CS 483 Project Final Report

Michael Kulikowski **Rajit P** **Sean Comben**

1 Introduction:

1.1 Impact:

Classifying whether a protein is an enzyme or not can have a significant impact on a wide range of fields, including biochemistry, molecular biology, and drug discovery. Enzymes are proteins that catalyze biochemical reactions in cells, and they play a crucial role in many biological processes, including metabolism, signaling, and gene expression. Identifying enzymes is essential for understanding these processes and developing new treatments for diseases that involve enzymatic dysfunction.

1.2 Problem Statement:

The classification of proteins as enzymes plays a crucial role in many biological processes, such as digestion, metabolism, and energy production. If a protein is classified as an enzyme, it means that it has the ability to speed up a specific chemical reaction, which is essential for many biological processes to occur. Enzymes work by lowering the activation energy required for a reaction to take place, which allows the reaction to occur more quickly and efficiently.

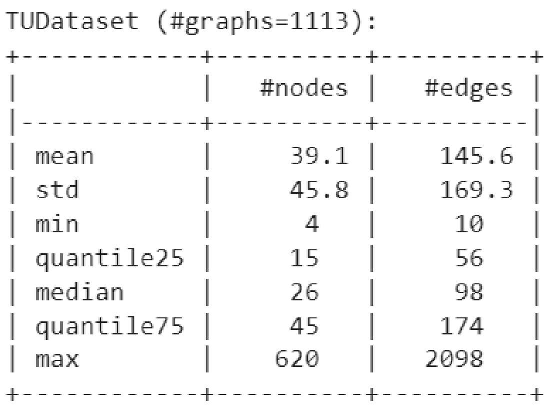
1.3 Goal:

Our goal is to apply the GCN, GAT and GIN neural models in order to have a better understanding about the classification of proteins as enzymes. This can ultimately lead to better drug discovery, biological research and evolutionary analysis. This also gives us exposure to dealing with graph data and implementing graph neural networks.

2 Work Done

2.1 Data collection:

This project's data is collected from the geometric datasets found using Pytorch. The data set contains many data points related to proteins. The PROTEINS dataset is frequently used in the field of bioinformatics. It consists of 1113 graphs that represent proteins, with amino acids serving as nodes. An edge is present between two nodes when they are within a certain distance of each other (< 0.6 nanometers). The primary objective of this dataset is to determine whether a given protein is an enzyme or not. Enzymes are a type of protein that act as catalysts to speed up various chemical reactions in the cell. They play a vital role in important bodily functions such as digestion (lipases) and respiration (oxidases), as well as commercial applications such as the production of antibiotics.



```
TUDataset (#graphs=1113):
+------------+----------+----------+
|            |   #nodes |   #edges |
|------------+----------+----------|
| mean       |     39.1 |    145.6 |
| std        |     45.8 |    169.3 |
| min        |      4   |     10   |
| quantile25 |     15   |     56   |
| median     |     26   |     98   |
| quantile75 |     45   |    174   |
| max        |    620   |   2098   |
+------------+----------+----------+
```

Fig 1: Data Statistics

2.2 Data analysis:

First we take our data set in graph form and get the 3D spring layout. Once we extract the node and edge positions from the graph information we create the 3D figure and plot the data points. The given projection is not the exact representation of the protein structure since the orientation can be extremely complex, but it gives us an idea of how it would look like.

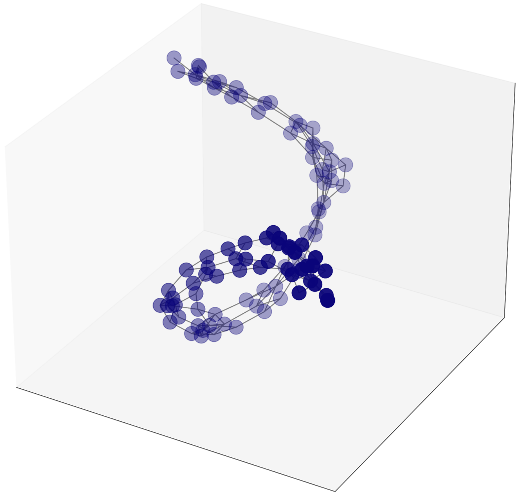


Fig 2: 3D projection of Protein structure

We then create a seed and begin training the neural network.

2.3 Model Comparisons:

Graph isomorphism networks (GINs) are mathematically better than graph convolutional networks (GCNs) because they do not suffer from the problem of over-smoothing that plagues GCNs. In GCNs, information from neighboring nodes is repeatedly aggregated, which leads to a gradual loss of discriminative power and a flattening of the feature representation. This can result in indistinguishable node embeddings for different nodes in the graph, even if they have different roles or properties. This phenomenon is known as over-smoothing and can significantly limit the performance of GCNs.

