}}$ that all node representations in both G_1 and G_2 are equal to.

The base case $t = 0$ is satisfied, because the input features for all nodes in both graphs are 1. Now assume that the induction hypothesis holds for $t - 1$. Then, for every $G \in \{G_1, G_2\}$ and $u \in V_G$:

$$\begin{aligned} \boldsymbol{h}_u^{(t)} &= \mathrm{MLP}^{(t)} \left(W_s^{(t)} \boldsymbol{h}_u^{(t-1)} + W_Q^{(t)} \sum_{v \in \mathcal{Q}(u)} \boldsymbol{h}_v^{(t-1)} + W_V^{(t)} \sum_{v \in V_G} \boldsymbol{h}_v^{(t-1)} + \boldsymbol{b}^{(t)} \right) \\ &= \mathrm{MLP}^{(t)} \left(W_s^{(t)} \boldsymbol{h}^{(t-1)} + W_Q^{(t)} \; 2\boldsymbol{h}^{(t-1)} + W_V^{(t)} \; 6\boldsymbol{h}^{(t-1)} + \boldsymbol{b}^{(t)} \right) \end{aligned} \tag{83}$$

Because for every $G \in \{G_1, G_2\}$ and every $u \in V_G$, $|\mathcal{Q}(u)| = 2$ and $|V_G| = 6$. The result of (83) does not depend on G, nor on u. So all $\boldsymbol{h}_u^{(t)}$ for $u \in V_{G_1} \cup V_{G_2}$ are equal to each other, and hence to some constant vector $\boldsymbol{h}^{(t)} \in \mathbb{R}^{d^{(t)}}$, which finishes the induction step.

It follows that

$$\left\{\!\!\left\{ \tilde{\mathcal{B}}(G_1)(u) \mid u \in V_{G_1} \right\}\!\!\right\} = \left\{\!\!\left\{ \boldsymbol{h}^{(T)}, \boldsymbol{h}^{(T)}, \boldsymbol{h}^{(T)}, \boldsymbol{h}^{(T)}, \boldsymbol{h}^{(T)}, \boldsymbol{h}^{(T)} \right\}\!\!\right\} = \left\{\!\!\left\{ \tilde{\mathcal{B}}(G_2)(u) \mid u \in V_{G_2} \right\}\!\!\right\} \tag{84}$$

So no parametrization of $\mathcal{B}$ can distinguish between G_1 and G_2.

□

From Theorems 2.9 and 2.10 now follows the desired conclusion that $\mathcal{A}$ is strictly more expressive than $\mathcal{B}$.

Question 3

(a) For this project, we compare two model architectures in terms of their ability to capture correlation between the input features and the target labels in an inductive node classification task. The first studied architecture is GATv2 ($\mathcal{A}$) [Brody et al., 2022], consisting of GATv2 layers interleaved with nonlinearities. The second studied architecture ($\mathcal{B}$) is an extension thereof, consisting of a slightly modified type of layers interleaved with nonlinearities:

$$\mathcal{A}_{\text{layer}} : \begin{cases} e_{ij}^{(t)} = \boldsymbol{a}^{(t)^T} \text{LeakyReLU}\left(\begin{pmatrix} W_l^{(t)} & W_r^{(t)} \end{pmatrix} \begin{pmatrix} \boldsymbol{h}_i^{(t-1)} \\ \boldsymbol{h}_j^{(t-1)} \end{pmatrix}\right) & \text{for } j \in \mathcal{N}_i \\ \alpha_{ij}^{(t)} = \text{softmax}_j(e_{ij}^{(t)}) \\ \boldsymbol{h}_i^{(t)} = \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(t)} W_r^{(t)} \boldsymbol{h}_j^{(t-1)} \end{cases} \tag{85}$$

$$\mathcal{B}_{\text{layer}} : \begin{cases} e_{ij}^{(t)} = \boldsymbol{a}^{(t)^T} \text{LeakyReLU}\left(\begin{pmatrix} W_l^{(t)} & W_r^{(t)} \end{pmatrix} \begin{pmatrix} \boldsymbol{h}_i^{(t-1)} \\ \boldsymbol{h}_j^{(t-1)} \end{pmatrix}\right) & \text{for } j \in \mathcal{N}_i \\ \alpha_{ij}^{(t)} = \text{softmax}_j(e_{ij}^{(t)}) \\ \boldsymbol{h}_i^{(t)} = W_l^{(t)} \boldsymbol{h}_i^{(t-1)} + \sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(t)} W_r^{(t)} \boldsymbol{h}_j^{(t-1)} \end{cases} \tag{86}$$

