Lecture 7: Message Passing Neural Networks and Randomisation

Relational Learning

İsmail İlkan Ceylan

Advanced Topics in Machine Learning, University of Oxford

10.02.2021

Overview

Overview

- The quest for expressive and scalable models

Overview

- The quest for expressive and scalable models
- MPNNs with random features

Overview

- The quest for expressive and scalable models
- MPNNs with random features
- Universality of MPNNs with random node initialisation

Overview

- The quest for expressive and scalable models
- MPNNs with random features
- Universality of MPNNs with random node initialisation
- Benchmarking expressiveness evaluation

Overview

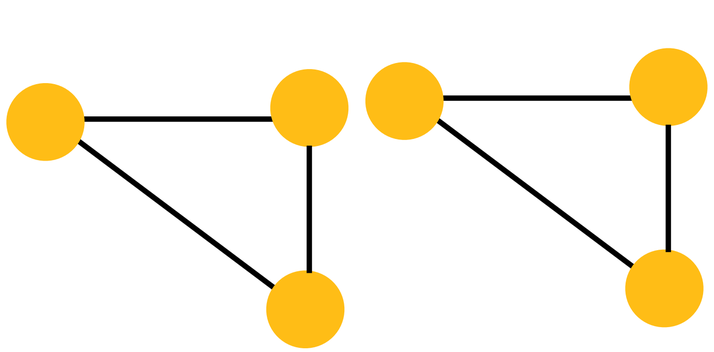
- The quest for expressive and scalable models
- MPNNs with random features
- Universality of MPNNs with random node initialisation
- Benchmarking expressiveness evaluation
- Discussions and outlook

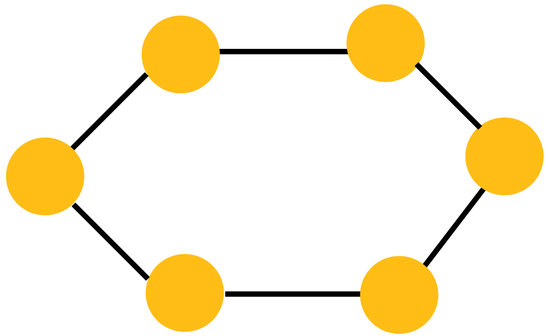
Overview

- The quest for expressive and scalable models
- MPNNs with random features
- Universality of MPNNs with random node initialisation
- Benchmarking expressiveness evaluation
- Discussions and outlook
- Summary

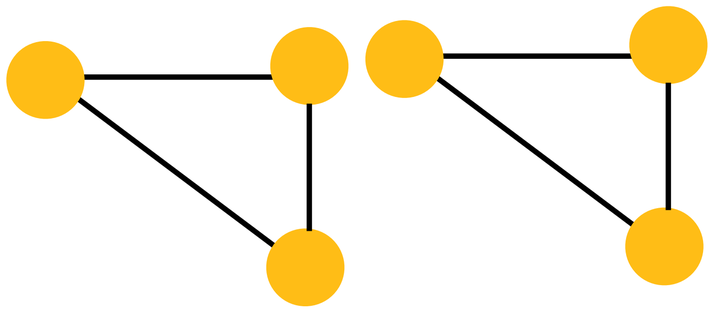
The Quest for Expressive and Scalable Models

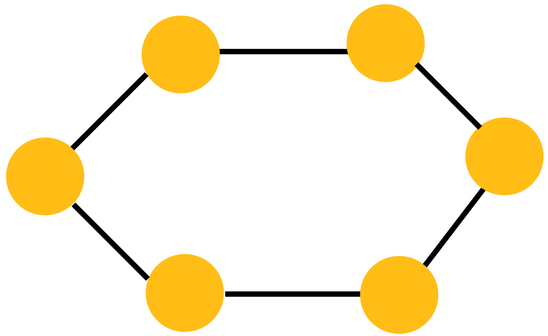
A Tale of Two Graphs





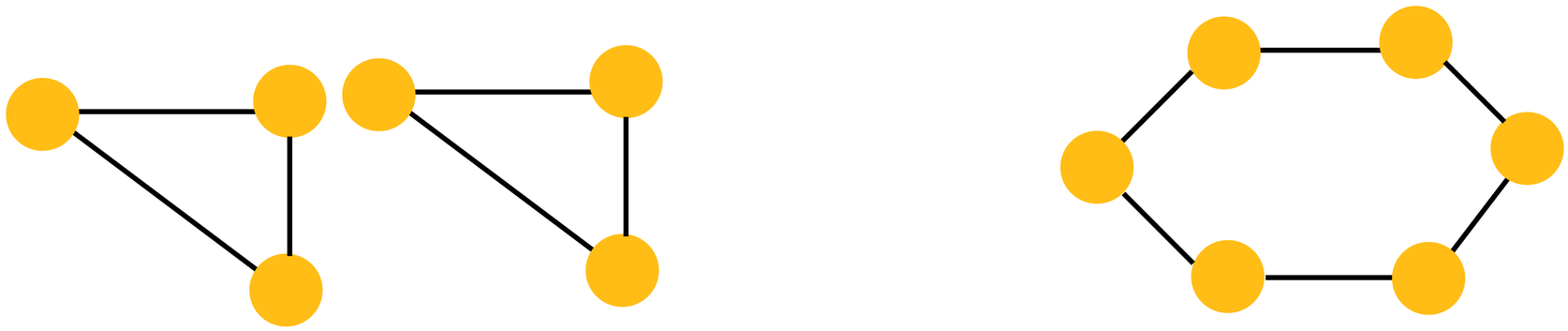
A Tale of Two Graphs





A brief recap

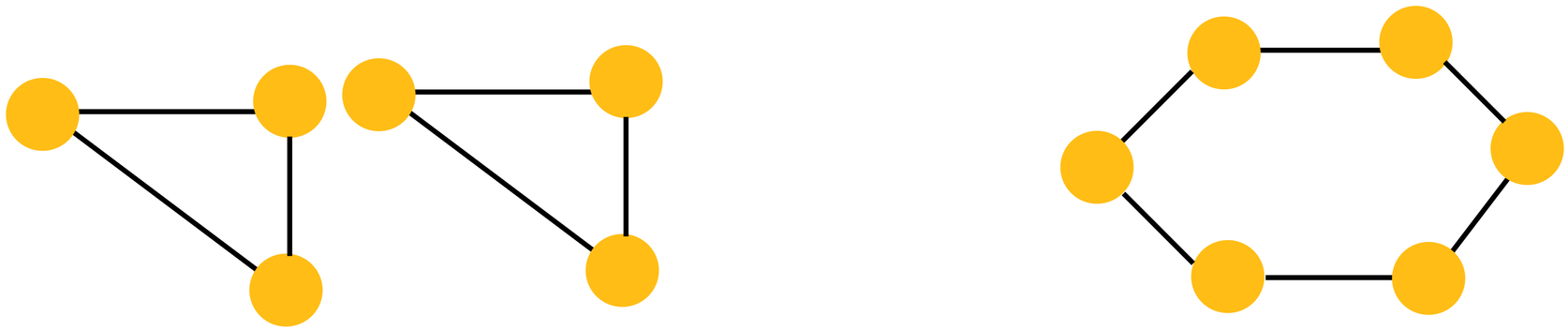
A Tale of Two Graphs



A brief recap

1. We have seen that 1-WL is insufficient and 2-WL is needed to distinguish these graphs.

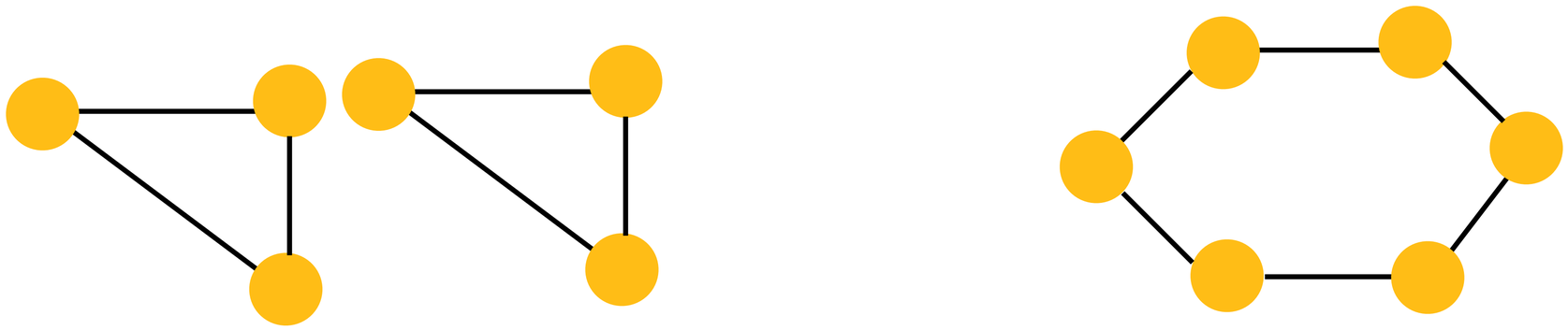
A Tale of Two Graphs



A brief recap

1. We have seen that 1-WL is insufficient and 2-WL is needed to distinguish these graphs.
2. The embedding learned for the graph on the left-hand side will be exactly the same as the embedding of the graph on the right-hand side for MPNNs!

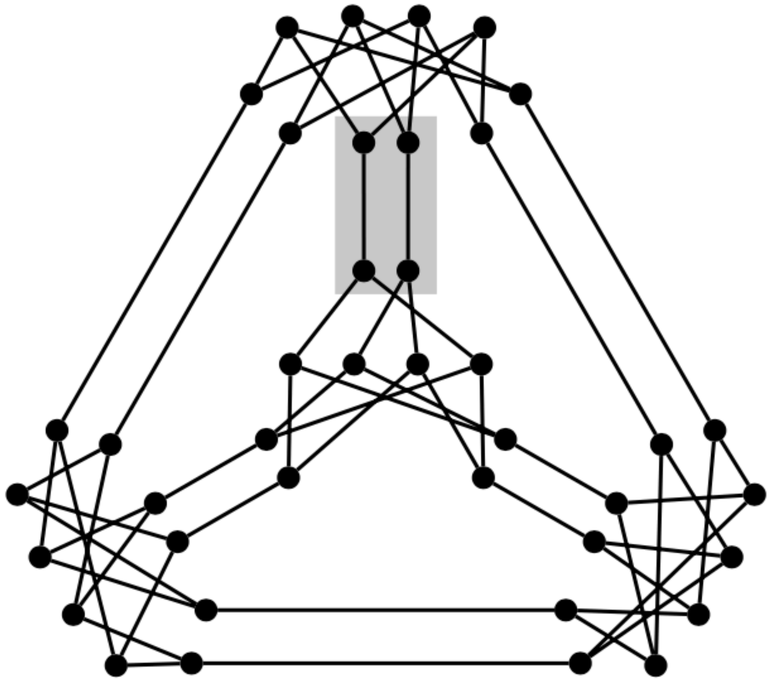
A Tale of Two Graphs

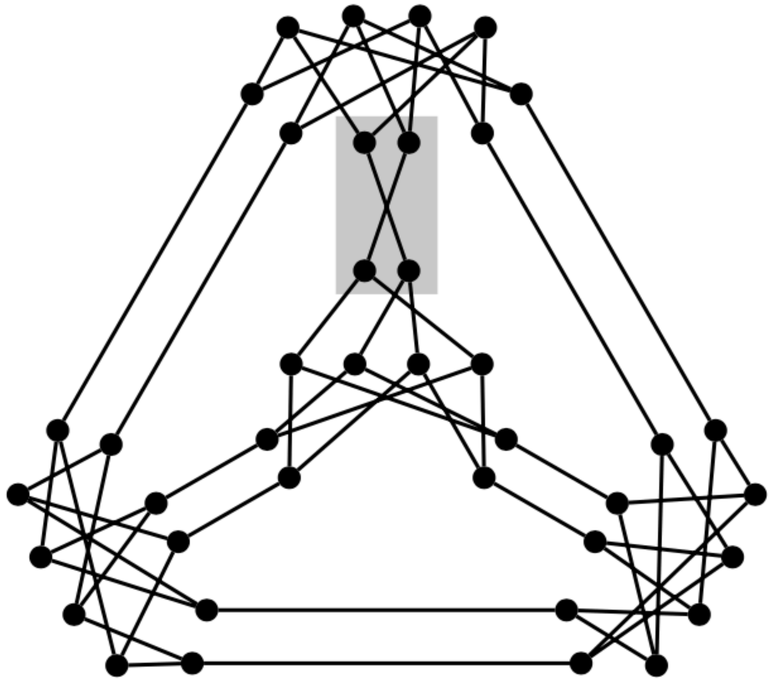


A brief recap

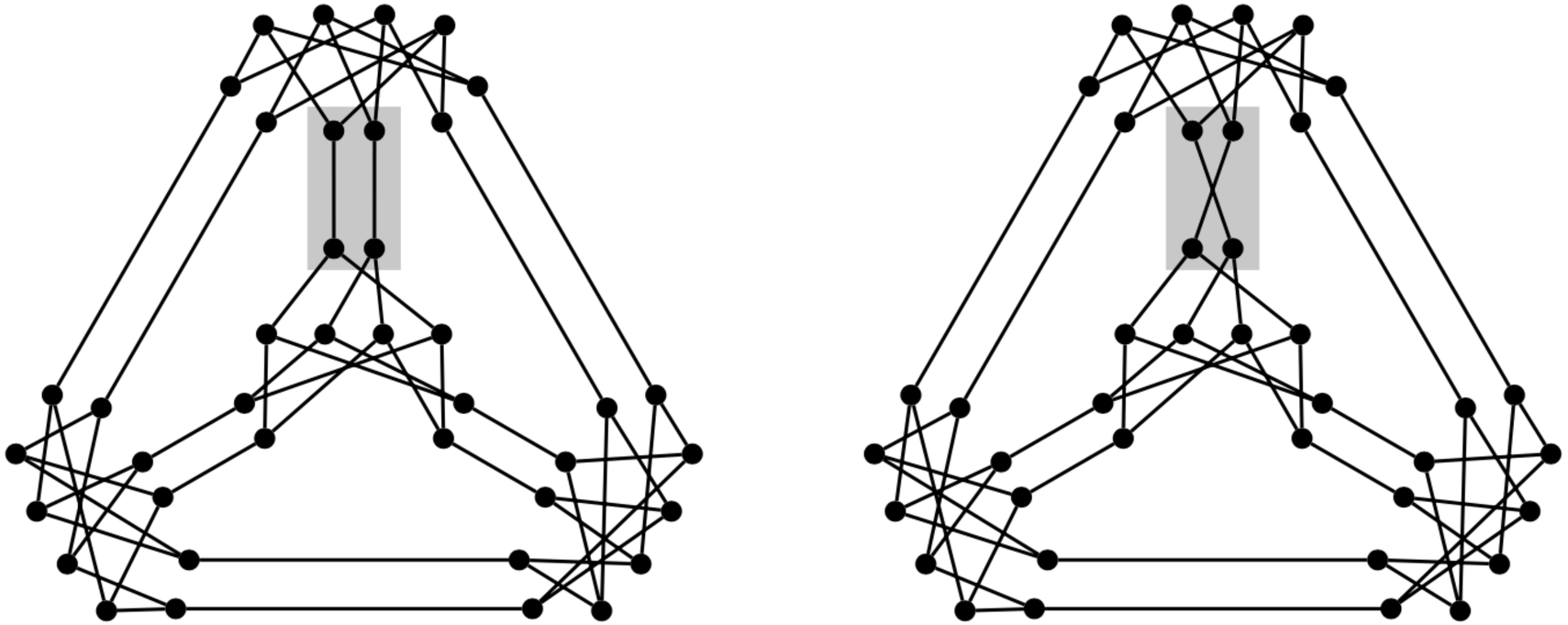
1. We have seen that 1-WL is insufficient and 2-WL is needed to distinguish these graphs.
2. The embedding learned for the graph on the left-hand side will be exactly the same as the embedding of the graph on the right-hand side for MPNNs!
3. There is a pair of non-isomorphic graphs distinguishable by $(k+1)$-WL but not by k-WL for each $k \geq 1$.

Graph Distinguishability





Graph Distinguishability



Example: 2WL cannot distinguish the shown graphs which differ only in the grey area (Grohe, 2017), i.e., even higher-order models such as 3-GNNs do not possess sufficient expressive power to distinguish these graphs.

A Tale of Two Coloured Graphs

A Tale of Two Coloured Graphs

Some observations:

A Tale of Two Coloured Graphs

Some observations:

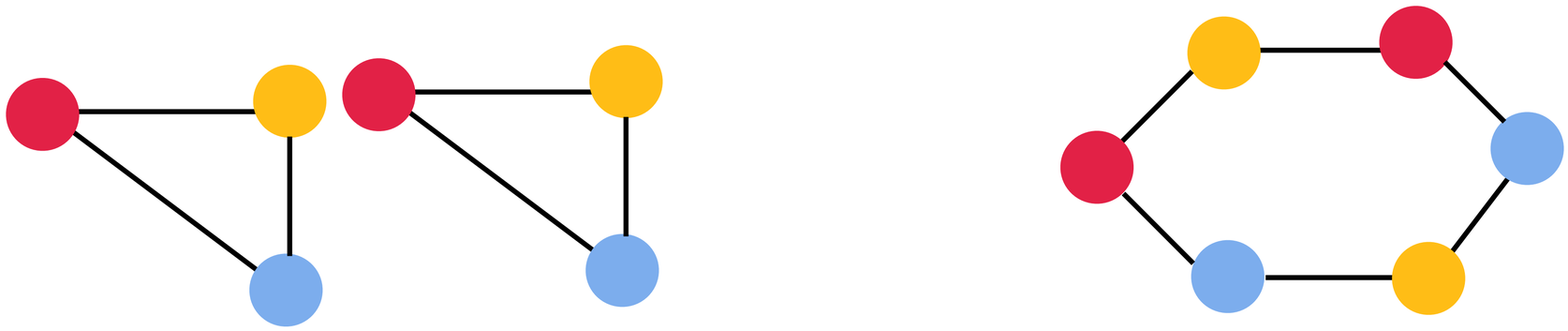
- The graphs we considered were not coloured, or equivalently, single-coloured.

A Tale of Two Coloured Graphs

Some observations:

- The graphs we considered were not coloured, or equivalently, single-coloured.
- The WL algorithm is defined in a more general way — we can start with any initial colouring.

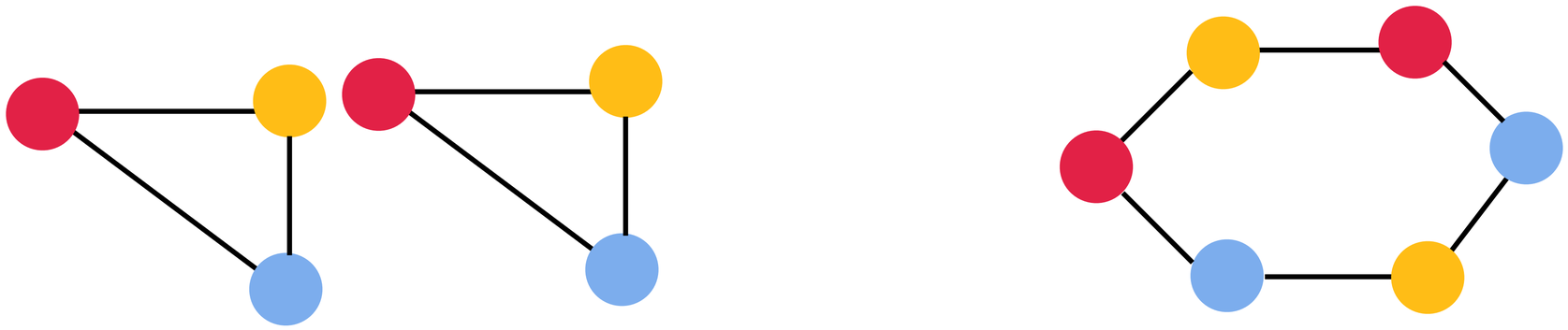
A Tale of Two Coloured Graphs



Some observations:

- The graphs we considered were not coloured, or equivalently, single-coloured.
- The WL algorithm is defined in a more general way — we can start with any initial colouring.

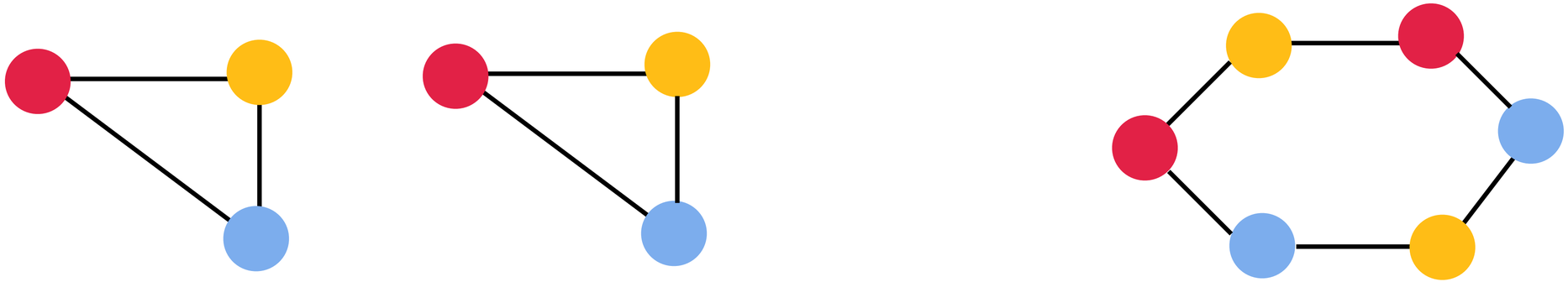
A Tale of Two Coloured Graphs



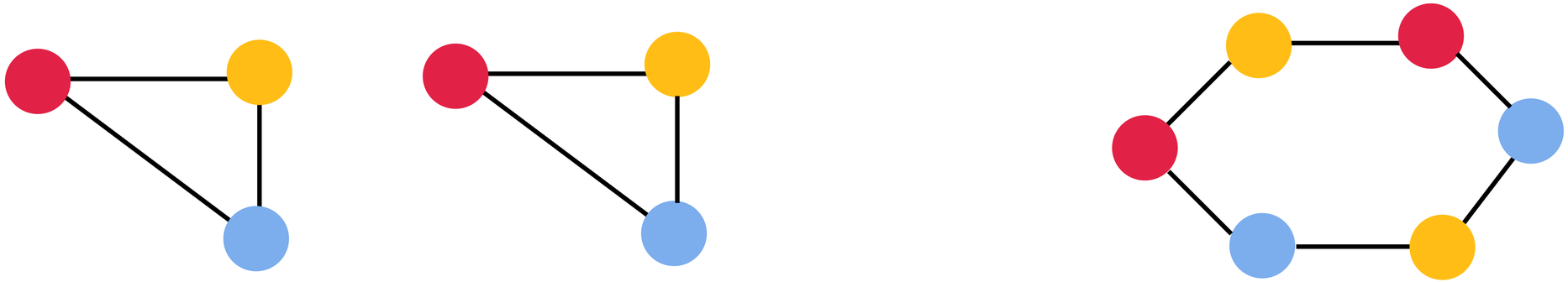
Some observations:

- The graphs we considered were not coloured, or equivalently, single-coloured.
- The WL algorithm is defined in a more general way — we can start with any initial colouring.
- The same is true for MPNNs — we can start with any node features.

