Lecture 7: Message Passing Neural Networks: Unique Features and Randomization

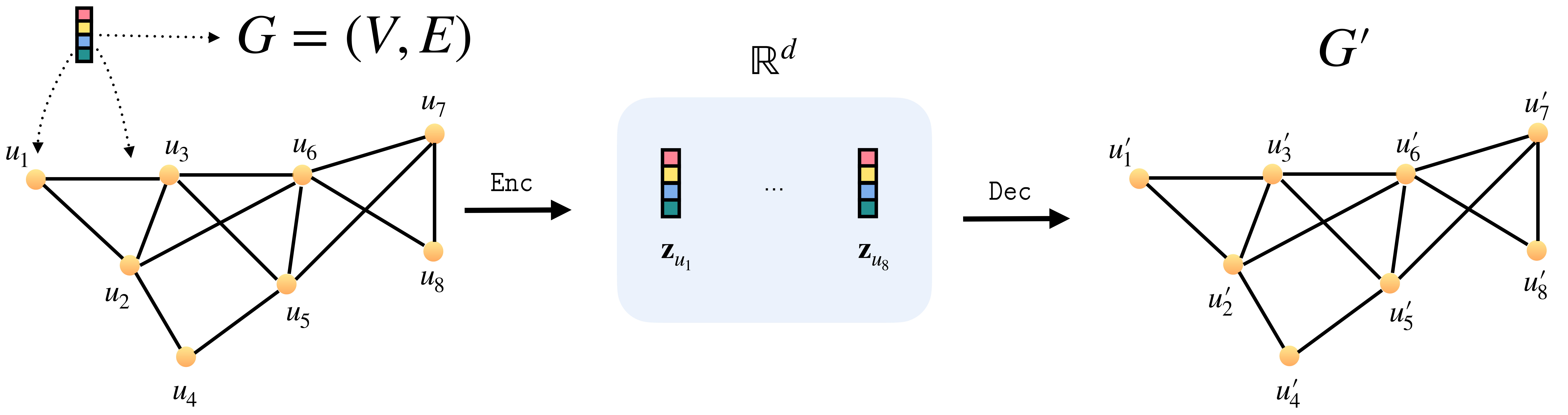
Relational Learning

İsmail İlkan Ceylan

Advanced Topics in Machine Learning, University of Oxford

31.01.2021

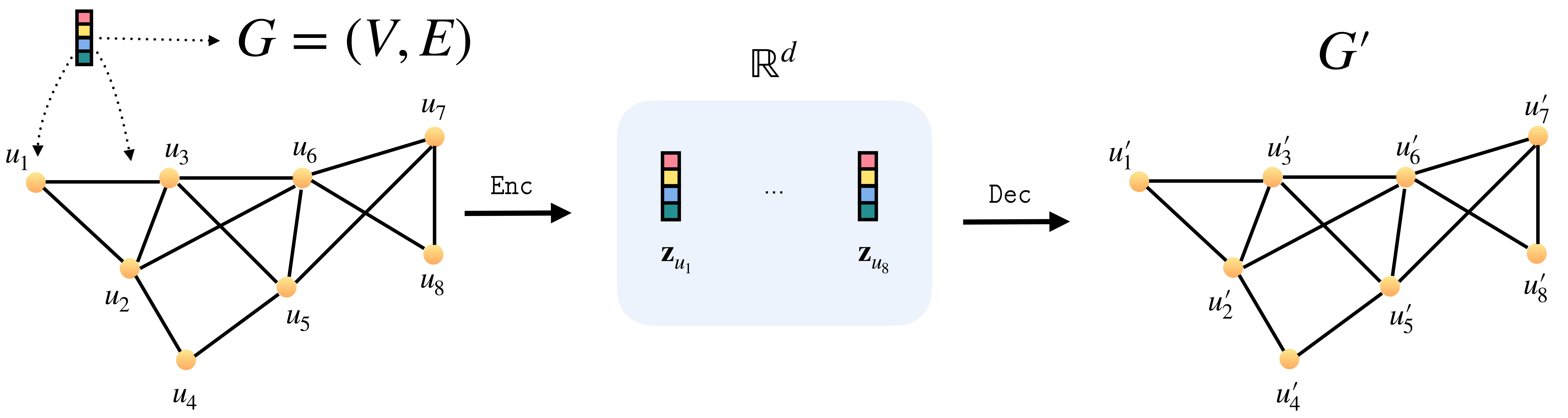
Graph Representation Learning



Expressive Power of MPNNs (Lecture 5)

$$\mathbf{h}_u^{(t)} = \sigma\left(\mathbf{W}_{self}^{(t)}\mathbf{h}_u^{(t-1)} + \mathbf{W}_{neigh}^{(t)}\sum_{v \in N(u)} \mathbf{h}_v^{(t-1)}\right)$$

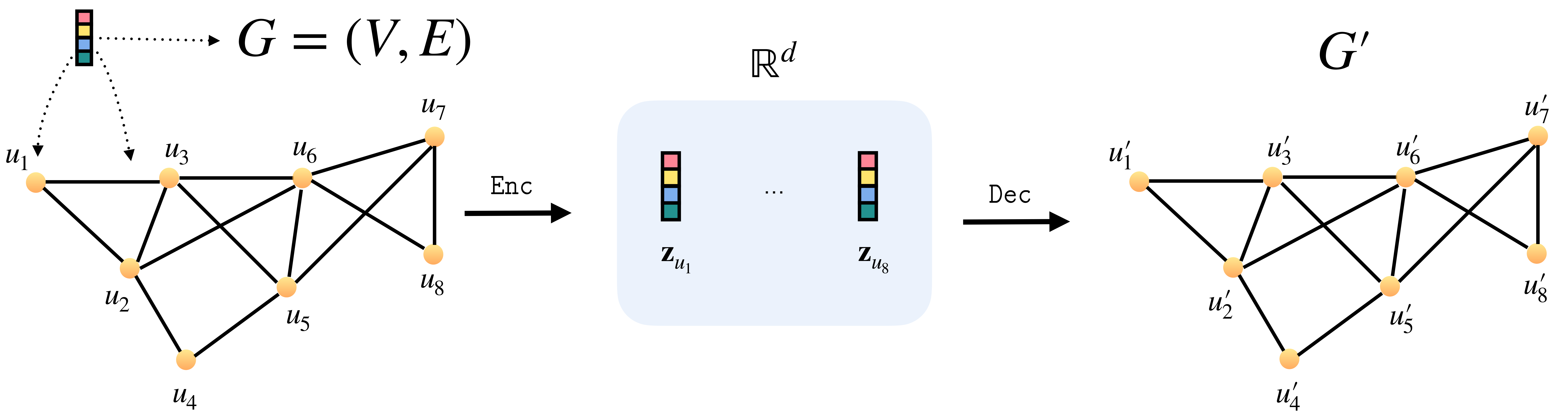
Graph Representation Learning



Higher-order GNNs (Lecture 6)

$$F = \mathsf{MLP} \circ \mathscr{H} \circ \mathscr{L}_d \circ \sigma \circ \cdots \circ \sigma \circ \mathscr{L}_1$$

Graph Representation Learning



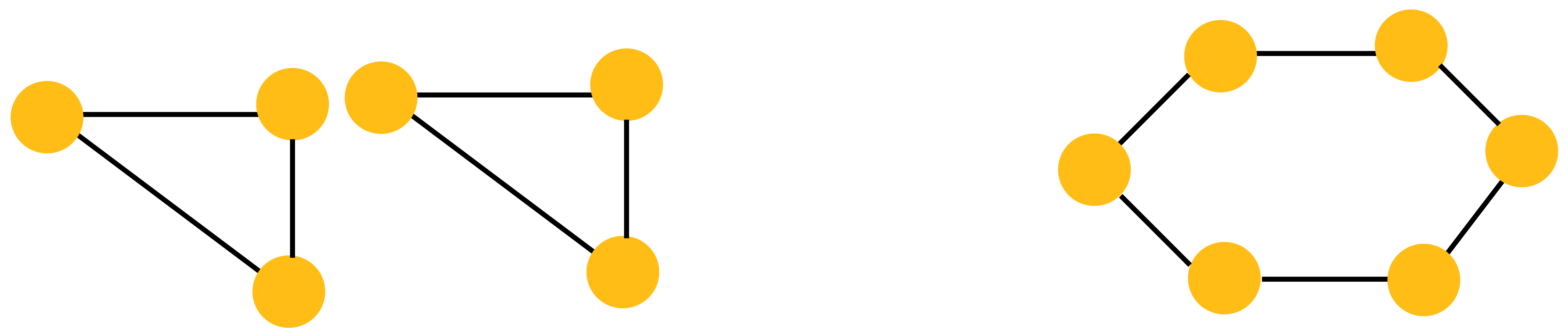
What is the expressive power of MPNNs with random features?

$$\mathbf{h}_u^{(t)} = \sigma\left(\mathbf{W}_{self}^{(t)}\mathbf{h}_u^{(t-1)} + \mathbf{W}_{neigh}^{(t)}\sum_{v\in N(u)}\mathbf{h}_v^{(t-1)}\right)$$