My motivation for this study is of a practical nature. In the GATv2 paper, I observed that (in their default implementation) $\boldsymbol{h}_i^{(t-1)}$ influences $\boldsymbol{h}_i^{(t)}$ solely through $e_{ij}^{(t)}$. Considering that many graphs exhibit a correlation between the node targets and the input features, it seemed to me that the model would benefit from a more explicit connection. Therefore, I devised $\mathcal{B}$ as a natural extension of $\mathcal{A}$ that does not introduce any new parameters and I hypothesized that it would be better able to capture the aforementioned input-target correlation.

(b) As both architectures are quite convoluted, it is hard to prove any theoretical bounds on their expressivity, certainly considering the three page limit of this project. Therefore, we opt for an empirical setup that consists of a dataset of synthetically generated graphs, where each graph has a known correlation between the input features and the targets. One-layer and two-layer models are trained on this dataset and evaluated on test graphs unseen during training, as the task is inductive. We use this setup to test the hypothesis that $\mathcal{B}$ is better able than $\mathcal{A}$ at classifying nodes correctly when there is a large input-target correlation. All code from this study is available at the following anonymized Git repository: https://anonymous.4open.science/r/GRL-mini-project-2023-7C24

(c) The node classification dataset consists of 10 sections, where each section contains 150 training graphs, 50 validation graphs and 50 test graphs. All graphs are undirected, have between 20 and 30 nodes, have no self-loops and were synthetically generated according to the Erdős-Rényi model with $r = 0.2$ [Erdős et al., 1960]. Graphs containing isolated nodes were excluded through rejection sampling, as $\mathcal{A}$ can generate no representations for such nodes apart from the zero vector. The per-node targets are uniformly chosen among 5 color classes, and the initial features are uniformly chosen vectors from

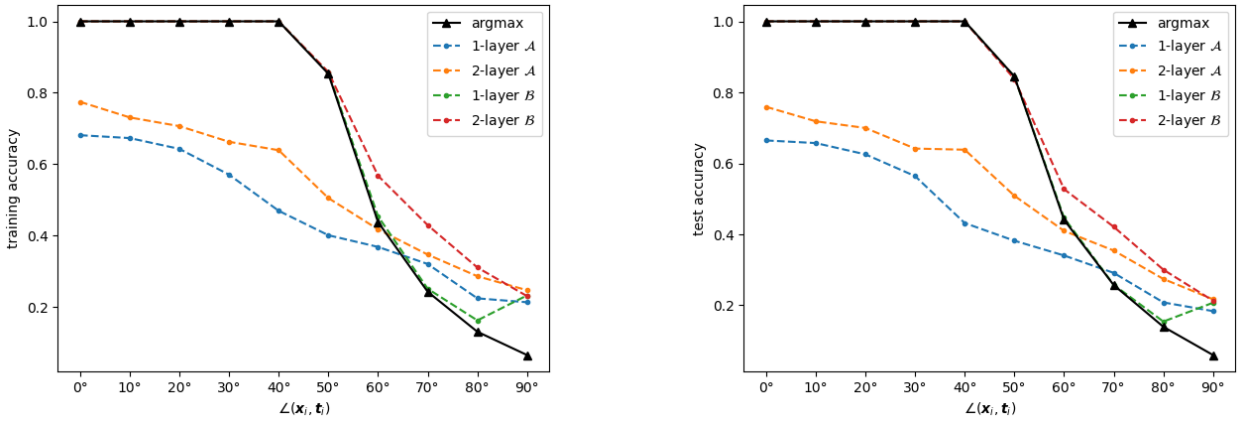
$$\left\{ \boldsymbol{x}_i \in \mathbb{R}^5 \mid \arccos \frac{\boldsymbol{x}_i^T \boldsymbol{t}_i}{\|\boldsymbol{x_i}\| \, \|\boldsymbol{t_i}\|} = \theta_s, \|\boldsymbol{x}\| = 1 \right\} \tag{87}$$

