1 Question 1

For the task of summing a multiset of integers using DeepSets, the optimal parameters are straightforward. The embedding layer $\phi(x)$ should learn an identity transformation, implying weights corresponding to an identity matrix and zero biases. The aggregation step computes a direct sum with no parameters to learn. Finally, the last layer ρ should perform a linear transformation with a weight of 1 and zero bias, ensuring the output matches the sum of the multiset elements.

2 Question 2

The DeepSets model operates as:

$$f(X) = \rho\left(\sum_{x \in X} \phi(x)\right), \quad \phi(x) = \tanh(W_2 \cdot (W_1 \cdot x + b_1) + b_2).$$

We choose the following weights and biases:

$$W_1 = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}, \quad b_1 = \begin{bmatrix} 0.5 \\ -0.2 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1 \end{bmatrix}, \quad b_2 = \begin{bmatrix} -0.1 \\ 0.3 \end{bmatrix}, \quad W_3 = [1, -1], \quad b_3 = 0.1.$$

For X_1: We compute $\phi(x)$ for each element in $X_1 = \{[1.2, -0.7]^\top, [-0.8, 0.5]^\top\}$:

$$\phi([1.2, -0.7]^\top) = \tanh\left(W_2 \cdot (W_1 \cdot [1.2, -0.7]^\top + b_1) + b_2\right),$$

where:

$$W_1 \cdot [1.2, -0.7]^\top = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1.2 \\ -0.7 \end{bmatrix} = \begin{bmatrix} 2.4 - 0.7 \\ -1.2 - 0.7 \end{bmatrix} = \begin{bmatrix} 1.7 \\ -1.9 \end{bmatrix}.$$

Adding b_1, we have:

$$W_1 \cdot [1.2, -0.7]^\top + b_1 = \begin{bmatrix} 1.7 \\ -1.9 \end{bmatrix} + \begin{bmatrix} 0.5 \\ -0.2 \end{bmatrix} = \begin{bmatrix} 2.2 \\ -2.1 \end{bmatrix}.$$

Next, applying W_2 and b_2:

$$W_2 \cdot \begin{bmatrix} 2.2 \\ -2.1 \end{bmatrix} + b_2 = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1 \end{bmatrix} \cdot \begin{bmatrix} 2.2 \\ -2.1 \end{bmatrix} + \begin{bmatrix} -0.1 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 2.2 - 1.05 \\ -1.1 - 2.1 \end{bmatrix} + \begin{bmatrix} -0.1 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 1.05 \\ -2.9 \end{bmatrix}.$$

Finally, applying the tanh activation:

$$\phi([1.2, -0.7]^\top) = \tanh \begin{bmatrix} 1.05 \\ -2.9 \end{bmatrix} = \begin{bmatrix} 0.781 \\ -0.993 \end{bmatrix}.$$

Similarly, for $\phi([-0.8, 0.5]^\top)$:

$$W_1 \cdot [-0.8, 0.5]^\top = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} -0.8 \\ 0.5 \end{bmatrix} = \begin{bmatrix} -1.6 + 0.5 \\ 0.8 + 0.5 \end{bmatrix} = \begin{bmatrix} -1.1 \\ 1.3 \end{bmatrix}.$$

Adding b_1:

$$W_1 \cdot [-0.8, 0.5]^\top + b_1 = \begin{bmatrix} -1.1 \\ 1.3 \end{bmatrix} + \begin{bmatrix} 0.5 \\ -0.2 \end{bmatrix} = \begin{bmatrix} -0.6 \\ 1.1 \end{bmatrix}.$$

Applying W_2 and b_2:

$$W_2 \cdot \begin{bmatrix} -0.6 \\ 1.1 \end{bmatrix} + b_2 = \begin{bmatrix} 1 & 0.5 \\ -0.5 & 1 \end{bmatrix} \cdot \begin{bmatrix} -0.6 \\ 1.1 \end{bmatrix} + \begin{bmatrix} -0.1 \\ 0.3 \end{bmatrix} = \begin{bmatrix} -0.6 + 0.55 \\ 0.3 + 1.1 \end{bmatrix} = \begin{bmatrix} -0.05 \\ 1.4 \end{bmatrix}.$$

