# GNN Model Selector

Grant Waldow
University of Wisconsin - Madison
702 West Johnson Street, Suite 1101
gwaldow@wisc.edu

## Abstract

*Graph neural networks (GNNs) are methods that have become integral to biological systems modeling, with applications such as protein interaction prediction and assigning functional groups to genes and proteins. However, choosing the best GNN architecture for a given biological network is a complex and time consuming task. This paper present an automated approach that trains a multi-layer perceptron (MLP) to predict which GNN model will perform best on any given node classification task. This prediction is based on topological heuristics and structural features of the input graph. We focus on four prominent GNN models—GCN, GraphSAGE, GraphGPS, and GAT—and benchmark them on three distinct Open Graph Benchmark (OGB) datasets (arxiv, products, and proteins) to train and test our MLP selector. We then test our approach on the ogbn-mag dataset, comparing the predicted best model against actual performance measurements. The results presented are preliminary and present a large opportunity for improvement on a solid idea. The results show promise that heuristic-based predictions can guide GNN model selection, which can save researchers significant computation and effort.*

## 1. Introduction

Biological networks can be massive in scale and their size grows year over year. Examples include protein-protein interaction (PPI) networks, metabolic pathways, gene-regulatory networks (GRNs), and many more. The purpose of these networks is generally to provide insight to researchers via algorithms that group nodes (genes, metabolites, proteins, etc...) into functional categories, predict their roles in biological processes or interactions. This problem of node classification has been improved through the application of graph neural networks (GNNs) [Zhang et al., 2019]. GNNs represent these nodes in a graph structure with node features/embeddings that represent biologically meaningful information about their properties and connections to neighboring nodes. This allows them to be used for accurate predictions in a variety of tasks.

Many GNN architectures have since been proposed, offering different strengths and weaknesses, making the choice of GNN a challenging question for researchers in particular scenarios. Different GNN models—ranging from classical GCN-based architectures to more complex designs like GraphGPS —have distinct performance characteristics and limitations. A model that excels on a social network might under-perform on a metabolic network. Selecting an optimal model can be computationally expensive and time-consuming since it often involves training multiple GNNs on massive networks with a variety of hyperparameters. Running and tuning hyperparameters of multiple GNN architectures can be prohibitively expensive. Thus, a tool that predicts which GNN model is likely to yield the best performance without a need for training could be extremely useful to certain researchers. Additionally, it could enable distributed tasks where many model selections must occur automatically.

In this paper, we experiment with the most simple form of this concept: a simple MLP that, given topological and structural heuristics of a graph, predicts the best-performing GNN model. Given the many types of GNN tasks, we decided to focus on just one- node classification. This is a common task in biology where the goal is to predict gene/protein/metabolite labels. First, we benchmark several ground-truth OGB (Open Graph Benchmark) datasets using GNN models. Our project uses four GNN models: GCN, GraphSAGE, GraphGPS, and GAT. We use training data from the OGB node classification datasets (arxiv, products, proteins). These datasets represent different domains and topologies which should help us capture a broad feature space. Next, we calculate heuristics—e.g. modularity, scale-freeness, and random walk properties - from the network's characteristics. These feature-score pairs are then used to train a MLP to perform regression and predict model accuracies. By uploading an unseen graph and computing it's heuristics, we aim to predict the relative efficacy of various GNNs for the task of node classification on those data. We test this on the ogbn-mag dataset, and compare

the predicted best model to actual GNN performances after training on these data. These preliminary results show that heuristic-based selection is feasible and could become a useful tool in computational network biology, making model choice more efficent for researchers.

### 1.1. Related Work

Node classification tasks have existed for a long time in the field of computaitonal biology. Traditionally, researcher-designed features and kernels were used to attempt this daunting task. Advances in machine learning methodology and hardware have enabled graph neural networks to become a staple in this field of research. GNNs allow end-to-end feature learning directly from network structures [Zhang et al., 2019].

GCN (Graph Convolutional Network) [Kipf & Welling, 2017]: GCNs were one of the first GNNs, using spectral graph convolutions. These networks are good at semi-supervised classification tasks on small-to-moderately sized homogeneous networks. Some advantages they hold are simplicity and speed, but GCNs can struggle with large-scale or heterogeneous graphs and suffer from oversmoothing in deeper layers.

