1 Question 1

For a given $G(n, p)$ graph, each node is connected to $n-1$ nodes. As the edges are independent, the random variable of the number of edges of node follow a Binomial law of parameter $(n-1, p)$. Hence, the expected number of edges, or the expected degree of a node is:

(n-1)p.

- For $n = 24$, $p = 0.2$, we have 4.8
- For $n = 24$, $p = 0.4$, we have 9.6

2 Question 2

As outlined in the document, the graphs within our dataset typically do not share the same number of nodes. Consequently, employing a trainable linear layer as a readout function poses challenges. Specifically, it becomes intricate to determine how to handle graphs with a node count lower than the number of parameters in the layer. Should we exclusively utilize the initial parameters and disregard the remainder? What impact might this strategy have on parameter learning?

To address this issue, we opt for a simpler non-trainable readout function, such as the sum or mean function. This choice helps circumvent the complexities associated with varying node counts and ensures a more straightforward approach to readout function implementation.

3 Question 3

We observe that when we choose as readout function the mean operator, we obtain for our graphs the same representation. Hence, our classification pipeline will not be able to obtain any information. This can be explained because we work with cycle graphs. In those graphs, every node is the same. Hence, their vector representation will be identical, and the readout function mean will only add the same representation n times before dividing it n times. Therefore, we will obtain the same representation for all the nodes.

4 Task 11

We observe that we obtain the same vector for the two graphs. Once again, our GNN cannot distinguish between the two graphs, although they are rather different.

5 Question 4

In Figure 2, the two displayed graphs share an identical vector representation achieved through the utilization of the sum operator, both for neighborhood aggregation and as the model's readout function. This similarity arises from the equivalence of the product AX, where X is the column matrix, i.e., the *ith* row represents the sum of the coefficients of matrix A in the *ith* row.

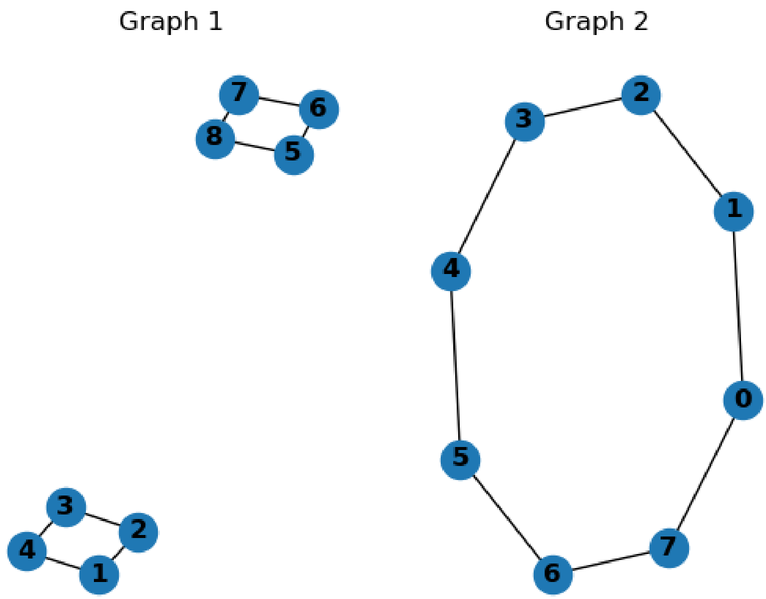

Figure 1: Give an example of two non-isomorphic graphs G1 and G2. The sum of their adjacency matrix over the axis 1 is the same, so we will obtain the same vector representation once again.

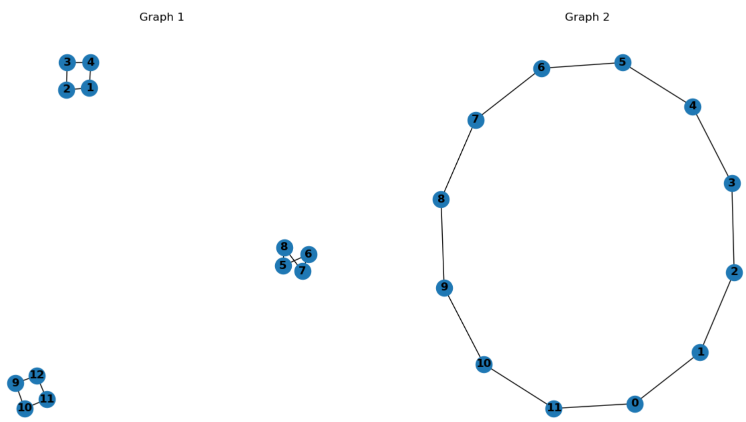

Figure 2: Give an example of two non-isomorphic graphs G1 and G2. The sum of their adjacency matrix over the axis 1 is the same, so we will obtain the same vector representation once again.