Applying tanh:

$$\phi([-0.8, 0.5]^\top) = \tanh \begin{bmatrix} -0.05 \\ 1.4 \end{bmatrix} = \begin{bmatrix} -0.050 \\ 0.886 \end{bmatrix}.$$

Summing over X_1:

$$\sum_{x \in X_1} \phi(x) = \begin{bmatrix} 0.781 \\ -0.993 \end{bmatrix} + \begin{bmatrix} -0.050 \\ 0.886 \end{bmatrix} = \begin{bmatrix} 0.731 \\ -0.107 \end{bmatrix}.$$

Finally:

$$f(X_1) = W_3 \cdot \begin{bmatrix} 0.731 \\ -0.107 \end{bmatrix} + b_3 = (1)(0.731) + (-1)(-0.107) + 0.1 = 0.938.$$

For X_2: Similar calculations for $X_2 = \{[0.2, -0.3]^\top, [0.2, 0.1]^\top\}$ yield:

$$\phi([0.2, -0.3]^\top) = \begin{bmatrix} 0.345 \\ -0.572 \end{bmatrix}, \quad \phi([0.2, 0.1]^\top) = \begin{bmatrix} 0.394 \\ 0.345 \end{bmatrix}.$$

$$\sum_{x \in X_2} \phi(x) = \begin{bmatrix} 0.739 \\ -0.227 \end{bmatrix}.$$

$$f(X_2) = W_3 \cdot \begin{bmatrix} 0.739 \\ -0.227 \end{bmatrix} + b_3 = (1)(0.739) + (-1)(-0.227) + 0.1 = 1.066.$$

Conclusion:
Since $f(X_1) \neq f(X_2)$, the DeepSets model maps X_1 and X_2 to different outputs, proving that it can embed the two sets into distinct vectors.

3 Task 7

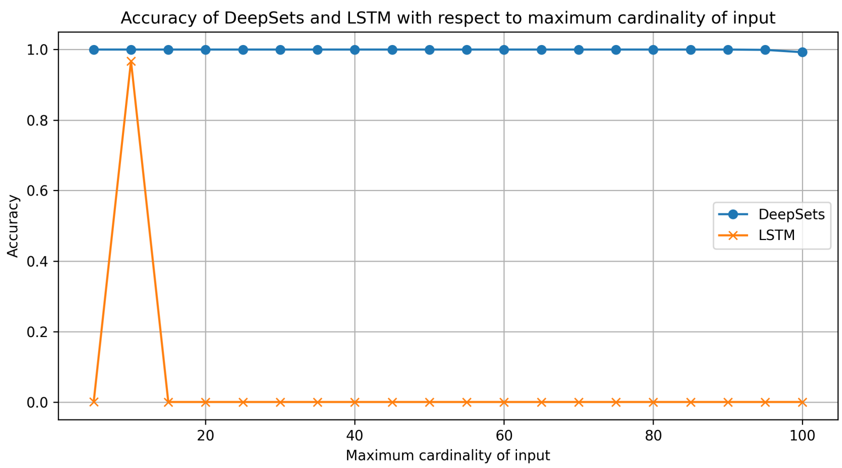


Figure 1: Figure 1: Accuracy of DeepSets and LSTM with respect to maximum cardinality of input.

4 Question 3

DeepSets can be integrated as a module in graph neural networks for graph classification. To classify graphs, we can embed representations of their nodes and handle varying numbers of nodes (different cardinalities) using DeepSets. Instead of embedding numbers, we could reduce the input nodes of a graph to a set $\{z_1, z_2, ..., z_n\}$, apply a readout function to combine these representations, and pass the result through an MLP with m output neurons for m-class classification.