A Tale of Two Coloured Graphs

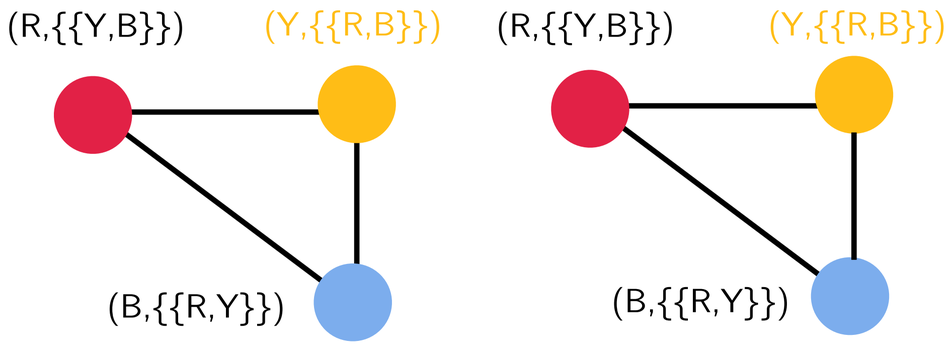


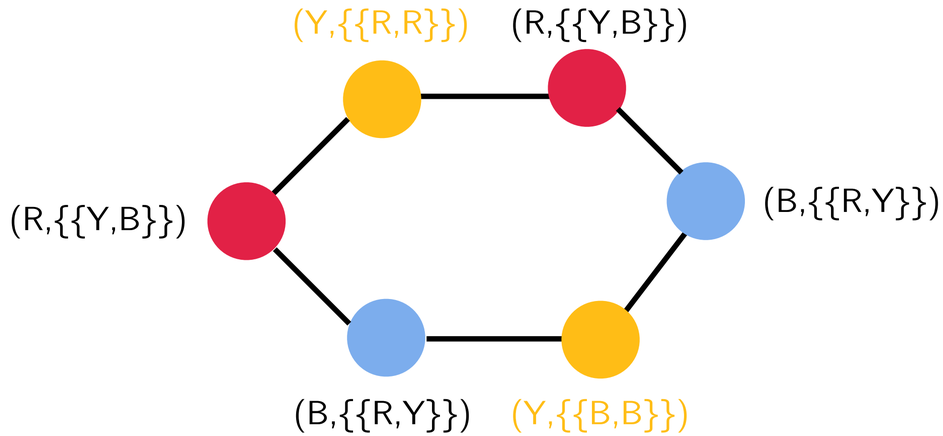
A Tale of Two Coloured Graphs



What happens when we colour the graph pairs?

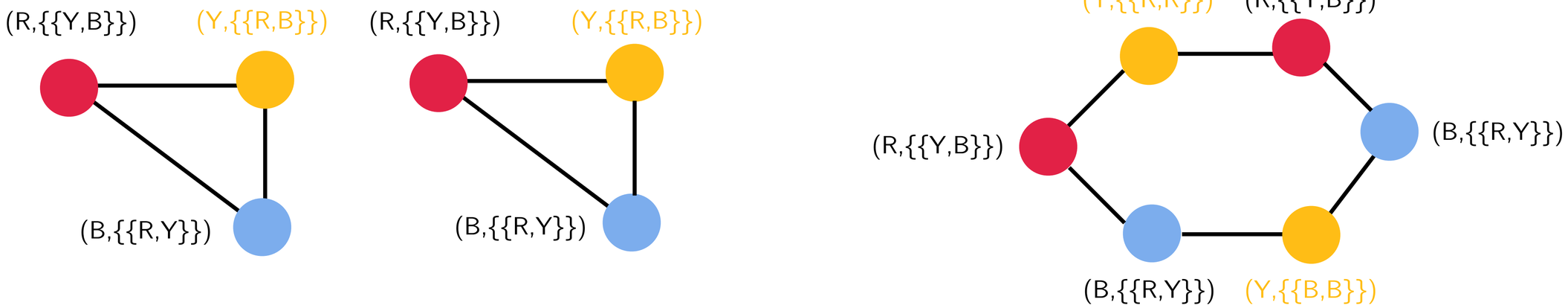
A Tale of Two Coloured Graphs





What happens when we colour the graph pairs?

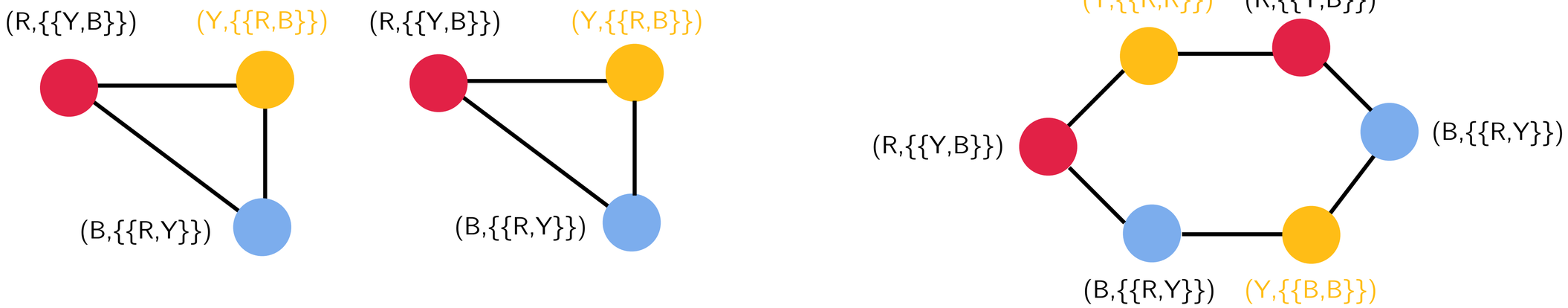
A Tale of Two Coloured Graphs



What happens when we colour the graph pairs?

- After the first iteration of the 1-WL algorithm the graphs are distinguished via the initially yellow nodes.

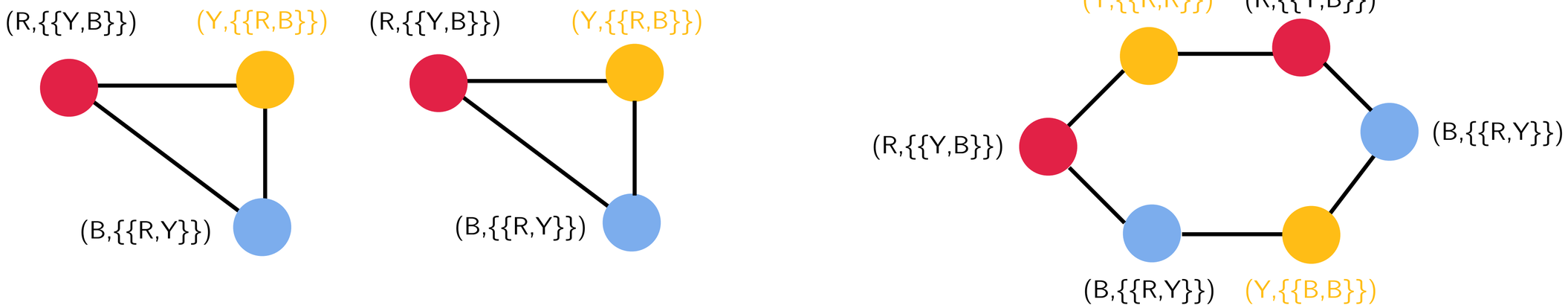
A Tale of Two Coloured Graphs



What happens when we colour the graph pairs?

- After the first iteration of the 1-WL algorithm the graphs are distinguished via the initially yellow nodes.
- After the second iteration the graphs will differ with respect to any node.

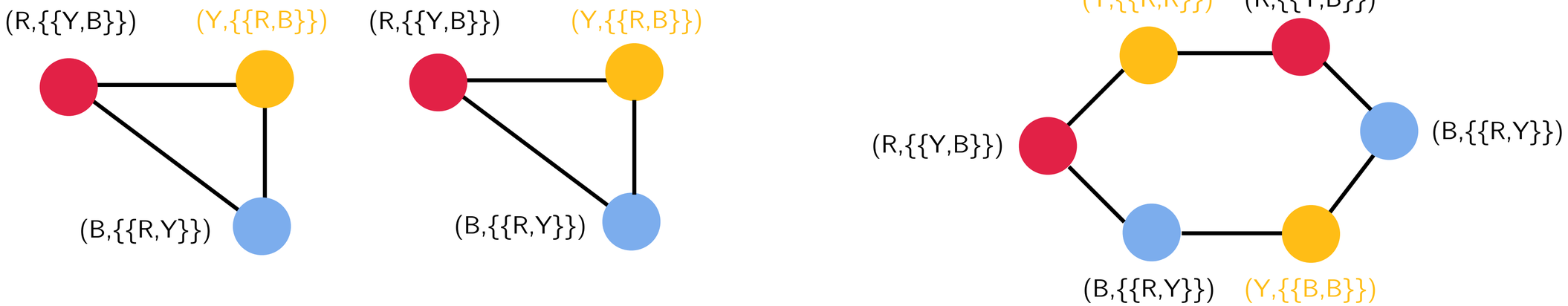
A Tale of Two Coloured Graphs



What happens when we colour the graph pairs?

- After the first iteration of the 1-WL algorithm the graphs are distinguished via the initially yellow nodes.
- After the second iteration the graphs will differ with respect to any node.
- The same is true for MPNNs — if we set the node features accordingly.

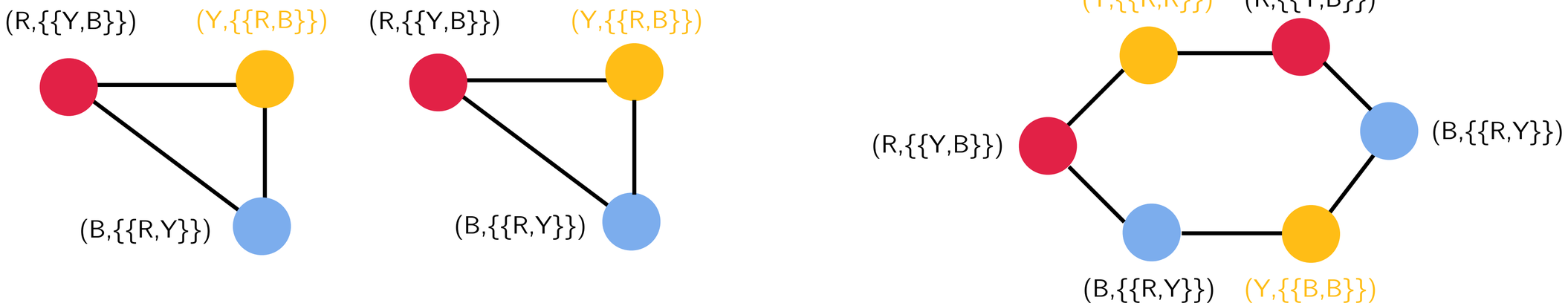
A Tale of Two Coloured Graphs



Wouldn't initialising node features in an MPNN to different colours be ideal?

This will yield an expressive model — 1-WL can distinguish any pair of ordered/coloured graphs.

A Tale of Two Coloured Graphs

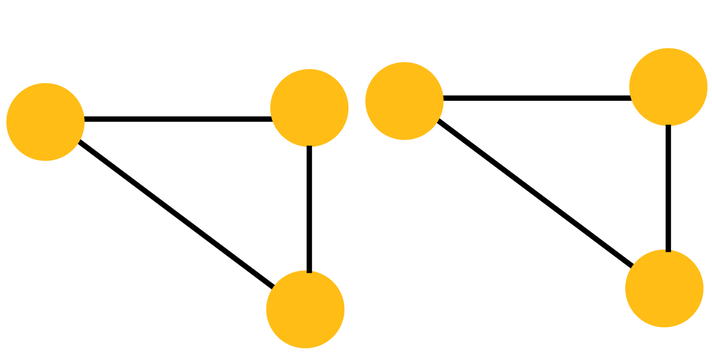


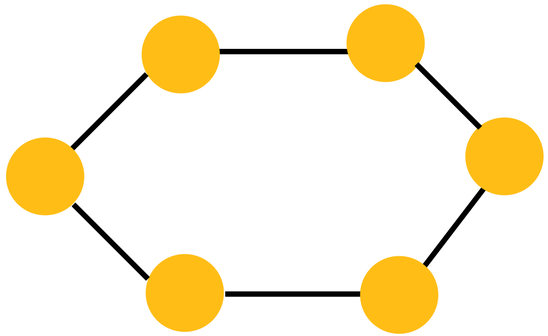
There are **problems** in initialising nodes with such features:

- We assign colours to nodes from a fixed class of colours which do not necessarily have a meaning.
- We change the meaning of the given node features — potentially losing valuable information.
- These features are deterministic and it is hard to generalise over these structures.

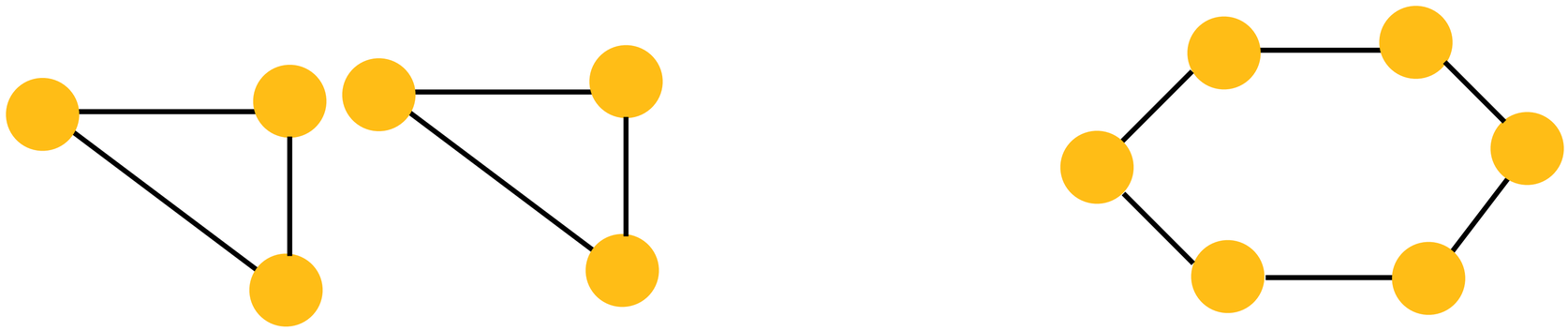
MPNNs with Random Features

Random Node Features and Coloured Graphs



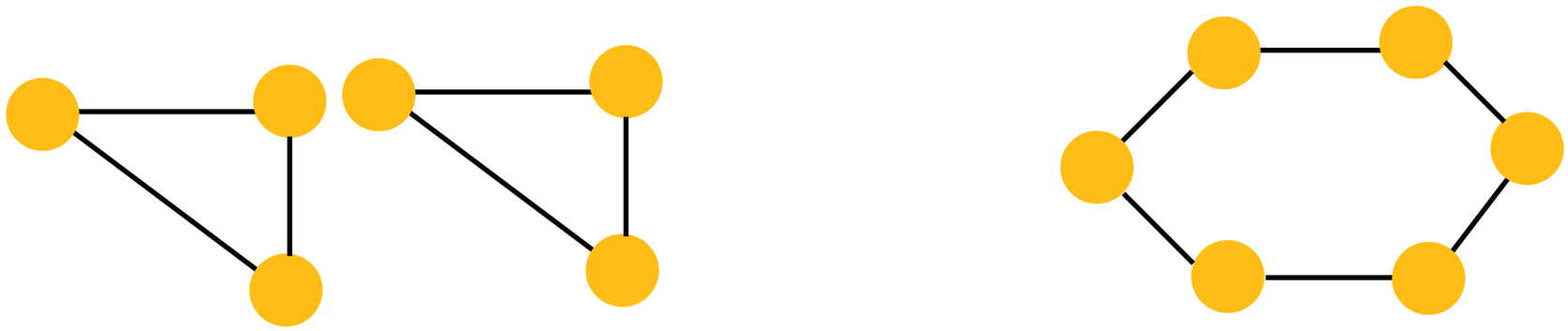


Random Node Features and Coloured Graphs



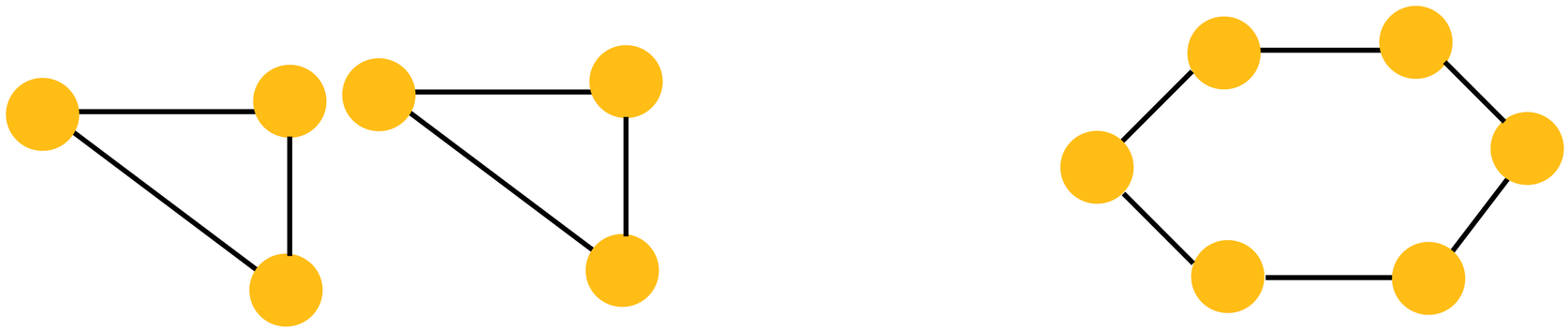
- In this example, we initialise an MPNN with, e.g., node degrees, which results in identical features for the nodes, and they cannot be distinguished.

Random Node Features and Coloured Graphs



- In this example, we initialise an MPNN with, e.g., node degrees, which results in identical features for the nodes, and they cannot be distinguished.
- If we initialise an MPNN with different colours, this results in different features for the nodes, and they can be distinguished.

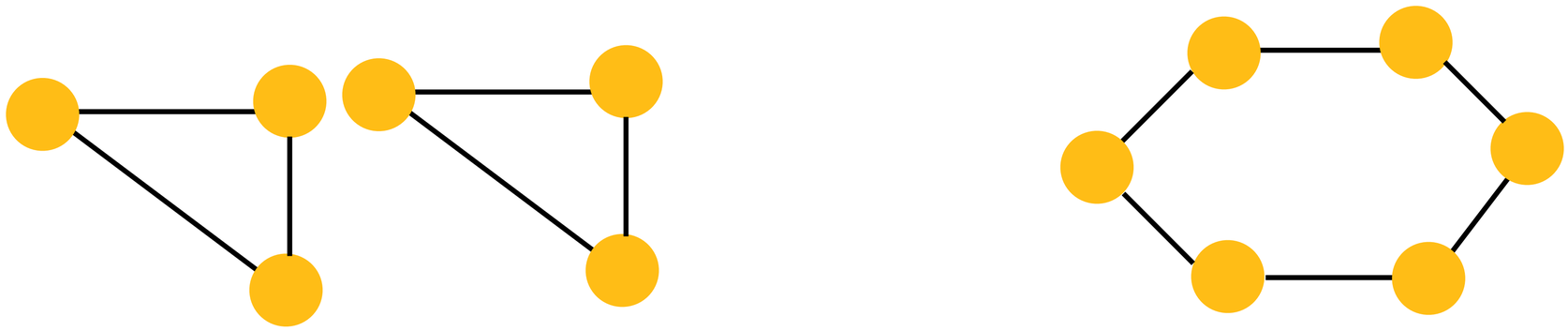
Random Node Features and Coloured Graphs



- In this example, we initialise an MPNN with, e.g., node degrees, which results in identical features for the nodes, and they cannot be distinguished.
- If we initialise an MPNN with different colours, this results in different features for the nodes, and they can be distinguished.

Question: What if we initialise an MPNN with random features instead?

Random Node Features and Coloured Graphs



- In this example, we initialise an MPNN with, e.g., node degrees, which results in identical features for the nodes, and they cannot be distinguished.
- If we initialise an MPNN with different colours, this results in different features for the nodes, and they can be distinguished.

Question: What if we initialise an MPNN with random features instead?

Intuition: Random features can implicitly induce a colouring, and yield a more expressive model.

MPNNs with Random Node Initialisation

MPNNs with Random Node Initialisation

MPNNs are enhanced with random node initialisation (RNI) such that the model trains and runs with (partially) randomised initial node features (Sato et al., 2020).

MPNNs with Random Node Initialisation

MPNNs are enhanced with random node initialisation (RNI) such that the model trains and runs with (partially) randomised initial node features (Sato et al., 2020).

The resulting model is called rGNNs, and more specifically, rGINs, as the base model which is extended with random features is GINs in this paper.

MPNNs with Random Node Initialisation

MPNNs are enhanced with random node initialisation (RNI) such that the model trains and runs with (partially) randomised initial node features (Sato et al., 2020).

The resulting model is called rGNNs, and more specifically, rGINs, as the base model which is extended with random features is GINs in this paper.

We will write MPNN-RNI to denote MPNNs with random node initialisation, and e.g., M-RNI to denote a specific MPNN model M extended with RNI.

MPNNs with Random Node Initialisation

MPNNs are enhanced with random node initialisation (RNI) such that the model trains and runs with (partially) randomised initial node features (Sato et al., 2020).

The resulting model is called rGNNs, and more specifically, rGINs, as the base model which is extended with random features is GINs in this paper.

We will write MPNN-RNI to denote MPNNs with random node initialisation, and e.g., M-RNI to denote a specific MPNN model M extended with RNI.

Remark: When we speak of features, it can refer to both node and edge features — In fact, edge features are very commonly used in the context of graph neural networks.

MPNNs with Random Node Initialisation

MPNNs are enhanced with random node initialisation (RNI) such that the model trains and runs with (partially) randomised initial node features (Sato et al., 2020).

