Exam Information

- **Percentage:** 35% of your course grade

- **Time:** a consecutive, **120-minute** slot from Nov 19, 10:00AM to Nov 20, 09:59AM
  - The make-up exam is 2 days prior

- **Exam Format:** The exam is administered through Gradescope
  - You can typeset your answers in **LaTeX** or handwrite your answers + upload them as **images**
  - The exam should take around 110 minutes, and you have 10 minutes to upload images
There will be 11 questions
- Some questions are easy, and some are harder
- Try to spend 5-15 minutes on each question
- If stuck on a particular question for too long, please skip that question and come back later

Types of questions:
- True/False questions with explanation
- Give examples of graphs
- Comparison of approaches
- Mathematical calculations and derivations

We feel that the exam is medium difficulty
General Advice for the Exam

- We suggest that you **read through all lecture slides** carefully.

- **Topics** that are **important** for the exam:
  - Node centrality measures, PageRank
  - GNN model and design space (e.g., message, aggregation, update)
  - Knowledge graph embeddings, Query2Box, recommender systems (LightGCN)

- **Lectures** that are **important** for the exam: lectures 2, 4, 6, 7, 8, 10, 11, 13
General Advice for the Exam

- We suggest that you **read through all lecture slides** carefully

- Lectures that are relatively unimportant for the exam: lectures 1, 3, 5, 9, 12, 14
  - You can spend less time studying these lectures
  - However, you should still read through them and understand the concepts as there may be miscellaneous questions
We use GNNs to execute the BFS algorithm
- Initially, all nodes have input features 0, except a source node with feature 1
- At every step, nodes reached by BFS have embedding 1, and nodes not reached by BFS have embedding 0
- Describe the message, aggregate, update functions
- Advice: Think from the perspective of nodes in the graph

Homework 1, Q4.6
(1) Message passing

Imagine you are a node in the graph. What information would you tell your neighbors?

- “I have been visited by the BFS algorithm!” or “I have not been visited!”
- Simply pass my embedding to my neighbors

$$\text{message}_{v \rightarrow u} \left( h^{(k-1)}_v, e_{v,u} \right) = h^{(k-1)}_v$$
(2) Aggregation

- What information should you get from your neighbors?
  - I want to know whether any of my neighbors have been visited

- Node $u$ aggregates neighbors’ information via:

$$\text{aggregate} \left( \{ \text{message}_{v \rightarrow u}, \forall v \in \mathcal{N}(u) \} \right) = \max_{v \in \mathcal{N}(u)} \text{message}_{v \rightarrow u}$$
(3) Update

- Don’t forget the **self-link** to the previous embedding for node \( u \)
  - BFS: I am visited if (1) I have been visited, or (2) any of my neighbors has been visited

\[
\text{update} \left( h_u^{(k-1)}, \text{aggregate}(\cdots) \right) = \\
\max \left( h_u^{(k-1)}, \text{aggregate}(\cdots) \right)
\]

- This is one solution to Q4.6, there are alternatives
There are **common patterns** in knowledge graph embeddings

- **Symmetry**: A is married to B, and B is married to A
- **Inverse**: A is teacher of B, and B is student of A
- **Composition**: A is son of B, and C is sister of B, then C is aunt of A

**KG method**: TransE

- Given a triplet $(h, l, t)$, TransE trains entity and relation embeddings to follow the equation $h + l \approx t$

Can we use TransE to model each of the relation patterns?
Given \((h, l, t)\), TransE equation is: \(h + l \approx t\)

**Key question:** For the given relation pattern, what equations should hold true?

**Symmetry:** A is married to B, and B is married to A

Can we use TransE to model symmetry? **No**

- For two triplets \((h, l, t)\) and \((t, l, h)\) to both hold true, we will have: \(h + l \approx t\) and \(t + l \approx h\)
- The only possibility for both equations to be true is if \(l = 0\) and \(h = t\), which is a problem since two different entities should have different embeddings
Given \((h, l, t)\), TransE equation is: \(h + l \approx t\)

**Inverse:** A is teacher to B, and B is student to A

Can we use TransE to model *inverse*? **Yes**

- For two triplets \((h, r1, t)\) and \((t, r2, h)\) to both hold true, we will have: \(h + r1 \approx t\) and \(t + r2 \approx h\)
- It suffices to set the inverse relation \(r2 = -r1\)
Given \((h, l, t)\), TransE equation is: \(h + l \approx t\)

**Composition**: A is son of B, and C is sister of B, then C is aunt of A

Can we use TransE to model *composition*? **Yes**

- Given three triplets, \((a, r1, b)\), \((b, r2, c)\), \((a, r3, c)\), where \(r3\) is the composition of \(r2\) and \(r1\)
- For all triplets to be true, we will have: \(a + r1 \approx b\), \(b + r2 \approx c\), \(a + r3 \approx c\)
- Set \(r3 = r1 + r2\) for composition
To produce an undirected graph $G = (V, E)$, the ER model uses a fixed likelihood to generate edges connecting any pair of nodes:

$$\mathbb{P}[(u, v) \in E] = r, \quad \forall u, v \in V, u \neq v$$

$n = 10$
$r = 1/6$
What is the expected average node degree, $E[d]$, of a graph generated by ER?

- **Key idea**: summing the edge connectivity over nodes to compute the expected node degree

\[
|E| = \frac{1}{2} \sum_{u \in V} \sum_{v \in V \setminus \{u\}} 1 \cdot 1[(u, v) \in E]
\]

\[
E[|E|] = \frac{1}{2} \sum_{u \in V} \sum_{v \in V \setminus \{u\}} 1 \cdot E[1[(u, v) \in E]]
\]

\[
= \frac{1}{2} \sum_{u \in V} \sum_{v \in V \setminus \{u\}} 1 \cdot r
\]

\[
= \frac{|V|(|V| - 1)}{2} r
\]

\[
= \binom{|V|}{2} r
\]

\[
d = \frac{2|E|}{|V|}
\]

\[
E[d] = \frac{2E[|E|]}{|V|}
\]

\[
= (|V| - 1)r
\]
All the Best

- All the best with your exam preparation!