Overview

The Quest for Expressive and Scalable Models

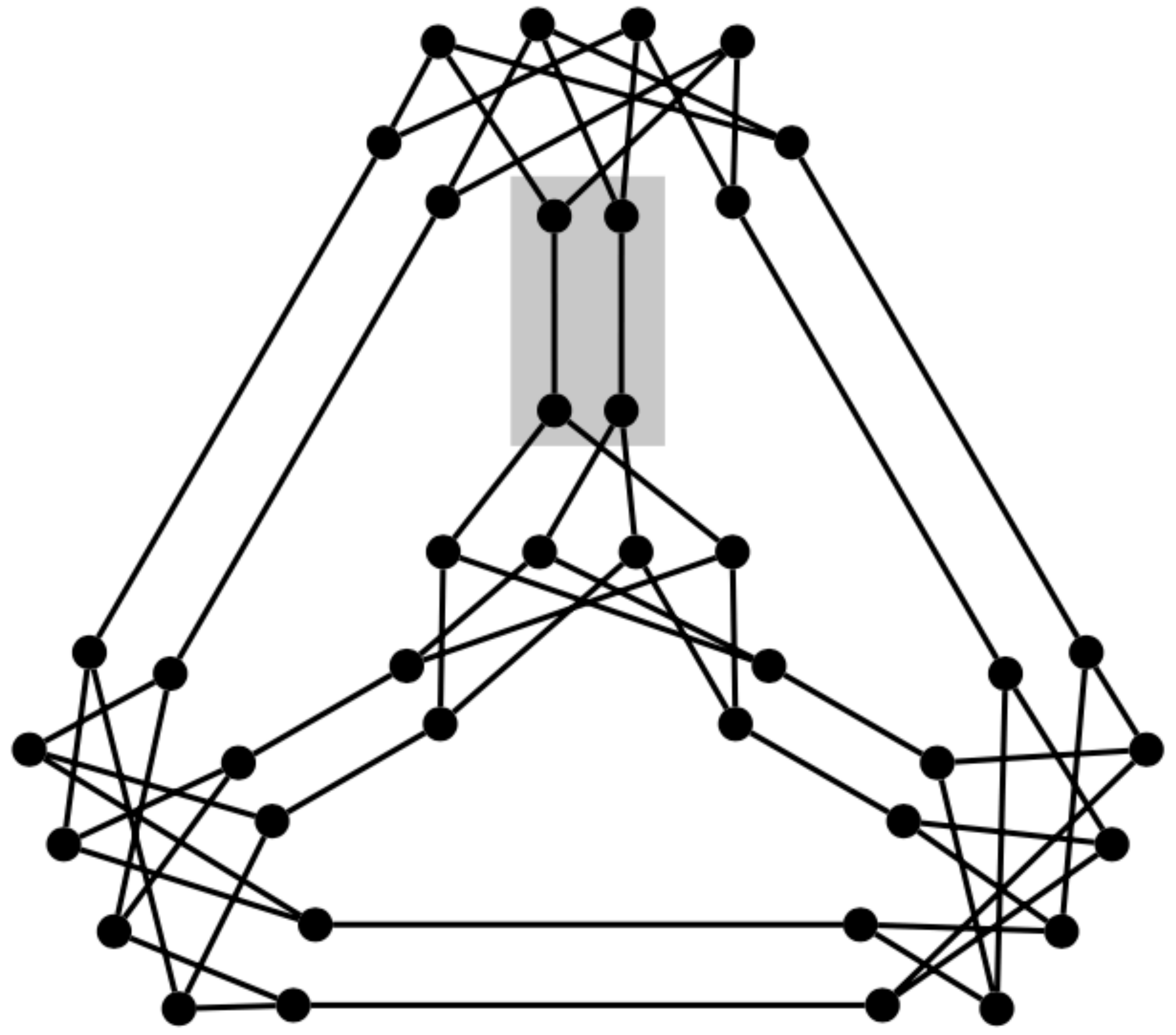
A Tale of Two Graphs

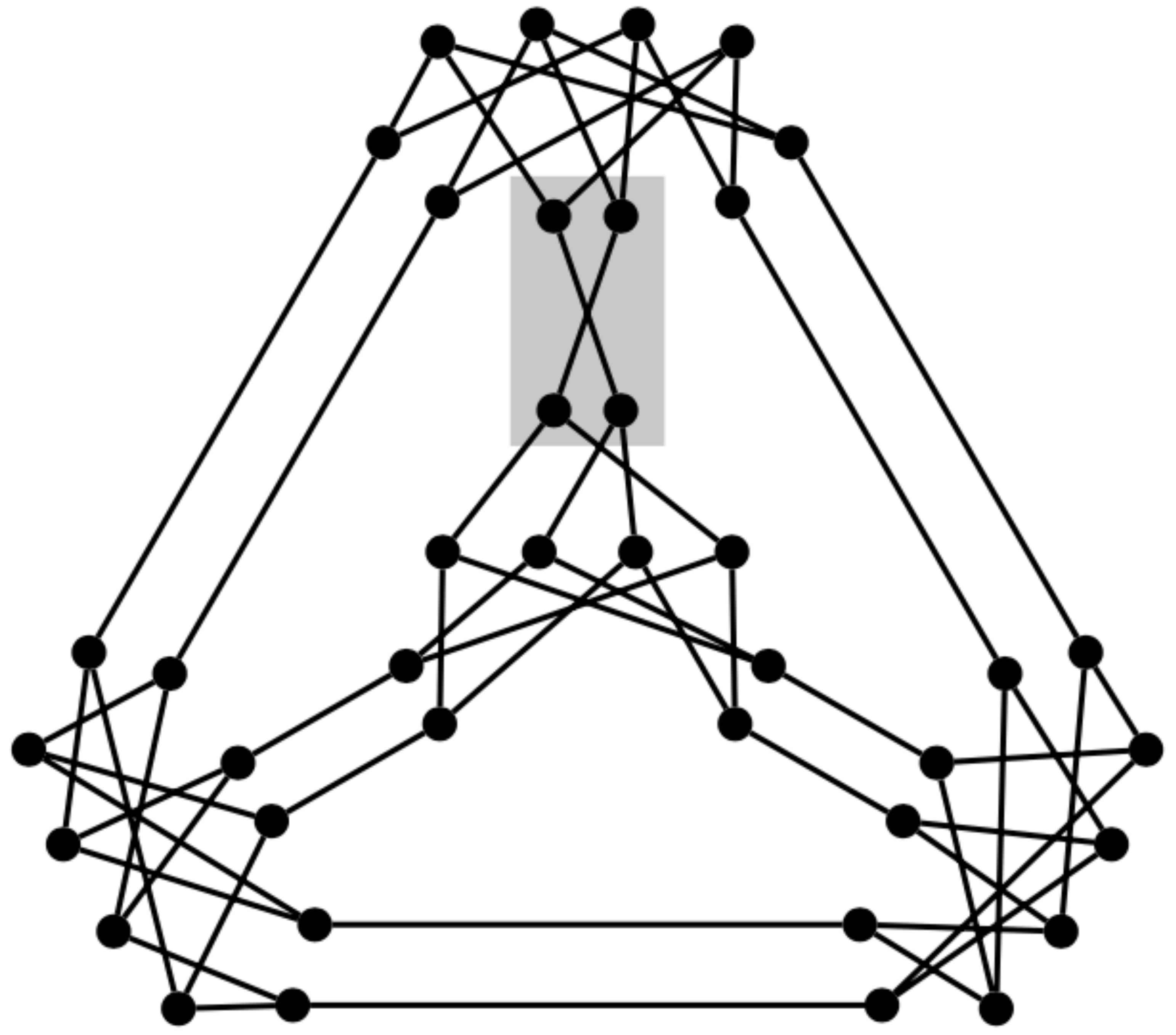


A brief recap

1. We have seen that 1-WL is insufficient and 2-WL is needed to distinguish these graphs.
2. MPNNs learn exact same embeddings for these graphs!
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

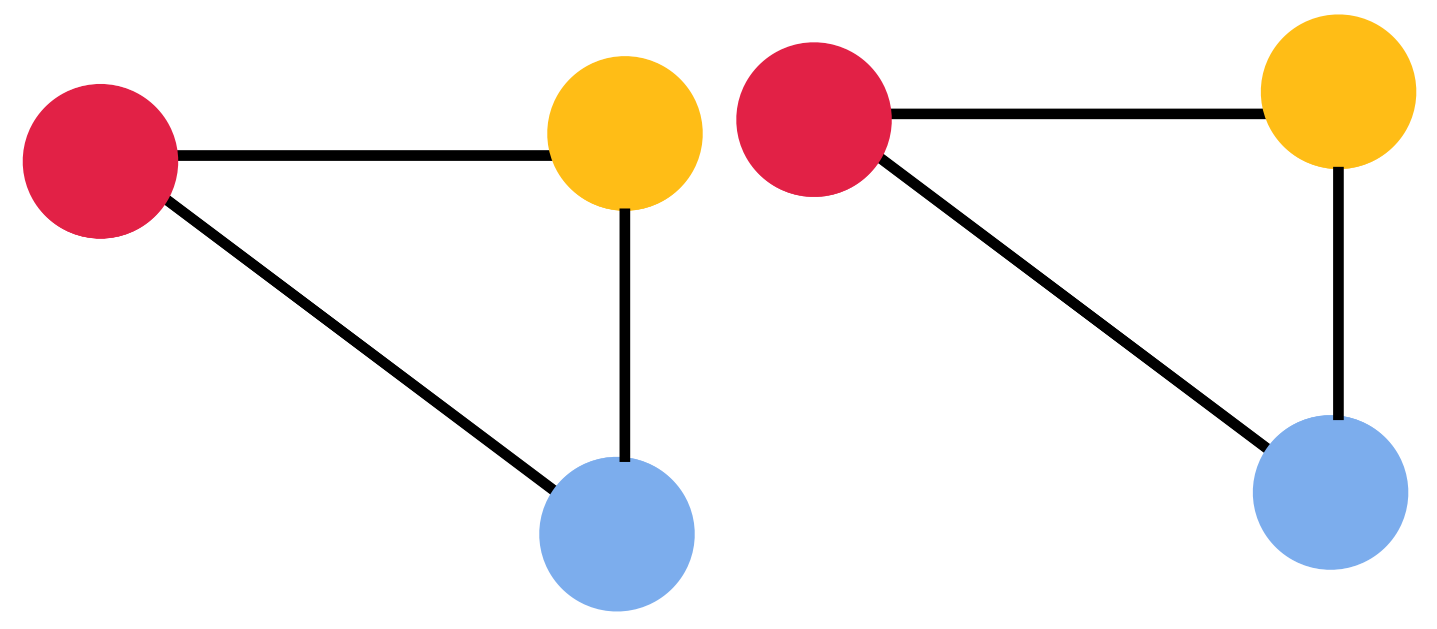


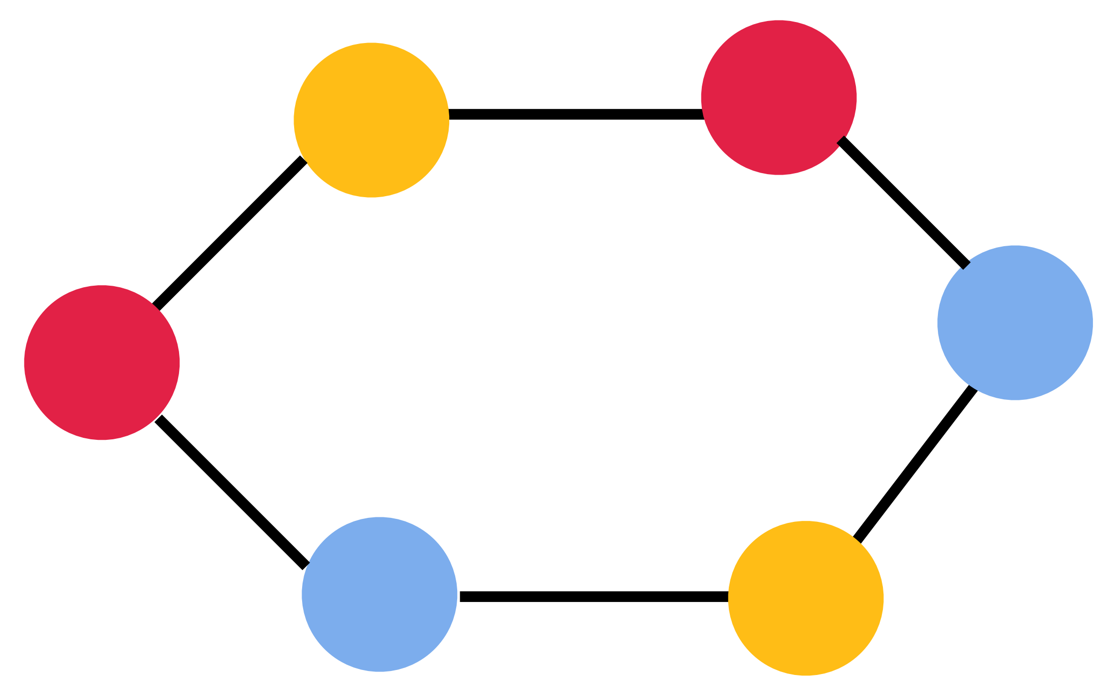


Example: 2WL cannot distinguish these graphs which differ only in the grey area (Grohe, 2017), i.e., even higher-order models such as 3-GNNs do not possess sufficient expressive power to distinguish these graphs.

Unique Node Identifiers

A Tale of Two Colored Graphs

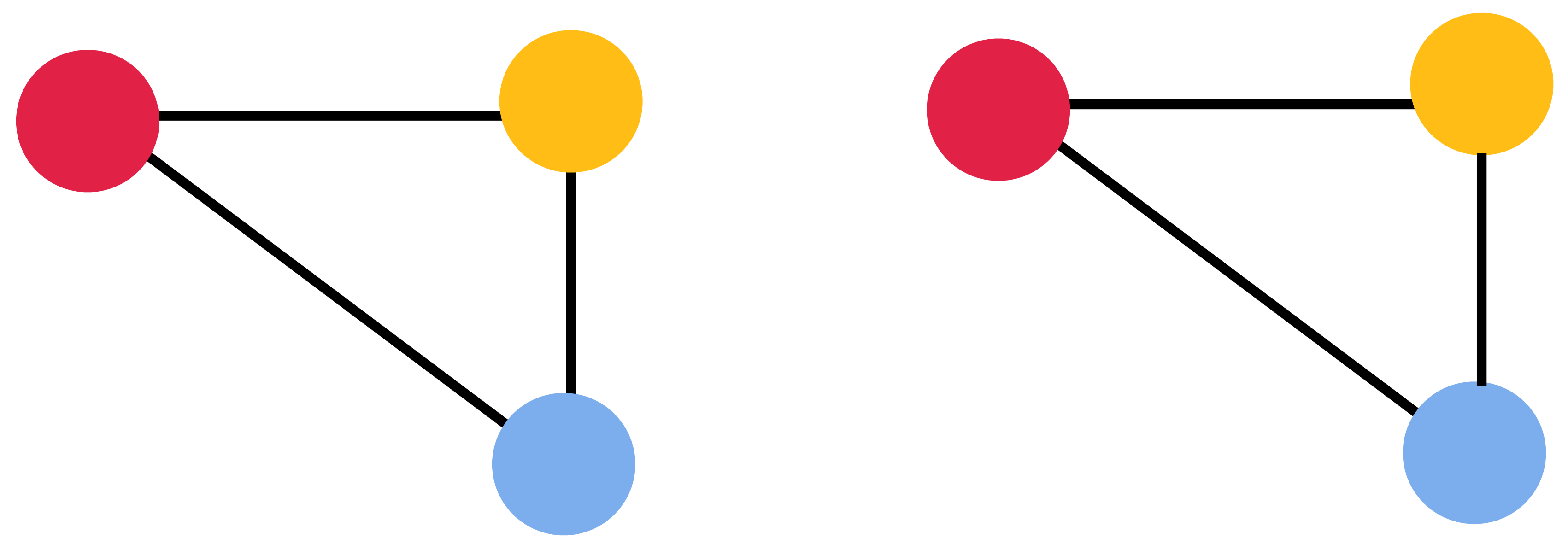


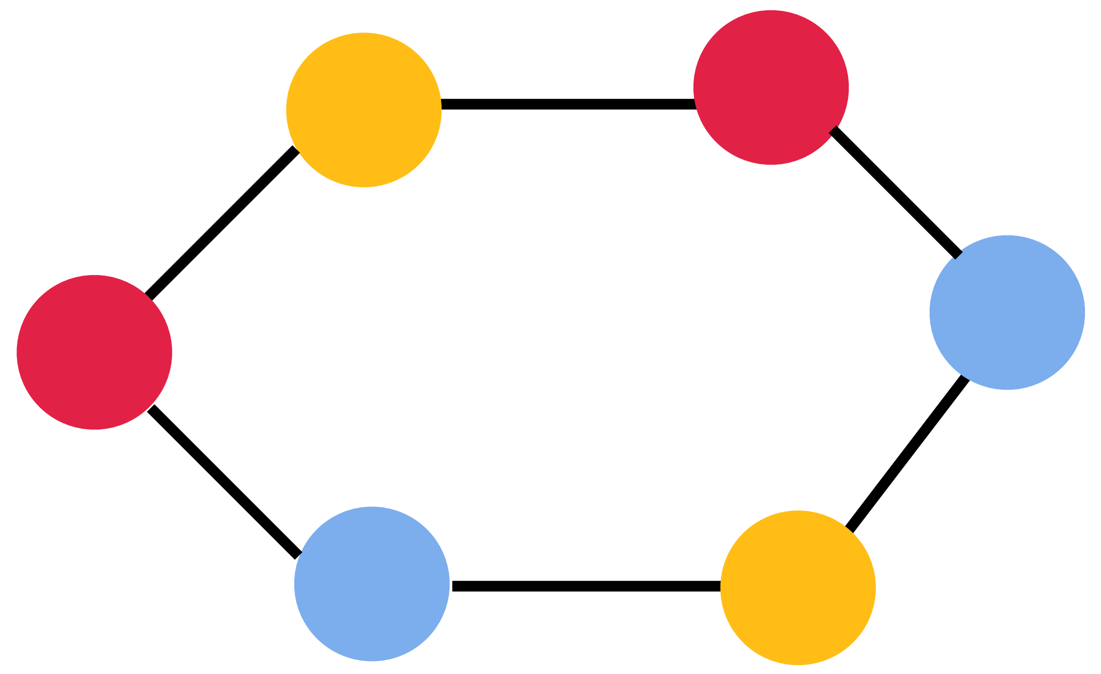


Some observations:

- The graphs we considered were not colored, or equivalently, single-colored.
- The WL algorithm can start with any initial coloring.
- The same is true for MPNNs — start with any node features.