The resulting model is called rGNNs, and more specifically, rGINs, as the base model which is extended with random features is GINs in this paper.

We will write MPNN-RNI to denote MPNNs with random node initialisation, and e.g., M-RNI to denote a specific MPNN model M extended with RNI.

Remark: When we speak of features, it can refer to both node and edge features — In fact, edge features are very commonly used in the context of graph neural networks.

To make the distinction clear, we speak of node features, and hence of random node initialisations here. Note that one can also consider random edge initialisations.

Substructure Detection with Random Node Initialisation

Substructure Detection with Random Node Initialisation

It has been shown that GIN-RNI models can detect characteristic sub-graphs in an input graph with high probability.

Substructure Detection with Random Node Initialisation

It has been shown that GIN-RNI models can detect characteristic sub-graphs in an input graph with high probability.

Informally, for a given class of (degree-bounded) graphs $\mathscr{G}$, and every fixed structure (G, v), where $v \in V_G$, Theorem 4.1 of (Sato et al., 2020) states that there exists a parametrisation θ for an GIN-RNI such that the resulting model can detect the structure (G, v) in the class of graph node pairs with high probability.

Substructure Detection with Random Node Initialisation

It has been shown that GIN-RNI models can detect characteristic sub-graphs in an input graph with high probability.

Informally, for a given class of (degree-bounded) graphs $\mathscr{G}$, and every fixed structure (G, v), where $v \in V_G$, Theorem 4.1 of (Sato et al., 2020) states that there exists a parametrisation θ for an GIN-RNI such that the resulting model can detect the structure (G, v) in the class of graph node pairs with high probability.

Example: If (G, v) characterises v being part of a triangle, then this theorem implies that GIN-RNI can classify the nodes w.r.t. the presence of the triangle structure.

Substructure Detection with Random Node Initialisation

It has been shown that GIN-RNI models can detect characteristic sub-graphs in an input graph with high probability.

Informally, for a given class of (degree-bounded) graphs $\mathscr{G}$, and every fixed structure (G, v), where $v \in V_G$, Theorem 4.1 of (Sato et al., 2020) states that there exists a parametrisation θ for an GIN-RNI such that the resulting model can detect the structure (G, v) in the class of graph node pairs with high probability.

Example: If (G, v) characterises v being part of a triangle, then this theorem implies that GIN-RNI can classify the nodes w.r.t. the presence of the triangle structure.

GIN-RNI models go beyond the capabilities of GINs which cannot determine the existence of a triangle.

Substructure Detection with Random Node Initialisation

It has been shown that GIN-RNI models can detect characteristic sub-graphs in an input graph with high probability.

Informally, for a given class of (degree-bounded) graphs $\mathscr{G}$, and every fixed structure (G, v), where $v \in V_G$, Theorem 4.1 of (Sato et al., 2020) states that there exists a parametrisation θ for an GIN-RNI such that the resulting model can detect the structure (G, v) in the class of graph node pairs with high probability.

Example: If (G, v) characterises v being part of a triangle, then this theorem implies that GIN-RNI can classify the nodes w.r.t. the presence of the triangle structure.

GIN-RNI models go beyond the capabilities of GINs which cannot determine the existence of a triangle.

Remark: This theorem does not imply universality, as it only asserts distinguishability w.r.t. a fixed structure. This is not the same as being able to approximate any function (which can depend on multiple, interacting structures) over this space.

Empirical Evaluation for Substructure Detection

Empirical Evaluation for Substructure Detection

These findings are empirically evaluated on two synthetic datasets (Sato et al., 2020) and we will briefly discuss one of these.

Empirical Evaluation for Substructure Detection

These findings are empirically evaluated on two synthetic datasets (Sato et al., 2020) and we will briefly discuss one of these.

Triangle: This dataset contains random 3-regular graphs for a binary node classification problem. Both training and test data contain 1000 graphs. The training graphs have 20 nodes, and test graphs have 20 nodes for the normal dataset and 100 nodes for the extrapolation dataset. A node v is positive if v has two neighbouring nodes that are adjacent to each other.

Empirical Evaluation for Substructure Detection

These findings are empirically evaluated on two synthetic datasets (Sato et al., 2020) and we will briefly discuss one of these.

Triangle: This dataset contains random 3-regular graphs for a binary node classification problem. Both training and test data contain 1000 graphs. The training graphs have 20 nodes, and test graphs have 20 nodes for the normal dataset and 100 nodes for the extrapolation dataset. A node v is positive if v has two neighbouring nodes that are adjacent to each other.

Summary of the results: The results are reported for GINs and GCNs and their respective RNI versions.

Empirical Evaluation for Substructure Detection

These findings are empirically evaluated on two synthetic datasets (Sato et al., 2020) and we will briefly discuss one of these.

Triangle: This dataset contains random 3-regular graphs for a binary node classification problem. Both training and test data contain 1000 graphs. The training graphs have 20 nodes, and test graphs have 20 nodes for the normal dataset and 100 nodes for the extrapolation dataset. A node v is positive if v has two neighbouring nodes that are adjacent to each other.

Summary of the results: The results are reported for GINs and GCNs and their respective RNI versions.

- Unsurprisingly, GINs as well as GCNs only achieve 50% accuracy on this dataset.

Empirical Evaluation for Substructure Detection

These findings are empirically evaluated on two synthetic datasets (Sato et al., 2020) and we will briefly discuss one of these.

Triangle: This dataset contains random 3-regular graphs for a binary node classification problem. Both training and test data contain 1000 graphs. The training graphs have 20 nodes, and test graphs have 20 nodes for the normal dataset and 100 nodes for the extrapolation dataset. A node v is positive if v has two neighbouring nodes that are adjacent to each other.

Summary of the results: The results are reported for GINs and GCNs and their respective RNI versions.

- Unsurprisingly, GINs as well as GCNs only achieve 50% accuracy on this dataset.
- GIN-RNI achieves >90% accuracy and GCN-RNI >85% achieves accuracy on normal and extrapolation datasets.

Empirical Evaluation for Substructure Detection

These findings are empirically evaluated on two synthetic datasets (Sato et al., 2020) and we will briefly discuss one of these.

Triangle: This dataset contains random 3-regular graphs for a binary node classification problem. Both training and test data contain 1000 graphs. The training graphs have 20 nodes, and test graphs have 20 nodes for the normal dataset and 100 nodes for the extrapolation dataset. A node v is positive if v has two neighbouring nodes that are adjacent to each other.

Summary of the results: The results are reported for GINs and GCNs and their respective RNI versions.

- Unsurprisingly, GINs as well as GCNs only achieve 50% accuracy on this dataset.
- GIN-RNI achieves >90% accuracy and GCN-RNI >85% achieves accuracy on normal and extrapolation datasets.
- Since the test graphs in the extrapolation dataset have more nodes than the training graphs, this shows that MPNN-RNI models can potentially extrapolate to variable size graphs.

Empirical Evaluation for Substructure Detection

Empirical Evaluation for Substructure Detection

Real-world datasets: Other empirical results are conducted on real-world datasets which do not necessarily require more than 1-WL expressivity.

Empirical Evaluation for Substructure Detection

Real-world datasets: Other empirical results are conducted on real-world datasets which do not necessarily require more than 1-WL expressivity.

Briefly, results on real-world datasets confirm that MPNN-RNI models perform either similarly to MPNNs, or marginally improve on them using a partial randomisation.

Empirical Evaluation for Substructure Detection

Real-world datasets: Other empirical results are conducted on real-world datasets which do not necessarily require more than 1-WL expressivity.

Briefly, results on real-world datasets confirm that MPNN-RNI models perform either similarly to MPNNs, or marginally improve on them using a partial randomisation.

Inspired by distributed local algorithms (Sato et al., 2020) also give algorithmic alignment results for certain combinatorial problems that admit such local algorithms, yielding near-optimally approximate solutions with graph neural networks.

Empirical Evaluation for Substructure Detection

Real-world datasets: Other empirical results are conducted on real-world datasets which do not necessarily require more than 1-WL expressivity.

Briefly, results on real-world datasets confirm that MPNN-RNI models perform either similarly to MPNNs, or marginally improve on them using a partial randomisation.

Inspired by distributed local algorithms (Sato et al., 2020) also give algorithmic alignment results for certain combinatorial problems that admit such local algorithms, yielding near-optimally approximate solutions with graph neural networks.

Overall, this means that the expressive power of MPNN-RNI models go beyond 1-WL, but it remains open to pinpoint the exact power gained by RNI.

Empirical Evaluation for Substructure Detection

Real-world datasets: Other empirical results are conducted on real-world datasets which do not necessarily require more than 1-WL expressivity.

Briefly, results on real-world datasets confirm that MPNN-RNI models perform either similarly to MPNNs, or marginally improve on them using a partial randomisation.

Inspired by distributed local algorithms (Sato et al., 2020) also give algorithmic alignment results for certain combinatorial problems that admit such local algorithms, yielding near-optimally approximate solutions with graph neural networks.

Overall, this means that the expressive power of MPNN-RNI models go beyond 1-WL, but it remains open to pinpoint the exact power gained by RNI.

Question: What is the expressive power of MPNN-RNI models, and can these be universal?

Universality of MPNNs with Random Node Initialisation

Approximating Functions with MPNNs

Approximating Functions with MPNNs

Question: What is the expressive power of MPNN-RNI models, and can they be universal?

Approximating Functions with MPNNs

Question: What is the expressive power of MPNN-RNI models, and can they be universal?

To make this question more concrete, let us focus on graph classification.

Approximating Functions with MPNNs

Question: What is the expressive power of MPNN-RNI models, and can they be universal?

To make this question more concrete, let us focus on graph classification.

Formally, let $\mathcal{G}_n$ be the class of all n-vertex graphs, i.e., graphs that consist of at most n vertices, and let us focus on the class of functions of the form $f : \mathcal{G}_n \mapsto \mathbb{R}$.

Approximating Functions with MPNNs

Question: What is the expressive power of MPNN-RNI models, and can they be universal?

To make this question more concrete, let us focus on graph classification.

Formally, let $\mathscr{G}_n$ be the class of all n-vertex graphs, i.e., graphs that consist of at most n vertices, and let us focus on the class of functions of the form $f : \mathscr{G}_n \mapsto \mathbb{R}$.

An MPNN-RNI can be viewed as computing random functions: We say that a randomised function $\mathscr{F}$ that associates with every graph $G \in \mathscr{G}_n$ a random variable $\mathscr{F}(G)$ is an (ϵ, δ)-approximation of f if for all $G \in \mathscr{G}_n$:

$$P(\,|\, f(G) - \mathscr{F}(G) \,|\, \leq \epsilon) \geq 1 - \delta\,.$$

If $\mathscr{F}$ is computed by an MPNN-RNI M, we say that M (ϵ, δ)-approximates f.

Approximating Functions with MPNNs

Question: What is the expressive power of MPNN-RNI models, and can they be universal?

To make this question more concrete, let us focus on graph classification.

Formally, let $\mathscr{G}_n$ be the class of all n-vertex graphs, i.e., graphs that consist of at most n vertices, and let us focus on the class of functions of the form $f : \mathscr{G}_n \mapsto \mathbb{R}$.

An MPNN-RNI can be viewed as computing random functions: We say that a randomised function $\mathscr{F}$ that associates with every graph $G \in \mathscr{G}_n$ a random variable $\mathscr{F}(G)$ is an (ϵ, δ)-approximation of f if for all $G \in \mathscr{G}_n$:

$$P(\mid f(G) - \mathscr{F}(G) \mid \leq \epsilon) \geq 1 - \delta\,.$$

If $\mathscr{F}$ is computed by an MPNN-RNI M, we say that M (ϵ, δ)-approximates f.

Given these, we can pose the following concrete question.

Question: Can MPNN-RNI models approximate all functions $f : \mathscr{G}_n \mapsto \mathbb{R}$?

A Universality Result

A Universality Result

It has been recently shown that MPNN-RNI models are universal, as stated in the following theorem.

A Universality Result

It has been recently shown that MPNN-RNI models are universal, as stated in the following theorem.

Theorem (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{R}$ be an invariant function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

A Universality Result

It has been recently shown that MPNN-RNI models are universal, as stated in the following theorem.

Theorem (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{R}$ be an invariant function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

This result relies on the result given for the special case of Boolean functions.

A Universality Result

It has been recently shown that MPNN-RNI models are universal, as stated in the following theorem.

Theorem (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{R}$ be an invariant function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

This result relies on the result given for the special case of Boolean functions.

Lemma (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{B}$ be an invariant Boolean function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

A Universality Result

It has been recently shown that MPNN-RNI models are universal, as stated in the following theorem.

Theorem (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{R}$ be an invariant function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

This result relies on the result given for the special case of Boolean functions.

Lemma (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{B}$ be an invariant Boolean function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

Once this result is obtained, it is not hard to lift this to the real domain and to conclude the theorem:

A Universality Result

It has been recently shown that MPNN-RNI models are universal, as stated in the following theorem.

Theorem (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{R}$ be an invariant function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

This result relies on the result given for the special case of Boolean functions.

Lemma (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{B}$ be an invariant Boolean function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

Once this result is obtained, it is not hard to lift this to the real domain and to conclude the theorem:

- Since $\mathcal{G}_n$ is finite, the range $Y = \{y_1, \ldots, y_s\}$ of the invariant function $f : \mathcal{G}_n \mapsto \mathbb{R}$ is finite.

A Universality Result

It has been recently shown that MPNN-RNI models are universal, as stated in the following theorem.

Theorem (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{R}$ be an invariant function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

This result relies on the result given for the special case of Boolean functions.

Lemma (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{B}$ be an invariant Boolean function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

Once this result is obtained, it is not hard to lift this to the real domain and to conclude the theorem:

- Since $\mathcal{G}_n$ is finite, the range $Y = \{y_1, \ldots, y_s\}$ of the invariant function $f : \mathcal{G}_n \mapsto \mathbb{R}$ is finite.
- We know that we can approximate any Boolean function $g : \mathcal{G}_n \mapsto \mathbb{B}$, by the lemma above.

A Universality Result

It has been recently shown that MPNN-RNI models are universal, as stated in the following theorem.

Theorem (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{R}$ be an invariant function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

This result relies on the result given for the special case of Boolean functions.

Lemma (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathcal{G}_n \mapsto \mathbb{B}$ be an invariant Boolean function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

Once this result is obtained, it is not hard to lift this to the real domain and to conclude the theorem:

- Since $\mathcal{G}_n$ is finite, the range $Y = \{y_1, \ldots, y_s\}$ of the invariant function $f : \mathcal{G}_n \mapsto \mathbb{R}$ is finite.
- We know that we can approximate any Boolean function $g : \mathcal{G}_n \mapsto \mathbb{B}$, by the lemma above.
- To approximate $f : \mathcal{G}_n \mapsto \mathbb{R}$, we can define a function g combining the Boolean functions $g_1, \ldots, g_s$ s.t.:

$$g_i(G) \mapsto 1, \text{ if } f(G) \mapsto y_i, \text{ and } g_i(G) \mapsto 0, \text{ otherwise.}$$

Logical Characterisation of MPNNs

Logical Characterisation of MPNNs

It is therefore sufficient to show the approximability of Boolean functions. The proof builds on the following result given for MPNNs (with global readout):

Logical Characterisation of MPNNs

It is therefore sufficient to show the approximability of Boolean functions. The proof builds on the following result given for MPNNs (with global readout):

Theorem (Barcelo et al., 2020). Each C^2 classifier can be captured by an MPNN.

Logical Characterisation of MPNNs

It is therefore sufficient to show the approximability of Boolean functions. The proof builds on the following result given for MPNNs (with global readout):

Theorem (Barcelo et al., 2020). Each C^2 classifier can be captured by an MPNN.

This result is stated for node classification and focuses on formulas with one free variable $\Phi(x)$: A graph G satisfies the property $\Phi(v)$ for a node $v \in V_G$ if $G \models \Phi(v)$.

Logical Characterisation of MPNNs

It is therefore sufficient to show the approximability of Boolean functions. The proof builds on the following result given for MPNNs (with global readout):

Theorem (Barcelo et al., 2020). Each C^2 classifier can be captured by an MPNN.

This result is stated for node classification and focuses on formulas with one free variable $\Phi(x)$: A graph G satisfies the property $\Phi(v)$ for a node $v \in V_G$ if $G \models \Phi(v)$.

This result also applies to graph classification, since we can simply pool the results of each $\Phi(v)$ for all nodes $v \in V_G$, and use this to classify the graph.

Logical Characterisation of MPNNs

It is therefore sufficient to show the approximability of Boolean functions. The proof builds on the following result given for MPNNs (with global readout):

Theorem (Barcelo et al., 2020). Each C^2 classifier can be captured by an MPNN.

This result is stated for node classification and focuses on formulas with one free variable $\Phi(x)$: A graph G satisfies the property $\Phi(v)$ for a node $v \in V_G$ if $G \models \Phi(v)$.

This result also applies to graph classification, since we can simply pool the results of each $\Phi(v)$ for all nodes $v \in V_G$, and use this to classify the graph.

A sentence Φ in C^2 expresses a graph property (i.e., there exists a triangle), and so it can be viewed as a Boolean function classifying the graphs with respect to this property.

Logical Characterisation of MPNNs

It is therefore sufficient to show the approximability of Boolean functions. The proof builds on the following result given for MPNNs (with global readout):

Theorem (Barcelo et al., 2020). Each C^2 classifier can be captured by an MPNN.

This result is stated for node classification and focuses on formulas with one free variable $\Phi(x)$: A graph G satisfies the property $\Phi(v)$ for a node $v \in V_G$ if $G \models \Phi(v)$.

This result also applies to graph classification, since we can simply pool the results of each $\Phi(v)$ for all nodes $v \in V_G$, and use this to classify the graph.

A sentence Φ in C^2 expresses a graph property (i.e., there exists a triangle), and so it can be viewed as a Boolean function classifying the graphs with respect to this property.

That is, the graph G satisfies the property specified by Φ if $G \models \Phi$, and we can simply denote this by $\Phi(G)$, viewing $\Phi : V_G \mapsto \mathbb{B}$.

Universality through Logical Characterisation

Universality through Logical Characterisation

Putting these together, and by making the error parameter explicit, we can restate this result as:

Universality through Logical Characterisation

Putting these together, and by making the error parameter explicit, we can restate this result as:

Theorem (Barcelo et al., 2020). For every C^2 sentence Φ and every $\epsilon > 0$ there is an MPNN that ϵ -approximates the Boolean function Φ.

Universality through Logical Characterisation

Putting these together, and by making the error parameter explicit, we can restate this result as:

Theorem (Barcelo et al., 2020). For every C^2 sentence Φ and every $\epsilon > 0$ there is an MPNN that ϵ -approximates the Boolean function Φ.

Remark: This result is stated for deterministic MPNNs, so the confidence parameter δ simply equal to 0 in this case and it is dropped.

Universality through Logical Characterisation

Putting these together, and by making the error parameter explicit, we can restate this result as:

Theorem (Barcelo et al., 2020). For every C^2 sentence Φ and every $\epsilon > 0$ there is an MPNN that ϵ -approximates the Boolean function Φ.

Remark: This result is stated for deterministic MPNNs, so the confidence parameter δ simply equal to 0 in this case and it is dropped.

This is useful to show the following result:

Universality through Logical Characterisation

Putting these together, and by making the error parameter explicit, we can restate this result as:

Theorem (Barcelo et al., 2020). For every C^2 sentence Φ and every $\epsilon > 0$ there is an MPNN that ϵ -approximates the Boolean function Φ.

Remark: This result is stated for deterministic MPNNs, so the confidence parameter δ simply equal to 0 in this case and it is dropped.

This is useful to show the following result:

Lemma (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathscr{G}_n \mapsto \mathbb{B}$ be an invariant Boolean function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

Universality through Logical Characterisation

Putting these together, and by making the error parameter explicit, we can restate this result as:

Theorem (Barcelo et al., 2020). For every C^2 sentence Φ and every $\epsilon > 0$ there is an MPNN that ϵ -approximates the Boolean function Φ.

Remark: This result is stated for deterministic MPNNs, so the confidence parameter δ simply equal to 0 in this case and it is dropped.

This is useful to show the following result:

Lemma (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathscr{G}_n \mapsto \mathbb{B}$ be an invariant Boolean function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

It may appear somewhat surprising, and even counter-intuitive, that randomly initialising node features would deliver such a gain in expressiveness.

Universality through Logical Characterisation

Putting these together, and by making the error parameter explicit, we can restate this result as:

Theorem (Barcelo et al., 2020). For every C^2 sentence Φ and every $\epsilon > 0$ there is an MPNN that ϵ -approximates the Boolean function Φ.

Remark: This result is stated for deterministic MPNNs, so the confidence parameter δ simply equal to 0 in this case and it is dropped.

This is useful to show the following result:

Lemma (Abboud et al., 2020). Let $n \geq 1$, and let $f : \mathscr{G}_n \mapsto \mathbb{B}$ be an invariant Boolean function. Then, for all $\delta > 0$, there is an MPNN-RNI that (ϵ, δ)-approximates f.

It may appear somewhat surprising, and even counter-intuitive, that randomly initialising node features would deliver such a gain in expressiveness.

The main connection comes from the fact that 1-WL can distinguish all ordered structures/graphs and so there is a C^2 sentence that can uniquely (up to isomorphism) identify an ordered graph.

Universality through Logical Characterisation

Universality through Logical Characterisation

Coloured graphs: We say a graph is ordered if there is a total order on the nodes of the graph. Observe that colouring the nodes of a graph with distinct colours induces a total order on the nodes of a graph. We can therefore refer to coloured graphs instead, where each node is assigned colours from a finite set of colours.

Universality through Logical Characterisation

Coloured graphs: We say a graph is ordered if there is a total order on the nodes of the graph. Observe that colouring the nodes of a graph with distinct colours induces a total order on the nodes of a graph. We can therefore refer to coloured graphs instead, where each node is assigned colours from a finite set of colours.

Individualised graphs: A coloured graph G is individualised if for any two distinct vertices $v, w \in V_G$ the sets $\pi(v), \pi(w)$ of colours they have are distinct. We say that a sentence ψ identifies a coloured graph G if for all coloured graphs H we have $H \models \psi$ if and only if H is isomorphic to G.

Universality through Logical Characterisation

Coloured graphs: We say a graph is ordered if there is a total order on the nodes of the graph. Observe that colouring the nodes of a graph with distinct colours induces a total order on the nodes of a graph. We can therefore refer to coloured graphs instead, where each node is assigned colours from a finite set of colours.

Individualised graphs: A coloured graph G is individualised if for any two distinct vertices $v, w \in V_G$ the sets $\pi(v), \pi(w)$ of colours they have are distinct. We say that a sentence ψ identifies a coloured graph G if for all coloured graphs H we have $H \models \psi$ if and only if H is isomorphic to G.