5 Question 4

For an Erdős–Rényi random graph with $n = 15$ nodes, the total number of possible edges is $\binom{15}{2} = 105$.
- When the edge probability is $p = 0.2$:

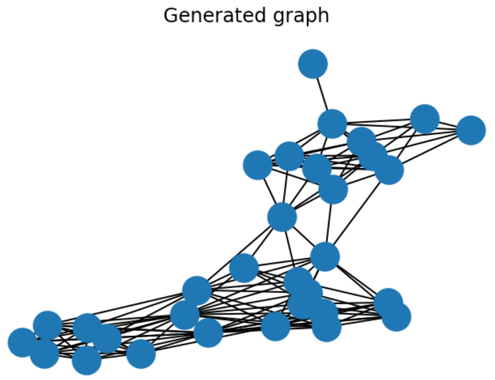
$$\mathbb{E}[E] = 0.2 \times 105 = 21, \quad \text{Var}(E) = 105 \times 0.2 \times 0.8 = 16.8.$$

- When the edge probability is $p = 0.4$:

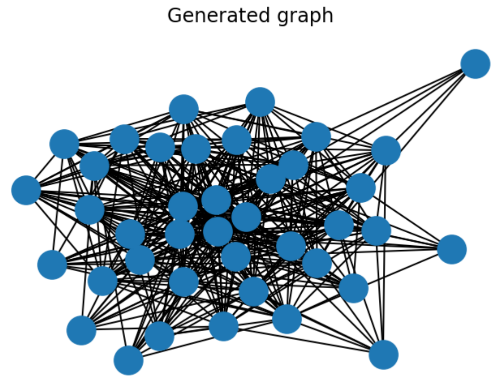
$$\mathbb{E}[E] = 0.4 \times 105 = 42, \quad \text{Var}(E) = 105 \times 0.4 \times 0.6 = 25.2.$$

6 Task 11

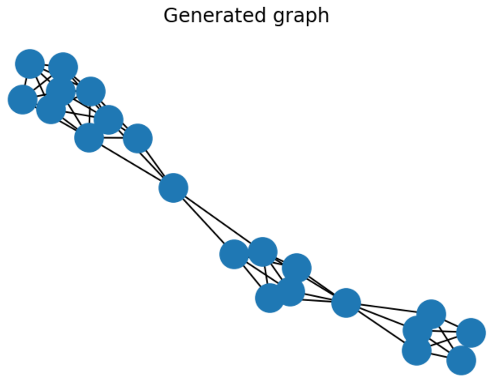
In this section, we present the graphs generated by the variational autoencoder. The following graphs illustrate the reconstructed structures from the input data. Each graph is accompanied by its descriptive caption.



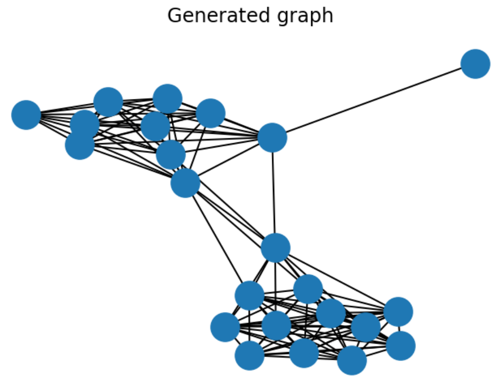
(a) Reconstructed graph 1 by the autoencoder.



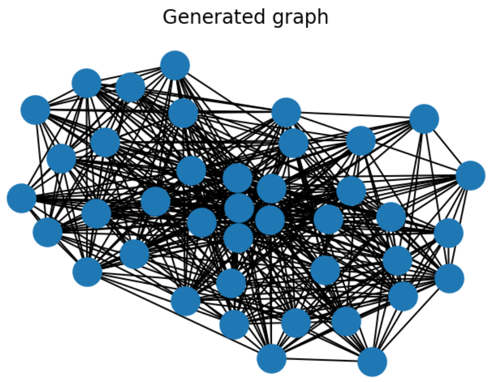
(b) Reconstructed graph 2 by the autoencoder.



(c) Reconstructed graph 3 by the autoencoder.



(d) Reconstructed graph 4 by the autoencoder.



(e) Reconstructed graph 5 by the autoencoder.

Figure 2: Graphs generated by the variational autoencoder. The graphs illustrate the structures obtained after reconstruction.