# Graph Neural Network Exploration of a C. Elegans Nervous System
# CS 575 Final Project

J. Wesley Borden

April 15, 2024

## Introduction

The modern field of neural connectomics traces its roots to a 1986 publication, *The mind of a worm*, which documented the meticulous mapping of the neural connections in a C. elegans nematode–the first detailed documentation of such a biological network of a nervous system. Our CS 575 course has spent this last semester reviewing graph data science principles that are implicit to such networks, being applicable to all networks, ranging from social circles to traffic navigation and to biological systems. I spent the first three projects characterizing generic neuronal networks with agent based modeling of simple and complex contagion, and in this project set the goal to analyze a more detailed real world data set with graph neural network-based machine learning techniques. While I had hoped to use data from the 2010-2016 human connectome project, the web application hosting that open-source dataset had a server failure that restricted my access to the data. Instead, I found another dataset through kaggle that was based on the decades of studies of the C. elegans nervous system that have continued since the landmark 1986 paper.

I collected data from the Open Worm Project using a repository available through Kaggle and a repository available directly from the Open Worm Project. These data sources were not fully cleaned for this project, so I cleaned the data by hand using Excel. The data included node and edge attributes, and I also used ChatGPT to generate descriptions of each node (search and reason enabled, prompt: "here is a list of cells in a nematode. for each, could you generate a brief description about its function, importance, neurotransmitter use, and other relevant details? then put it all into a csv file with two columns: cell name and description. [list of cells]"). Each cell has been individually indexed with resources like Worm Atlas, so ChatGPT was able to compile descriptions of these results. Lastly, I used a function that read the ChatGPT descriptions to classify neurons by their involvement in sensory and/or motor activity.

Using this cleaned data set, I demonstrated a one-mode projection, visualized the network, and trained a graph autoencoder to generate encoding of the neurons. Most notably, this analysis suggested that classification of sensory-, motor-, and inter-neurons could be inferred with reasonable accuracy based only on the structure of the network and minimal additional information.

## One-mode Projection

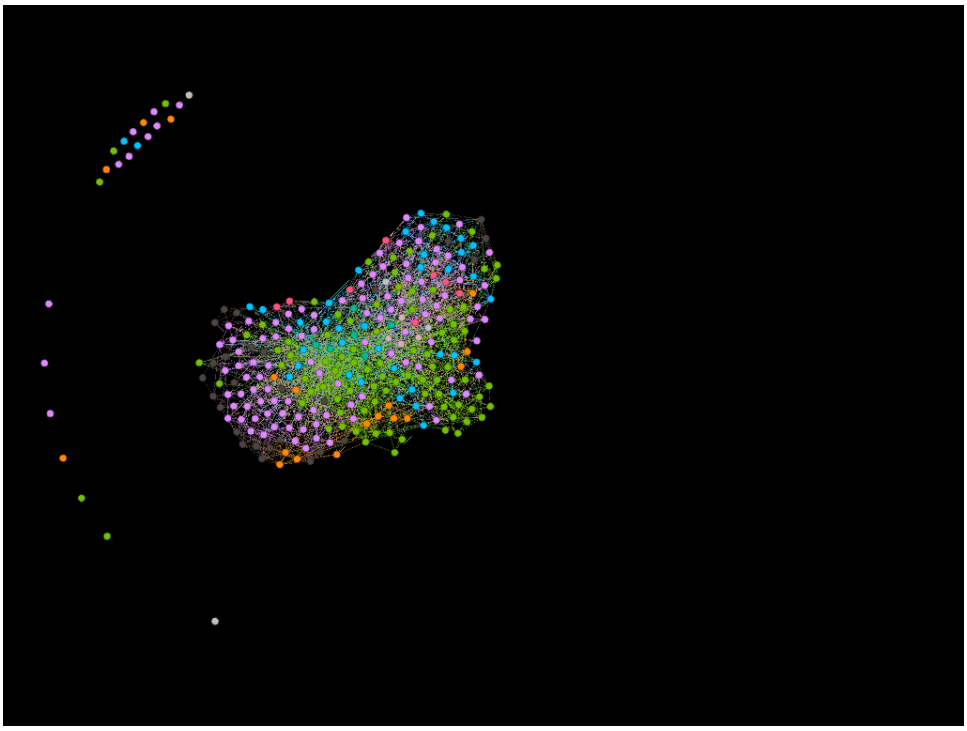
I hypothesized that:

- "One-mode proj. eliminates specified nodes and effectively shows relationships between immediately adjacent nodes"

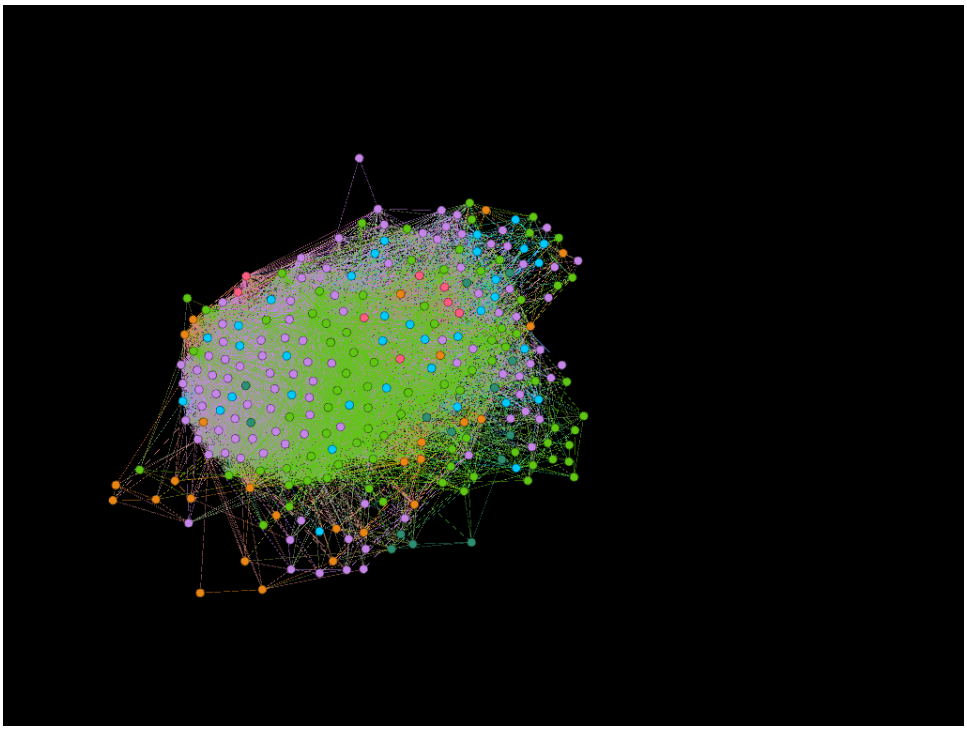
In other words, I anticipated that this projection would show relationships between a node and its neighbors' neighbors. I considered this conservative, as it was nearly a rephrasing of the definition of one-mode projection. However, I had yet to validate a one-mode projection in my own data set, which did not have the same multi-entity schema that the discord dataset did. I built a function that accepted a graph and a python dictionary specifying criteria of nodes to remove from the graph (e.g., neurotransmitter type), and that would return a graph with the projection performed such that neurons matching the input criteria were excluded. The anticipate utility of this method would be a general-purpose one-step projection.

While my testing appears to have showed functionality of the projection function, I found that the networks are so large and nuanced, with many cross-connections, such that a one-mode projection did not seem to provide many insights through visualization. Images below demonstrate this lack of obvious structure change via the projection.

The full, un-projected graph:



The projected graph:



# Graph Autoencoder Analysis

The most significant analysis of my data implemented a graph autoencoder that was trained on the network as in Dr. Goodrich's jupyter notebook tutorials. This autoencoder used many attributes about each node, including a text description of the cell from ChatGPT (Every cell in this network has been indexed and studied in detail) and additional attributes such as neurotransmitter category. These attributes were translated into a ¿400-dimensional encoding for use in the first graph convolutional layer of the encoder. As in tutorials, the encoder used three layers, and a simple training strategy was followed, including use of an Adam optimizer (learning rate of 0.01) and 2000 training steps.

