# Graph representations for relational deep learning

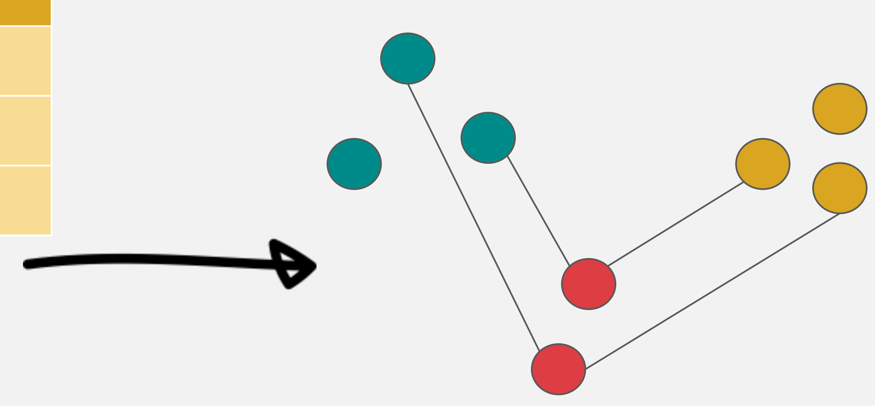
**Tamara Cucumides, Floris Geerts, Pablo Barceló**

## Learning on relational data

| id_A | Name |
|---|---|
| a1 | name_1 |
| a2 | name_2 |
| a3 | name_3 |

| id_B | Tag |
|---|---|
| b1 | tag_1 |
| b2 | tag_2 |
| b3 | tag_3 |

| id_A | id_B |
|---|---|
| a1 | b2 |
| a2 | b3 |



**Relational Deep Learning** transforms **databases into graphs**, eliminating the need for **feature engineering**. Tuples become **nodes**, and relationships define **edges**, enabling **Graph Neural Networks (GNNs)** to learn directly from data.

## Our objective

**Our objective is to investigate how different ways of representing relational data as graphs impact model performance in learning and predictive tasks.**

**In particular,** we aim to determine **which entities in relational data should be embedded as nodes and which should be encoded as features of these nodes**.

This choice directly affects **how information flows** using **message passing,** and impacts **learning performance**.

## Embedding the relational data into graphs

| driver_id | driver | nationality | team_id |
|---|---|---|---|
| 001 | Hamilton | British | 101 |
| 002 | Norris | British | 102 |
| 003 | Verstappen | Dutch | 103 |

| team_id | team | Engine |
|---|---|---|
| 101 | Mercedes | Mercedes |
| 102 | McLaren | Mercedes |
| 103 | Red Bull | Honda |


📌 **Graph Embedding:** Each **tuple** becomes a **node**, **attributes** are **node features**, and **foreign keys** define the **edges**. This structure allows **GNNs** to learn from relational data.

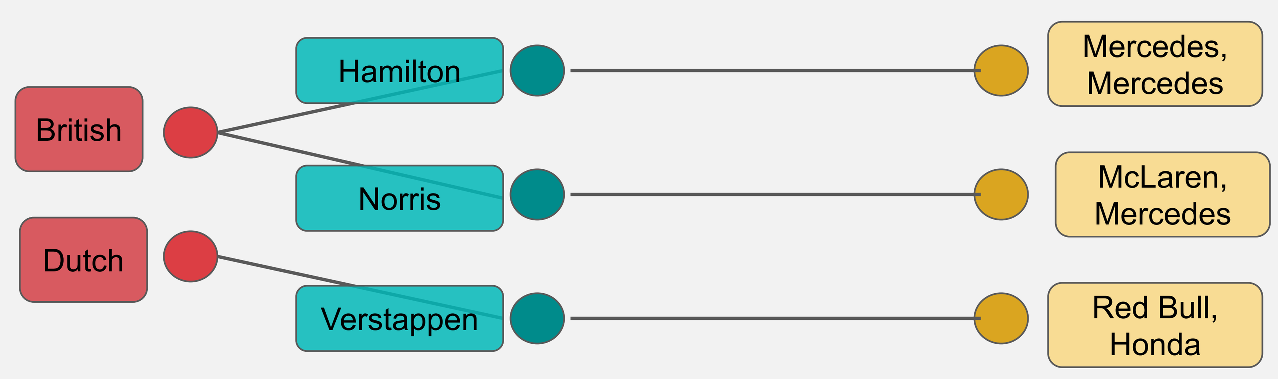
📌 **Message Passing:** Tuples with foreign key relationships become nodes that **share information**, allowing the model to learn from the relational structure of the data.

$$h^{(t)}_{\text{Hamilton}} = \sigma\left(W^{(t)} h^{(t-1)}_{\text{Hamilton}} + W^{(t)}_m h^{(t-1)}_{\text{Mercedes}}\right)$$

| n_id | nationality |
|---|---|
| 010 | British |
| 011 | Dutch |

| driver_id | driver | n_id | team_id |
|---|---|---|---|
| 001 | Hamilton | 010 | 101 |
| 002 | Norris | 010 | 102 |
| 003 | Verstappen | 011 | 103 |

| team_id | team | Engine |
|---|---|---|
| 101 | Mercedes | Mercedes |
| 102 | McLaren | Mercedes |
| 103 | Red Bull | Honda |




We can choose to represent some attributes, like **nationality**, directly as nodes in the graphs, instead of as a node feature.

1. Create a **Nationality** table with unique **n_id** values.
2. Replace nationality in the **Drivers** table with **n_id**.
3. Add a foreign key between **Drivers** and **Nationality**, creating **edges** in the graph.

Now, **driver nodes** interact with their **nationality nodes**, linking drivers of the same nationality.

$$h^{(t)}_{\text{Hamilton}} = \sigma\left(W^{(t)} h^{(t-1)}_{\text{Hamilton}} + W^{(t)}_m h^{(t-1)}_{\text{Mercedes}} + W^{(t)}_n h^{(t-1)}_{\text{British}}\right)$$

## Early experimental results

We experimented with a **Formula 1 dataset (1)**, predicting tasks like driver position, race completion, and podium finishes. We changed the representation of **nationality** from a **node label** to a **node** in the graph.

🚀 **Improved performance** across all tasks.

🧠 **New graph** benefits from more complex models (e.g., deeper GNNs, larger MLPs).

⚠️ **Original graph** suffers from overfitting with more complex models.

**(1) Robinson, J., et al. (2025).** Relbench: A benchmark for deep learning on relational databases. *Advances in Neural Information Processing Systems*, 37, 21330-21341.

## Future work and questions

🌐 Extend results to **new datasets** and **tasks**.

🧠 Develop **criteria** for deciding which attributes should be encoded as **nodes** or **node labels**.

⏳ Explore how **temporal attributes** differ from **static ones** and whether they require special handling in graph construction to better capture **time-dependent relationships**.

🚀 Investigate if these changes enhance the **expressiveness** of the message passing schema.