1 Question 1

The approach presented in the following was described in class, during Lecture 6 (Deep Learning for Graphs - I), slide 22. In order to avoid immediate visits of the same nodes within a path, we may try to assign different probabilities over "jumping" to new nodes. More precisely, when we select a new node, instead of sampling from an uniform distribution (every neighbour of a node is equally likely to be chosen as the next node), we will sample from a probability distribution that assigns a higher probability to the neighbours that were not already visited. Therefore, the transition probability from one node to one of its neighbours will be higher if the neighbours was not recently visited and, otherwise, it will be lower.

2 Question 2

In order to perform graph classification, one possible approach is depicted in the following. We will take the output of the second message passing layer with (h_2 hidden units), which is a matrix of shape ($number\_of\_nodes$, h_2). This matrix will be input to a pooling layer that will aggregate every row of the matrix into a single one (we can use, for example, sum pooling or average pooling). We can interpret the feature vector obtained in this way as containing all the synthesised information about our graph and we can use it in order to perform graph classification. This feature representation can then be input into a fully connected layer, which will map it to another vector consisting of a number of elements equal to the number of classes. Afterwards, we can use a softmax layer, for example, to transform this vector into a probability distribution and we can select the class with the highest probability as being the class corresponding to the graph.

3 Question 3

GNN considerably outperforms the Deep Walk + logistic regression approach on the node classification task (on the karate dataset), as we can see from Table 1. GNN achieves a perfect score, while the DeepWalk + LR obtains a result which is comparable to that of a random classifier. A possible explanation might be the fact that the embeddings obtained from DeepWalk (using word2Vec) are not optimized for the node classification task (instead, they are optimized for an unsupervised objective). Meanwhile, the GNN is trained in a supervised fashion for the specific task of node classification, resulting in node embeddings specifically optimized for the desired task. More generally, an end-to-end system trained for classification in a supervised manner usually performs better than a pipeline containing separate blocks, trained with multiple, different objectives (here, embeddings trained for an unsupervised objective and logistic regression trained for the node classification task). Furthermore, the node revisiting issue mentioned in Question 1 is also problematic for the DeepWalk + LR approach.

Method	Accuracy (%)
DeepWalk + LR	42.8
GNN	**100.0**

Table 1: Comparison between results achieved with the DeepWalk + logistic regression and GNN on the karate dataset.

4 Question 4

The described modification concerning the feature initialization has a drastic negative effect on the GNN's performance, as we can see from Table 2. When the features are initialized with the same value (all of them are set to 1 in our case), we can see that the accuracy decreases to 28.57% from the perfect accuracy that we had before (when the features are initialized with different values). A possible explanation is suggested by the fact that in the first message passing layer ($Z^0 = f(\hat{A} \cdot X \cdot W^0)$), the normalized adjacency matrix of the graph $\hat{A}$ is multiplied by the feature vector X for which the initialization is changed. When $\hat{A}$ is multiplied with a matrix filled with identical values, the resulting matrix will contain identical values on each row (the first row

is entirely filled with a single value, the second one with another value, and so on). This might result in the loss of information which can be important for the classification task. For an even more illustrative argument, suppose $\hat{A}$ is the unnormalized adjacency matrix. Then, using the identical values feature initialization results in a matrix $\hat{A} \cdot X$ which contains on row i the repeated value of the degree of node i. This drastic loss of information would suggest a potentially serious impact on the classification performance, as we indeed observe.

Method	Accuracy (%)
GNN with different initialization for features	**100.00**
GNN with identical initialization for features	28.57

Table 2: Comparison between results achieved by GNN when its features are initialized with different values vs. with the same value.

References