A Tale of Two Colored Graphs

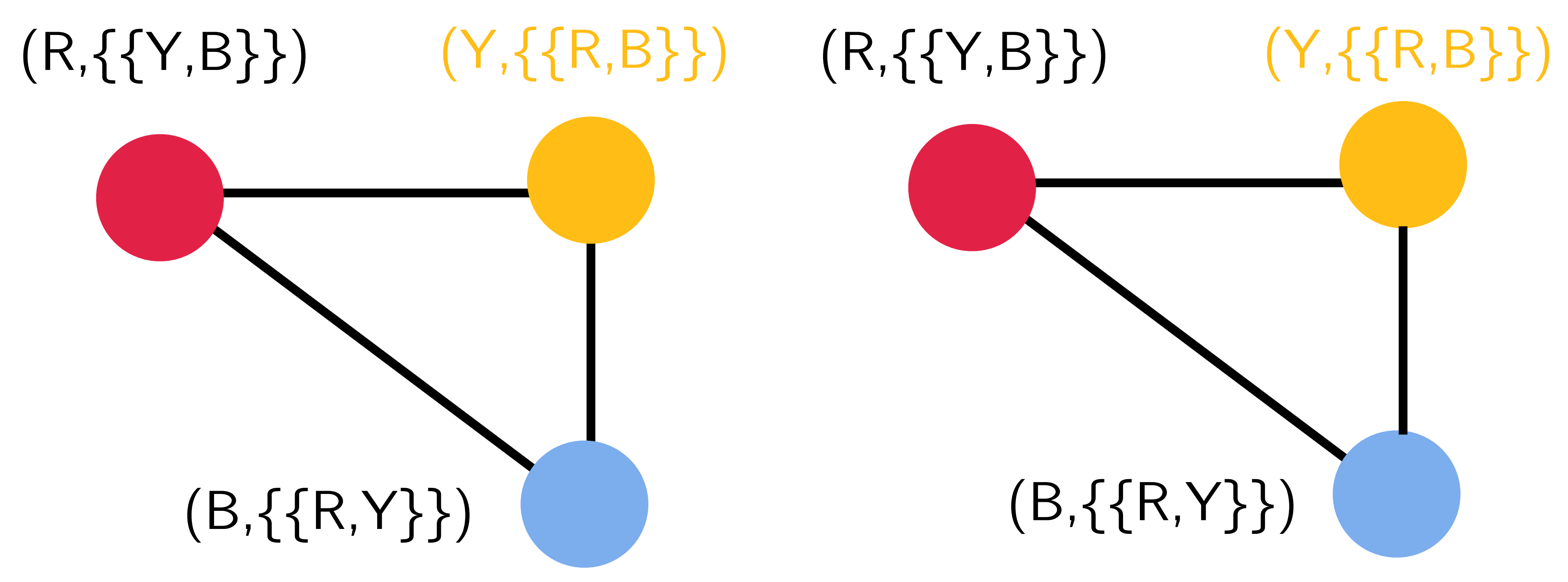


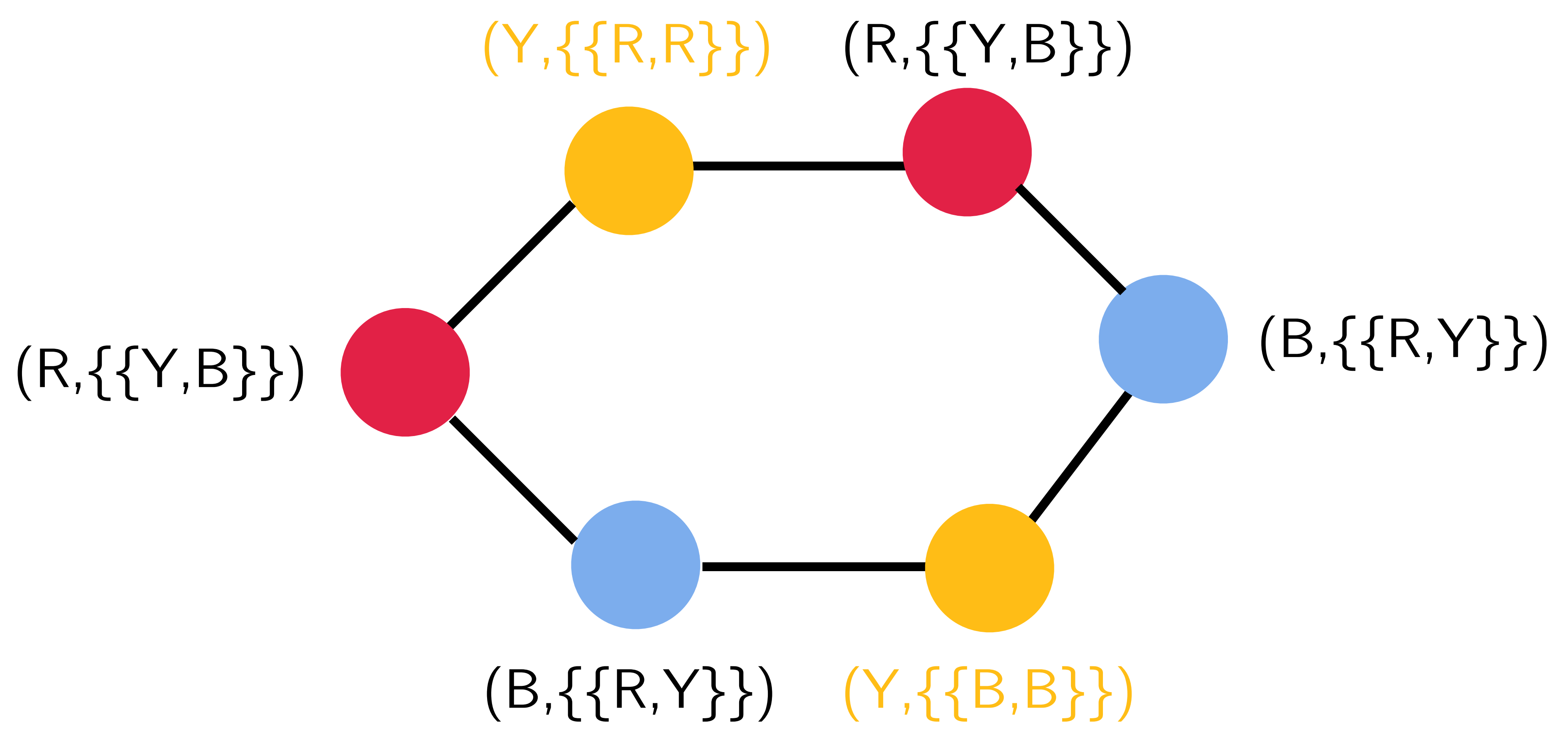


What happens when we color the graph pairs?

- 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 — by setting the node features accordingly.

A Tale of Two Colored Graphs

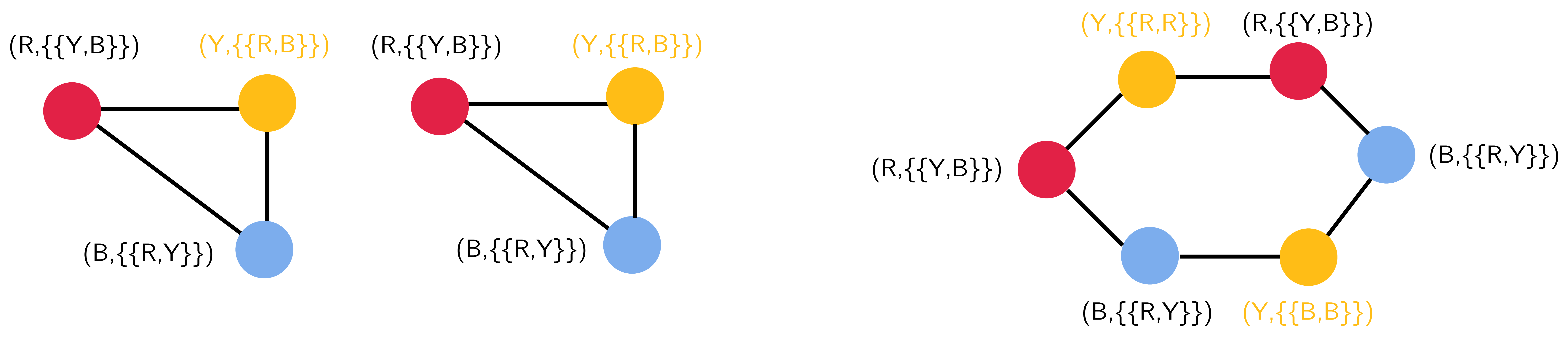




What happens when we color the graph pairs?

- 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 — by setting the node features accordingly.

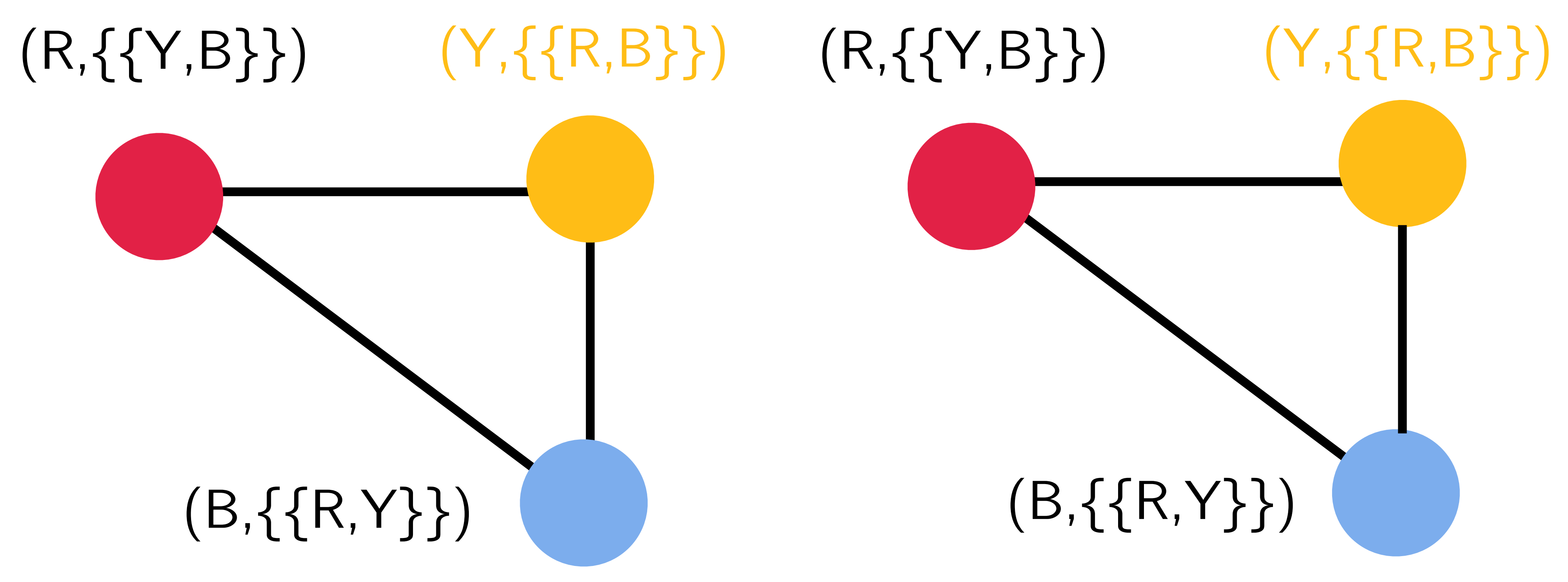
A Tale of Two Colored Graphs

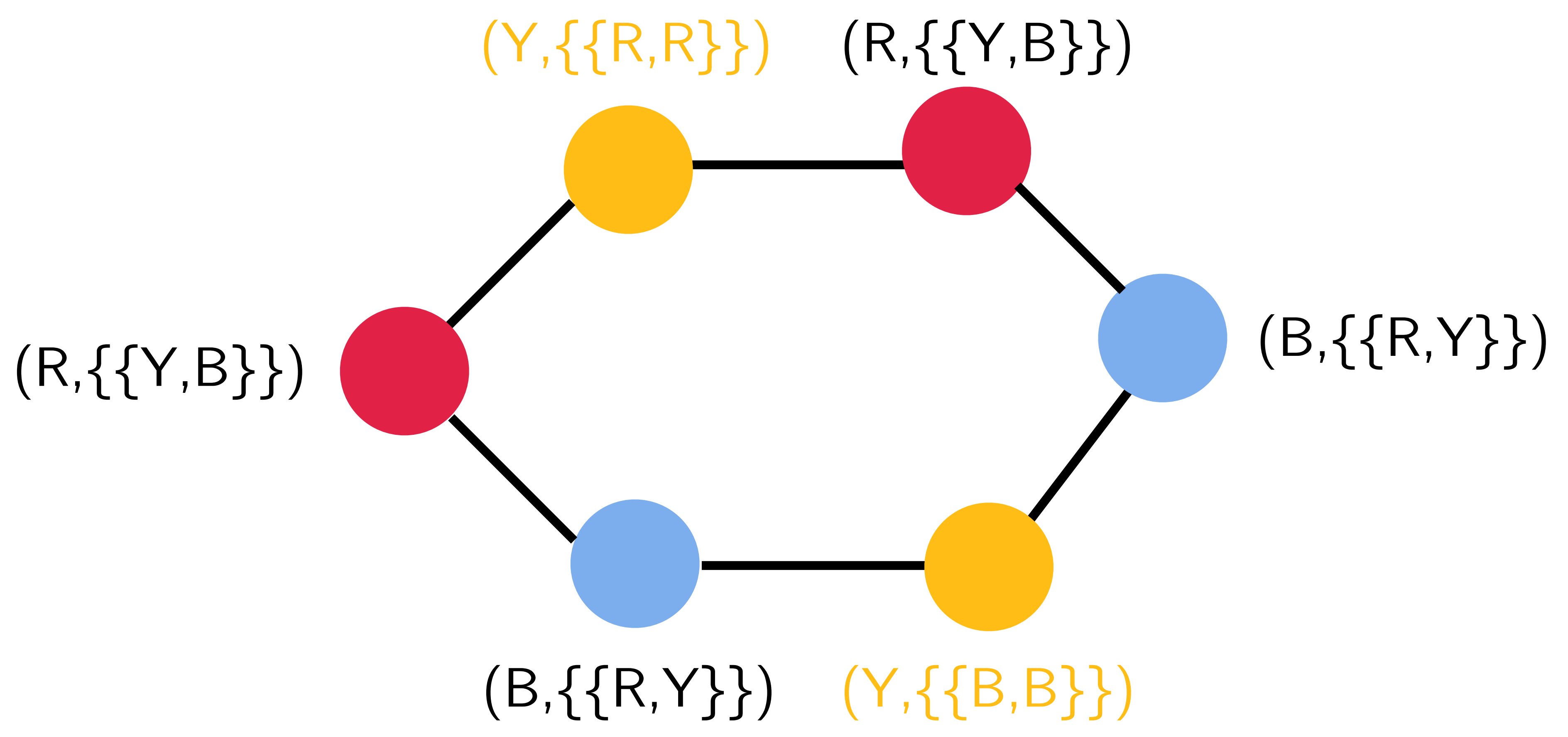


Initializing node features in an MPNN to different colors (i.e., unique node identifiers)?

This yields an expressive model — 1-WL can distinguish any pair of ordered/colored graphs.