where $\boldsymbol{t}_i$ is the target for node i and θ_s is the angle specific to the database section s. The angles for the different sections are $\{10^\circ \cdot s \mid 0 \leq s < 10\}$ and their cosines correspond with the correlation between $\boldsymbol{x}_i$ and $\boldsymbol{t}_i$ [Rodgers and Nicewander, 1988].

In this study, we compare four models: a one-layer and a two-layer model of each architecture. Call these models A_1, A_2, B_1 and B_2. For each section of the dataset, all four models are trained on the training graphs and evaluated on the test graphs. The validation graphs are used to monitor the training process and apply early stopping when the validation loss has not improved for 10 epochs. This process is repeated three times. The used hyperparameters are summarized in Table 2, most of these are adopted from the annotated implementation in [Brody et al., 2022] that applies a GATv2 model on the Cora dataset. I believe these hyperparameters are relevant, as Cora is another node-level classification task. The main deviations in this study are the removal of the dropout layer, which is inappropriate because many nodes have low degree, and the smaller size of the models.

Table 2: Hyperparameters

Hyperparameter	**Value**
Loss Function	Cross-Entropy
Optimizer	Adam
Learning Rate	5×10^{-3}
Weight Decay	5×10^{-4}
Dropout	0
Max Epochs	1000
Patience	10
Batch Size	15
Hidden Layer Dimension	5
Inter-layer Nonlinearity	Tanh



(a) Training Accuracies

(b) Test Accuracies

Figure 1: Training and Test Accuracies

(d) Figure 1 plots the median training and test accuracies after training three instances of each model for every section of the dataset, and compares them to a baseline model *argmax*. This baseline predicts that a node belongs to the color class with the same index as the largest entry in its initial feature vector, and can classify nodes with 100% accuracy as long as $\theta = \angle(\boldsymbol{x}_i, \boldsymbol{t}_i) < 45°$.

The test accuracies confirm the hypothesis that $\mathcal{B}$ is better than $\mathcal{A}$ at capturing input-target correlation. Both B_1 and B_2 match the accuracy of *argmax* for $\theta \leq 50°$. B_2 even improves upon it for higher θ, possibly by filtering for the feature vector entry that equals $\cos(\theta)$. In contrast, both $\mathcal{A}$ models are clearly incapable of learning the identity function: even when the targets are identical to the features ($\theta = 0$), they struggle to classify the nodes correctly. The fact that the training accuracies closely match the test accuracies shows that this is indeed an expressiveness issue rather than a generalisation issue. While increasing the hidden layer dimension may enhance the performance of the 2-layer $\mathcal{A}$ model, the B models exhibit a notable superiority even without additional parameters.

These results can be theoretically motivated by considering the one-layer parametrization of $\mathcal{B}$ with parameters $\boldsymbol{a}^{(1)} = \mathbf{0}^d, W_l^{(1)} = I_d$ and $W_r^{(1)} = \mathbf{0}_{d\times d}$. This parametrization implements the identity operation on the node features, which after taking the maximum entry as its predictions amounts to the *argmax* model. Hence, it is not surprising that B_1 performs at least as well as the baseline. Further, B_2 is provably strictly more expressive than B_1: it can use any parametrization of B_1 as a first layer, apply a monotone nonlinearity and then apply the aforementioned identity layer to obtain the same classifications as the used B_1 parametrization. Therefore, it is in line with expectations that B_2 further outperforms B_1.

For the $\mathcal{A}$ models, there is not such a straightforward parametrization of the identity operation. Instead, A_1 must learn to manipulate the e_{ij} such that $\sum_{j\in\mathcal{N}_i} \alpha_{ij}^{(t)} W_r^{(t)} \boldsymbol{x}_j$ has the same maximal element as $\boldsymbol{x}_i$, for all i. This is a much harder task, both intuitively and evidenced by the results. Thanks to its second layer, A_2 has the additional option to propagate the information of a node's features through its neighbours, but it still cannot match the performance of the $\mathcal{B}$ models.