- If we have a sentence ψ that can identify a coloured graph uniquely up to isomorphism, then we have the power of distinguishing all such graphs.

Universality through Logical Characterisation

Coloured graphs: We say a graph is ordered if there is a total order on the nodes of the graph. Observe that colouring the nodes of a graph with distinct colours induces a total order on the nodes of a graph. We can therefore refer to coloured graphs instead, where each node is assigned colours from a finite set of colours.

Individualised graphs: A coloured graph G is individualised if for any two distinct vertices $v, w \in V_G$ the sets $\pi(v), \pi(w)$ of colours they have are distinct. We say that a sentence ψ identifies a coloured graph G if for all coloured graphs H we have $H \models \psi$ if and only if H is isomorphic to G.

- If we have a sentence ψ that can identify a coloured graph uniquely up to isomorphism, then we have the power of distinguishing all such graphs.
- If, furthermore, all sentences that can identify the class of coloured graphs are in C^2, then we can leverage the result of (Barcelo et al., 2020) to claim that MPNNs can identify the class of coloured graphs.

Universality through Logical Characterisation

Coloured graphs: We say a graph is ordered if there is a total order on the nodes of the graph. Observe that colouring the nodes of a graph with distinct colours induces a total order on the nodes of a graph. We can therefore refer to coloured graphs instead, where each node is assigned colours from a finite set of colours.

Individualised graphs: A coloured graph G is individualised if for any two distinct vertices $v, w \in V_G$ the sets $\pi(v), \pi(w)$ of colours they have are distinct. We say that a sentence ψ identifies a coloured graph G if for all coloured graphs H we have $H \models \psi$ if and only if H is isomorphic to G.

- If we have a sentence ψ that can identify a coloured graph uniquely up to isomorphism, then we have the power of distinguishing all such graphs.
- If, furthermore, all sentences that can identify the class of coloured graphs are in C^2, then we can leverage the result of (Barcelo et al., 2020) to claim that MPNNs can identify the class of coloured graphs.

Problem: The input to MPNNs is not individualised graphs!

Universality through Logical Characterisation

Coloured graphs: We say a graph is ordered if there is a total order on the nodes of the graph. Observe that colouring the nodes of a graph with distinct colours induces a total order on the nodes of a graph. We can therefore refer to coloured graphs instead, where each node is assigned colours from a finite set of colours.

Individualised graphs: A coloured graph G is individualised if for any two distinct vertices $v, w \in V_G$ the sets $\pi(v), \pi(w)$ of colours they have are distinct. We say that a sentence ψ identifies a coloured graph G if for all coloured graphs H we have $H \models \psi$ if and only if H is isomorphic to G.

- If we have a sentence ψ that can identify a coloured graph uniquely up to isomorphism, then we have the power of distinguishing all such graphs.
- If, furthermore, all sentences that can identify the class of coloured graphs are in C^2, then we can leverage the result of (Barcelo et al., 2020) to claim that MPNNs can identify the class of coloured graphs.

Problem: The input to MPNNs is not individualised graphs!

Randomisation: RNI yields individualised graphs from input graphs with high probability!

Universality through Logical Characterisation

Universality through Logical Characterisation

Given these, the roadmap of the proof is then as follows:

Universality through Logical Characterisation

Given these, the roadmap of the proof is then as follows:

1. Establish that for every individualised coloured graph G there is a C^2-sentence ψ that identifies G, and let us call these graph sentences.

Universality through Logical Characterisation

Given these, the roadmap of the proof is then as follows:

1. Establish that for every individualised coloured graph G there is a C^2-sentence ψ that identifies G, and let us call these graph sentences.

2. Extend this to Boolean functions over sets of individualised graphs, by showing that these functions can also be represented by a C^2 sentence, namely the disjunction of all constituent graph sentences.

Universality through Logical Characterisation

Given these, the roadmap of the proof is then as follows:

1. Establish that for every individualised coloured graph G there is a C^2-sentence ψ that identifies G, and let us call these graph sentences.

2. Extend this to Boolean functions over sets of individualised graphs, by showing that these functions can also be represented by a C^2 sentence, namely the disjunction of all constituent graph sentences.

3. Following this, provide a construction based on MPNN-RNI:

Universality through Logical Characterisation

Given these, the roadmap of the proof is then as follows:

1. Establish that for every individualised coloured graph G there is a C^2-sentence ψ that identifies G, and let us call these graph sentences.

2. Extend this to Boolean functions over sets of individualised graphs, by showing that these functions can also be represented by a C^2 sentence, namely the disjunction of all constituent graph sentences.

3. Following this, provide a construction based on MPNN-RNI:

    - Colours corresponds to node features vectors in MPNN-RNI, initialised randomly, and based on these, show that, with high probability, RNI makes the input graphs individualised.

Universality through Logical Characterisation

Given these, the roadmap of the proof is then as follows:

1. Establish that for every individualised coloured graph G there is a C^2-sentence ψ that identifies G, and let us call these graph sentences.

2. Extend this to Boolean functions over sets of individualised graphs, by showing that these functions can also be represented by a C^2 sentence, namely the disjunction of all constituent graph sentences.

3. Following this, provide a construction based on MPNN-RNI:

    - Colours corresponds to node features vectors in MPNN-RNI, initialised randomly, and based on these, show that, with high probability, RNI makes the input graphs individualised.

    - Since all such functions can be captured by a sentence in C^2, and an MPNN with a global readout can capture C^2 (Barcelo et al., 2020), MPNN-RNI can capture arbitrary Boolean functions.

Universality through Logical Characterisation

Given these, the roadmap of the proof is then as follows:

1. Establish that for every individualised coloured graph G there is a C^2-sentence ψ that identifies G, and let us call these graph sentences.

2. Extend this to Boolean functions over sets of individualised graphs, by showing that these functions can also be represented by a C^2 sentence, namely the disjunction of all constituent graph sentences.

3. Following this, provide a construction based on MPNN-RNI:

    - Colours corresponds to node features vectors in MPNN-RNI, initialised randomly, and based on these, show that, with high probability, RNI makes the input graphs individualised.

    - Since all such functions can be captured by a sentence in C^2, and an MPNN with a global readout can capture C^2 (Barcelo et al., 2020), MPNN-RNI can capture arbitrary Boolean functions.

    - Lift the result of Boolean functions to real functions as described earlier.

Universality through Logical Characterisation

Given these, the roadmap of the proof is then as follows:

1. Establish that for every individualised coloured graph G there is a C^2-sentence ψ that identifies G, and let us call these graph sentences.

2. Extend this to Boolean functions over sets of individualised graphs, by showing that these functions can also be represented by a C^2 sentence, namely the disjunction of all constituent graph sentences.

3. Following this, provide a construction based on MPNN-RNI:

    - Colours corresponds to node features vectors in MPNN-RNI, initialised randomly, and based on these, show that, with high probability, RNI makes the input graphs individualised.

    - Since all such functions can be captured by a sentence in C^2, and an MPNN with a global readout can capture C^2 (Barcelo et al., 2020), MPNN-RNI can capture arbitrary Boolean functions.

    - Lift the result of Boolean functions to real functions as described earlier.

Remark: This result holds even with partial node initialisations — even with a single randomised dimension.

Permutation Invariance

Permutation Invariance

Recall that we did not prefer embedding graphs to an MLP, due to the lack of right inductive bias, and properties such as permutation-invariance. The following question is then in place:

Permutation Invariance

Recall that we did not prefer embedding graphs to an MLP, due to the lack of right inductive bias, and properties such as permutation-invariance. The following question is then in place:

Question: Are MPNN-RNIs permutation-invariant, or, more generally, do MPNN-RNIs have the right inductive bias?

Permutation Invariance

Recall that we did not prefer embedding graphs to an MLP, due to the lack of right inductive bias, and properties such as permutation-invariance. The following question is then in place:

Question: Are MPNN-RNIs permutation-invariant, or, more generally, do MPNN-RNIs have the right inductive bias?

- On the surface, MPNN-RNI models no longer preserves the invariance of MPNNs, since the result of the computation of an MPNN-RNI not only depends on the structure (i.e., the isomorphism type) of the input graph, but also on RNI.

Permutation Invariance

Recall that we did not prefer embedding graphs to an MLP, due to the lack of right inductive bias, and properties such as permutation-invariance. The following question is then in place:

Question: Are MPNN-RNIs permutation-invariant, or, more generally, do MPNN-RNIs have the right inductive bias?

- On the surface, MPNN-RNI models no longer preserves the invariance of MPNNs, since the result of the computation of an MPNN-RNI not only depends on the structure (i.e., the isomorphism type) of the input graph, but also on RNI.
- The broader picture is, however, rather subtle: We can view such a model as computing a random variable (or as generating an output distribution), and this random variable would still be invariant.

Permutation Invariance

Recall that we did not prefer embedding graphs to an MLP, due to the lack of right inductive bias, and properties such as permutation-invariance. The following question is then in place:

Question: Are MPNN-RNIs permutation-invariant, or, more generally, do MPNN-RNIs have the right inductive bias?

- On the surface, MPNN-RNI models no longer preserves the invariance of MPNNs, since the result of the computation of an MPNN-RNI not only depends on the structure (i.e., the isomorphism type) of the input graph, but also on RNI.
- The broader picture is, however, rather subtle: We can view such a model as computing a random variable (or as generating an output distribution), and this random variable would still be invariant.
- This means that the outcome of the computation of an MPNN-RNI does still not depend on the specific representation of the input graph, which fundamentally maintains invariance.

Permutation Invariance

Permutation Invariance

Observation: MPNN-RNIs are permutation-invariant in expectation!

Permutation Invariance

Observation: MPNN-RNIs are permutation-invariant in expectation!

- Indeed, random features vary around a mean which, in expectation, will inform model predictions, and is identical across all nodes.

Permutation Invariance

Observation: MPNN-RNIs are permutation-invariant in expectation!

- Indeed, random features vary around a mean which, in expectation, will inform model predictions, and is identical across all nodes.
- However, the variability between different samples, and the variability of a random sample relative to this mean, enable graph discrimination and improve expressiveness.

Permutation Invariance

Observation: MPNN-RNIs are permutation-invariant in expectation!

- Indeed, random features vary around a mean which, in expectation, will inform model predictions, and is identical across all nodes.
- However, the variability between different samples, and the variability of a random sample relative to this mean, enable graph discrimination and improve expressiveness.
- Overall, in expectation, all samples over training and evaluation fluctuate around a unique value, preserving invariance, whereas single-sample variance achieves the improved expressiveness.

Permutation Invariance

Observation: MPNN-RNIs are permutation-invariant in expectation!

- Indeed, random features vary around a mean which, in expectation, will inform model predictions, and is identical across all nodes.
- However, the variability between different samples, and the variability of a random sample relative to this mean, enable graph discrimination and improve expressiveness.
- Overall, in expectation, all samples over training and evaluation fluctuate around a unique value, preserving invariance, whereas single-sample variance achieves the improved expressiveness.

Observe also that MPNN-RNIs still have the structural encoding of the graph, and explicit message passing, and the structure-induced bias is preserved.

Permutation Invariance

Observation: MPNN-RNIs are permutation-invariant in expectation!

- Indeed, random features vary around a mean which, in expectation, will inform model predictions, and is identical across all nodes.
- However, the variability between different samples, and the variability of a random sample relative to this mean, enable graph discrimination and improve expressiveness.
- Overall, in expectation, all samples over training and evaluation fluctuate around a unique value, preserving invariance, whereas single-sample variance achieves the improved expressiveness.

Observe also that MPNN-RNIs still have the structural encoding of the graph, and explicit message passing, and the structure-induced bias is preserved.

Together with the arguments above, MPNN-RNI models, allowing variability, are universal models, and preserve the good inductive bias of MPNNs.

Benchmarking Expressiveness Evaluation

Datasets for Expressiveness Evaluation

Datasets for Expressiveness Evaluation

Recall that existing benchmarks are unlikely to include the cases MPNNs cannot distinguish, and so it is hard to evaluate the models against this criteria.

Datasets for Expressiveness Evaluation

Recall that existing benchmarks are unlikely to include the cases MPNNs cannot distinguish, and so it is hard to evaluate the models against this criteria.

We have seen experiments conducted on randomly generated graphs with, e.g., triangles as substructures. These datasets require more expressivity, but they are still dealing with an inherently local problem.

Datasets for Expressiveness Evaluation

Recall that existing benchmarks are unlikely to include the cases MPNNs cannot distinguish, and so it is hard to evaluate the models against this criteria.

We have seen experiments conducted on randomly generated graphs with, e.g., triangles as substructures. These datasets require more expressivity, but they are still dealing with an inherently local problem.

To evaluate whether the universality result is viable practically, the dataset EXP is proposed (Abboud et al., 2020) which aims to evaluate the expressiveness of GNN models and it is based on the well-known propositional satisfiability problem.

Datasets for Expressiveness Evaluation

Recall that existing benchmarks are unlikely to include the cases MPNNs cannot distinguish, and so it is hard to evaluate the models against this criteria.

We have seen experiments conducted on randomly generated graphs with, e.g., triangles as substructures. These datasets require more expressivity, but they are still dealing with an inherently local problem.

To evaluate whether the universality result is viable practically, the dataset EXP is proposed (Abboud et al., 2020) which aims to evaluate the expressiveness of GNN models and it is based on the well-known propositional satisfiability problem.

Briefly, given a propositional formula ϕ, the satisfiability problem (SAT) is to determine whether the formula has a satisfying assignment, and it is the most prototypical NP-complete problem.

Datasets for Expressiveness Evaluation

Recall that existing benchmarks are unlikely to include the cases MPNNs cannot distinguish, and so it is hard to evaluate the models against this criteria.

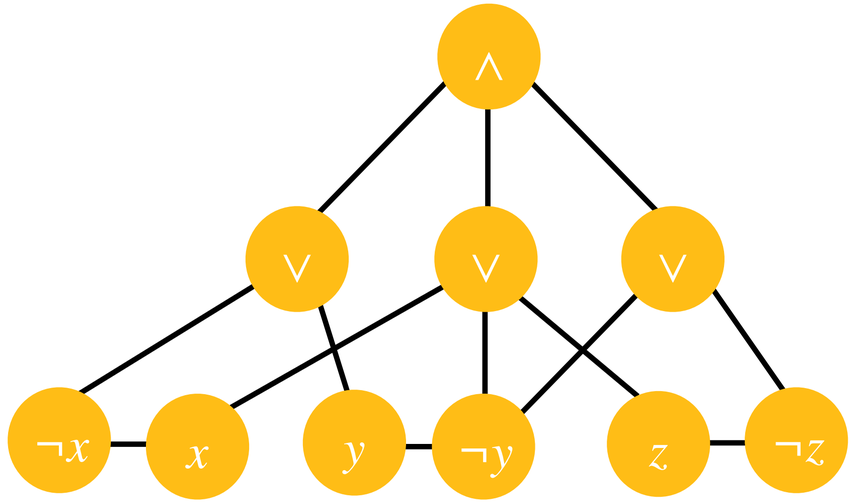
We have seen experiments conducted on randomly generated graphs with, e.g., triangles as substructures. These datasets require more expressivity, but they are still dealing with an inherently local problem.

To evaluate whether the universality result is viable practically, the dataset EXP is proposed (Abboud et al., 2020) which aims to evaluate the expressiveness of GNN models and it is based on the well-known propositional satisfiability problem.

Briefly, given a propositional formula ϕ, the satisfiability problem (SAT) is to determine whether the formula has a satisfying assignment, and it is the most prototypical NP-complete problem.

SAT is combinatorial by its nature and is not local, i.e., the satisfiability of a formula cannot be determined by local properties alone.

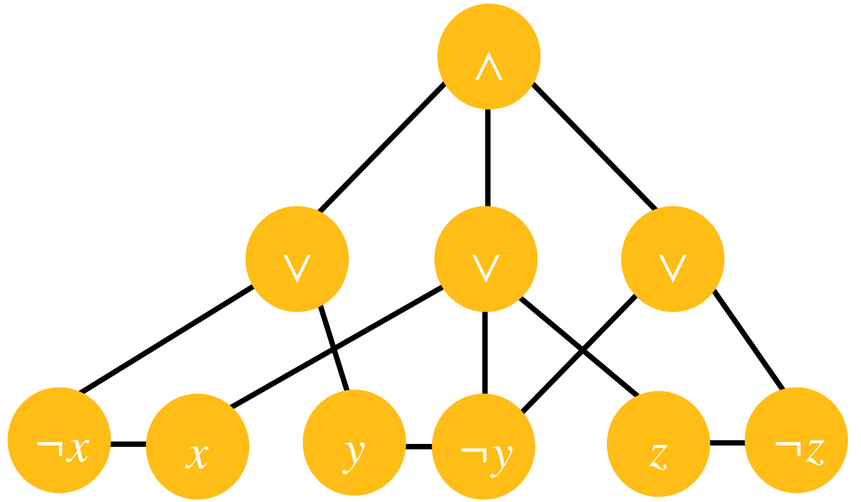
Datasets for Expressiveness Evaluation



$$\phi = (\neg x \vee y) \wedge (x \vee \neg y \vee z) \wedge (\neg y \vee \neg z)$$

Datasets for Expressiveness Evaluation

Idea: Encode each SAT instance as a graph (using well-known transformations) and formulate the satisfiability problem as a Boolean graph classification problem.

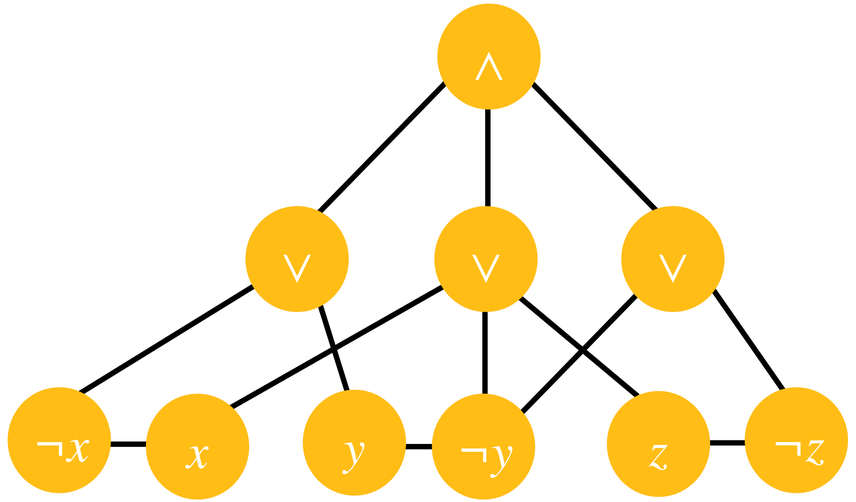


$$\phi = (\neg x \vee y) \wedge (x \vee \neg y \vee z) \wedge (\neg y \vee \neg z)$$

Datasets for Expressiveness Evaluation

Idea: Encode each SAT instance as a graph (using well-known transformations) and formulate the satisfiability problem as a Boolean graph classification problem.

The model then needs to classify graphs that represent satisfiable instances as true and graphs that represent unsatisfiable instances as false.



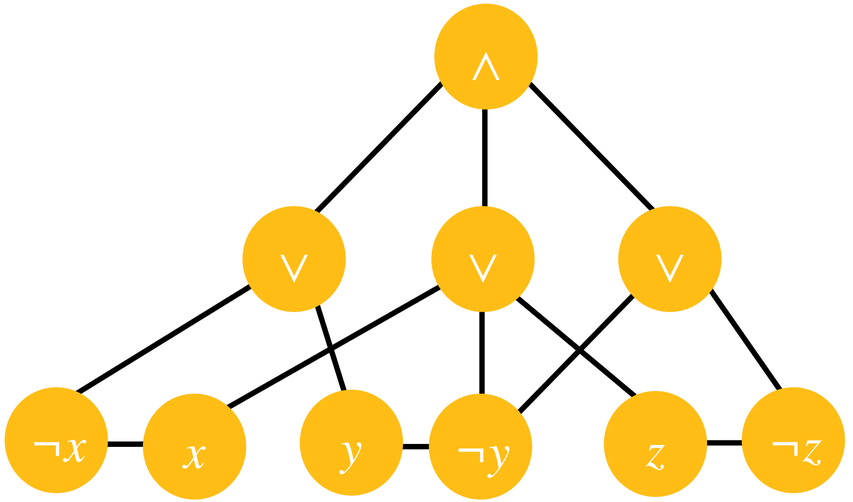
$$\phi = (\neg x \vee y) \wedge (x \vee \neg y \vee z) \wedge (\neg y \vee \neg z)$$

Datasets for Expressiveness Evaluation

Idea: Encode each SAT instance as a graph (using well-known transformations) and formulate the satisfiability problem as a Boolean graph classification problem.

The model then needs to classify graphs that represent satisfiable instances as true and graphs that represent unsatisfiable instances as false.

A simple graph encoding of a propositional formula is shown on the right hand side.



$$\phi = (\neg x \vee y) \wedge (x \vee \neg y \vee z) \wedge (\neg y \vee \neg z)$$

Datasets for Expressiveness Evaluation

Datasets for Expressiveness Evaluation

The precise details are cumbersome, since it is not sufficient to come up with satisfiable/unsatisfiable instances, but these should also be indistinguishable using 1-WL, once mapped to graph representation.

Datasets for Expressiveness Evaluation

The precise details are cumbersome, since it is not sufficient to come up with satisfiable/unsatisfiable instances, but these should also be indistinguishable using 1-WL, once mapped to graph representation.

Briefly, EXP consists of a set of graph instances $\{G_1, \ldots, G_n, H_1, \ldots, H_n\}$, such that each instance is a graph encoding of a propositional formula, and each pair (G_i, H_i) respects the following properties:

Datasets for Expressiveness Evaluation

The precise details are cumbersome, since it is not sufficient to come up with satisfiable/unsatisfiable instances, but these should also be indistinguishable using 1-WL, once mapped to graph representation.

Briefly, EXP consists of a set of graph instances $\{G_1, \ldots, G_n, H_1, \ldots, H_n\}$, such that each instance is a graph encoding of a propositional formula, and each pair (G_i, H_i) respects the following properties:

- G_i and H_i are non-isomorphic,

Datasets for Expressiveness Evaluation

The precise details are cumbersome, since it is not sufficient to come up with satisfiable/unsatisfiable instances, but these should also be indistinguishable using 1-WL, once mapped to graph representation.

Briefly, EXP consists of a set of graph instances $\{G_1, \ldots, G_n, H_1, \ldots, H_n\}$, such that each instance is a graph encoding of a propositional formula, and each pair (G_i, H_i) respects the following properties:

- G_i and H_i are non-isomorphic,
- G_i and H_i have different SAT outcomes: G_i encodes a satisfiable formula, while H_i encodes an unsatisfiable formula,

Datasets for Expressiveness Evaluation

The precise details are cumbersome, since it is not sufficient to come up with satisfiable/unsatisfiable instances, but these should also be indistinguishable using 1-WL, once mapped to graph representation.

