1 Question 1

LSTMs are not permutation invariant. They are a kind of RNN which are inherently sensible to permutations and depend on the order of the input sequence. They maintain a hidden states that is computed based on the previous hidden state and current input. Indeed, we have the following equations for the cell state, candidate cell state and hidden state are :

$$\tilde{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c)$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$$

$$h_t = o_t * tanh(c_t)$$

We clearly see that h_t depends on h_{t-1}.

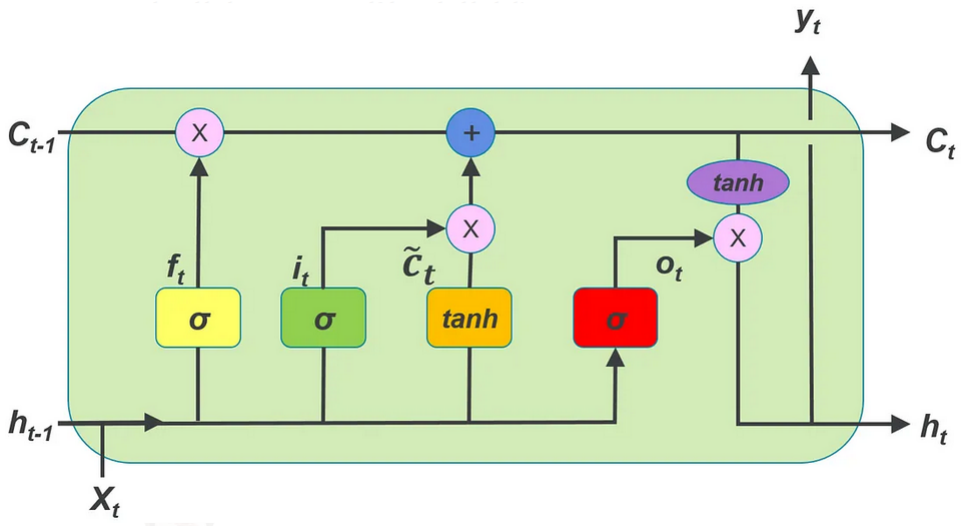

Figure 1: LSTM architecture

Therefore, using a LSTM model for sets that are not order sensitive is not recommended as there will not be the same output depending on the order of the sets.

2 Question 2

The GNNs models we implemented during lab6 explicitly models graph structures while DeepSets do set processing whithout an explicit consideration of graph structures. In GNNs there are message-passing layers to aggregate information from neighboring nodes in a graph and a readout function to aggregate information from all nodes to produce a graph-level representation. On the other hand, Deepset doesn't consider the input as graph but rather as an unordered set of elements. Deepsets is also permutation invariant so it will produce the same output no matter the order of the input.

Both sets and graphs without edges represent collections of objects, but they do so in different ways. Sets emphasize the unordered nature of the collection, while graphs without edges highlight the isolation of the objects.

3 Question 3

In general, an edge probability matrix P where $P_{ii} > P_{ij} (i \neq j)$ allows to sample homophilic graphs (higher probability for edges in communities rather than between communuties. On the contrary an edge probability matrix P where $P_{ii} < P_{ij} (i \neq j)$ allows to sample heterophilic graphs.

We give the following exemples based on the P matrix given in the handout :

- to sample **homophilic** graphs, we could use the following P edge probability matrix :

$$P_{i,j} = \begin{cases} 0.8 & \text{if } i = j \\ 0.05 & \text{otherwise} \end{cases}$$

  giving :

$$P = \begin{bmatrix} 0.8 & 0.05 \\ 0.05 & 0.8 \end{bmatrix}$$

- to sample **heterophilic** graphs, we could use the following P edge probability matrix :

$$P_{i,j} = \begin{cases} 0.05 & \text{if } i = j \\ 0.8 & \text{otherwise} \end{cases}$$

  giving :

$$P = \begin{bmatrix} 0.05 & 0.8 \\ 0.8 & 0.05 \end{bmatrix}$$

  We now consider stochastic block model with $n = 20$ nodes containing $r = 4$ blocks of 5 nodes each and with edge probability matrix P such as :

$$P = \begin{bmatrix} 0.8 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.8 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.8 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.8 \end{bmatrix}$$

We want to determine the expected number of edges between nodes in different blocks. A node from a community (block) has a probability of 0.05 to have an edge with a node from a different community. If we consider a node N from a specific community, the number of edges between node N and nodes from an other community follows a binomial distribution $B(5, 0.05)$.

Therefore, the expected number of edges between one node and another block is :

$$E[\text{\# edges 1 node - 1 block }] = 5 * 0.05 = 0.25$$

Then, the expected number of edgest between one block and another block is :

$$E[\text{\# edges 1 block - 1 block}] = 5 * E[\text{\# edges 1 node - 1 block}] = 1.25$$

Finally, the expected number of edges between nodes from differents communities is :

$$\begin{aligned} E[\text{\# edges nodes from different blocks}] &= 3 \times E[\text{\# edges 1 block - 1 block }] \\ &+ 2 \times E[\text{\# edges 1 block - 1 block }] \\ &+ 1 \times E[\text{\# edges 1 block - 1 block }] \\ &= 7.5 \end{aligned}$$

In this calculation, we count the expected number of edges between block B_1 and blocks B_2, B_3 and B_4 (hence the 3 factor). Then we count the expected number of edges between block B_2 and blocks B_3 and B_4 (hence factor 2 because edges with B_1 were already counted. Finally we count the expected number of edges between block B_3 and B_4.

4 Question 4

A loss function better suited for weighted graphs might be the Mean Square Error (MSE). The MSE would measure the average squared difference between the corresponding entries of the original adjacency matrix A and the reconstructed adjacency matrix $\hat{A}$ and can take continuous values as input.

$$L = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} (A_{ij} - \hat{A}_{ij})^2$$

However we might want to consider using