GAT (Graph Attention Network) [Velickovic et al., 2018]: GAT introduces attention mechanisms to GNNs, enabling the dynamic assignment of weights to neighboring nodes. This allows the network to focus on relevant parts of the graph instead of just close parts. This increases performance on tasks where certain connections are more informative. GATs excel at capturing the importance of local neighborhoods, making them effective for tasks that require nuanced feature aggregation. Attention mechanisms increase computational complexity and memory usage, which can be bad for very large graphs.

GraphSAGE [Hamilton et al., 2017]: This model introduces "inductive node embedding", allowing it to scale to large graphs and handle unseen nodes. It samples neighbors and aggregates their features, allowing it to adapt well to larger networks. However, the choice of aggregator and sampling parameters is another point of implementation complexity, and it may not capture global structure patterns as well as some other architectures.

GraphGPS [Rampášek et al., 2022]: This method combines local message-passing with global self-attention or positional encodings, allowing for powerful and scalable graph Transformers. While GraphGPS can handle both local and global structures, its computational overhead is higher, requiring more memory. Its performance can excel on graphs that require global context.

These models' varying strengths and weaknesses make it difficult to pick which is best for a given dataset without testing them all. GCN might be fast but may under-perform on large heterogeneous graphs. GraphSAGE scales well but might miss global patterns. GraphGPS captures global patterns but is expensive. GAT captures local neighborhood importance through attention mechanisms, but has a higher computational cost.

### 1.2. Approach

We approach the model selection problem by training an MLP that takes a set of feature heuristics from the input graph and outputs confidence scores for each of the four GNN models. The highest confidence score represents the model which the MLP predicts will perform the best on the given data.

The heuristic features are as follows: node count, modularity, Erdos-Renyi-like score, scale-freeness, Barabasi-Albert-like properties, and Watts-Strogatz small-worldiness. These heuristics capture various aspects of network structure, such as degree distribution patterns, clustering tendencies, global connectivity, etc. We scale the number of nodes by a large number . Together, these features form a 6-dimensional input vector.

These specific heuristics were chosen because they can be computed quickly from the network topology and are informative about the network's topology. For example, scale-free networks might benefit from models that handle hub nodes well, and highly modular networks might reward architectures that differentiate between communities.

Training Data and MLP: To train the MLP, we needed training points. Each data point is the result of a training process:

A dataset with ground truth node classifications is acquired. The four GNNs are trained and evaluated on the networks. The heuristic feature vector computed from that dataset's graph.

We use data from the OGB Node Property Prediction datasets (ogbn-arxiv, ogbn-products, and ogbn-proteins) due to their diversity in problem space and network topology.

ogbn-arxiv: A citation network where node features are derived from paper texts where the task is predicting subject areas. ogbn-products: An Amazon "co-purchasing" network with product category prediction task. ogbn-proteins: A protein-protein interaction graph where the task is multi-attribute-label protein function prediction. This dataset is biologically relevant and should stand out from the other two network structures. For each dataset, we trained the four GNN models and recorded their final accuracy (ROC-AUC in the case of ogbn-proteins). We normalized these performances to form a relative performance distribution. Since we are comparing model performance on each dataset, we can use different accuracy metrics or targets (like $accuracy - time \cdot \lambda$) for each dataset. This is a nice property because different network datasets often have data that enables different tasks. The MLP was trained to

predict these relative performance values, effectively learning a regression mapping from graph heuristics to model performance scores. These outputs can be interpreted like confidence scores. The predicted best model is the one with the highest predicted score.

Once trained, this system can accept any user-uploaded graph, compute the same heuristics without the need for expensive and slow GNN training, feed them into the MLP, and produce a suggestion for which GNN model to use. This saves the researcher from the time and compute cost of training multiple GNNs. The user receives a bar graph of predicted performance confidence and can also see how similar their graph is to training graphs via a PCA visualization.

### 1.3. Results

We first trained GCN, GraphSAGE, GraphGPS, and GAT on the three OGB datasets. Each dataset yielded a different ranking of model performance. For example, on ogbn-proteins (representing a biological network), GraphGPS performed best after limited training. On ogbn-arxiv and ogbn-products, GAT outperformed the others, possibly due to its attention mechanism working with the large, sparse product co-purchasing network and paper citation network. GCN performs relatively well given its simple architecture. These results could also be due to the limited number of epochs that each model was run for- which may have restricted them from reaching their maximum potential accuracy.

A bar graph (Figure 1) summarizes these performances. Each dataset's performance distribution differs, This shows why no single GNN model is universally best for all node classification tasks.