Briefly, EXP consists of a set of graph instances $\{G_1, \ldots, G_n, H_1, \ldots, H_n\}$, such that each instance is a graph encoding of a propositional formula, and each pair (G_i, H_i) respects the following properties:

- G_i and H_i are non-isomorphic,
- G_i and H_i have different SAT outcomes: G_i encodes a satisfiable formula, while H_i encodes an unsatisfiable formula,
- G_i and H_i are 1-WL indistinguishable, so are guaranteed to be classified in the same way by standard MPNNs, and

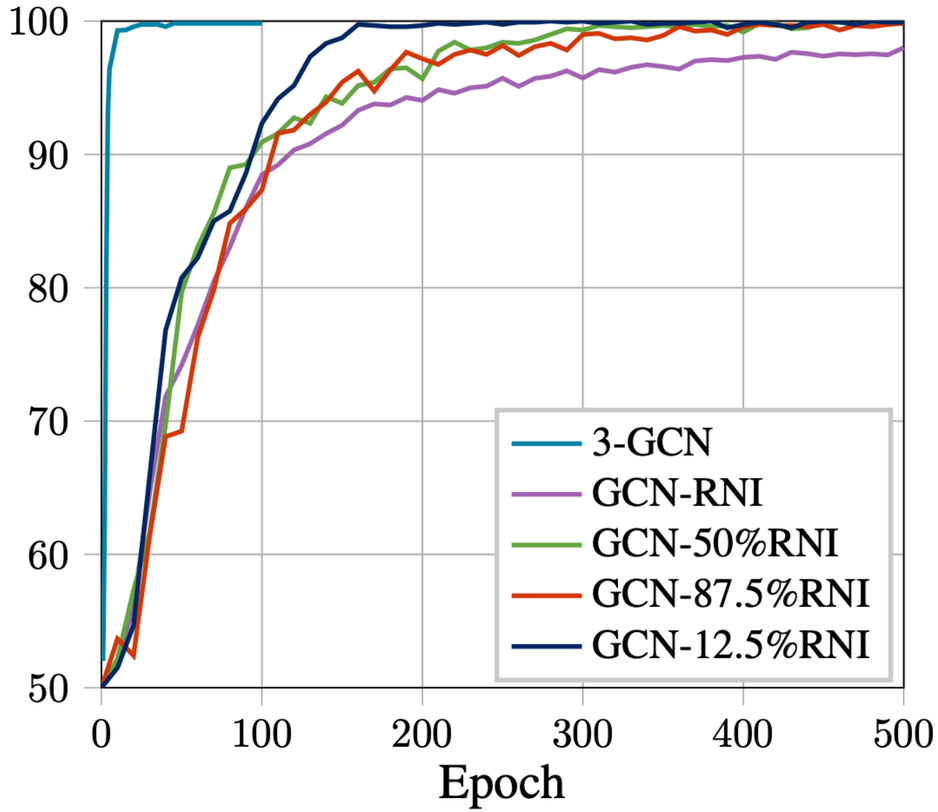
Datasets for Expressiveness Evaluation

The precise details are cumbersome, since it is not sufficient to come up with satisfiable/unsatisfiable instances, but these should also be indistinguishable using 1-WL, once mapped to graph representation.

Briefly, EXP consists of a set of graph instances $\{G_1, \ldots, G_n, H_1, \ldots, H_n\}$, such that each instance is a graph encoding of a propositional formula, and each pair (G_i, H_i) respects the following properties:

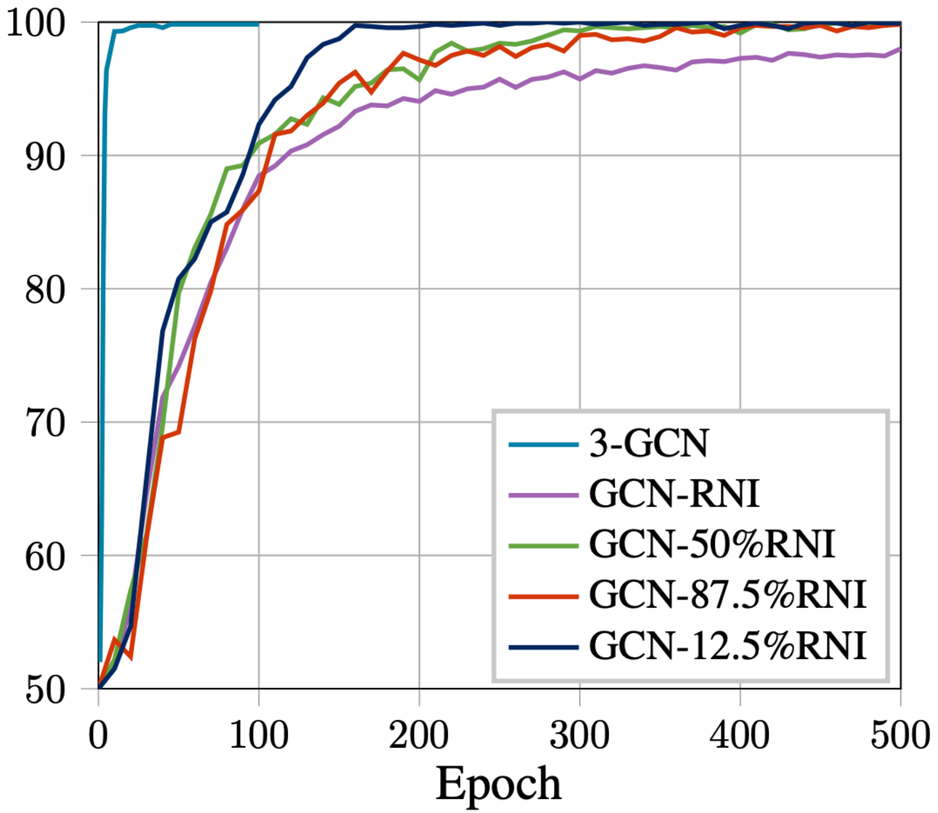
- G_i and H_i are non-isomorphic,
- G_i and H_i have different SAT outcomes: G_i encodes a satisfiable formula, while H_i encodes an unsatisfiable formula,
- G_i and H_i are 1-WL indistinguishable, so are guaranteed to be classified in the same way by standard MPNNs, and
- G_i and H_i are 2-WL distinguishable, so can be classified differently by higher-order GNNs that have 2-WL expressive power.

Random Node Initialisation is Powerful



Random Node Initialisation is Powerful

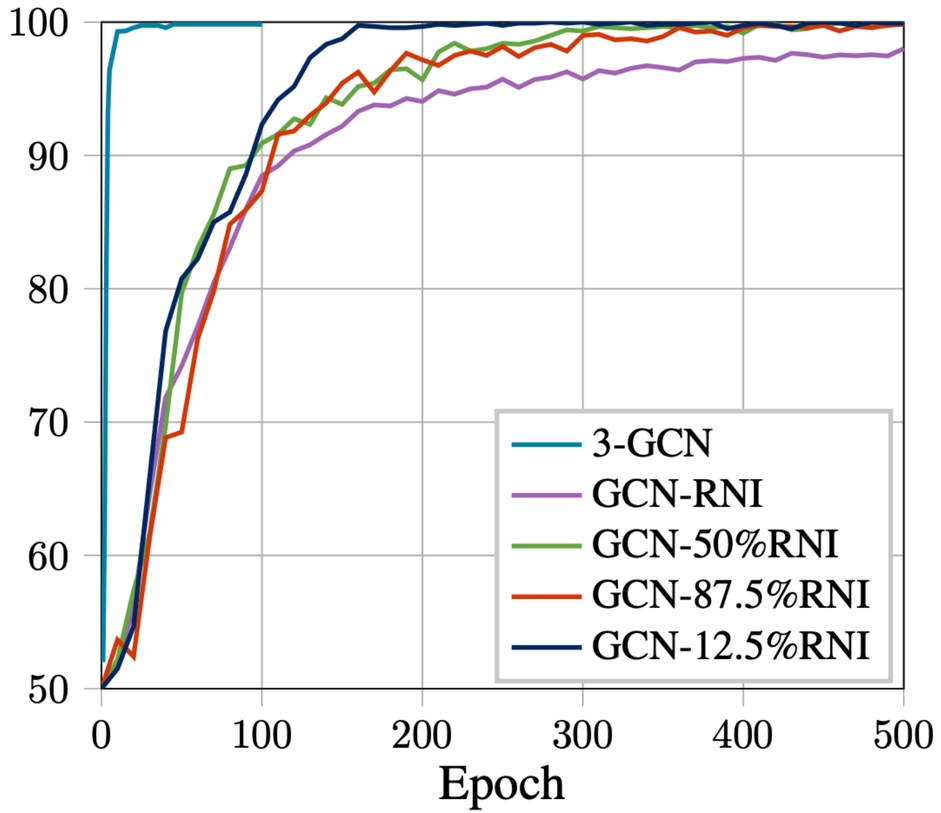
Results: Reported for GCNs and its randomised variants. GCN-$x\,\%$ RNI denotes a GCN-RNI model, where only $x\,\%$ of initial node embeddings are randomised.



Random Node Initialisation is Powerful

Results: Reported for GCNs and its randomised variants. GCN-$x\,\%$ RNI denotes a GCN-RNI model, where only $x\,\%$ of initial node embeddings are randomised.

The extreme value $100\,\%$, the intermediate value $50\,\%$, as well as near-edge cases of $87.5\,\%$ and $12.5\,\%$, are reported.

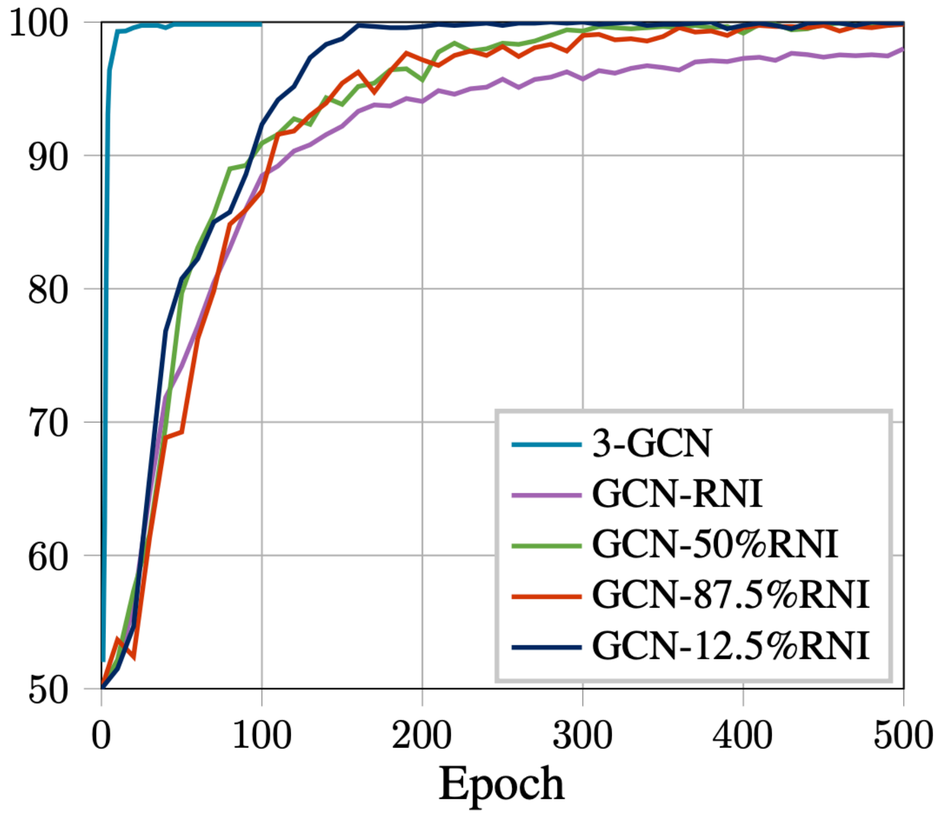


Random Node Initialisation is Powerful

Results: Reported for GCNs and its randomised variants. GCN-$x\,\%$ RNI denotes a GCN-RNI model, where only $x\,\%$ of initial node embeddings are randomised.

The extreme value $100\,\%$, the intermediate value $50\,\%$, as well as near-edge cases of $87.5\,\%$ and $12.5\,\%$, are reported.

Figure from (Abboud et al., 2020) depicts the learning curves of the respective models on the dataset EXP.



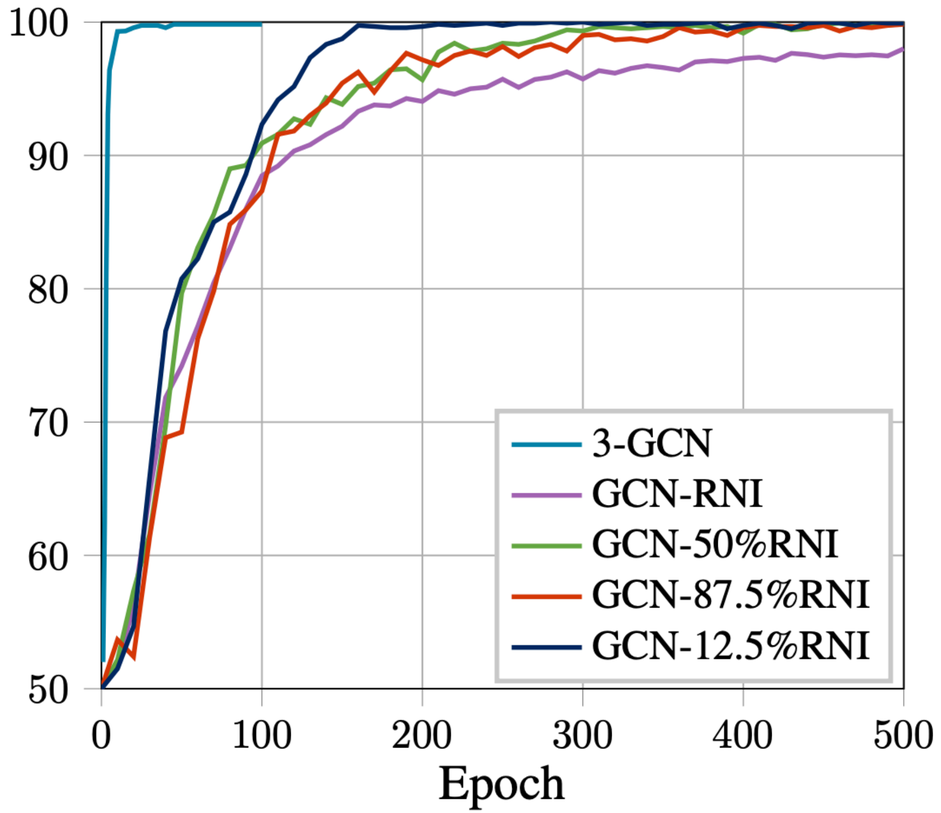
Random Node Initialisation is Powerful

Results: Reported for GCNs and its randomised variants. GCN-$x\,\%$ RNI denotes a GCN-RNI model, where only $x\,\%$ of initial node embeddings are randomised.

The extreme value $100\,\%$, the intermediate value $50\,\%$, as well as near-edge cases of $87.5\,\%$ and $12.5\,\%$, are reported.

Figure from (Abboud et al., 2020) depicts the learning curves of the respective models on the dataset EXP.

GCN model achieves exactly $50\,\%$ (omitted in the figure), and 3-GCN model achieves near-perfect accuracy very quickly.



Random Node Initialisation is Powerful

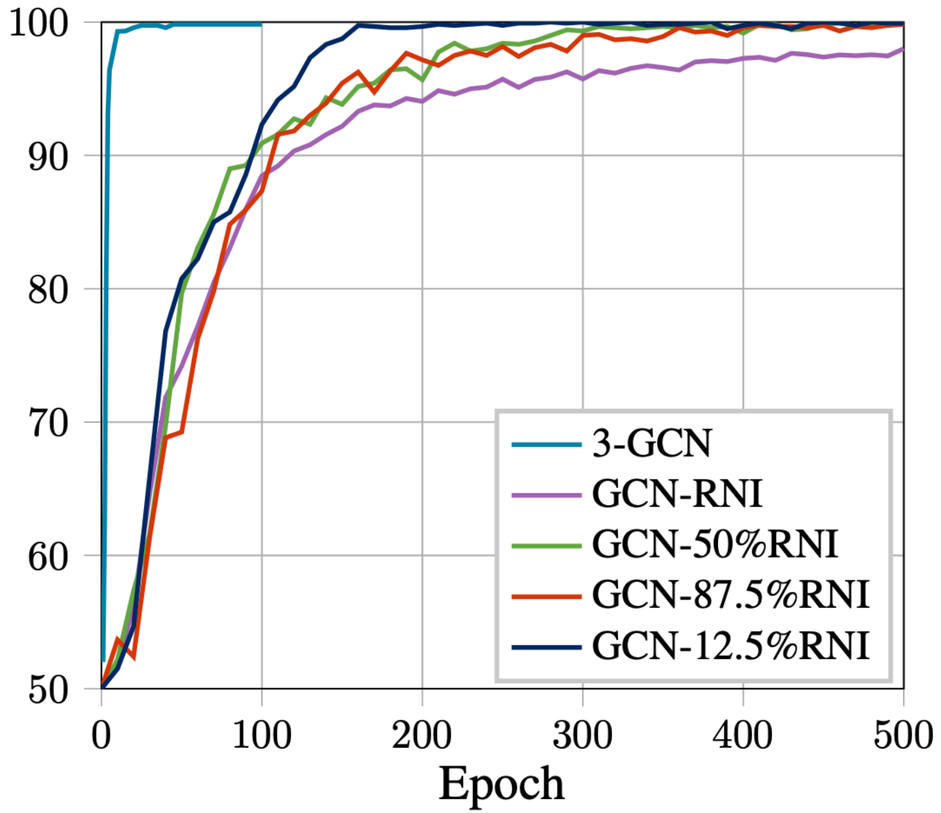
Results: Reported for GCNs and its randomised variants. GCN-$x\,\%$ RNI denotes a GCN-RNI model, where only $x\,\%$ of initial node embeddings are randomised.

The extreme value $100\,\%$, the intermediate value $50\,\%$, as well as near-edge cases of $87.5\,\%$ and $12.5\,\%$, are reported.

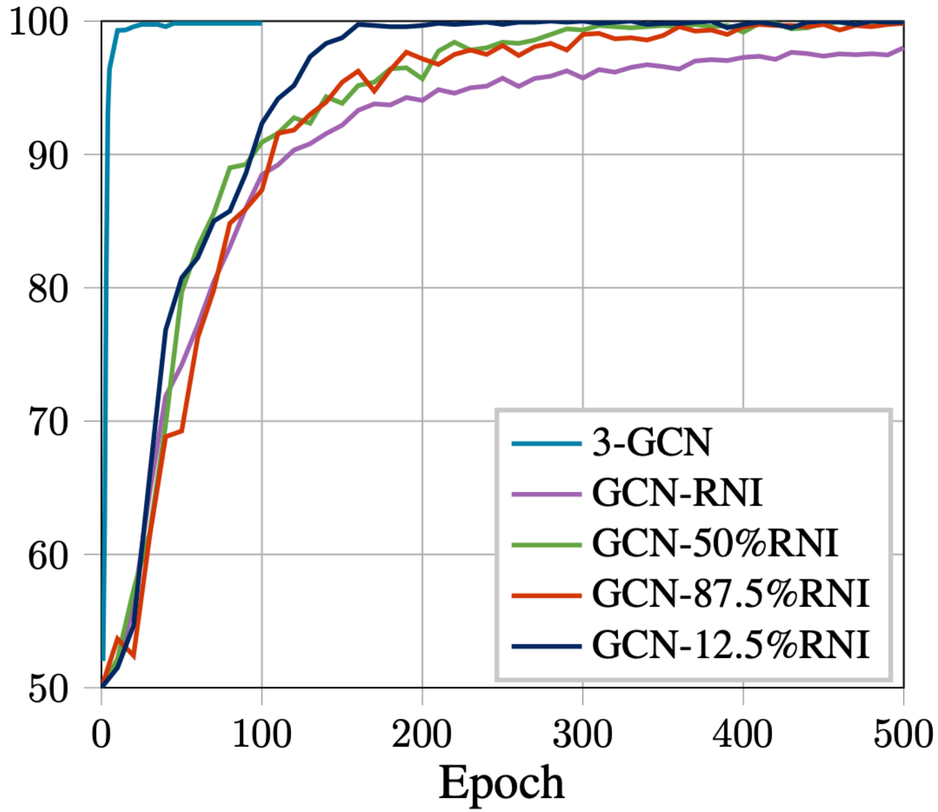
Figure from (Abboud et al., 2020) depicts the learning curves of the respective models on the dataset EXP.

GCN model achieves exactly $50\,\%$ (omitted in the figure), and 3-GCN model achieves near-perfect accuracy very quickly.

All other GCN-$x\,\%$ RNI models achieve also near-perfect accuracy!