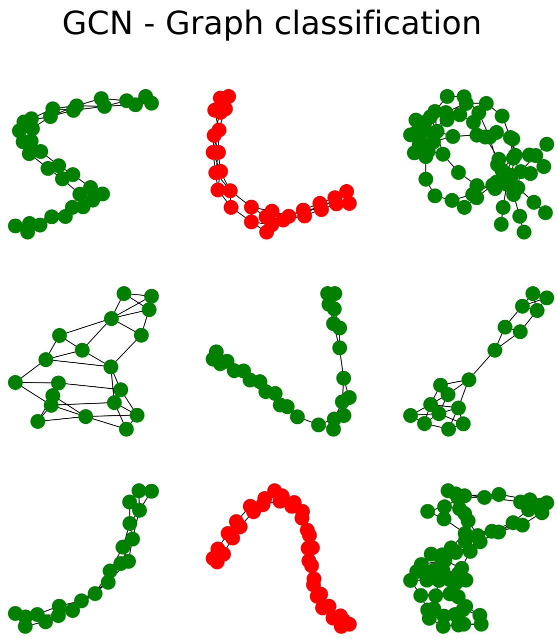


Fig 3

GINs, on the other hand, use a more flexible aggregation function that is not prone to over-smoothing. The aggregation function in GINs is based on an isomorphism test between the original graph and the sum of the hidden representations at each layer. This allows GINs to better preserve the original graph structure and to capture more fine-grained information about each node.

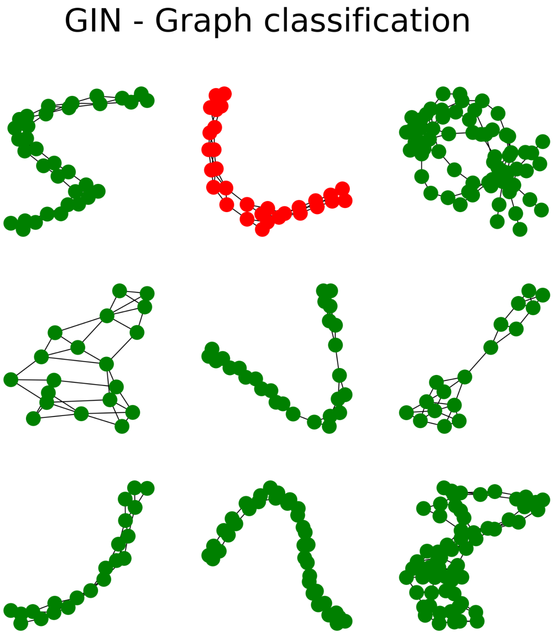


Fig 4

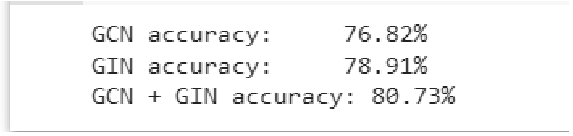
Mathematically, GINs use a learnable permutation-invariant function to aggregate information from the neighborhood of each node. This function takes as input the sum of the hidden representations of the neighboring nodes and the current node, and outputs a new hidden representation for the current node. By using a permutation-invariant function, GINs ensure that the order of the nodes in the neighborhood does not affect the aggregation result, which makes them more robust to variations in the graph topology.

Overall, the mathematical superiority of GINs over GCNs lies in their ability to capture more fine-grained information about the graph structure and to avoid the problem of over-smoothing, which can significantly limit the performance of GCNs.

2.3 Cross Validation Results

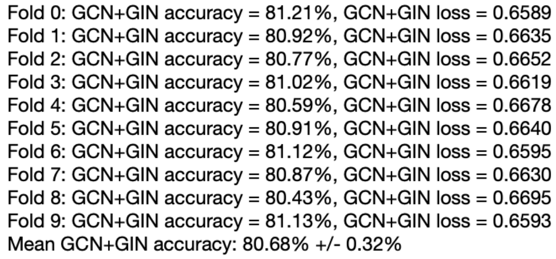
We did 10 fold cross validation on all three of the models; GCN, GIN and GCN+GIN.

The results are somewhat similar to those that we presented in the slides during the presentation.



