

1 Question 1

LSTM are not permutation invariant. Indeed, they are designed to process time series sequences (so sequences that are arranged in chronological order). If we permute the input, the sequence will give a different output in general (we can see it with the operations made by the LSTM which treats the inputs in ascending order). While working with sets, permutation should not change the final results, as sets are unordered. That's why it is not recommended to use LSTM on sets, because if we give the same set but with elements in different positions, we would get different outputs.

2 Question 2

The DeepSets architecture process each element of a set separately and then sum the outputs, to obtain a unique output for the whole set. Each element of the set can give the same amount of information to the final output.

GNN for graph-level tasks uses two message passing layers to retrieve information from neighbors nodes in a graph, and finally sum them to have a unique output for the whole graph.

The difference between these two models is that GNN uses the graph structure and edge informations to process the output, where DeepSets model only uses the elements of the set.

Mathematically, graphs without edges and finite sets are in bijection. A graph without edges can be identified with a set. But the way we are processing the sets in the DeepSets model is treating each element separately before summing them, and the way we are processing the graphs on GNN emphasizes the

3 Question 3

For the first part of the question, we first treat homophilic graphs.

If we take the matrix $P = \begin{bmatrix} 0.8 & 0.05 \\ 0.05 & 0.8 \end{bmatrix}$ given in the lab topic, we'll have far more edges between nodes in same community compared to the number of edges linking nodes of different communities.

More generally, to sample homophilic graphs, we can take a matrix P such that the coefficients verify $P_{ij} = \alpha P_{kk}$ (for $i \neq j$ and α far lower than 1).

Then, we notice that for heterophilic graphs, we want to have the inverse phenomenon (far more edges between nodes from different communities), and we can sample such a graph from the matrix $P = \begin{bmatrix} 0.05 & 0.8 \\ 0.8 & 0.05 \end{bmatrix}$. More generally, to sample heterophilic graphs, we can take a matrix P such that the coefficients verify $\alpha P_{ij} = P_{kk}$ (for $i \neq j$ and α far lower than 1).

We first compute the expected number of edge linking a given node to nodes of other communities. As there are 15 nodes in different communities, this expected number is $(15 \cdot 0.05) = 0.75$.

To get the expected number of edges between nodes of different communities, as there are 20 nodes which plays the same role in a graph, we have to multiply the previous value by 10 and then divide it by 2 because the graph is undirected (and if we were not doing this, we would count the edges two times):

$$\frac{1}{2} 20 \cdot (15 \cdot 0.05) = 7.5$$

4 Question 4

The Mean Square Error loss ($L = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n (\hat{A}_{ij} - A_{ij})^2$) would be more suitable, as it will encourage the neural network to lower the absolute difference between A_{ij} and \hat{A}_{ij} .