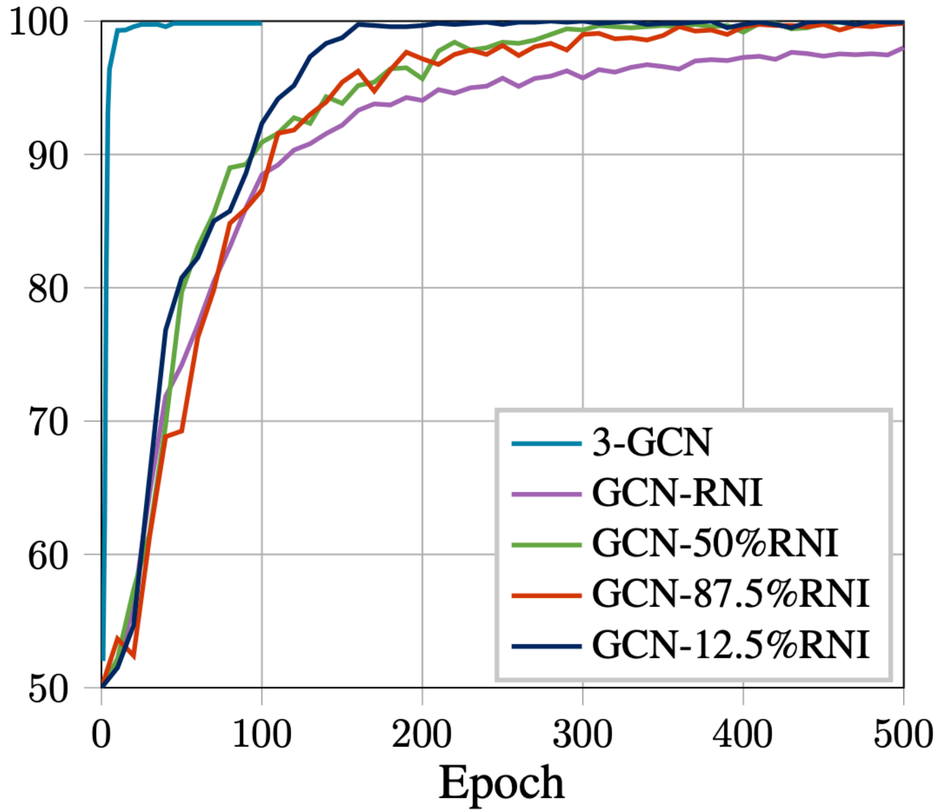


Random Node Initialisation is Powerful



Random Node Initialisation is Powerful

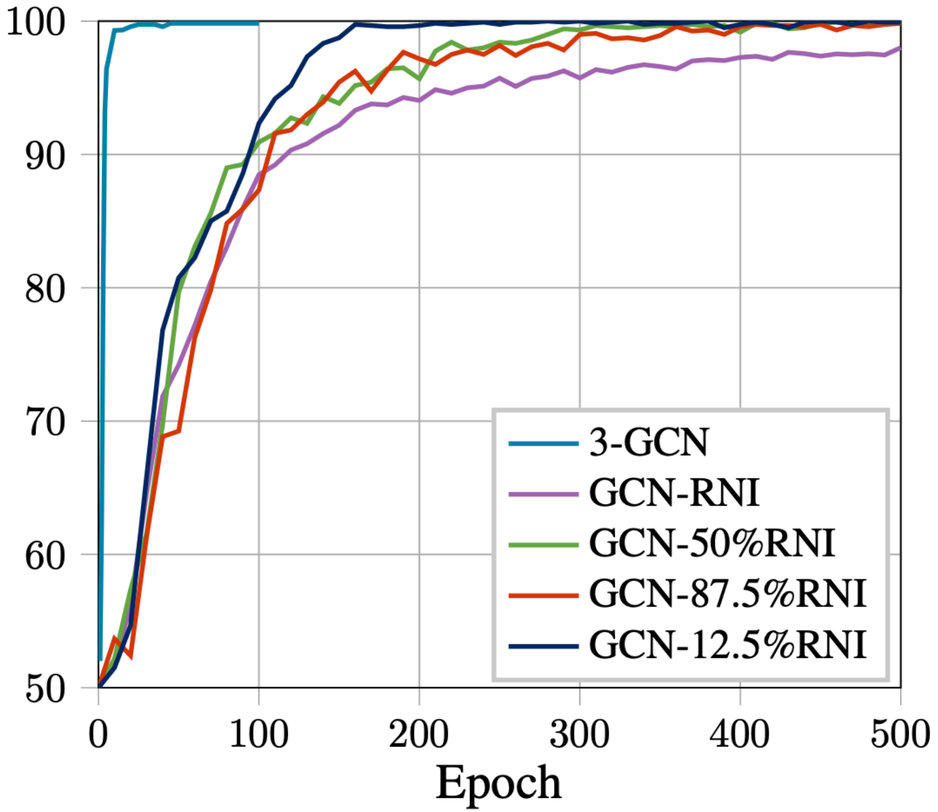
Space-efficiency: This suggest that RNI can practically improve the expressiveness of MPNNs, and make them competitive with higher-order models, despite being significantly less demanding computationally.



Random Node Initialisation is Powerful

Space-efficiency: This suggest that RNI can practically improve the expressiveness of MPNNs, and make them competitive with higher-order models, despite being significantly less demanding computationally.

MPNNs with RNI are space efficient, unlike higher-order GNNs, and combine expressiveness with efficiency in practice.

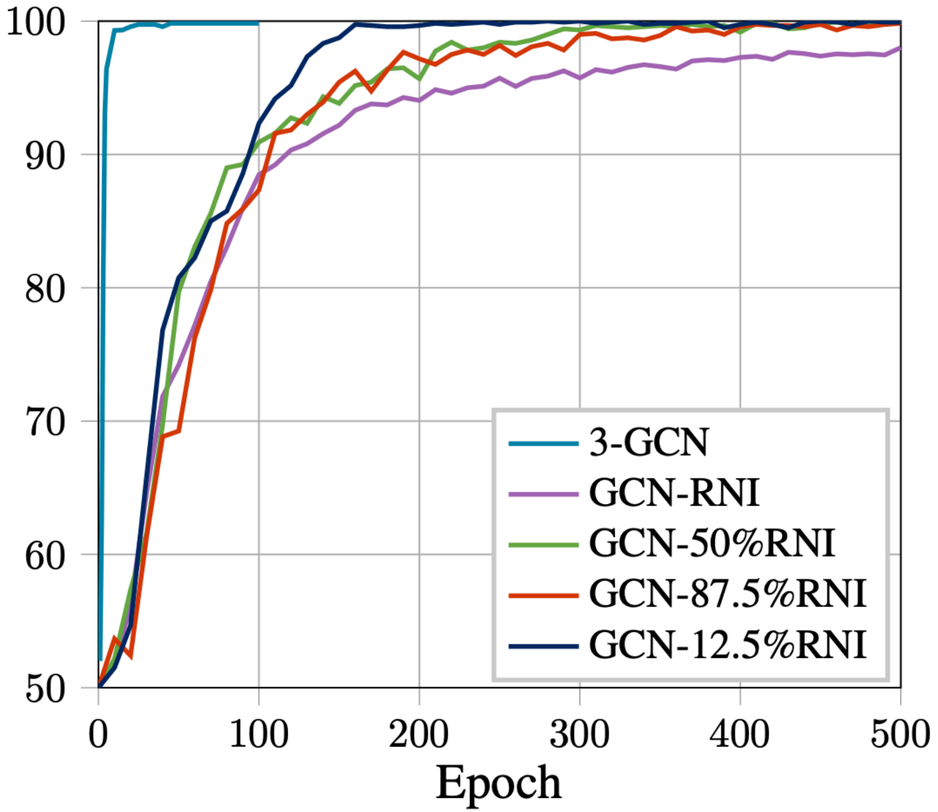


Random Node Initialisation is Powerful

Space-efficiency: This suggest that RNI can practically improve the expressiveness of MPNNs, and make them competitive with higher-order models, despite being significantly less demanding computationally.

MPNNs with RNI are space efficient, unlike higher-order GNNs, and combine expressiveness with efficiency in practice.

Indeed, for a typical EXP instance with 50 nodes, with an embedding dimensionality of 64, GCN-RNI only requires 3200 parameters, whereas 3-GCN requires 1,254,400 parameters!



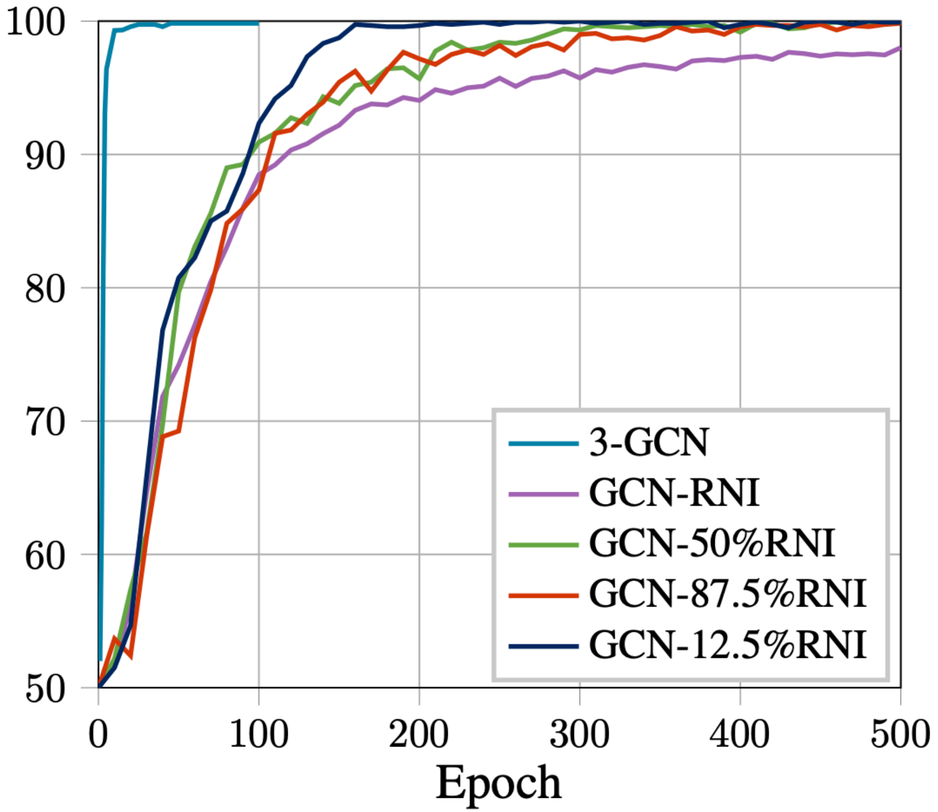
Random Node Initialisation is Powerful

Space-efficiency: This suggest that RNI can practically improve the expressiveness of MPNNs, and make them competitive with higher-order models, despite being significantly less demanding computationally.

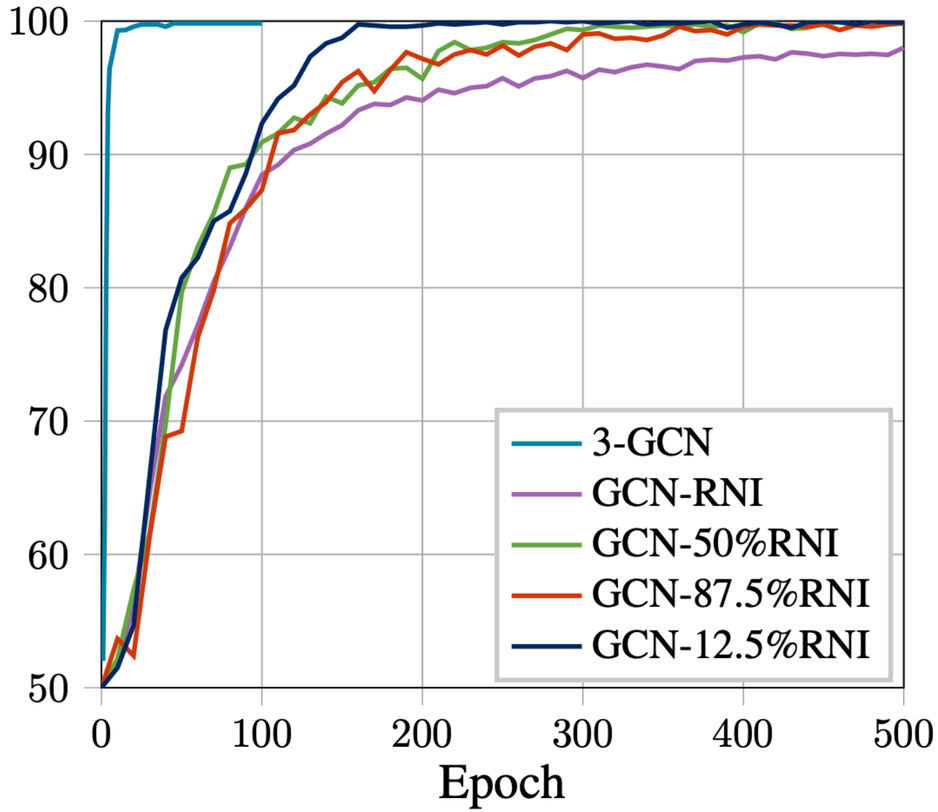
MPNNs with RNI are space efficient, unlike higher-order GNNs, and combine expressiveness with efficiency in practice.

Indeed, for a typical EXP instance with 50 nodes, with an embedding dimensionality of 64, GCN-RNI only requires 3200 parameters, whereas 3-GCN requires 1,254,400 parameters!

Somewhat surprisingly, GCN-RNI closely matches the performance of 3-GCN, and can easily scale to larger instances that are not within reach for 3-GCN.

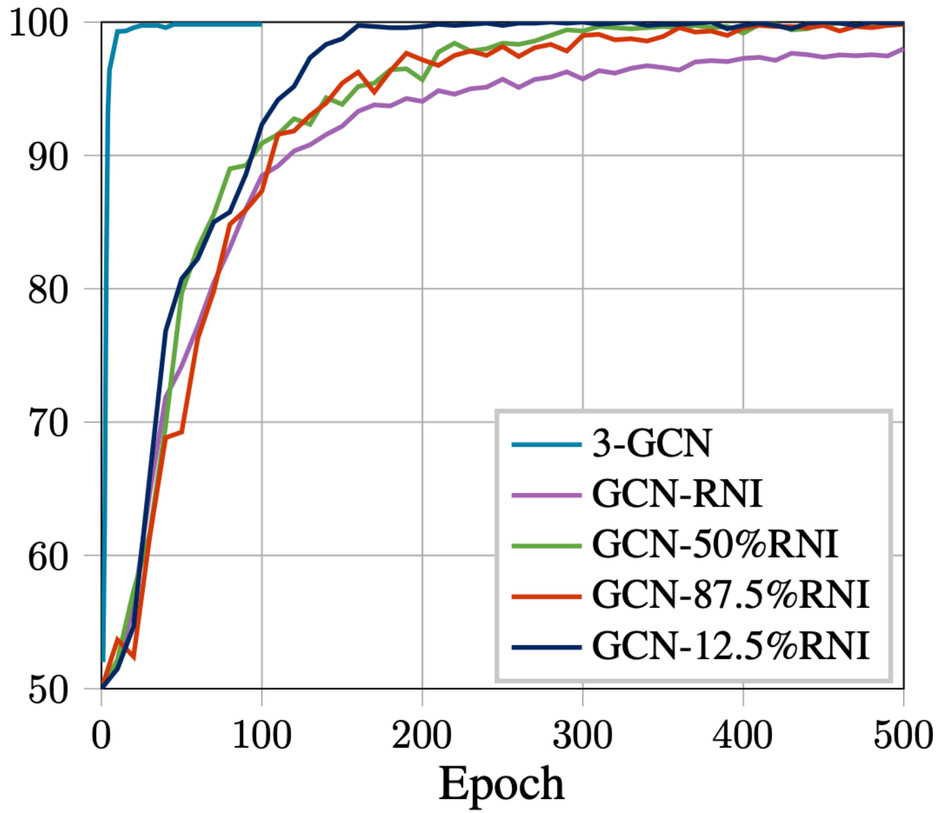


Randomised Models Converge Slower



Randomised Models Converge Slower

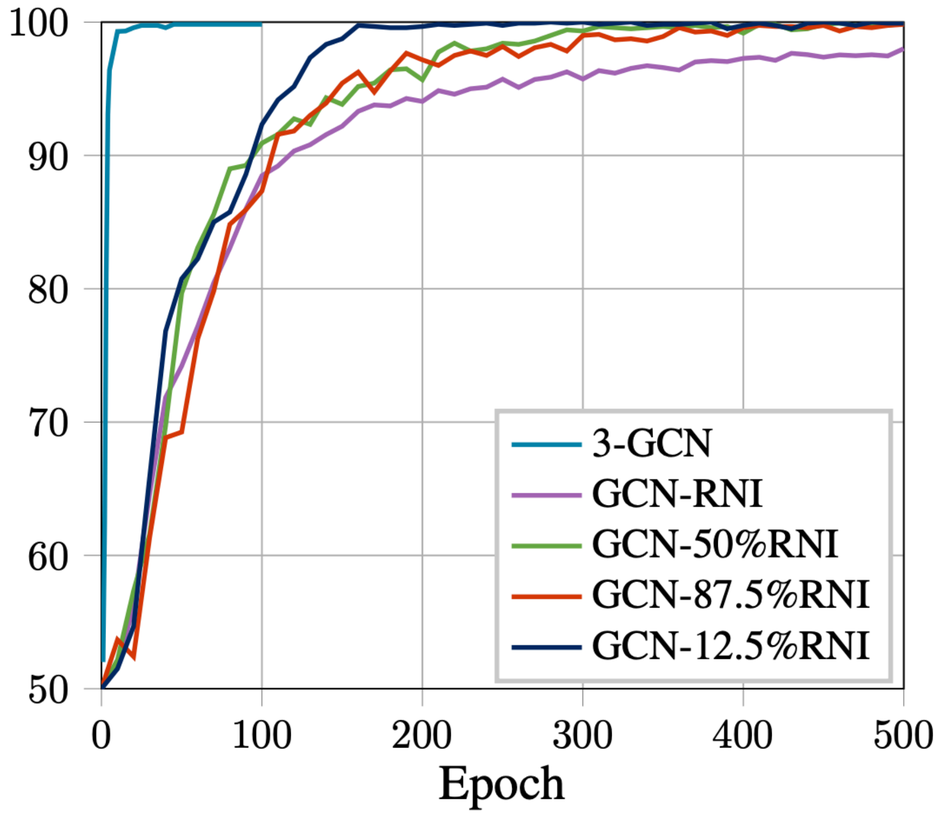
Convergence: Model convergence is slower for GCN-RNI and this is the price to pay, quoting (Abboud et al., 2020):



Randomised Models Converge Slower

Convergence: Model convergence is slower for GCN-RNI and this is the price to pay, quoting (Abboud et al., 2020):

"3-GCN only requires about 10 epochs to achieve its optimal performance, whereas GCN-RNI models all require in excess of 100 epochs. Intuitively, the slower convergence of GCN-RNI can be attributed to a significantly harder learning task compared to 3-GCN: Whereas 3-GCN must learn from a deterministic set of node embeddings, and is naturally capable of discerning between dataset cores, GCN-RNI relies on RNI to discern between data points in EXP, via an artificial node ordering. This in turn implies that GCN-RNI must first leverage RNI to detect structure, then subsequently learn robustness against the variability of RNI, which makes the learning task for GCN-RNI especially challenging."



A More Variable Dataset

A More Variable Dataset

EXP is solely designed for expressiveness evaluation, and this leaves multiple questions open: How does RNI impact learning when data contains instances with varying expressiveness requirements, and how does RNI affect model generalisation on more variable datasets?

A More Variable Dataset

EXP is solely designed for expressiveness evaluation, and this leaves multiple questions open: How does RNI impact learning when data contains instances with varying expressiveness requirements, and how does RNI affect model generalisation on more variable datasets?

CEXP dataset is proposed to address these questions, and it can be seen as a combination of two datasets: EXP dataset and CORRUPT dataset, which a minimally corrupted version of EXP.

A More Variable Dataset

EXP is solely designed for expressiveness evaluation, and this leaves multiple questions open: How does RNI impact learning when data contains instances with varying expressiveness requirements, and how does RNI affect model generalisation on more variable datasets?

CEXP dataset is proposed to address these questions, and it can be seen as a combination of two datasets: EXP dataset and CORRUPT dataset, which a minimally corrupted version of EXP.

While EXP graphs are not 1-WL distinguishable, CORRUPT graphs are 1-WL distinguishable. Importantly, CORRUPT graphs are still very similar to their uncorrupted variants, making the overall learning task harder.

A More Variable Dataset

EXP is solely designed for expressiveness evaluation, and this leaves multiple questions open: How does RNI impact learning when data contains instances with varying expressiveness requirements, and how does RNI affect model generalisation on more variable datasets?

CEXP dataset is proposed to address these questions, and it can be seen as a combination of two datasets: EXP dataset and CORRUPT dataset, which a minimally corrupted version of EXP.

While EXP graphs are not 1-WL distinguishable, CORRUPT graphs are 1-WL distinguishable. Importantly, CORRUPT graphs are still very similar to their uncorrupted variants, making the overall learning task harder.

CEXP is thus well-suited for evaluating the efficacy of RNI more holistically, as it allows the evaluation of the contribution of RNI jointly on EXP and CORRUPT:

A More Variable Dataset

EXP is solely designed for expressiveness evaluation, and this leaves multiple questions open: How does RNI impact learning when data contains instances with varying expressiveness requirements, and how does RNI affect model generalisation on more variable datasets?

CEXP dataset is proposed to address these questions, and it can be seen as a combination of two datasets: EXP dataset and CORRUPT dataset, which a minimally corrupted version of EXP.

While EXP graphs are not 1-WL distinguishable, CORRUPT graphs are 1-WL distinguishable. Importantly, CORRUPT graphs are still very similar to their uncorrupted variants, making the overall learning task harder.

CEXP is thus well-suited for evaluating the efficacy of RNI more holistically, as it allows the evaluation of the contribution of RNI jointly on EXP and CORRUPT:

- EXP requires 2-WL expressiveness, and any model without this power will get $50\,\%$ accuracy on this subset.

A More Variable Dataset

EXP is solely designed for expressiveness evaluation, and this leaves multiple questions open: How does RNI impact learning when data contains instances with varying expressiveness requirements, and how does RNI affect model generalisation on more variable datasets?

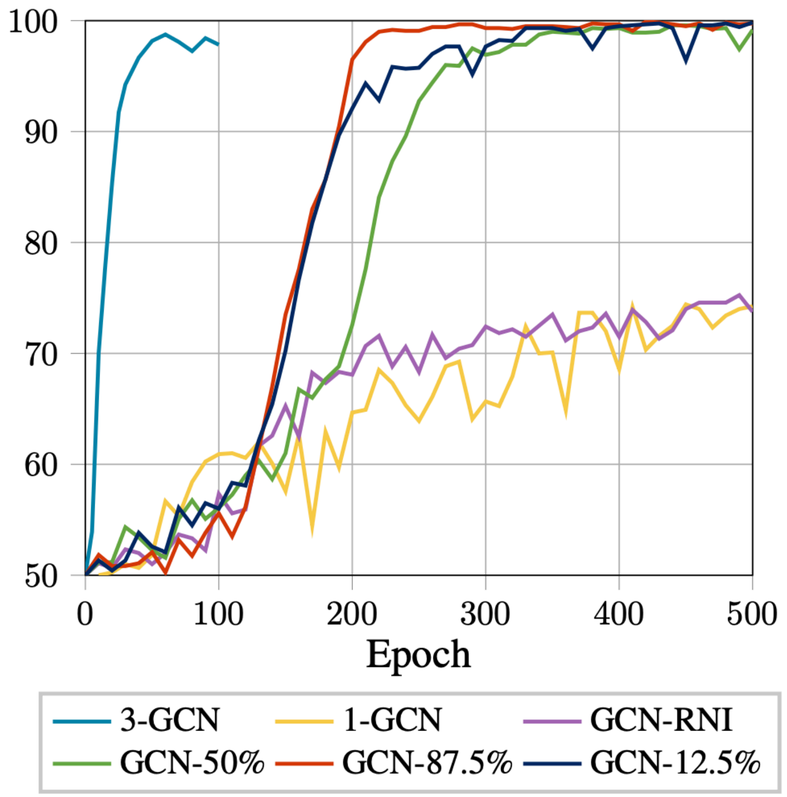
CEXP dataset is proposed to address these questions, and it can be seen as a combination of two datasets: EXP dataset and CORRUPT dataset, which a minimally corrupted version of EXP.