## Hypotheses

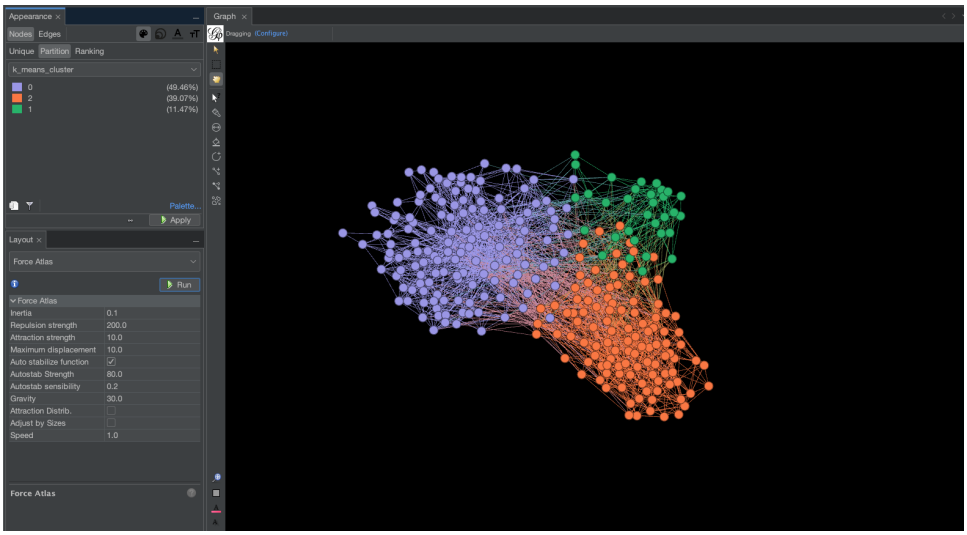
I hypothesized:

- "Applying a GAE with k-means clustering will effectively differentiate sensory vs. motor neurons"
- "Applying semi-supersized learning with neurotransmitter classification of neurons with known neurotransmitter will provide plausible NT classifications for neurons with unknown neurotransmitters"

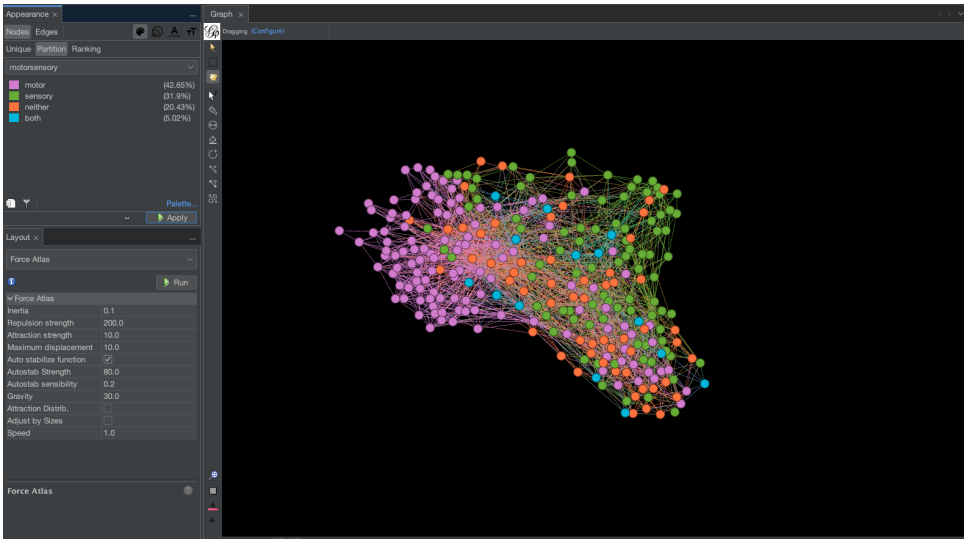
## Results

After training the autoencoder, I exported the graph to visualize in Gephi, keeping attributes for k-means grouping (assuming 3 groups), motor/sensory classification, and neurotransmitter. As a note, I realized after training that there was no existing label for motor/sensory classification, so I added that label at the end, using a function that set labels based on if the words "motor" or "sensory" were in the ChatGPT description I've discussed. Results are as follows:

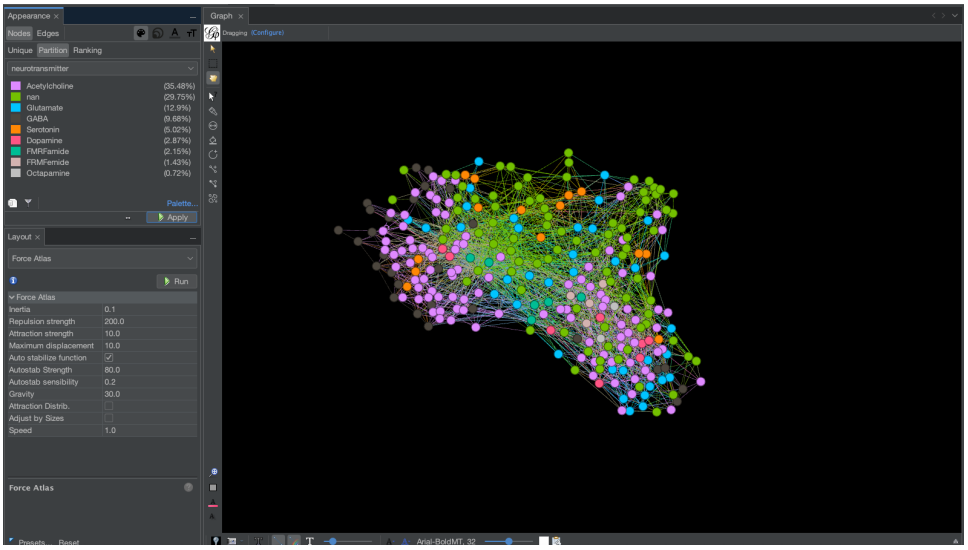
The k-means groups based on learned embeddings:



The motor-sensory classifications:




The neurotransmitter labels (green indicates a missing label)



### Discussion

The first figure above displays three clean communities in the k-means clustering visualization, while the second (corresponding to motor-sensory classification) shows two analogous communities in the top left and top right, but some blurring of the communities where the bottom right/third cluster would be. This supports my hypothesis of a strong association between k-means clustering and sensorimotor classification, while showing there is also more nuance, with the action of interneurons not segregated into a separate community, but rather distributed across the system. I would anticipate that this could lend to resilience of the system.

Considering my second hypothesis, I was not able to build a semi-supervised learning model due to time constraints, but the neurotransmitter labels in the third figure above show that pattern learning would likely be very complex, and potentially not feasible in the scale and maturity of our models. I would no longer anticipate that hypothesis to hold.

## Time and Effort

The following table summarizes my time and effort spent on this project.

| Date | Duration (Hours) | Category | Description |
|---|---|---|---|
| 2025-04-04 | 0.5 | Prep | Read specs, prep repo, plan project |
| 2025-04-04 | 1.75 | Data Prep | Find dataset |
| 2025-04-04 | 1.75 | Data Prep | Clean data from two public data sources |
| 2025-04-04 | 0.75 | Data Handling | Build function to load data to an *nx.Graph* object |
| 2025-04-07 | 1.75 | Data Handling | Finish function to load data to an *nx.Graph* object |
| 2025-04-07 | 1 | Data Handling | Build a one-mode projection function |
| 2025-04-07 | 0.75 | Visualization | Write jupyter notebook to visualize a full or projected graph |
| 2025-04-07 | 2.25 | GNN | Build GNN based on Dr. Goodrich tutorial notebooks |
| 2025-04-07 | 1 | GNN | Complete and test GNN |
| 2025-04-15 | 1.25 | GNN | Finish graph autoencoder and analysis |
| 2025-04-15 | 2.5 | Report | Write this report |
| **TOTAL** | 15.25 | | |

## LLM Acknowledgement

This report was generated using LaTeX, adapted from prior LaTeXfiles which had been written with the help of LLM's such as ChatGPT for formatting, etc. LLM's (ChatGPT, Google Search AI Overview) provided LaTeXformatting advice and may have provided information for the development of code during the project (e.g., to answer questions about function call syntax), but did not directly write the code. Finally, ChatGPT may have been used to refine the report's tone, directly updating the text of the report as specified in the project description. My work reflects my best efforts at learning the content and building a promising simulation while leveraging the LLM's for things like productivity, speed, and technical writing tone.