A Tale of Two Colored Graphs

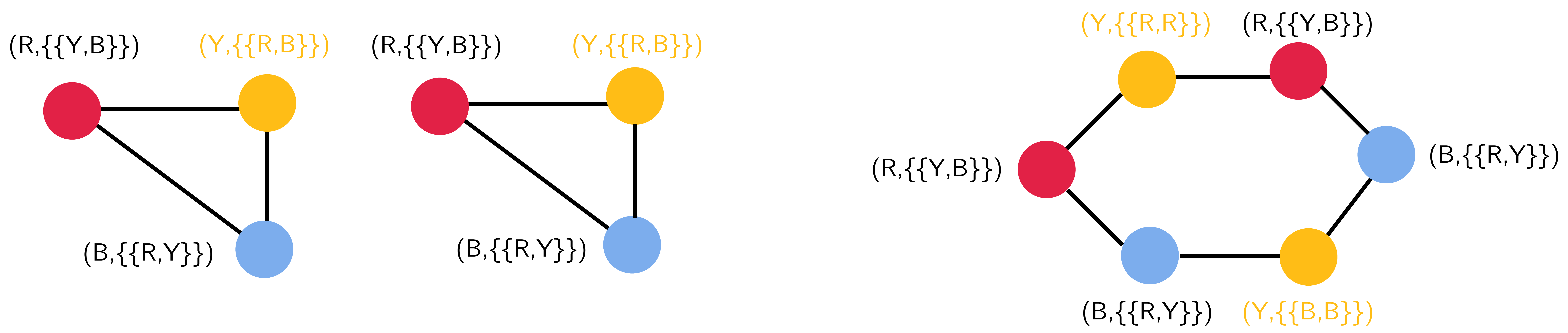




Unique node identifiers are used in understanding the potential and limits of GNNs (Loukas, 2020):

- GNNs with unique node identifiers are Turing-universal with unbounded width.
- GNNs cannot compute well-known functions (i.e., cycle detection, shortest path) unless $d \times w = O(p(n))$, where d is the depth, w is the width, and $p(n)$ is a polynomial of the graph size n.

A Tale of Two Colored Graphs

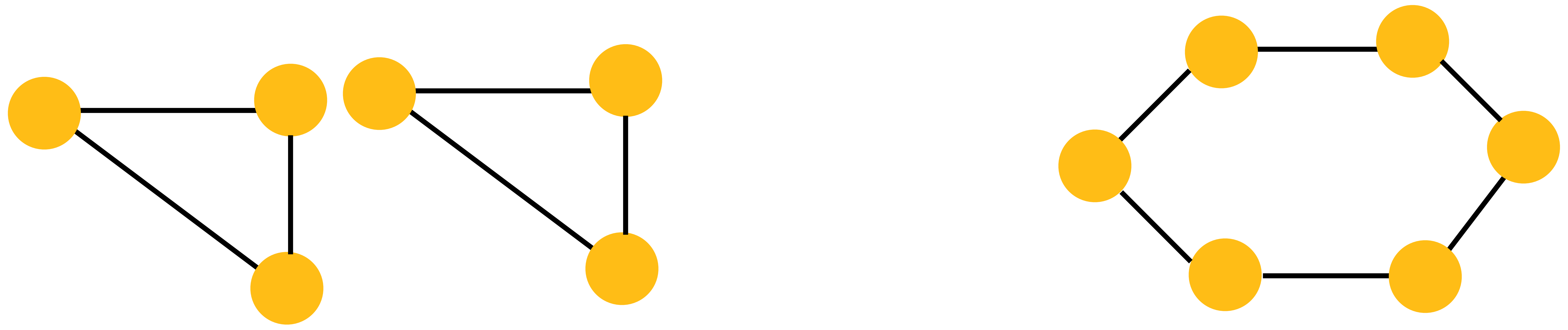


Problems in initializing nodes with unique node identifiers:

- We assign colors (or, identifiers) to nodes which do not necessarily have a meaning.
- We potentially change the meaning of the given node features — potentially losing valuable information.
- These features are deterministic and it is hard to generalize over these structures.

MPNNs with Random Features

Random Node Features and Colored Graphs



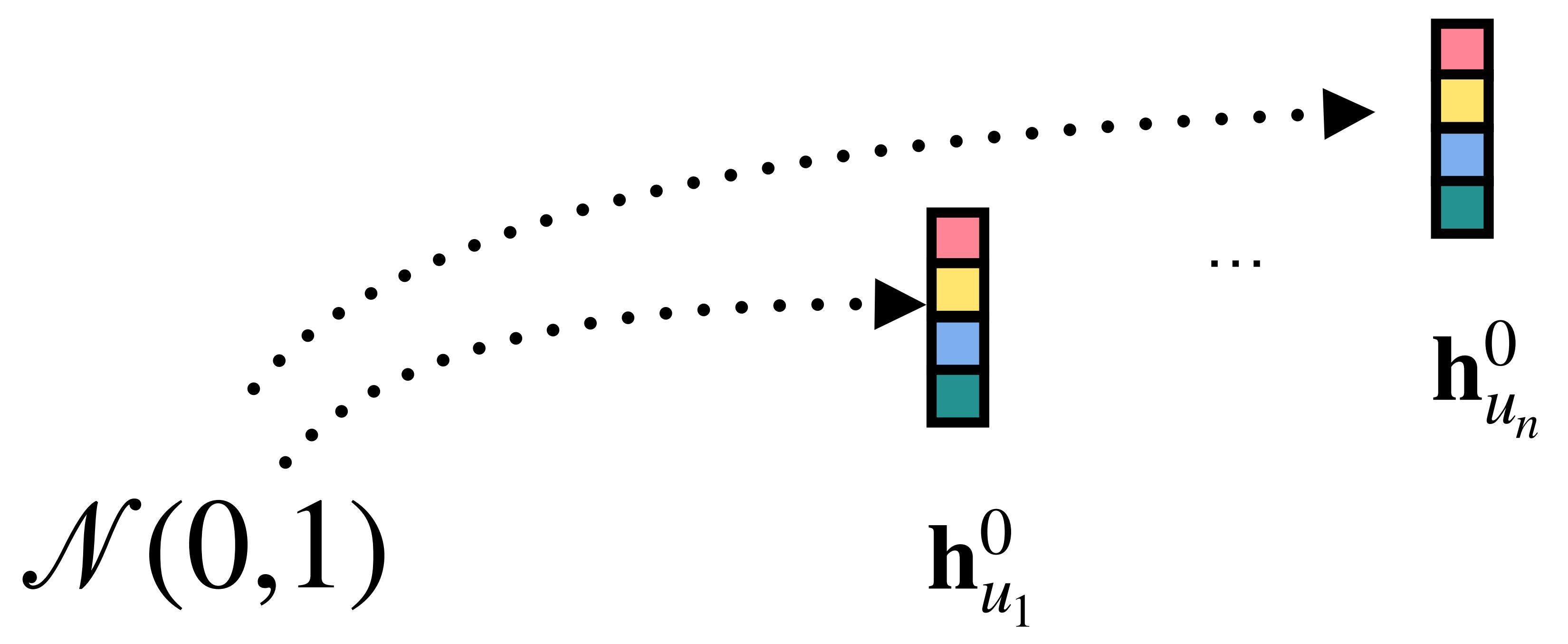
MPNN initialized with identical node features for these graphs - they cannot be distinguished.

MPNN initialized with distinct node features for these graphs - they can be distinguished.

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

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

MPNNs with Random Node Initialization

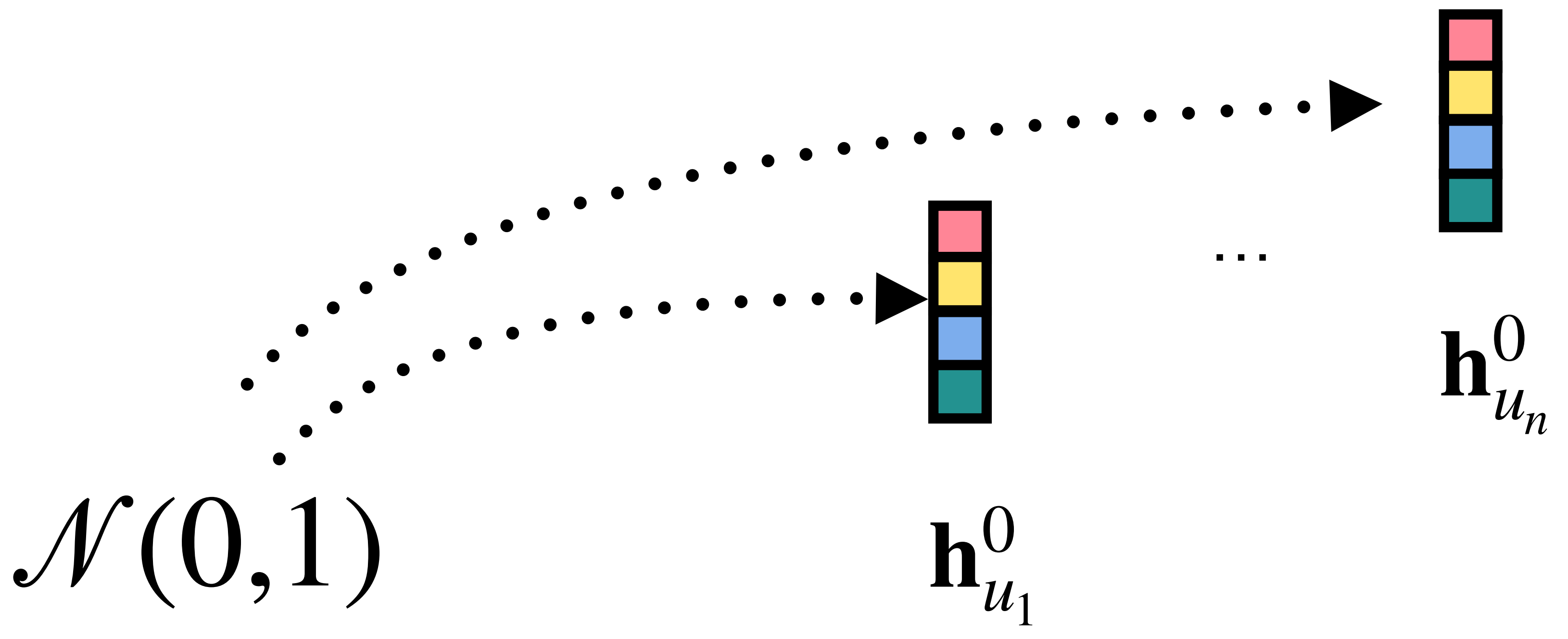


MPNNs with random node initialization: Model trains and runs with (partially) randomized initial node features (Sato et al., 2021; Abboud et al., 2021).

Notation: MPNN-RNI denotes MPNNs with random node initialization, and e.g., M-RNI denotes a specific MPNN model M extended with RNI, e.g., GIN-RNI.

Context: Features refer to node features — and our context is random node initializations.

MPNNs with Random Node Initialization

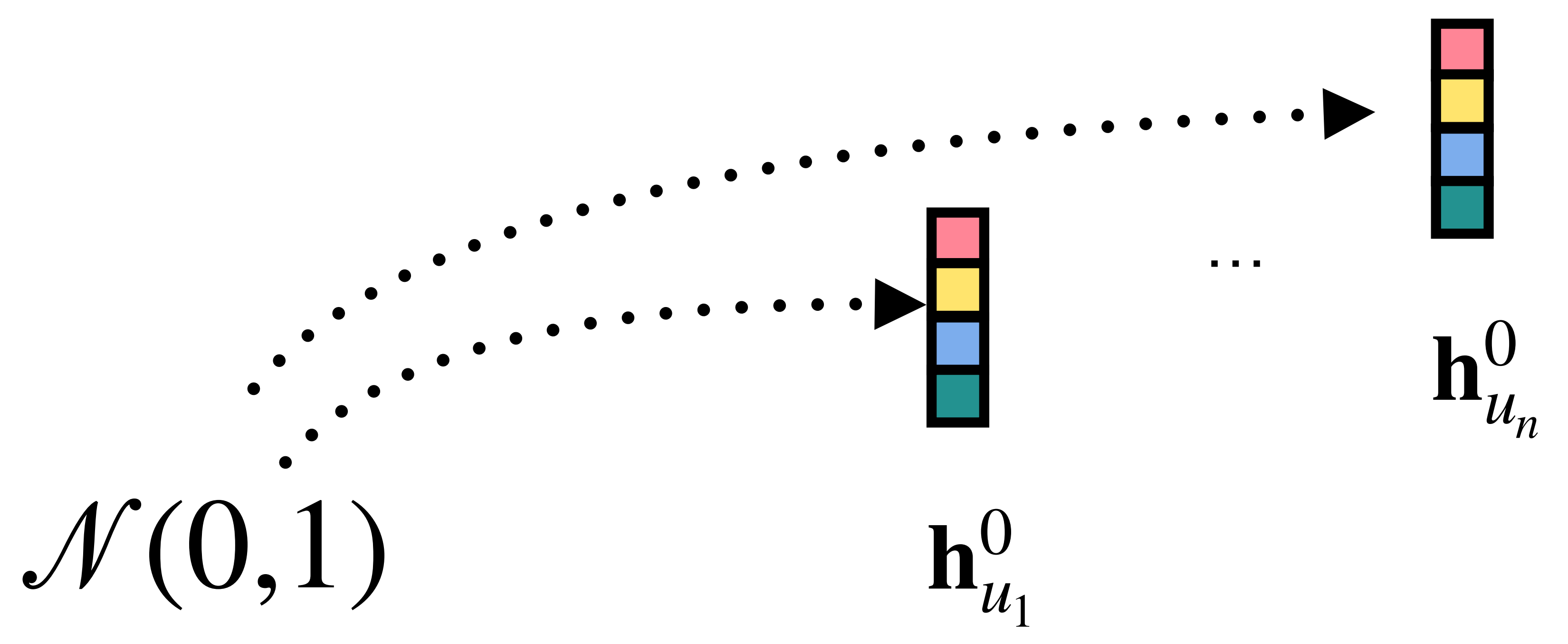


Take-away: GIN-RNI models can detect characteristic sub-graphs in an input graph with high probability.

Detecting substructures (Sato et al., 2021): For a given class of (degree-bounded) graphs $\mathcal{G}$, and every fixed structure (G, v), where $v \in V_G$, states that there exists a parametrization 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) characterizes 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.