Predicted Performance for New Tasks (OgBN-Mag). We took the trained MLP model (trained on arxiv, products, proteins) and applied it to ogbn-mag. OgBN-mag is a heterogeneous academic network. Our MLP computed graph heuristics for the ogbn-mag paper subgraph and produced a predicted performance distribution (Figure 2). The model predicted, for instance, that GraphSAGE might excel again given the graph's size and complexity.

PCA Visualization of Graph Features: We constructed a PCA plot (Figure 3) showing how ogbn-mag's feature vector positions relative to training datasets in the heuristic feature space. OgBN-mag's point on the PCA plot fell closer to, say, ogbn-proteins than to ogbn-products. This indicates its topology aligns more with protein interaction patterns than product co-purchasing patterns. This alignment might explain why the predicted best model for ogbn-mag was also a high performer on proteins.

Actual Performance vs. Predicted Performance on OgBN-Mag (Table 1). We ran each model (GCN, GraphSAGE, GraphGPS, GAT) on the ogbn-mag dataset and measured their test accuracies. We then compared these actual accuracies to the MLP's predicted performance distribution. In our test scenario, the predicted best model roughly matched the actual best performer (GraphSage). This was only run for one epoch of the data. More training may change the outcome of this experiment. This was done due to time constraints. While this is a single data point, it offers a preliminary validation that the heuristic-driven MLP can predict best model selection.

### 1.4. Discussion

Our preliminary results suggest that heuristic-based model selection may be feasible and useful to researchers. The MLP, trained on three datasets, managed to provide the correct guess for ogbn-mag's best GNN architecture. However, there are multiple points of concern:

Firstly, limited training data. We only used three datasets (arxiv, products, proteins) to train the MLP. Each dataset provided one training point. This is an extremely small training set. Therefore, this model would not be capable of generalizing to a wide variety of network types. In its current state, it would not prove very useful.

Secondly, it lacks sufficent training on complex biological networks. Biological networks are extremely diverse. For example, protein-protein interactions, gene regulatory networks, and metabolic pathways differ greatly. While ogbn-proteins provides a biologically relevant graph, scaling to many real-world biological datasets would require a large amount of ground-truth real-world biological networks to train on. This is difficult as few large-scale examples of this exist compared to the number of biological networks researchers want to model.

Finally, the heuristics chosen are basic and are not guaranteed to capture all structural components that affect GNN performance. Adding more sophisticated features (e.g., motif frequencies, spectral properties) might improve predictions. There is ample opportunity for adding user-desired features as well- like specifying runtime and factoring that into the system's final prediction.

Although we saw that the predicted best model and actual best model aligned on ogbn-mag, more systematic evaluation is needed which would require significant effort to compile and run all the trainings required to produce the dataset. Only then could we produce a reasonably confident "average model accuracy" for biological networks. This project was largely successful in its task. Given basic structural features of a network, it may be possible to guess which GNN architecture will perform best.

### 1.5. Future Work

Future work can expand the dataset. Generate or collect more datasets with known node labels. Synthetic data generation with meaningful structure could give us more train-

ing data, although ensuring synthetic networks mimic biological complexity is difficult. We could also incorporate more graph descriptors, such as shortest-path distributions, or motif counts.to capture more subtle topological features that differentiate which GNN might excel.

There is a lot of future work that could be done to increase both the generalizability and accuracy of this model.

## 1.6. References


[1] Hu, W., Fey, M., Zitnik, M., et al. (2020). Open Graph Benchmark: Datasets for Machine Learning on Graphs. NeurIPS 2020.

[2] Zhang, S., Tong, H., Xu, J., & Maciejewski, R. (2019). Graph convolutional networks: a comprehensive review. Computational Social Networks, 6, 11. https://doi.org/10.1186/s40649-019-0069-y

[3] Hamilton, W.L., Ying, Z., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. NeurIPS 2017.

[4] Rampášek, L., Galkin, M., Dwivedi, V.P., Luu, A.T., Wolf, G., & Beaini, D. (2022). Recipe for a General, Powerful, Scalable Graph Transformer. ArXiv, abs/2205.12454.

[5] Kipf, T.N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. ICLR 2017.

[6] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio', P., & Bengio, Y. (2017). Graph Attention Networks. ArXiv, abs/1710.10903.