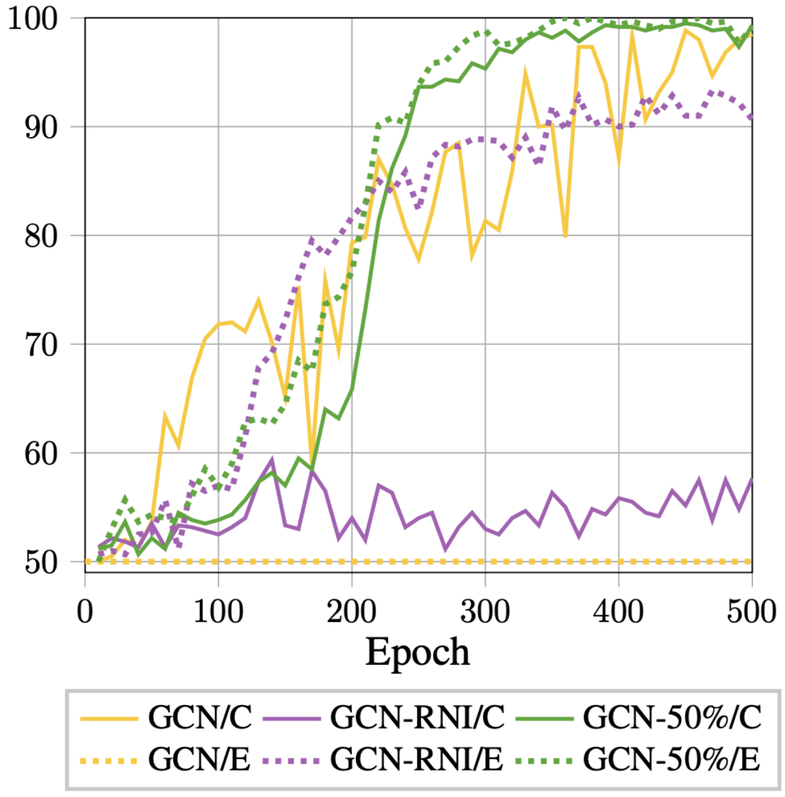
While EXP graphs are not 1-WL distinguishable, CORRUPT graphs are 1-WL distinguishable. Importantly, CORRUPT graphs are still very similar to their uncorrupted variants, making the overall learning task harder.

CEXP is thus well-suited for evaluating the efficacy of RNI more holistically, as it allows the evaluation of the contribution of RNI jointly on EXP and CORRUPT:

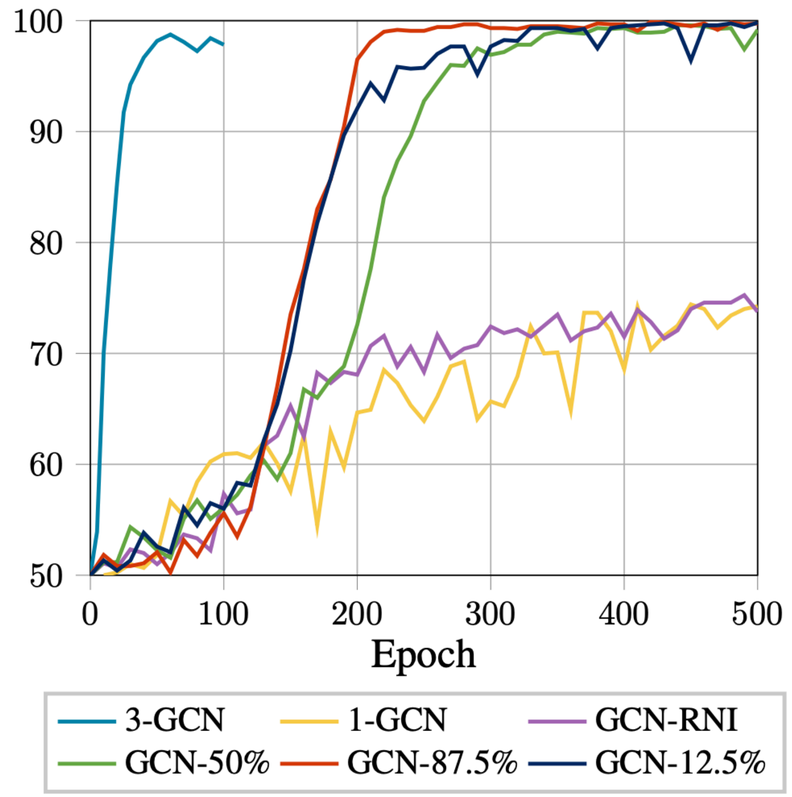
- EXP requires 2-WL expressiveness, and any model without this power will get $50\,\%$ accuracy on this subset.
- CORRUPT does not require expressiveness, but makes the overall learning task harder due to the structural similarities to EXP instances — Hence, an expressive model that generalises poorly can be identified by poor performance on this dataset.

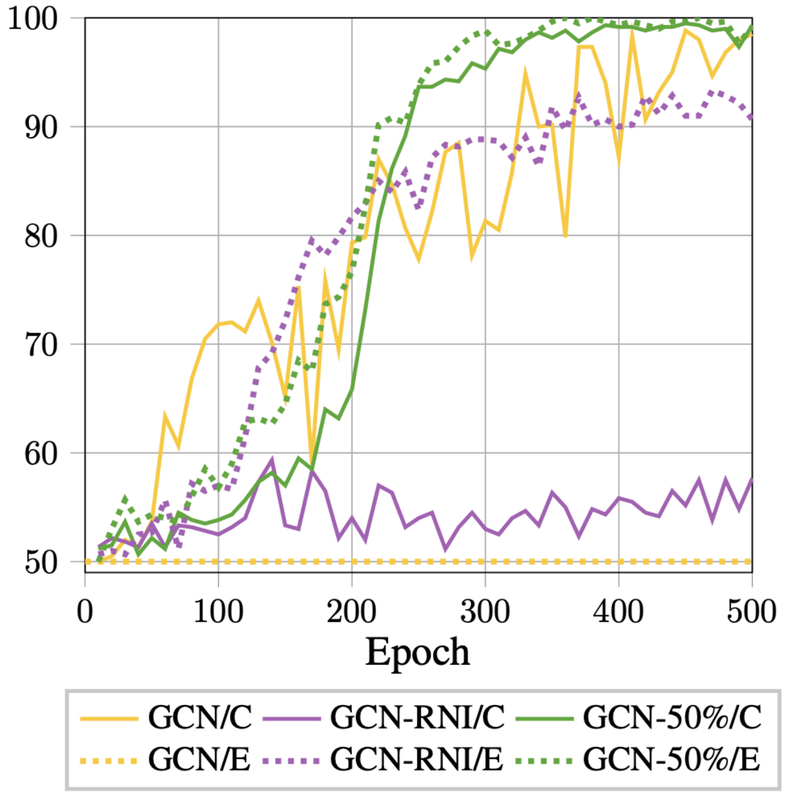
A More Variable Dataset





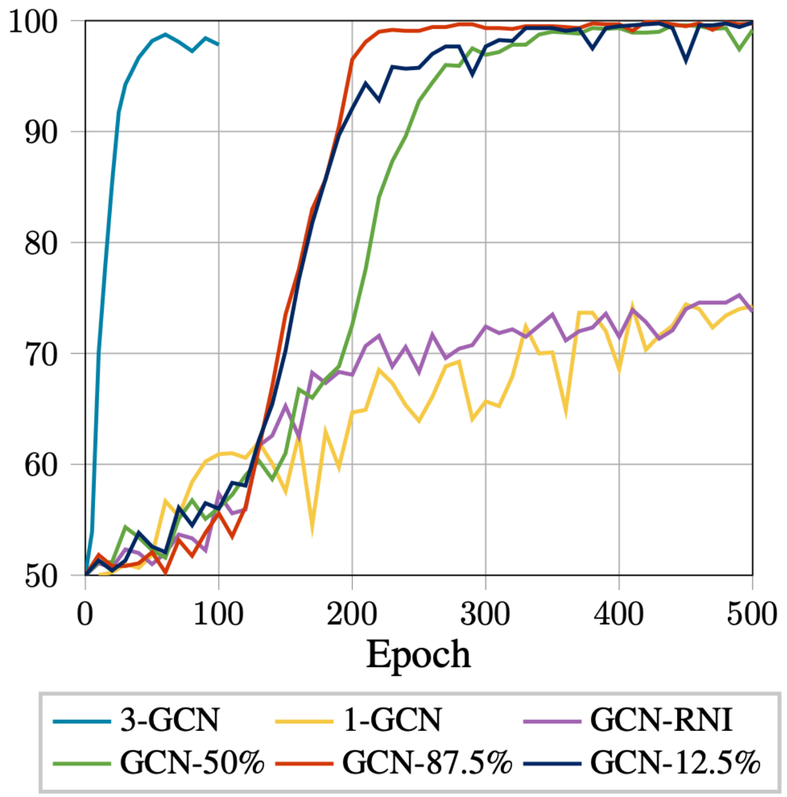
A More Variable Dataset

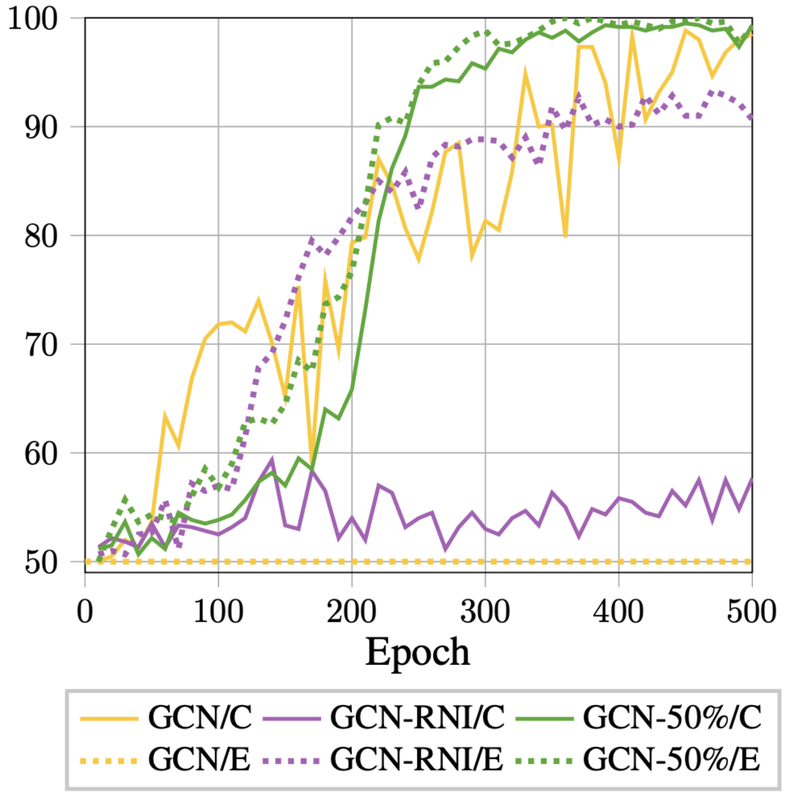




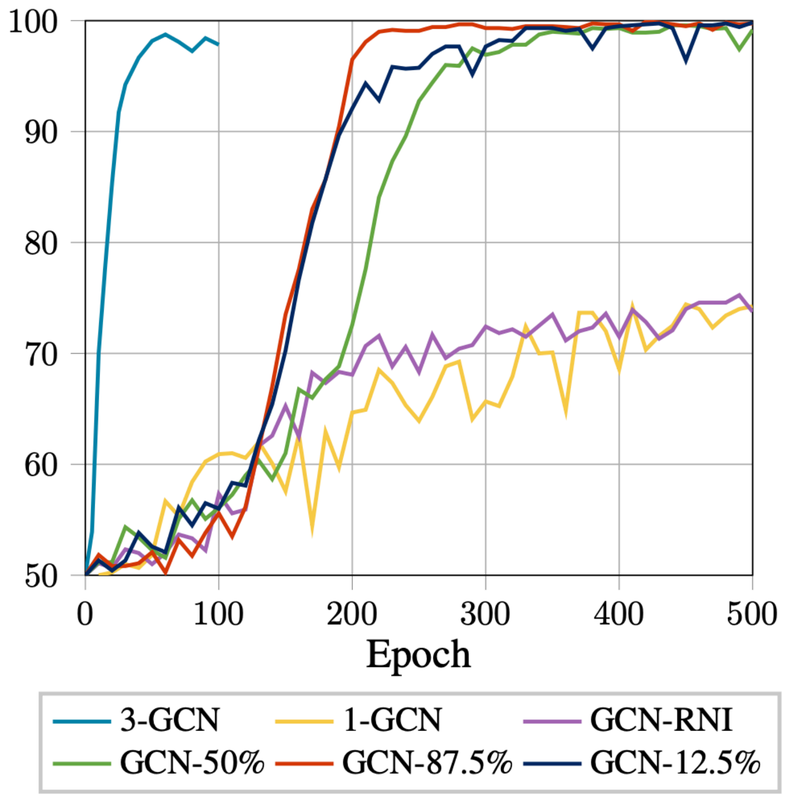
Fully randomised GCN-RNI model achieves $91\,\%$ accuracy on the EXP subset (dotted-purple line on the RHS), but struggles on CORRUPT subset (purple line on the RHS), slightly below $60\,\%$.

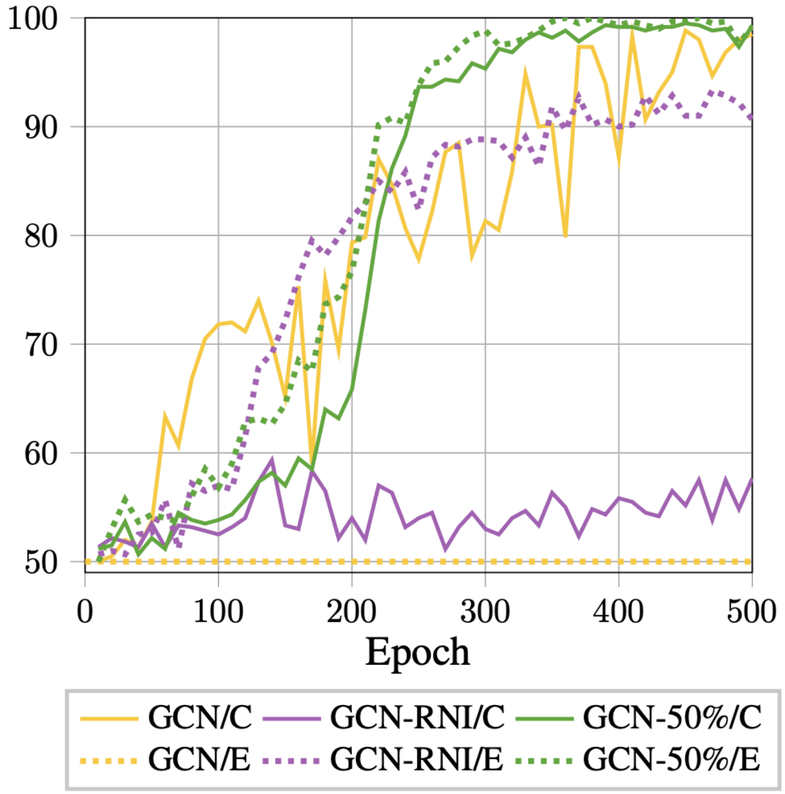
A More Variable Dataset





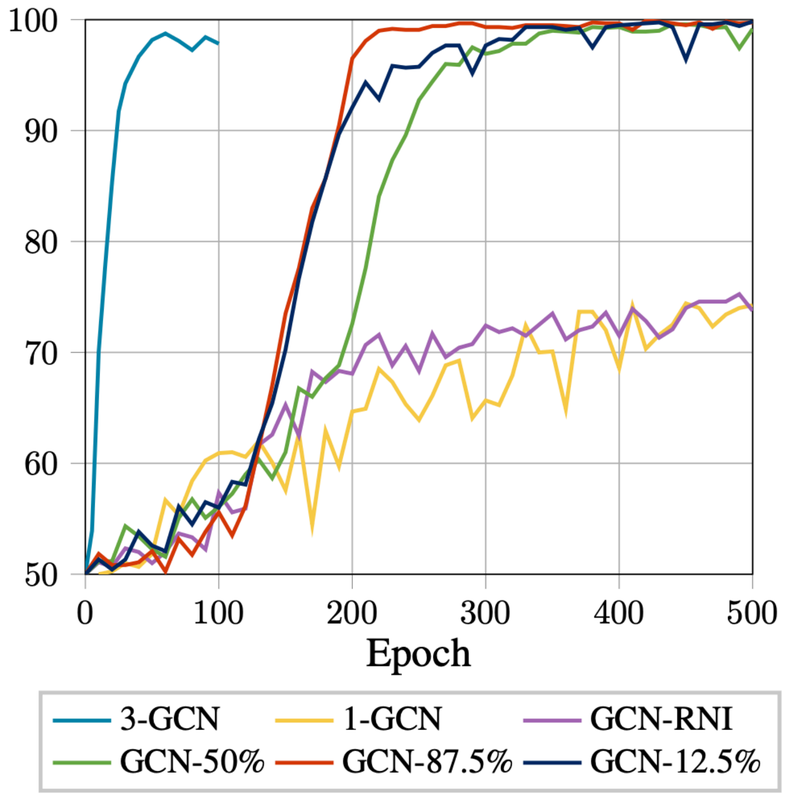
A More Variable Dataset

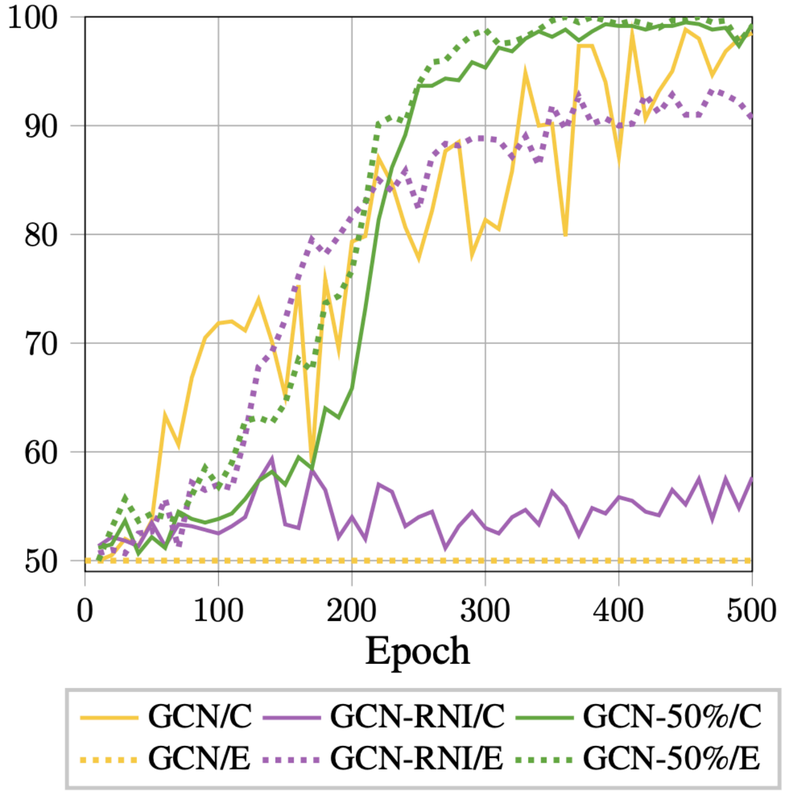




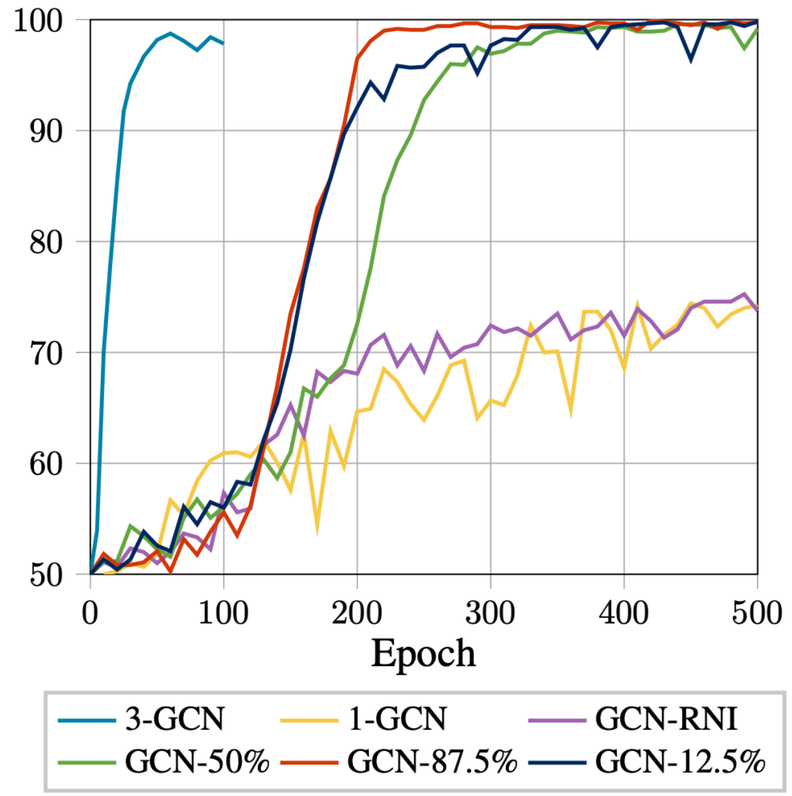
Observation: Fully randomised GCN-RNI loses all node type information, which is valuable for making robust predictions, and therefore struggles on CORRUPT, and converges much slower.