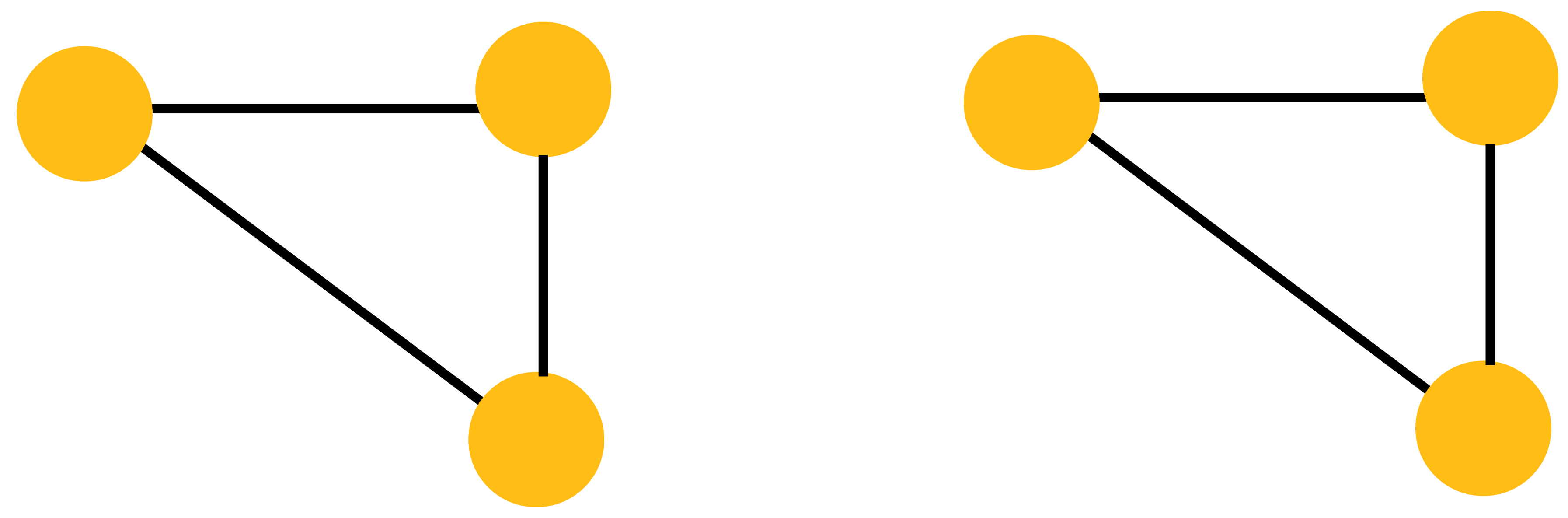
MPNNs with Random Node Initialization



Observation: GIN-RNI models go clearly beyond the capabilities of GINs by this result.

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

Empirical Evaluation for Substructure Detection

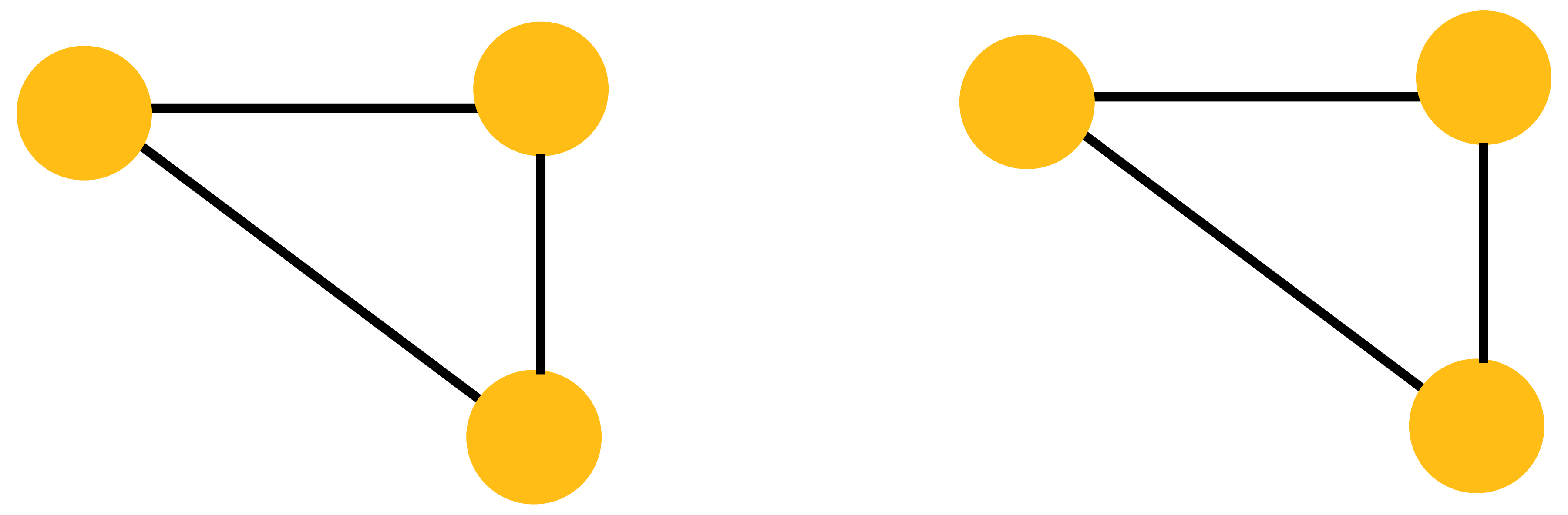


Triangle: Random 3-regular graphs for binary node classification. Training and test set contain 1000 graphs. Training graphs have 20 nodes, and test graphs have 20 nodes for the normal dataset and 100 nodes for the extrapolation dataset. A node v is positive if v has two neighboring nodes that are adjacent to each other.

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

- Unsurprisingly, GINs /GCNs only achieve 50% accuracy on this dataset.
- GIN-RNI achieves >90% accuracy and GCN-RNI >85% accuracy on normal and extrapolation datasets.
- MPNN-RNI models can potentially extrapolate to variable size graphs.

Empirical Evaluation: Real-World



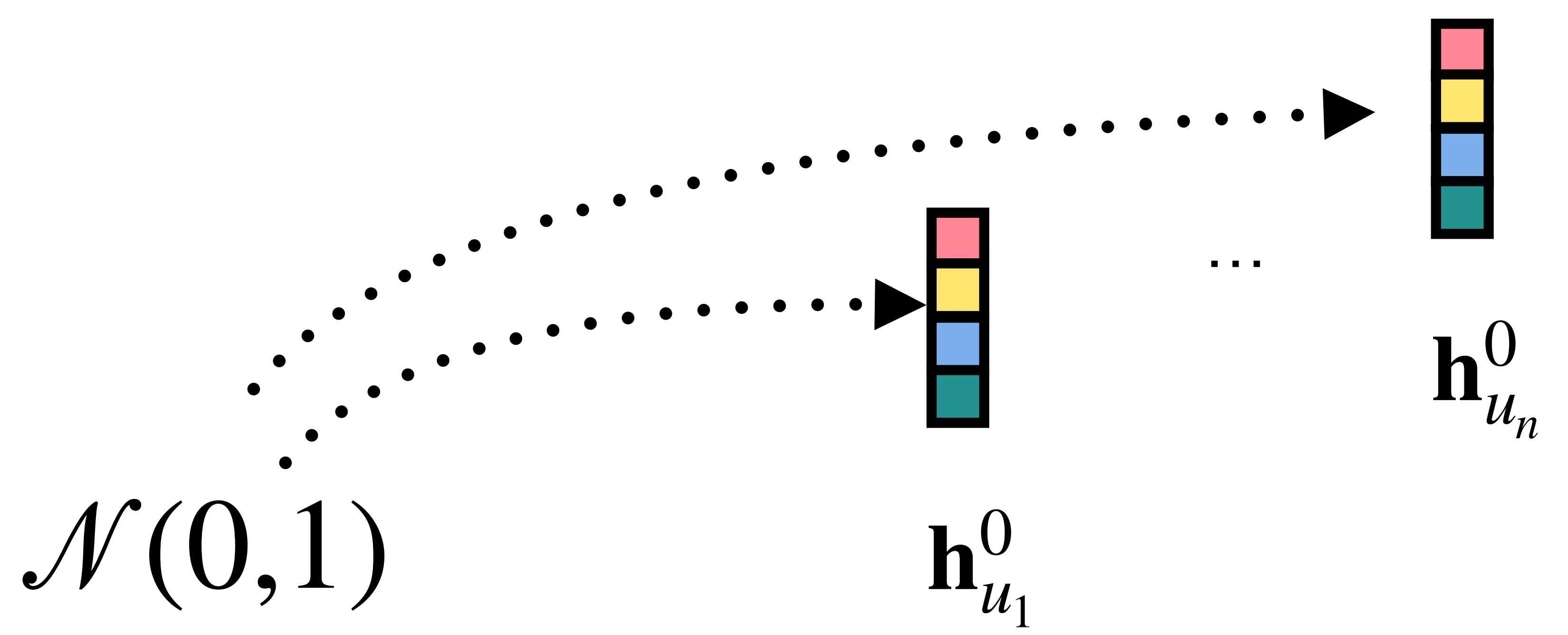
Real-world datasets: MPNN-RNI models perform either similarly to MPNNs, or marginally improve on them using a partial randomization.

Other results (Sato et al., 2021): Inspired by distributed local algorithms also give algorithmic alignment results for certain combinatorial problems that admit such local algorithms.

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

Universality of MPNNs with Random Node Initialization

Expressive Power of MPNNs with Randomization



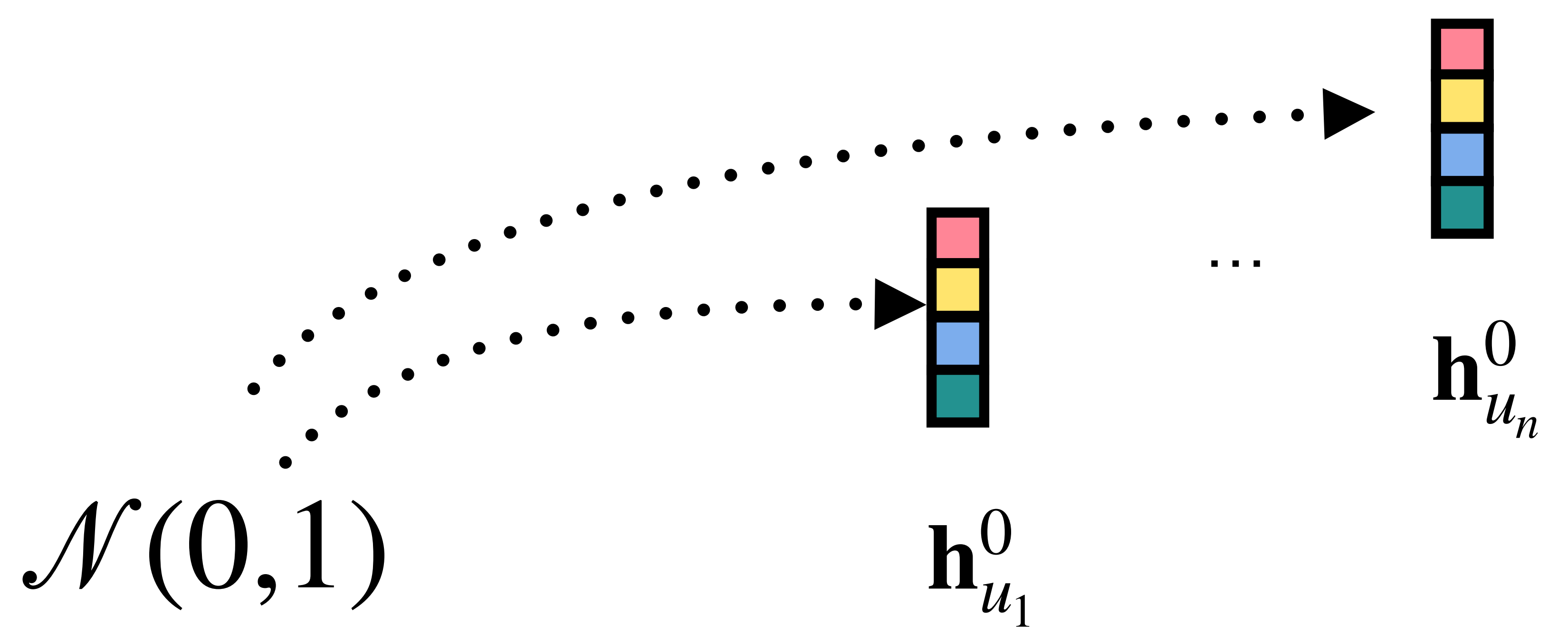
Question: What is the expressive power of MPNN-RNI models, and can they be universal?

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

Let $\mathcal{G}_n$ be the class of all n-vertex graphs, and let us focus on the class of functions of the form $f : \mathcal{G}_n \mapsto \mathbb{R}$.

MPNN-RNI computes a random function - how to approximate functions?