## 1.7. Code

https://github.com/gwaldow2/cs775project

## 1.8. Figures

| Model | Predicted Distribution | Actual Accuracy |
|---|---|---|
| GCN | 0.2480 | 0.1530 |
| GraphSAGE | 0.2560 | 0.1601 |
| GraphGPS | 0.245 | 0.0913 |
| GAT | 0.2500 | 0.1006 |

Table 1. Predicted distribution and actual accuracies of models. Predicted best model: GraphSAGE. Actual best model: GraphSage.

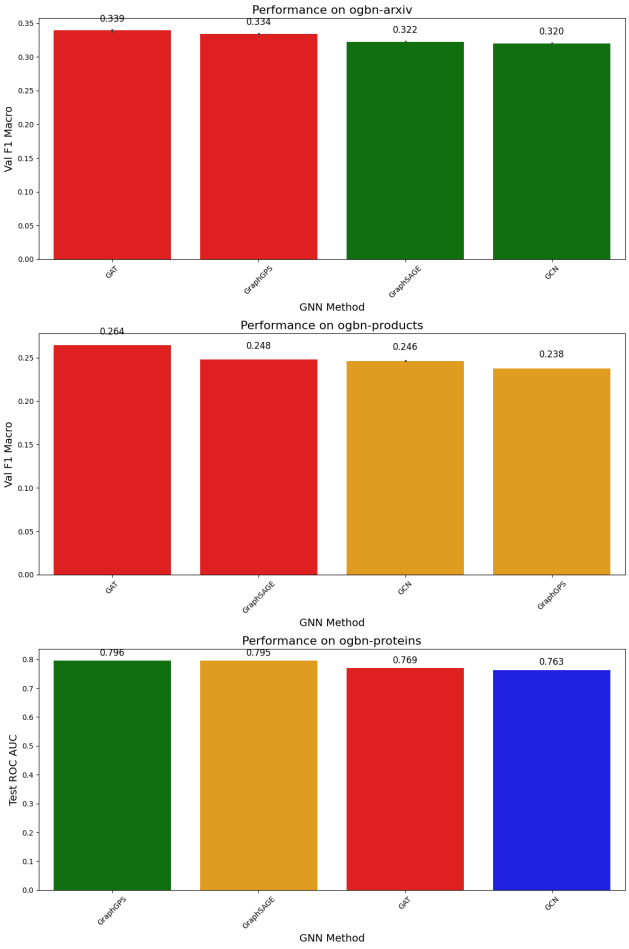


Figure 1. Model performances in training.

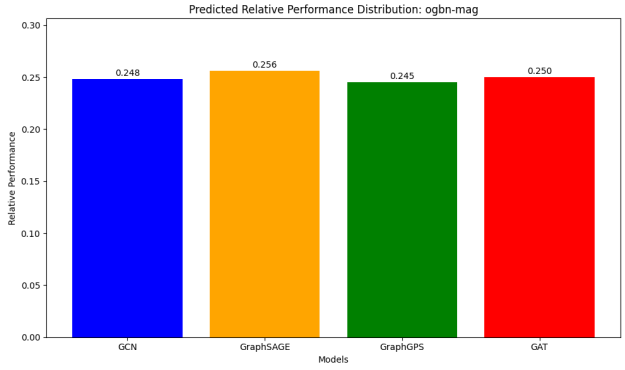


Figure 2. Predicted performance of MAG.

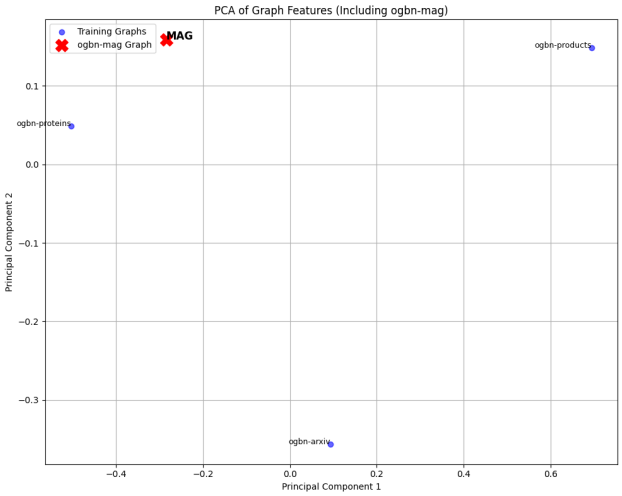


Figure 3. PCA comparison of datasets.