It is quite standard in graph neural networks to have a direct dependence of $\boldsymbol{h}_u^{(t)}$ on $\boldsymbol{h}_u^{(t-1)}$, most notably in the Basic GNN Model [Hamilton, 2020], the Gated GNN Model [Li et al., 2017] and the GraphSAGE framework [Hamilton et al., 2017]. Where this dependency is not natively present, adding self-loops is a common and well-studied practice to improve performance. For example in the case of Graph Convolutional Networks, [Kipf and Welling, 2017] observe experimentally that adding self-loops improves accuracy on popular benchmarks and [Wu et al., 2019] provide theoretical support for these results. I considered doing this project about the effect of self-loops on the expressiveness of GATv2. However, I opted against this idea because [Brody et al., 2022] do use self-loops in some of their experiments and I preferred to examine a more innovative approach.

(e) The experimental results support the hypothesis that $\mathcal{B}$ outperforms $\mathcal{A}$ at capturing input-target correlation. However, it should be noted that $\mathcal{B}$'s benefit may not extend to more realistic datasets, as the presented task is unrealistic in at least two ways. First, it has been argued that the ER-model does not generate realistic graphs [Saberi, 2015]. The only reason why this model was chosen is for the sake of simplicity. Second, in the presented task there is no correlation between the graph stucture and the targets whatsoever. This situation never occurs in real-world graphs, as that would contradict the purpose of modelling the dataset as a graph.

Moreover, this study only considers models with one or two layers. While I hypothesize similar outcomes when using more layers, further study is required to assert this more confidently.

(f) The results in this study are not too surprising as the experiment was explicitly designed to test the aspect in which $\mathcal{B}$ surpasses $\mathcal{A}$. Therefore, it is imperative to compare the models on more realistic benchmarks to reach more conclusive statements. For this purpose, I suggest utilizing the *ogbn-products* benchmark [Hu et al., 2020] because of its information-rich input features.

Next, it is essential to compare the expressiveness of multi-layer models, both on the presented task and on more realistic benchmarks. This broader evaluation is necessary to extend the conclusion about the expressiveness of $\mathcal{A}$ and $\mathcal{B}$ beyond their one-layer and two-layer instances.

Finally, it would be useful to theoretically compare both architectures in the spirit of question 2d. It could be an intermediate step (and would be useful in its own right) to prove theoretical upper and lower bounds for $\mathcal{A}$ and $\mathcal{B}$ or simplications thereof. The 1-WL test is certainly an upper bound for both architectures as they are both MPNNs [Xu et al., 2018] and perhaps one can prove that this bound is also tight, like [Morris et al., 2021] did for the basic GNN model.

References

[Brody et al., 2022] Brody, S., Alon, U., and Yahav, E. (2022). How attentive are graph attention networks?

[Erdős et al., 1960] Erdős, P., Rényi, A., et al. (1960). On the evolution of random graphs. *Publ. math. inst. hung. acad. sci*, 5(1):17–60.

[Hamilton et al., 2017] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

[Hamilton, 2020] Hamilton, W. L. (2020). *Graph representation learning*. Morgan & Claypool Publishers.

[Hu et al., 2020] Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. (2020). Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133.

[Kipf and Welling, 2017] Kipf, T. N. and Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17.

[Leshno et al., 1993] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, 6(6):861–867.

[Li et al., 2017] Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. (2017). Gated graph sequence neural networks.

[Morris et al., 2021] Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. (2021). Weisfeiler and leman go neural: Higher-order graph neural networks.

[Rodgers and Nicewander, 1988] Rodgers, J. L. and Nicewander, W. A. (1988). Thirteen ways to look at the correlation coefficient. *American statistician*, pages 59–66.

[Saberi, 2015] Saberi, A. A. (2015). Recent advances in percolation theory and its applications. *Physics Reports*, 578:1–32.

[Wu et al., 2019] Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., and Weinberger, K. (2019). Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR.

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