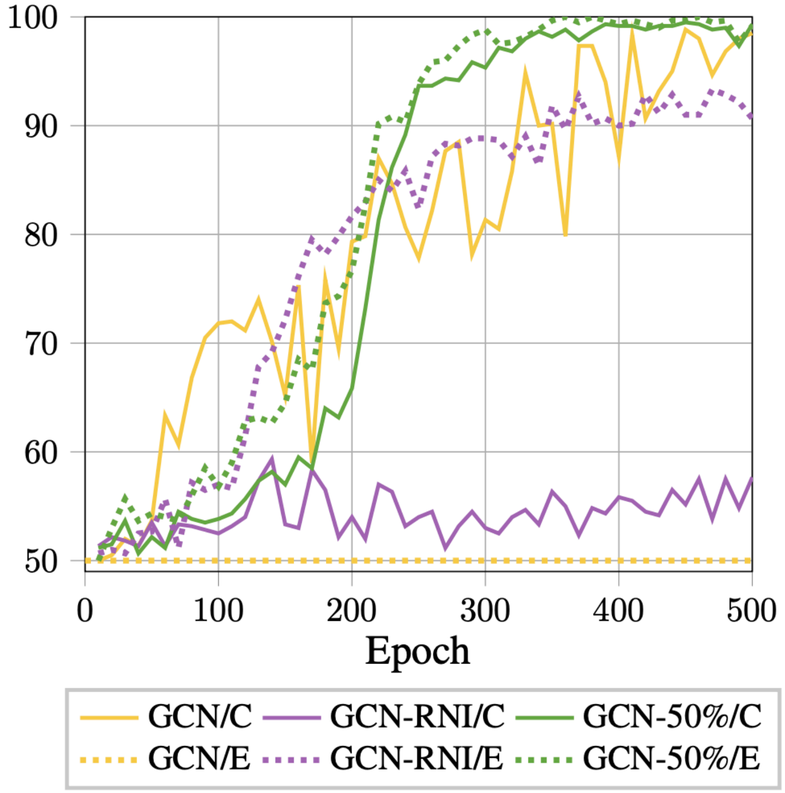
A More Variable Dataset





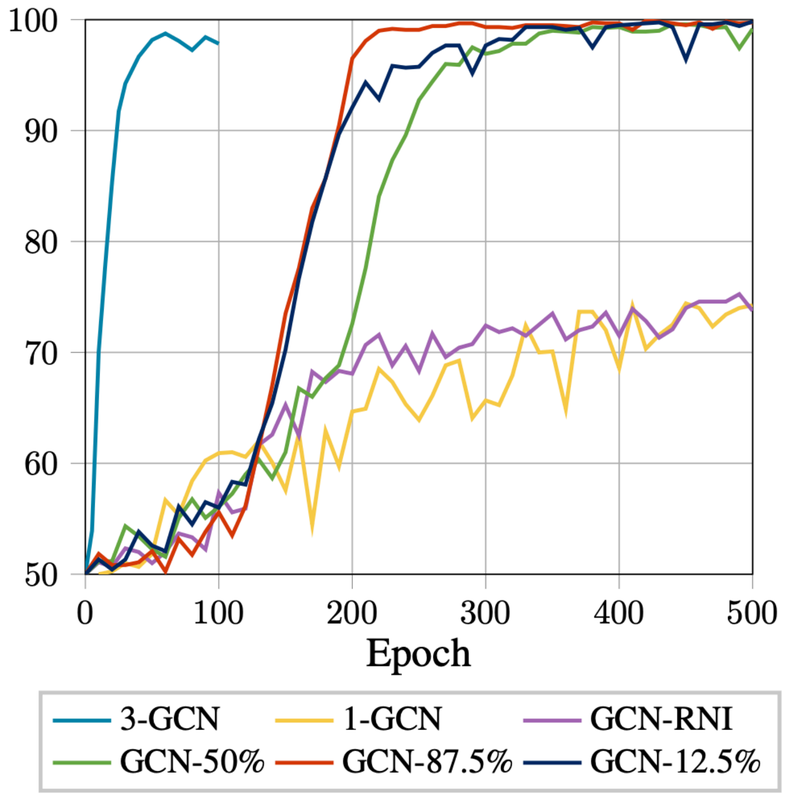
A More Variable Dataset

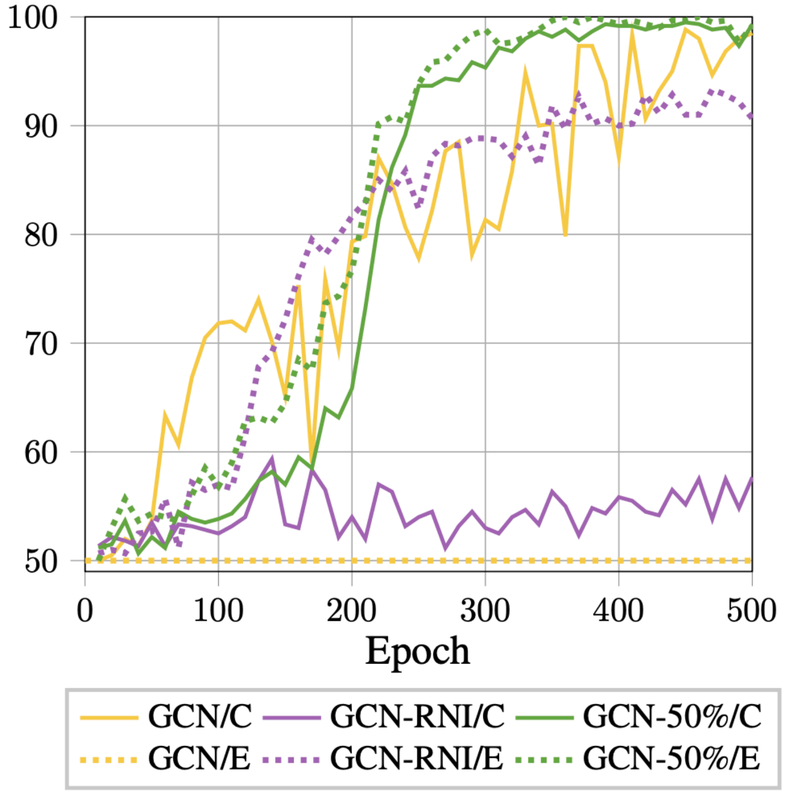




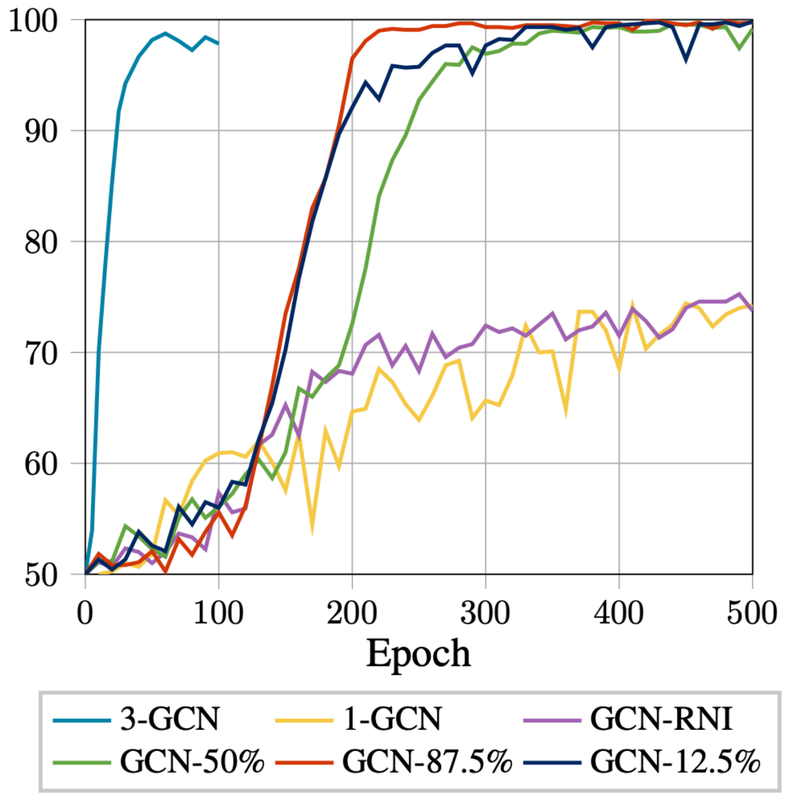
On the other hand, all partially randomised models, perform near-perfect also on this challenging dataset!

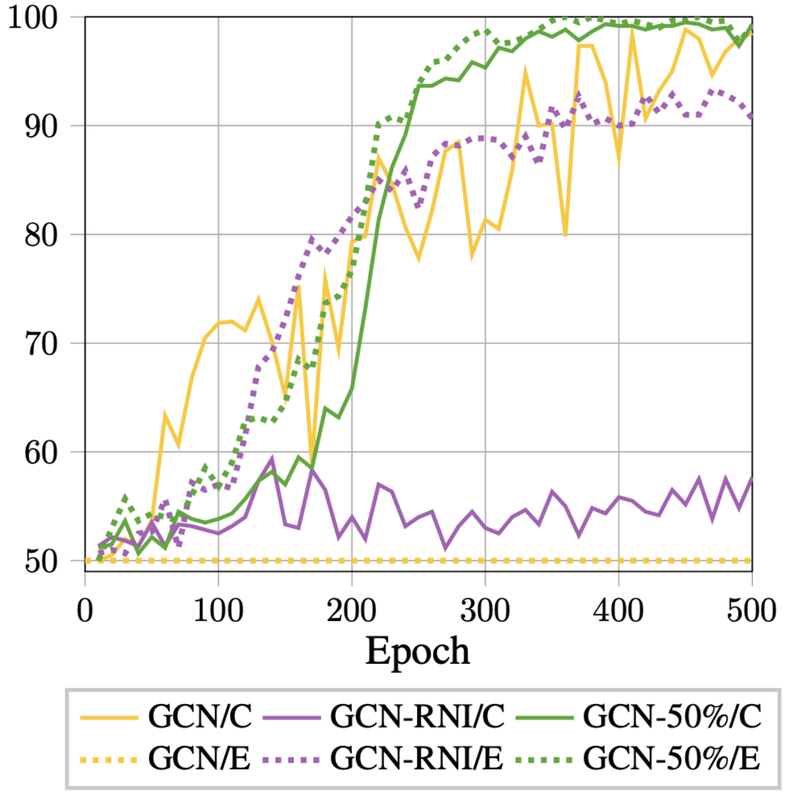
A More Variable Dataset





A More Variable Dataset





Observation: Partially randomised models achieve the best of both worlds on CEXP, leveraging inductive bias from deterministic node embeddings, while harnessing the power of random embeddings to gain expressiveness.

Tying Things Together

Tying Things Together

The overall behaviour of MPNN-RNI models can be intuitively described as follows:

Tying Things Together

The overall behaviour of MPNN-RNI models can be intuitively described as follows:

- MPNN-RNI models theoretically extend MPNN capabilities with RNI and enable individualisation of graphs with high probability.

Tying Things Together

The overall behaviour of MPNN-RNI models can be intuitively described as follows:

- MPNN-RNI models theoretically extend MPNN capabilities with RNI and enable individualisation of graphs with high probability.
- Since at every epoch (a subset of) node features are re-initialised randomly, and intuitively, each sample yields a different order (i.e., coloured graph), and after "sufficiently many" iterations, the model will become robust to different orderings — yielding strong generalisation empirically!

Tying Things Together

The overall behaviour of MPNN-RNI models can be intuitively described as follows:

- MPNN-RNI models theoretically extend MPNN capabilities with RNI and enable individualisation of graphs with high probability.
- Since at every epoch (a subset of) node features are re-initialised randomly, and intuitively, each sample yields a different order (i.e., coloured graph), and after "sufficiently many" iterations, the model will become robust to different orderings — yielding strong generalisation empirically!
- For an MPNN-RNI model to converge, it needs to see different orderings — and so it is solving a harder task than MPNNs and converges slower.

Tying Things Together

The overall behaviour of MPNN-RNI models can be intuitively described as follows:

- MPNN-RNI models theoretically extend MPNN capabilities with RNI and enable individualisation of graphs with high probability.
- Since at every epoch (a subset of) node features are re-initialised randomly, and intuitively, each sample yields a different order (i.e., coloured graph), and after "sufficiently many" iterations, the model will become robust to different orderings — yielding strong generalisation empirically!
- For an MPNN-RNI model to converge, it needs to see different orderings — and so it is solving a harder task than MPNNs and converges slower.
- Partially randomised MPNN-RNI models both perform better and converge faster than fully random MPNN-RNI models — attributed to the fact that they combine the best of both worlds.

Tying Things Together

The overall behaviour of MPNN-RNI models can be intuitively described as follows:

- MPNN-RNI models theoretically extend MPNN capabilities with RNI and enable individualisation of graphs with high probability.
- Since at every epoch (a subset of) node features are re-initialised randomly, and intuitively, each sample yields a different order (i.e., coloured graph), and after "sufficiently many" iterations, the model will become robust to different orderings — yielding strong generalisation empirically!
- For an MPNN-RNI model to converge, it needs to see different orderings — and so it is solving a harder task than MPNNs and converges slower.
- Partially randomised MPNN-RNI models both perform better and converge faster than fully random MPNN-RNI models — attributed to the fact that they combine the best of both worlds.
- Partial RNI is sufficient, and this is more so for real-world datasets that do not require to handle so many edge cases jointly.

Discussions and Outlook

Descriptive Complexity and Network Size

Descriptive Complexity and Network Size

Universality results come in many flavours, as we have seen. One important aspect is their implications on the size of the models, i.e., what is the depth and width of the network needed to capture the target function?

Descriptive Complexity and Network Size

Universality results come in many flavours, as we have seen. One important aspect is their implications on the size of the models, i.e., what is the depth and width of the network needed to capture the target function?

- The given construction of (Abboud et al., 2020) implies an exponential blow-up in the size of the MPNN. This is unsurprising, since there are no restrictions on the target function f that is being learned — It can be a function that requires exponential time/space etc.

Descriptive Complexity and Network Size

Universality results come in many flavours, as we have seen. One important aspect is their implications on the size of the models, i.e., what is the depth and width of the network needed to capture the target function?

- The given construction of (Abboud et al., 2020) implies an exponential blow-up in the size of the MPNN. This is unsurprising, since there are no restrictions on the target function f that is being learned — It can be a function that requires exponential time/space etc.
- Interestingly, however, when we focus on Boolean functions, the size of the MPNN entirely correlates with the descriptive complexity of the logical representation of the target representation:

Descriptive Complexity and Network Size

Universality results come in many flavours, as we have seen. One important aspect is their implications on the size of the models, i.e., what is the depth and width of the network needed to capture the target function?

- The given construction of (Abboud et al., 2020) implies an exponential blow-up in the size of the MPNN. This is unsurprising, since there are no restrictions on the target function f that is being learned — It can be a function that requires exponential time/space etc.
- Interestingly, however, when we focus on Boolean functions, the size of the MPNN entirely correlates with the descriptive complexity of the logical representation of the target representation:
    - If the target function can be represented with a formula Φ in C^2 then the depth of the resulting MPNN will be bounded with the quantifier depth of Φ.

Descriptive Complexity and Network Size

Universality results come in many flavours, as we have seen. One important aspect is their implications on the size of the models, i.e., what is the depth and width of the network needed to capture the target function?

- The given construction of (Abboud et al., 2020) implies an exponential blow-up in the size of the MPNN. This is unsurprising, since there are no restrictions on the target function f that is being learned — It can be a function that requires exponential time/space etc.
- Interestingly, however, when we focus on Boolean functions, the size of the MPNN entirely correlates with the descriptive complexity of the logical representation of the target representation:
    - If the target function can be represented with a formula Φ in C^2 then the depth of the resulting MPNN will be bounded with the quantifier depth of Φ.
    - The width of the resulting MPNN depends polynomially on the confidence parameter δ, as this directly determines the dimensions of the state vectors to reach the desired accuracy

Descriptive Complexity and Network Size

Universality results come in many flavours, as we have seen. One important aspect is their implications on the size of the models, i.e., what is the depth and width of the network needed to capture the target function?

- The given construction of (Abboud et al., 2020) implies an exponential blow-up in the size of the MPNN. This is unsurprising, since there are no restrictions on the target function f that is being learned — It can be a function that requires exponential time/space etc.
- Interestingly, however, when we focus on Boolean functions, the size of the MPNN entirely correlates with the descriptive complexity of the logical representation of the target representation:
    - If the target function can be represented with a formula Φ in C^2 then the depth of the resulting MPNN will be bounded with the quantifier depth of Φ.
    - The width of the resulting MPNN depends polynomially on the confidence parameter δ, as this directly determines the dimensions of the state vectors to reach the desired accuracy
- This gives rise to direct bounds on the size of MPNN-RNI models for special classes of functions, and paves the way for a principled and formal analysis of MPNN-RNI models.

Inductive Capacity and Expressive Power

Inductive Capacity and Expressive Power

MPNN-RNI models exhibit an interesting trade-off between expressive power and inductive capacity.

Inductive Capacity and Expressive Power

MPNN-RNI models exhibit an interesting trade-off between expressive power and inductive capacity.

Flexibility: MPNN-RNI models are universal, but they are not designed to target a specific level of expressiveness (unlike, e.g., k-GNNs), so their precise expressive power is governed by the particular dataset.

Inductive Capacity and Expressive Power

MPNN-RNI models exhibit an interesting trade-off between expressive power and inductive capacity.

Flexibility: MPNN-RNI models are universal, but they are not designed to target a specific level of expressiveness (unlike, e.g., k-GNNs), so their precise expressive power is governed by the particular dataset.

For example, the discernment power may not be used if the dataset does not require higher expressiveness, in which case, the model can even degenerate into an MPNN. Taking this perspective, MPNN-RNI models can be seen as faithful extensions of MPNNs.

Inductive Capacity and Expressive Power

MPNN-RNI models exhibit an interesting trade-off between expressive power and inductive capacity.

Flexibility: MPNN-RNI models are universal, but they are not designed to target a specific level of expressiveness (unlike, e.g., k-GNNs), so their precise expressive power is governed by the particular dataset.

For example, the discernment power may not be used if the dataset does not require higher expressiveness, in which case, the model can even degenerate into an MPNN. Taking this perspective, MPNN-RNI models can be seen as faithful extensions of MPNNs.

Local vs global: While MPNN-RNI models have the capacity to learn global properties using randomisation, they can behave similar to MPNNs w.r.t other properties we discussed. For example, homophily is likely captured by MPNN-RNI models similarly to MPNNs, as they are still based on local neighbourhood aggregation — and they can flexibly adapt to focus on local properties.

Inductive Capacity and Expressive Power

MPNN-RNI models exhibit an interesting trade-off between expressive power and inductive capacity.

Flexibility: MPNN-RNI models are universal, but they are not designed to target a specific level of expressiveness (unlike, e.g., k-GNNs), so their precise expressive power is governed by the particular dataset.

For example, the discernment power may not be used if the dataset does not require higher expressiveness, in which case, the model can even degenerate into an MPNN. Taking this perspective, MPNN-RNI models can be seen as faithful extensions of MPNNs.

Local vs global: While MPNN-RNI models have the capacity to learn global properties using randomisation, they can behave similar to MPNNs w.r.t other properties we discussed. For example, homophily is likely captured by MPNN-RNI models similarly to MPNNs, as they are still based on local neighbourhood aggregation — and they can flexibly adapt to focus on local properties.

Full vs partial RNI: Empirical evidence suggests that full randomisation can hurt inductive bias — as the randomness increases, the overall learning task becomes harder, and the presence of deterministic features prove helpful to preserve inductive bias, while gaining expressiveness.

Summary

Summary

- The quest for expressive models and coloured graphs

Summary

- The quest for expressive models and coloured graphs
- MPNNs with random node initialisation

Summary

- The quest for expressive models and coloured graphs
- MPNNs with random node initialisation
- Random node initialisation makes MPNNs universal even with partial RNI

Summary

- The quest for expressive models and coloured graphs
- MPNNs with random node initialisation
- Random node initialisation makes MPNNs universal even with partial RNI
- Permutation-invariance is preserved in expectation

Summary

- The quest for expressive models and coloured graphs
- MPNNs with random node initialisation
- Random node initialisation makes MPNNs universal even with partial RNI
- Permutation-invariance is preserved in expectation
- MPNN-RNI models are expressive and scalable — no space inefficiency

Summary

- The quest for expressive models and coloured graphs
- MPNNs with random node initialisation
- Random node initialisation makes MPNNs universal even with partial RNI
- Permutation-invariance is preserved in expectation
- MPNN-RNI models are expressive and scalable — no space inefficiency
- MPNN-RNI models converge slower — to see different colourings to become robust to them!

Summary

- The quest for expressive models and coloured graphs
- MPNNs with random node initialisation
- Random node initialisation makes MPNNs universal even with partial RNI
- Permutation-invariance is preserved in expectation
- MPNN-RNI models are expressive and scalable — no space inefficiency
- MPNN-RNI models converge slower — to see different colourings to become robust to them!
- The size of the MPNN-RNI model is correlated with the descriptive complexity of the target function

Summary

- The quest for expressive models and coloured graphs
- MPNNs with random node initialisation
- Random node initialisation makes MPNNs universal even with partial RNI
- Permutation-invariance is preserved in expectation
- MPNN-RNI models are expressive and scalable — no space inefficiency
- MPNN-RNI models converge slower — to see different colourings to become robust to them!
- The size of the MPNN-RNI model is correlated with the descriptive complexity of the target function
- Empirical evaluation suggests partial MPNN-RNI models as a strong alternative

Summary

- The quest for expressive models and coloured graphs
- MPNNs with random node initialisation
- Random node initialisation makes MPNNs universal even with partial RNI
- Permutation-invariance is preserved in expectation
- MPNN-RNI models are expressive and scalable — no space inefficiency
- MPNN-RNI models converge slower — to see different colourings to become robust to them!
- The size of the MPNN-RNI model is correlated with the descriptive complexity of the target function
- Empirical evaluation suggests partial MPNN-RNI models as a strong alternative
- There are more questions than answers in this context — more research needed!

Summary

- The quest for expressive models and coloured graphs
- MPNNs with random node initialisation
- Random node initialisation makes MPNNs universal even with partial RNI
- Permutation-invariance is preserved in expectation
- MPNN-RNI models are expressive and scalable — no space inefficiency
- MPNN-RNI models converge slower — to see different colourings to become robust to them!
- The size of the MPNN-RNI model is correlated with the descriptive complexity of the target function
- Empirical evaluation suggests partial MPNN-RNI models as a strong alternative
- There are more questions than answers in this context — more research needed!
- With this, we have covered all theoretical and foundational aspects, for practical aspects: Lecture 8.

References

- H. Maron, H. Ben-Hamu, H. Serviansky, and Y. Lipman. Provably powerful graph networks. *NeurIPS*, 2019a.
- C. Morris, M. Ritzert, M. Fey, W. Hamilton,J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and Leman go neural: Higher-order graph neural networks. *AAAI*, 2019.
- M. Grohe. Descriptive Complexity, Canonisation, and Definable Graph Structure Theory. *Cambridge University Press,* 2017.
- R. Sato, M. Yamada, and H. Kashima. Random features strengthen graph neural networks. CoRR, abs/ 2002.03155, 2020.
- R. Abboud, İ. İ. Ceylan, M. Grohe, T. Lukasiewicz, The Surprising Power of Graph Neural Networks with Random Node Initialization, arXiv:2010.01179, 2020
- M. Grohe. Descriptive Complexity, Canonisation, and Definable Graph Structure Theory. *Cambridge University Press,* 2017.