Random Functions and Approximations



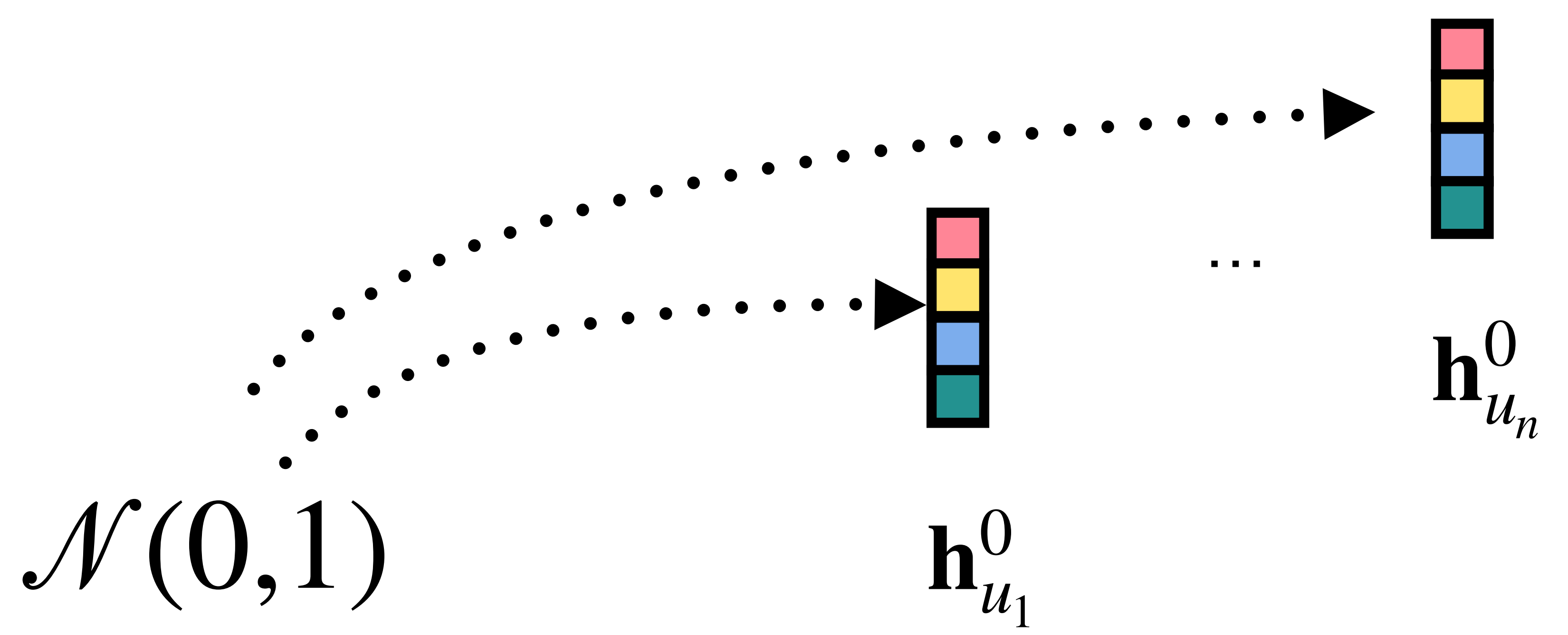
Random function: An MPNN-RNI computes random functions. A randomized function $\mathcal{F}$ that associates with every graph $G \in \mathcal{G}_n$ a random variable $\mathcal{F}(G)$ is an (ϵ, δ)-approximation of f if for all $G \in \mathcal{G}_n$:

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

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

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

A Universality Result



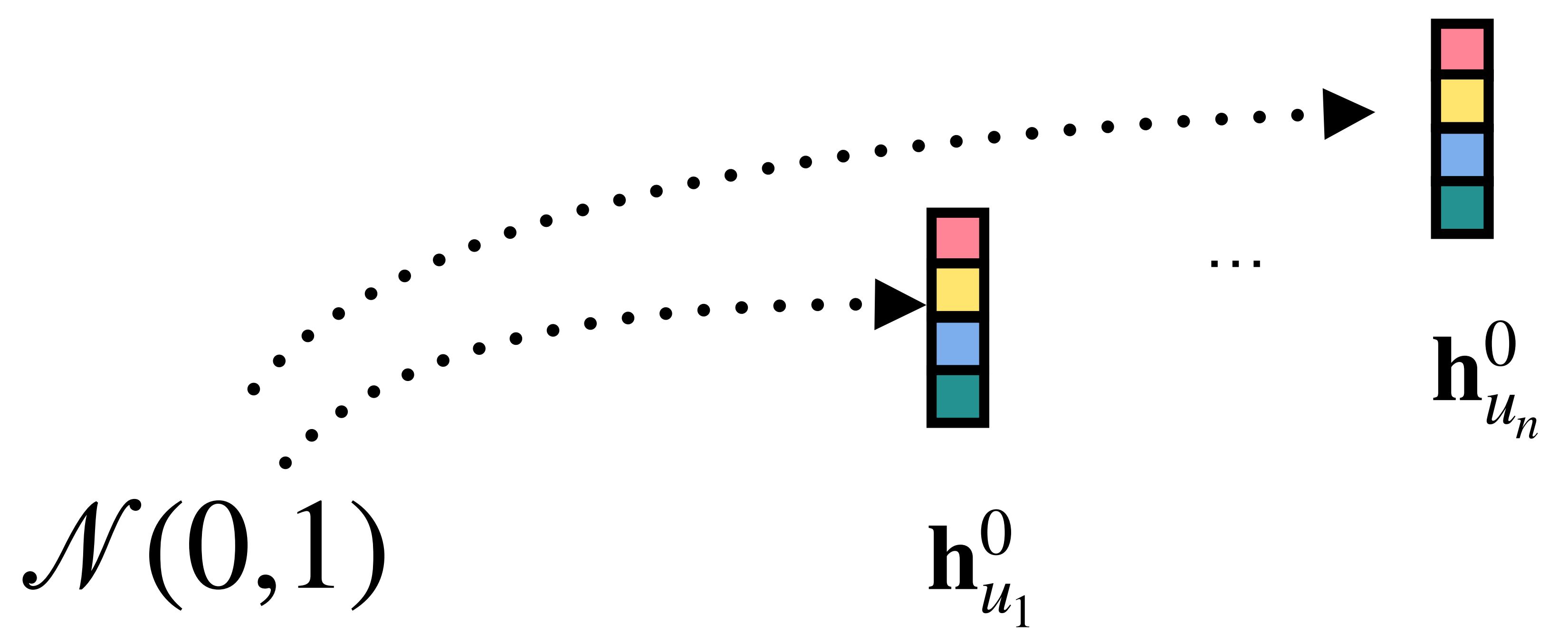
It has been recently shown that MPNN-RNI models are universal:

Theorem (Abboud et al., 2021). 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., 2021). 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



Once this lemma 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.:

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

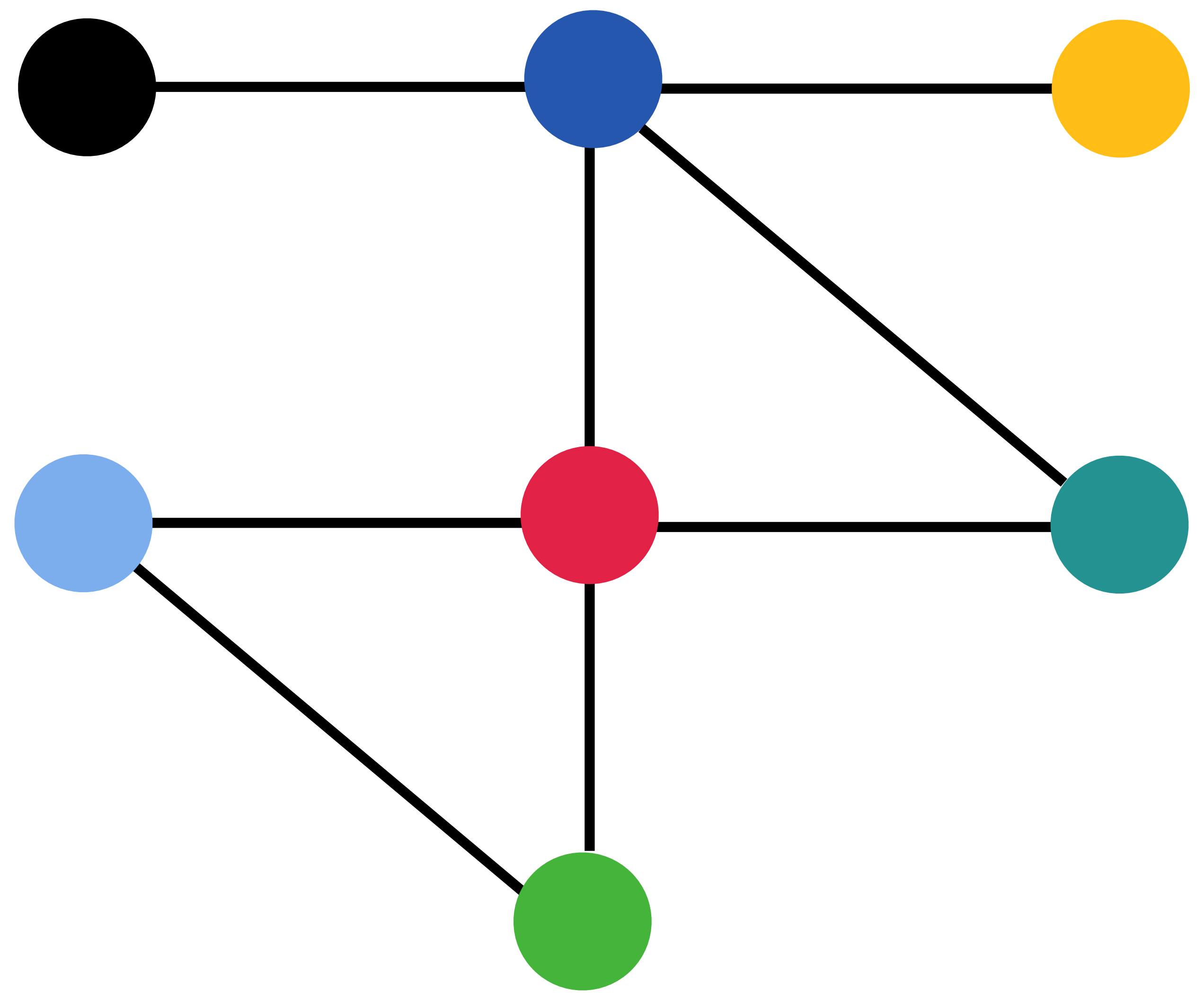
Individualized Graphs

Individualized graphs: Graphs with unique node identifiers, e.g., individualized colored graphs.

Or, a colored graph G is individualized if for any two distinct vertices $v, w \in V_G$ the sets $\pi(v), \pi(w)$ of colors they have are distinct.

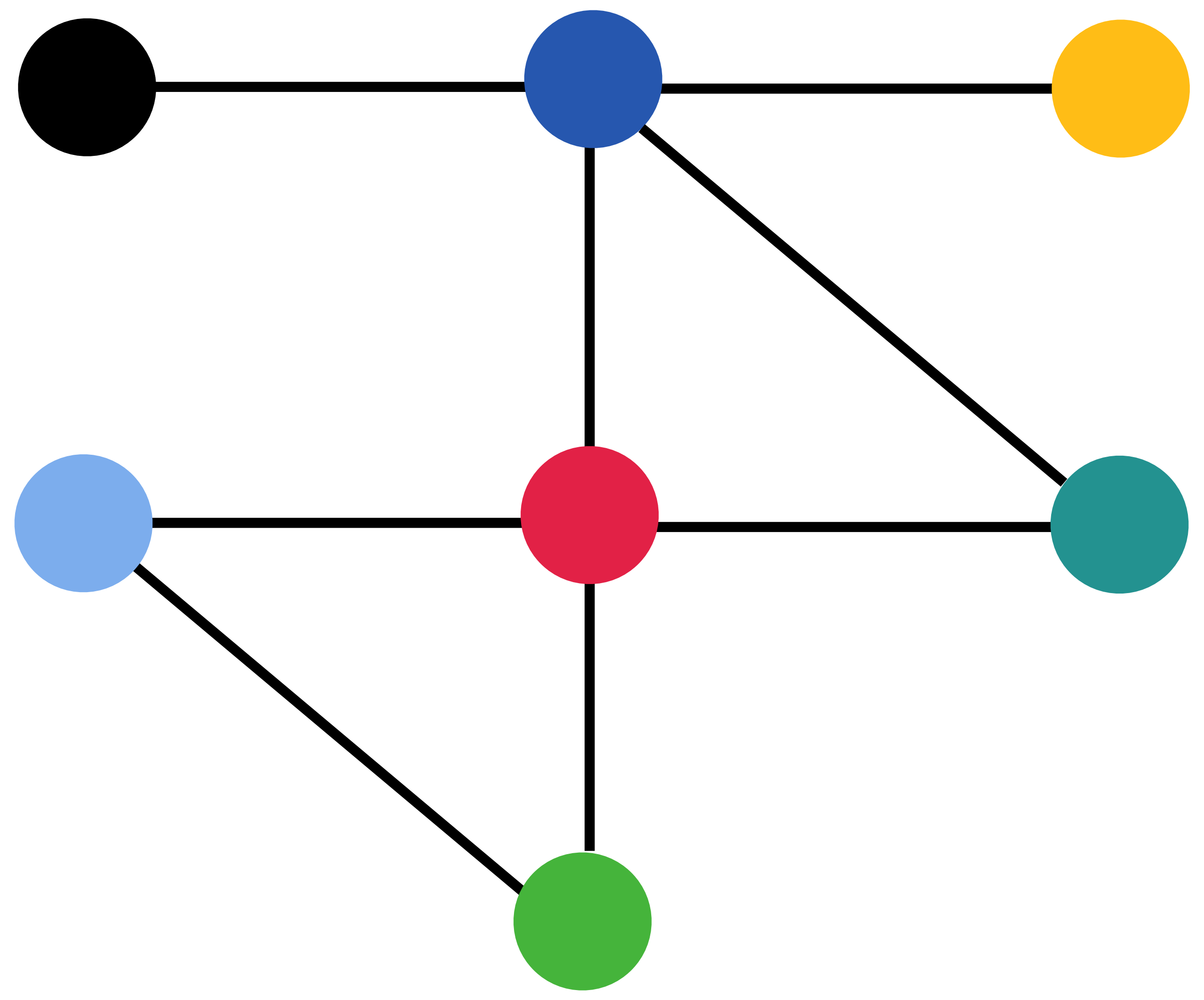
Identifying: A sentence ψ identifies an individualized graph G if for all individualized graphs H:

$$H \models \psi \text{ if and only if } H \text{ is isomorphic to } G$$



Outline of the Result

1. Random node initialization produces individualized graphs with high probability.
2. **Folklore**: For every individualized graph G there is a C^2-sentence ψ that identifies G.
3. To capture Boolean functions over sets of individualized graphs, show that these functions can also be represented by a C^2 sentence, namely the disjunction of all constituent sentences.
4. Leverge the result of Barcelo et al. (2020): Each C^2 graph classifier can be captured by an MPNN (with a global readout).



