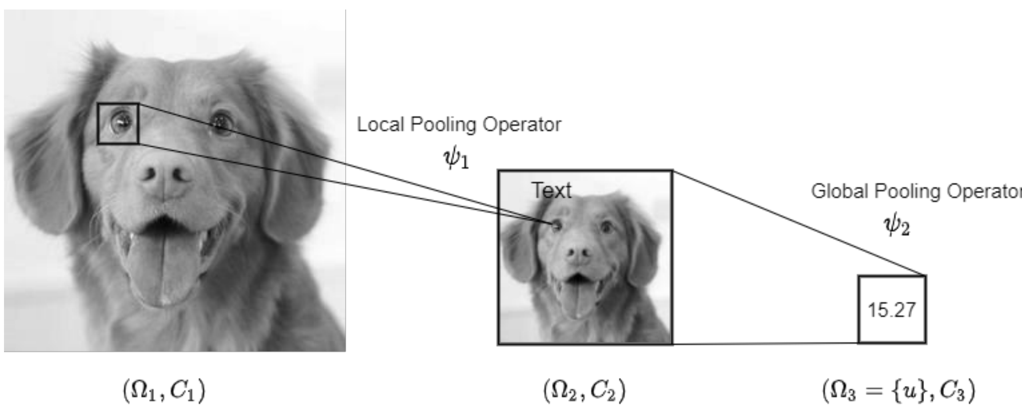
# Midterm Homework

Phillip Wang pw1287

October 2021

## 1.1 GDL Blueprint

### 1.




As shown in the figure, $\Phi$ is the composition of the two operators $\psi_1 \circ \psi_2$

### 2.

## 1.2 Invariant Polynomials

### 1.

Let $d = 3$, if $p(x_1, x_2, x_3) = a_1x_2 + a_2x_3$. Then we have

$$p(x_1, x_2, x_3) = a_1x_2 + a_2x_3 \tag{1}$$
$$p(x_3, x_1, x_2) = a_1x_1 + a_2x_2 \tag{2}$$
$$p(x_2, x_3, x_1) = a_1x_3 + a_2x_1 \tag{3}$$
$$p(x_1, x_3, x_2) = a_1x_3 + a_2x_2 \tag{4}$$
$$p(x_3, x_2, x_1) = a_1x_2 + a_2x_1 \tag{5}$$
$$p(x_2, x_1, x_3) = a_1x_1 + a_2x_3 \tag{6}$$

(1), (2) and (3) demonstrates that the given polynomial is not invariant to cyclic group $C_3$, and all the equation demonstrates that it is also not invariant to $S_3$.

### 2.

Let $d = 3$, if $p(x_1, x_2, x_3) = x_1x_2^2 + x_2x_3^2 + x_3x_1^2$. Then we have

$$p(x_1, x_2, x_3) = x_1x_2^2 + x_2x_3^2 + x_3x_1^2 \tag{7}$$
$$p(x_3, x_1, x_2) = x_3x_1^2 + x_1x_2^2 + x_2x_3^2 \tag{8}$$
$$p(x_2, x_3, x_1) = x_2x_3^2 + x_3x_1^2 + x_1x_2^2 \tag{9}$$
$$\tag{10}$$

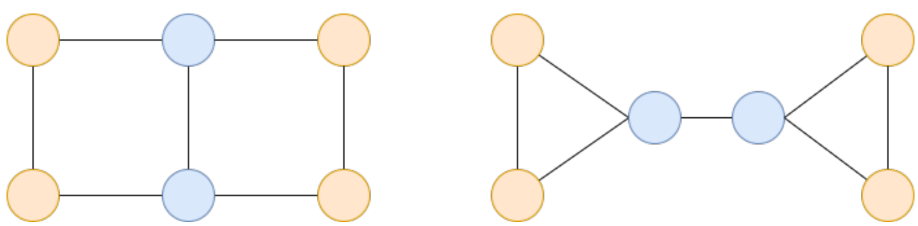
Eq. 7-9 is sufficient to prove that the given polynomial is invariant to cyclic group $C_3$. However, $p(x_1, x_3, x_2) = x_1x_3^2 + x_3x_2^2 + x_2x_1^2 \neq p(x_1, x_2, x_3)$, thus proving that it is not invariant to symmetric group $S_3$.