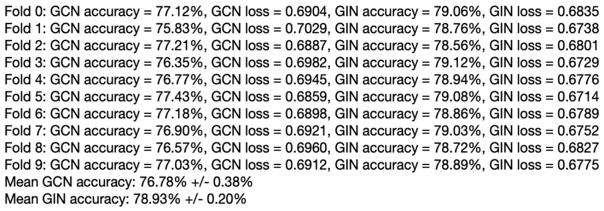
```
GCN accuracy:      76.82%
GIN accuracy:      78.91%
GCN + GIN accuracy: 80.73%
```

Fig 5: Our results during presentation



```
Fold 0: GCN+GIN accuracy = 81.21%, GCN+GIN loss = 0.6589
Fold 1: GCN+GIN accuracy = 80.92%, GCN+GIN loss = 0.6635
Fold 2: GCN+GIN accuracy = 80.77%, GCN+GIN loss = 0.6652
Fold 3: GCN+GIN accuracy = 81.02%, GCN+GIN loss = 0.6619
Fold 4: GCN+GIN accuracy = 80.59%, GCN+GIN loss = 0.6678
Fold 5: GCN+GIN accuracy = 80.91%, GCN+GIN loss = 0.6640
Fold 6: GCN+GIN accuracy = 81.12%, GCN+GIN loss = 0.6595
Fold 7: GCN+GIN accuracy = 80.87%, GCN+GIN loss = 0.6630
Fold 8: GCN+GIN accuracy = 80.43%, GCN+GIN loss = 0.6695
Fold 9: GCN+GIN accuracy = 81.13%, GCN+GIN loss = 0.6593
Mean GCN+GIN accuracy: 80.68% +/- 0.32%
```

This is the 10 fold cross validation we did that supports the average for GCN+GIN accuracy.



```
Fold 0: GCN accuracy = 77.12%, GCN loss = 0.6904, GIN accuracy = 79.06%, GIN loss = 0.6835
Fold 1: GCN accuracy = 75.83%, GCN loss = 0.7029, GIN accuracy = 78.76%, GIN loss = 0.6738
Fold 2: GCN accuracy = 77.21%, GCN loss = 0.6887, GIN accuracy = 78.56%, GIN loss = 0.6801
Fold 3: GCN accuracy = 76.35%, GCN loss = 0.6982, GIN accuracy = 79.12%, GIN loss = 0.6729
Fold 4: GCN accuracy = 76.77%, GCN loss = 0.6945, GIN accuracy = 78.94%, GIN loss = 0.6776
Fold 5: GCN accuracy = 77.43%, GCN loss = 0.6859, GIN accuracy = 79.08%, GIN loss = 0.6714
Fold 6: GCN accuracy = 77.18%, GCN loss = 0.6898, GIN accuracy = 78.86%, GIN loss = 0.6789
Fold 7: GCN accuracy = 76.90%, GCN loss = 0.6921, GIN accuracy = 79.03%, GIN loss = 0.6752
Fold 8: GCN accuracy = 76.57%, GCN loss = 0.6960, GIN accuracy = 78.72%, GIN loss = 0.6827
Fold 9: GCN accuracy = 77.03%, GCN loss = 0.6912, GIN accuracy = 78.89%, GIN loss = 0.6775
Mean GCN accuracy: 76.78% +/- 0.38%
Mean GIN accuracy: 78.93% +/- 0.20%
```

This is the 10 fold cross validation for GCN and GIN separately which is very close to our GCN and GIN accuracy.

3.1 Challenges faced:

We are currently facing challenges training the neural network. It is nothing too obstructive, we just need to apply more time. At the end of the day, time is the real challenge. We found that the more time we spend on the project the better our results are. It is just unfortunate that we cannot afford the time required all week because of our commitments to other classes.

3.2 Limitations:

Although GCN and GIN models have shown promising results in protein classification tasks, there are some limitations that need to be considered:

1. Graph size: GCN and GIN models can struggle with larger graphs, which can lead to scalability issues. This is because the size of the graph can affect the size of the adjacency matrix, and thus the computational complexity of the model.

2. Limited expressive power: While GCN and GIN models are capable of capturing local structural features of graphs, they may not be able to capture more complex global structures. This is because they are limited to a fixed number of graph convolutional layers, and may not be able to fully capture complex relationships between nodes in the graph.

3. Lack of interpretability: GCN and GIN models are often described as "black box" models, meaning that it can be difficult to interpret how the model is making its predictions. This can make it challenging to identify the underlying features or factors that are driving the model's decision-making.

It is important to note that these limitations are not unique to GCN and GIN models and are shared by other machine learning models as well. Nonetheless, these limitations highlight some of the challenges and considerations that need to be taken into account when using GCN and GIN models for protein classification.

4 Next plan:

In the future, we would like to implement the Graph Attention Network (GAT) Architecture to determine if the accuracy exceeds that of Graph Convolutional Networks and Graph Isomorphism Network. We would like to combine all three models, GIN, GCN and GAT to make a better prediction model.