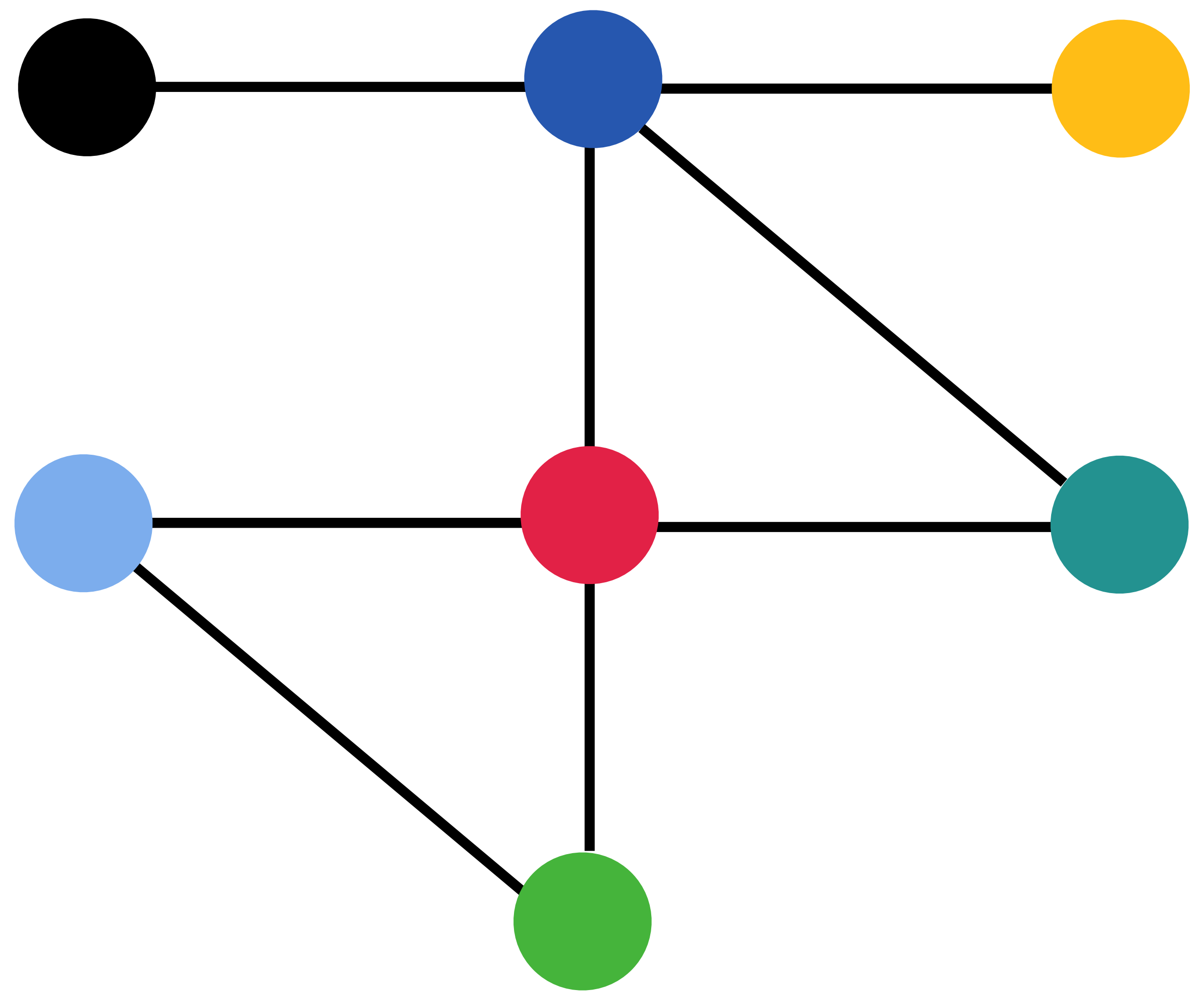
Graph Classification

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: $\Phi(G)$, viewing $\Phi : \mathscr{G} \mapsto \mathbb{B}$.

We can rephrase the result of Barcelo et al., (2020) to graph classification, by pooling:

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.



How Effective is the Construction?

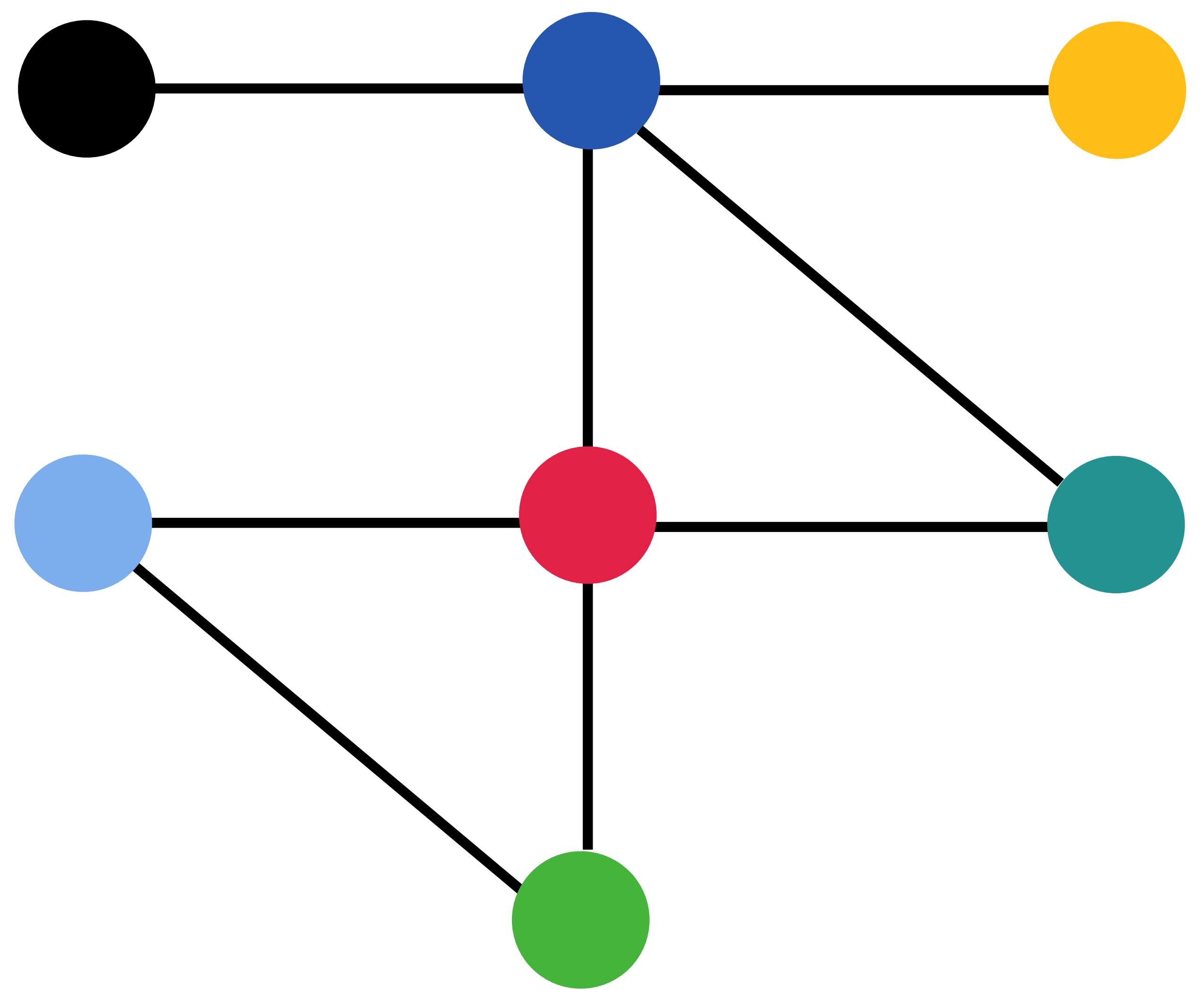
What is the depth and width of the network needed to capture A target function?

- 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.
- For Boolean functions, the size of the MPNN correlates with the descriptive complexity of the logical representation of the target function:
    - 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
- Direct bounds on the size of MPNN-RNI models for special classes of functions: paves the way for a principled and formal analysis of MPNN-RNI models.

Permutation-Invariance

Question: Are MPNN-RNIs permutation-invariant?

- 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.
- Nevertheless, MPNN-RNI computes a random variable (or as generating an output distribution), and this random variable would still be invariant.
- 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.

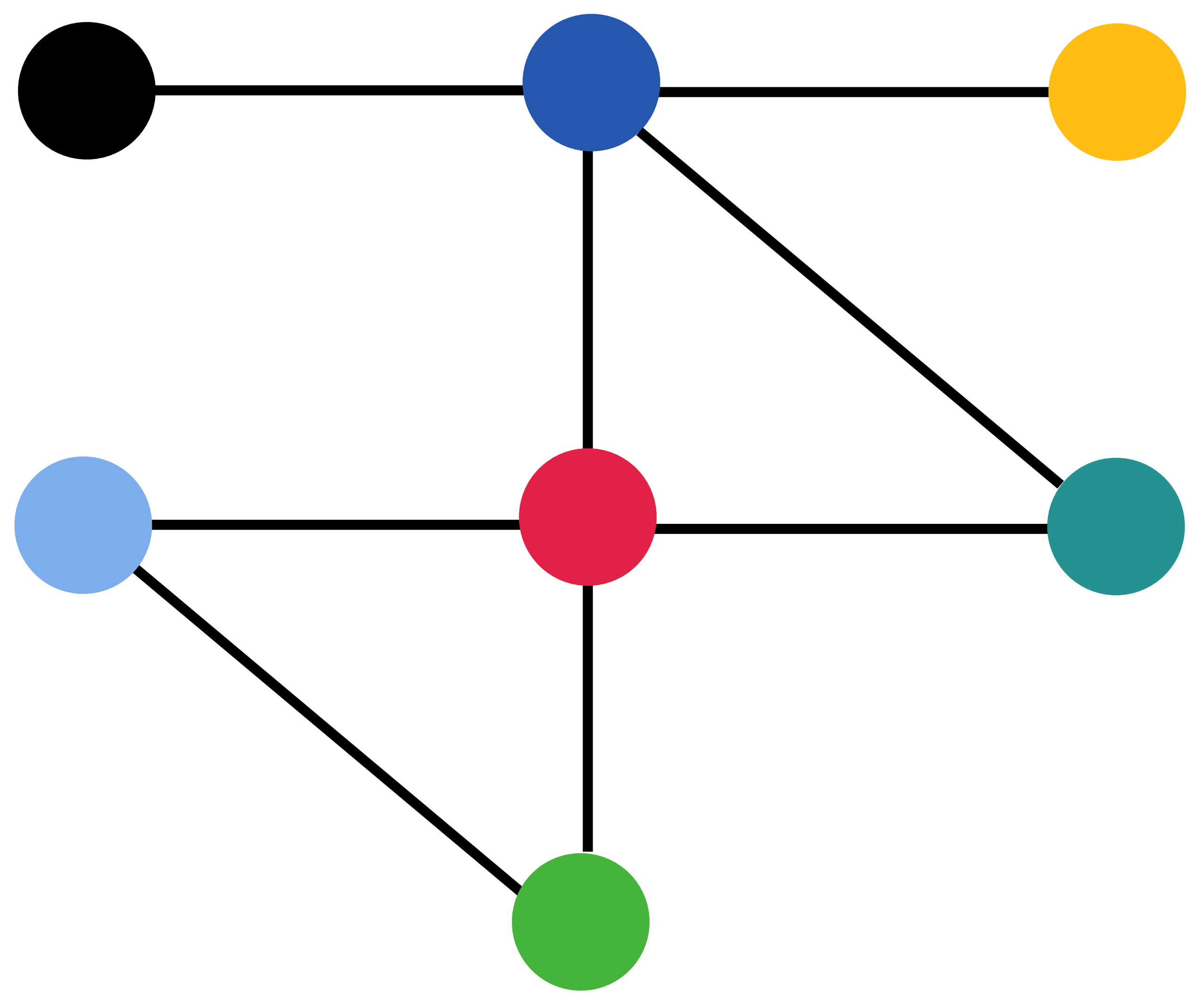


Permutation-Invariance

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

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

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



Benchmarking Expressiveness Evaluation

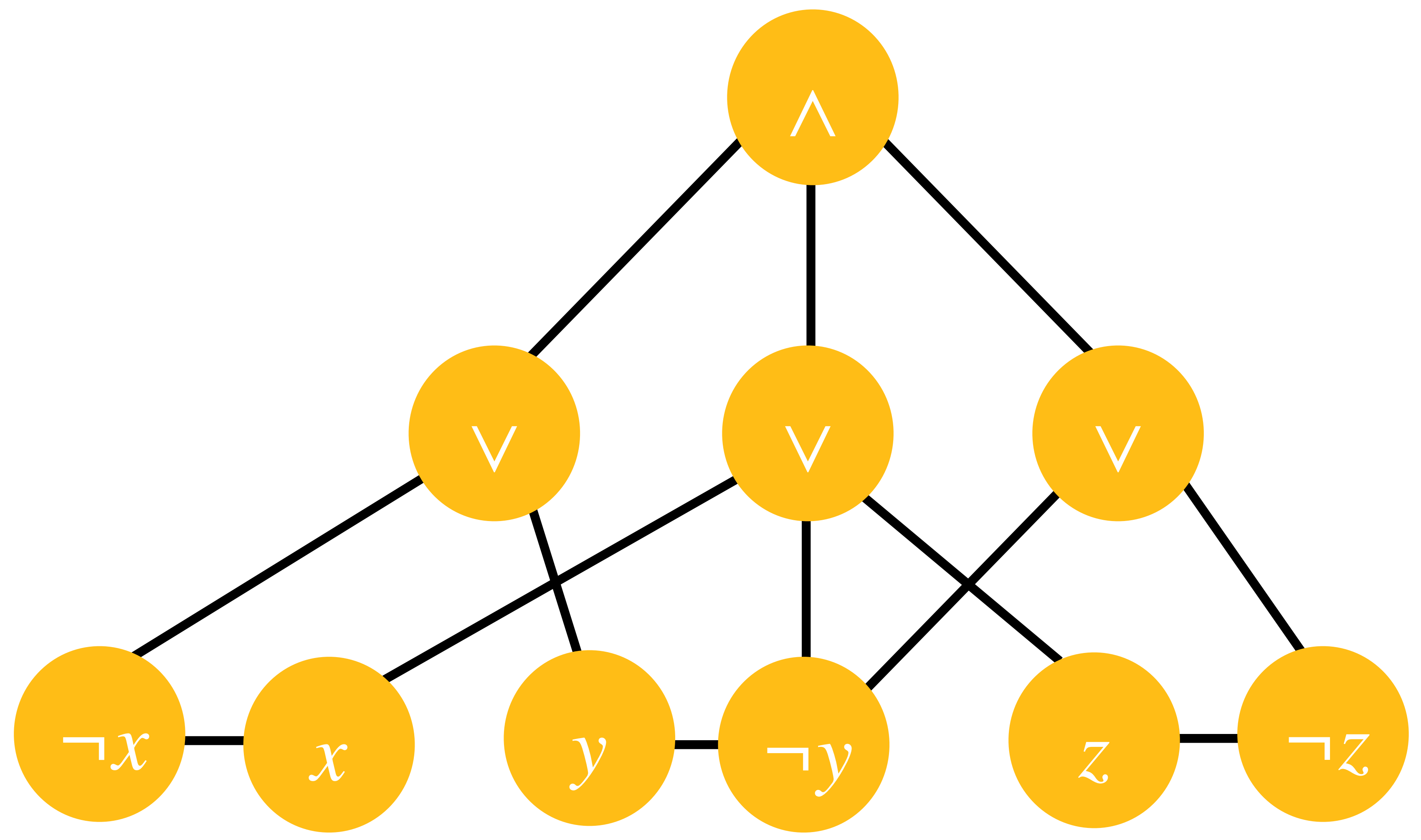
Datasets for Expressiveness Evaluation

EXP dataset (Abboud et al., 2020): Evaluate the expressiveness of GNN models based on the well-known propositional satisfiability (SAT) problem.