### 3.

No. This is impossible since $C_d \subseteq S_d$. We can prove this by contradiction, say we have a given $p(x_1 \dots x_d)$ that is invariant to $S_d$ but not to $C_d$. Since $C_d \subseteq S_d$ then $p(x_1 \dots x_d)$ is also invariant to $C_d$, thus contradicting our original assumption.

## 1.3 Graph Isomorphism

### 1.3.1



Consider the two graphs above. Each graph has the same amount of vertices, and each of them has the same order (i.e. the orange node denote order of 2, and blue node denote order of 3). Observe trivially that in the left-hand graph, the two yellow nodes connected to the same blue nodes are not directly connected; while this is not true on the right-hand graph. Since there does not exist such mapping that maps each node from the one graph to the other, the two graphs are non-isomorphic.

It can be proved that $\Phi(G) \neq \Phi(G')$ since for each orange node in the two graphs, we have $x_i^1 = \sigma(\alpha_k x_i^0 + \beta_k \sum_j x_j^0)$, where $(i, j) \in E \implies x_i^1 =$

$\sigma(2\alpha_1 + 5\beta_1)$; for blue nodes, we have $x_i^1 = \sigma(3\alpha_1 + 7\beta_1)$. Therefore, we have the identical input vector for $\Phi(\cdot)$, which means $\Phi(G)$ and $\Phi(G')$ are the same.

### 1.3.2

Consider the same two graph from the section above. We have

$$\Phi(G) = \Psi(G') = \frac{1}{6}\sum_{i\in V} x_i^{(1)} = \frac{1}{6}\sum_{i\in V} \sigma(\alpha_2 x_i^1 + \beta_2 \sum_j x_j^1) = \frac{1}{6}\cdot(4\sigma(2\alpha_1+5\beta_1)+2\sigma(3\alpha_1+7\beta_1))$$

From the above equation, we conclude that $\Phi(\cdot)$ is unable to differentiate between $G$ and $G'$ with its parameters. If we set our graph prediction task to be classifying graphs isomorphic to $G$, which should return 1 for isomorphic graphs and 0 for non-isomorphic graphs. In this setting, $G$ and $G'$ are non-isomorphic graphs, which contradicts the result from $\Phi(\cdot)$.

### 1.3.3

As $\Phi$ is provably unable to solve task $\tilde{f}$ when K=1 but able to solve when K=2, it means that $\Phi_1(G) = \Phi_1(G'), \Phi_2(G) \neq \Phi_2(G')$ for two graph $G, G'$ such that $\tilde{f}(G) \neq \tilde{f}(G')$. As such, we can construct two graphs that has two different height (i.e. longest path in the graph), each is larger than 2. Since when K=1, the diffusion layer limits the GCN to only look at a node and its immediate neighbors, there is no way for it to know the height for certain. So if we construct two graphs, with the same number of nodes, each node with the same local neighborhood, but with different height, we are able to produce $\Phi_1(G) = \Phi_1(G')$ such that cannot differentiate between the two graphs. It is only possible for the GCN to see the full longest path when diffusion layer is set to 2, which allows the graph to see the different local neighborhood at a divergent nodes in the two graphs. Hence, we have $\Phi_2(G) \neq \Phi_2(G')$ that could distinguish between the two graphs with appropriate $\alpha_k, \beta_k$.

## 2 GNNs for the SBM Model

### 1.

Since the numeric value of the label $y$ is trivial, as long as we could label the nodes belonging to the same community under the same banner. If we have a optimal $\Phi(G, \Theta)$ that gives 1 for all nodes belonging to -1, and -1 for all nodes belonging to 1, then the loss function $\tilde{L}(\Theta)$ gives high loss for such consequence, which is undesirable for training.