SAT: Combinatorial by nature and not local.

Approach: Encode each SAT instance as a graph and formulate the satisfiability problem as a Boolean graph classification problem.

Task: Classify graphs that represent satisfiable instances as true and graphs that represent unsatisfiable instances as false.

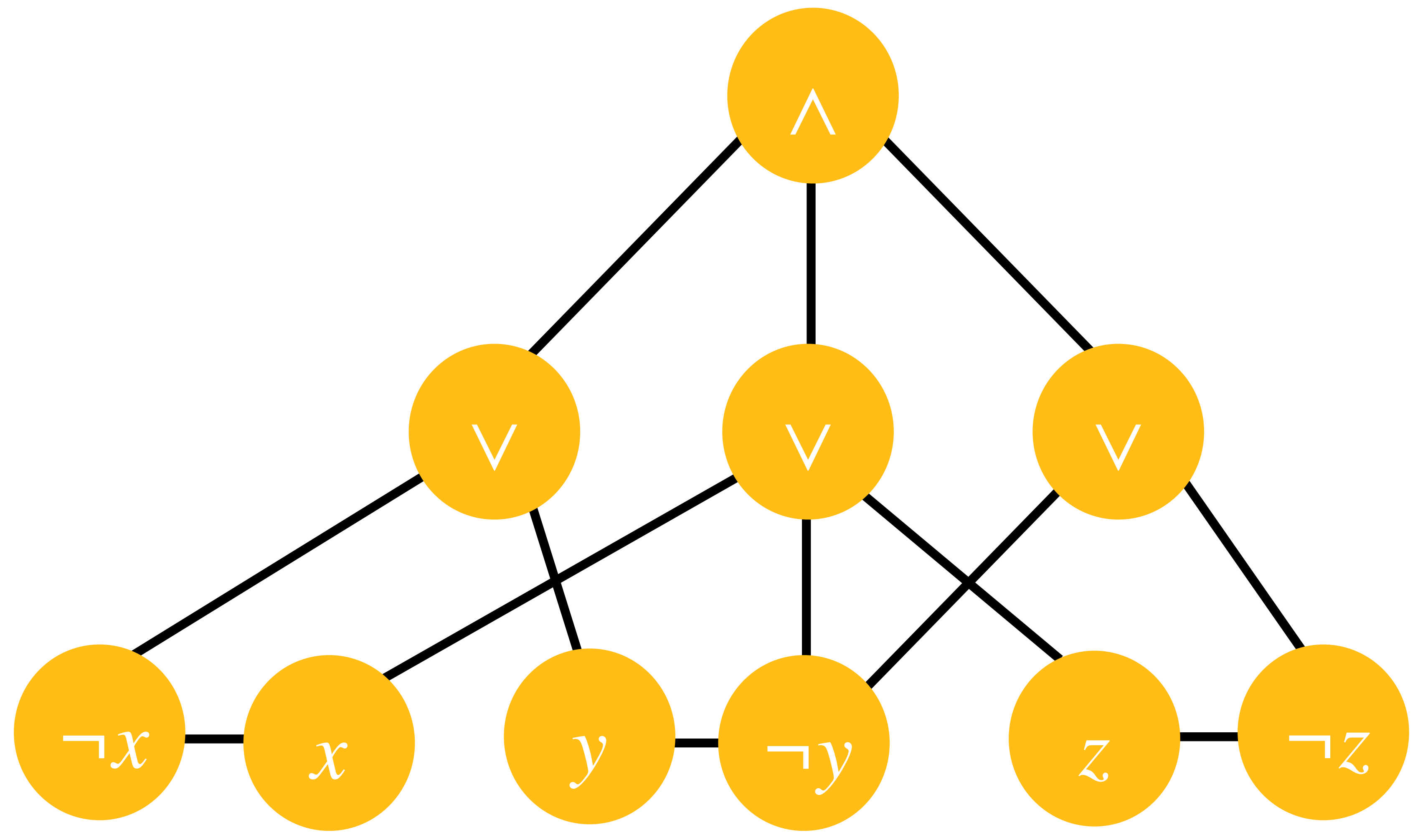


$$\phi = (\neg x \vee y) \wedge (x \vee \neg y \vee z) \wedge (\neg y \vee \neg z)$$

Datasets for Expressiveness Evaluation

EXP consists of a set of graph instances $\{G_1, \dots, G_n, H_1, \dots, H_n\}$: 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 GNNs that have 2-WL expressive power.



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

Random Node Initialization is Powerful

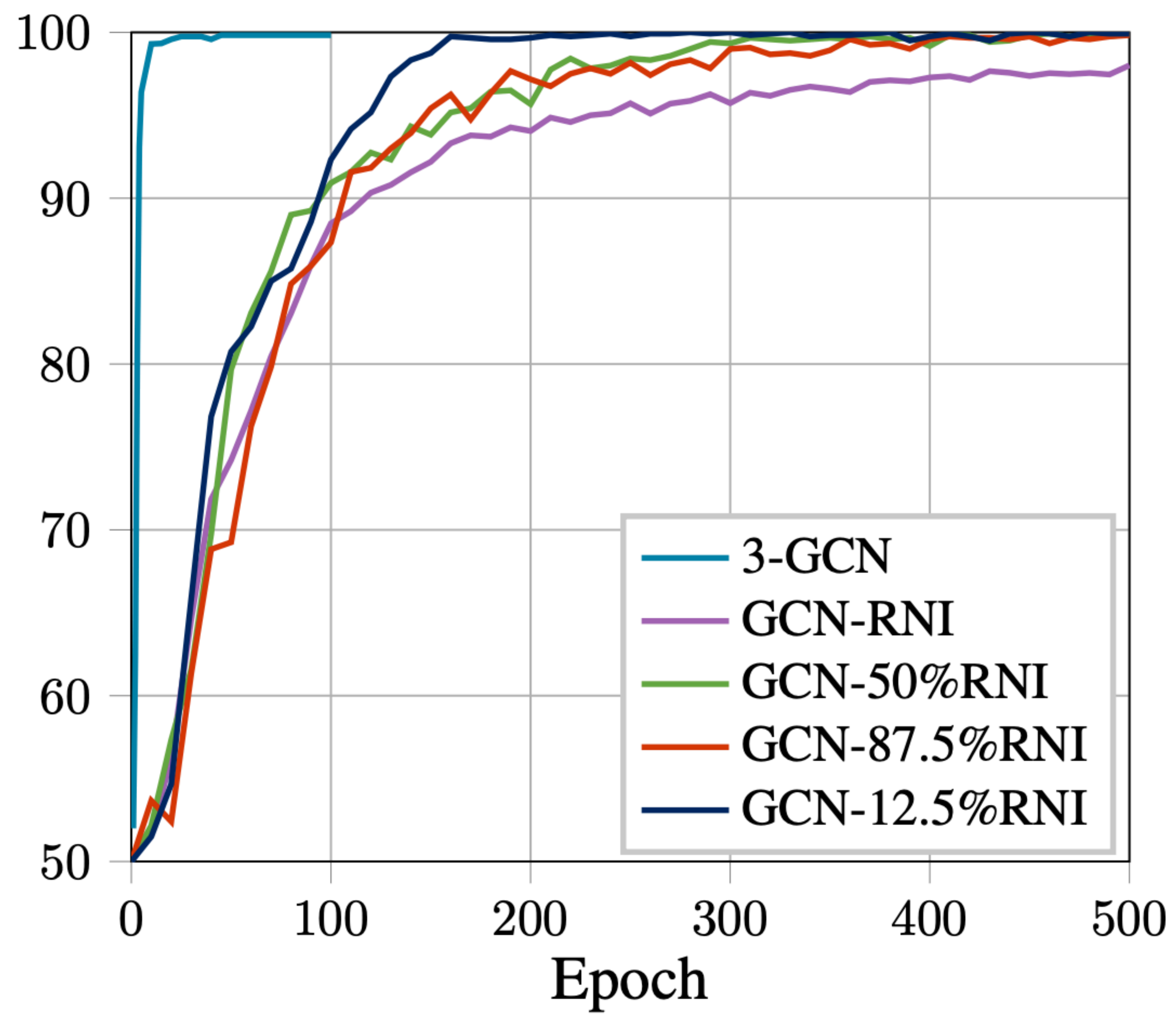
Results: GCN-$x\%$ RNI denotes a GCN-RNI model, where only $x\%$ of initial node embeddings are randomized.

Partial RNI: 100%, 50%, 87.5% and 12.5%, are reported.

Learning curves: Figure 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!



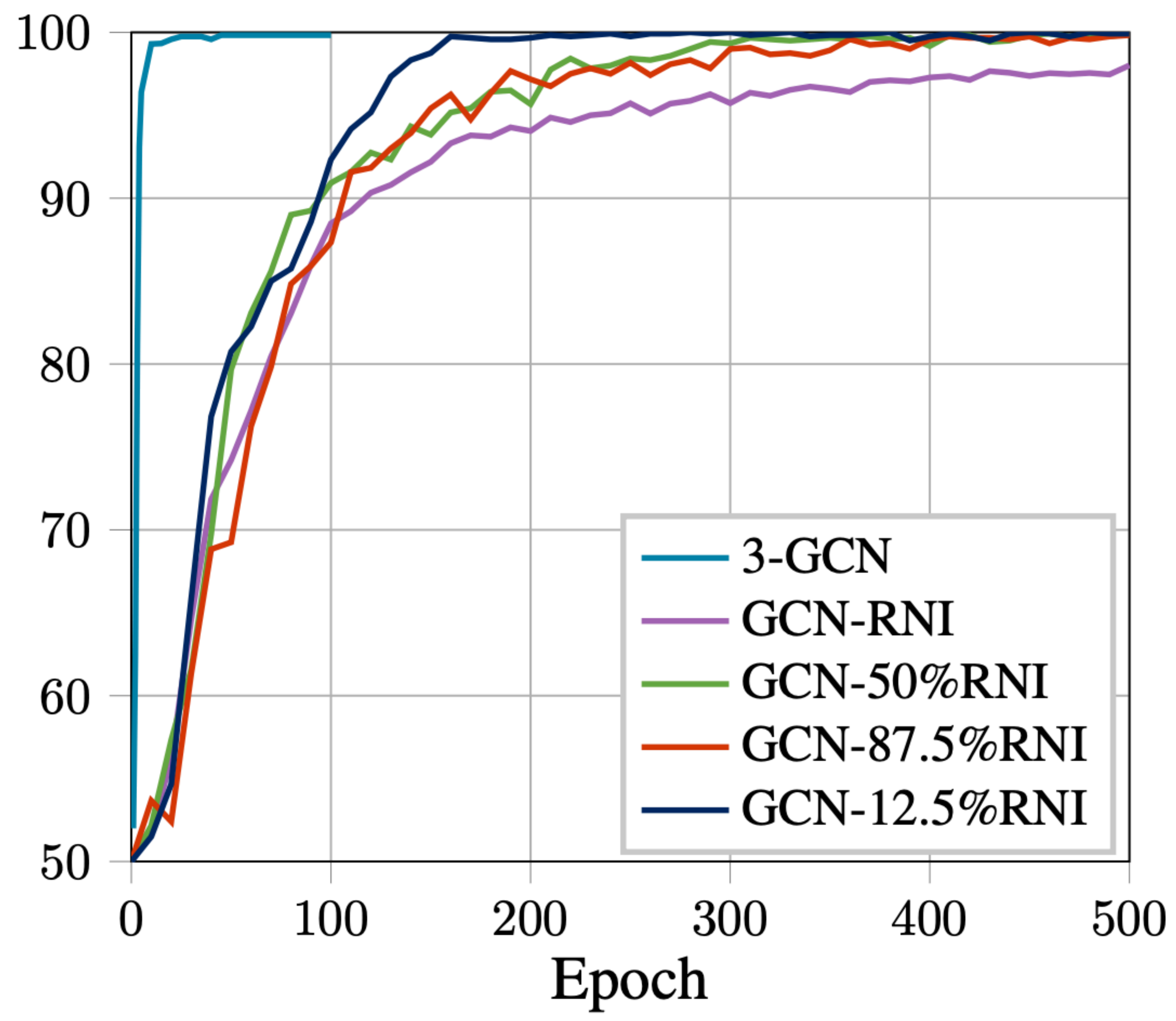
Random Node Initialization is Powerful

Space and time efficiency: MPNN-RNI improves the expressiveness of MPNNs - competitive with higher-order models, despite being significantly less demanding computationally.

Convergence: Model convergence is slower for GCN-RNI and this is the price to pay.

Harder learning task: MPNN-RNI must first leverage RNI to detect structure, then subsequently learn robustness against the variability of RNI.

Observation: Experiments on more variable datasets show that partially randomization achieves the best of both worlds.



Tying Things Together

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

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

Summary

- The quest for expressive models
- Unique node identifiers
- MPNNs with random node initialization and individualized graphs
    - Random node initialization 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
- There are more questions than answers in this context — more research needed!
- Next lecture on generative models: 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. *SDM*, 2021.
- R. Abboud, İ. İ. Ceylan, M. Grohe, T. Lukasiewicz, The Surprising Power of Graph Neural Networks with Random Node Initialization, *IJCAI*, 2021
- M. Grohe. Descriptive Complexity, Canonisation, and Definable Graph Structure Theory. *Cambridge University Press,* 2017.
- A. Loukas, What Graph Neural Networks Cannot Learn: Depth vs Width. *ICLR*, 2020.