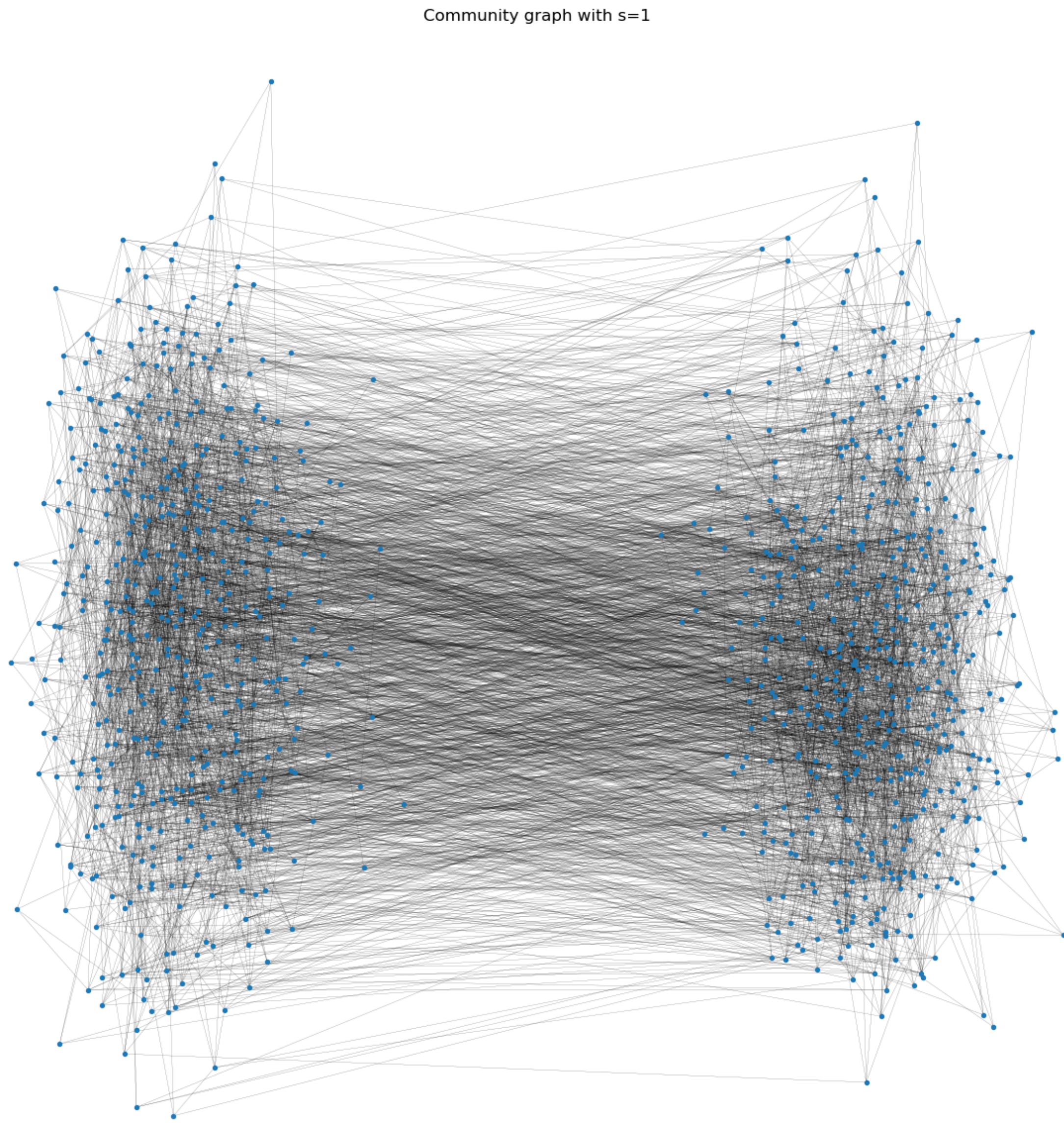
### 2.

The given loss function $L(\Theta)$ addresses the issue above by picking the minimal loss of either assuming a given community is labeled as -1 or labeled as 1. The loss function is 0 even if our estimator produces the exact opposite of the labels associated with every vertex.

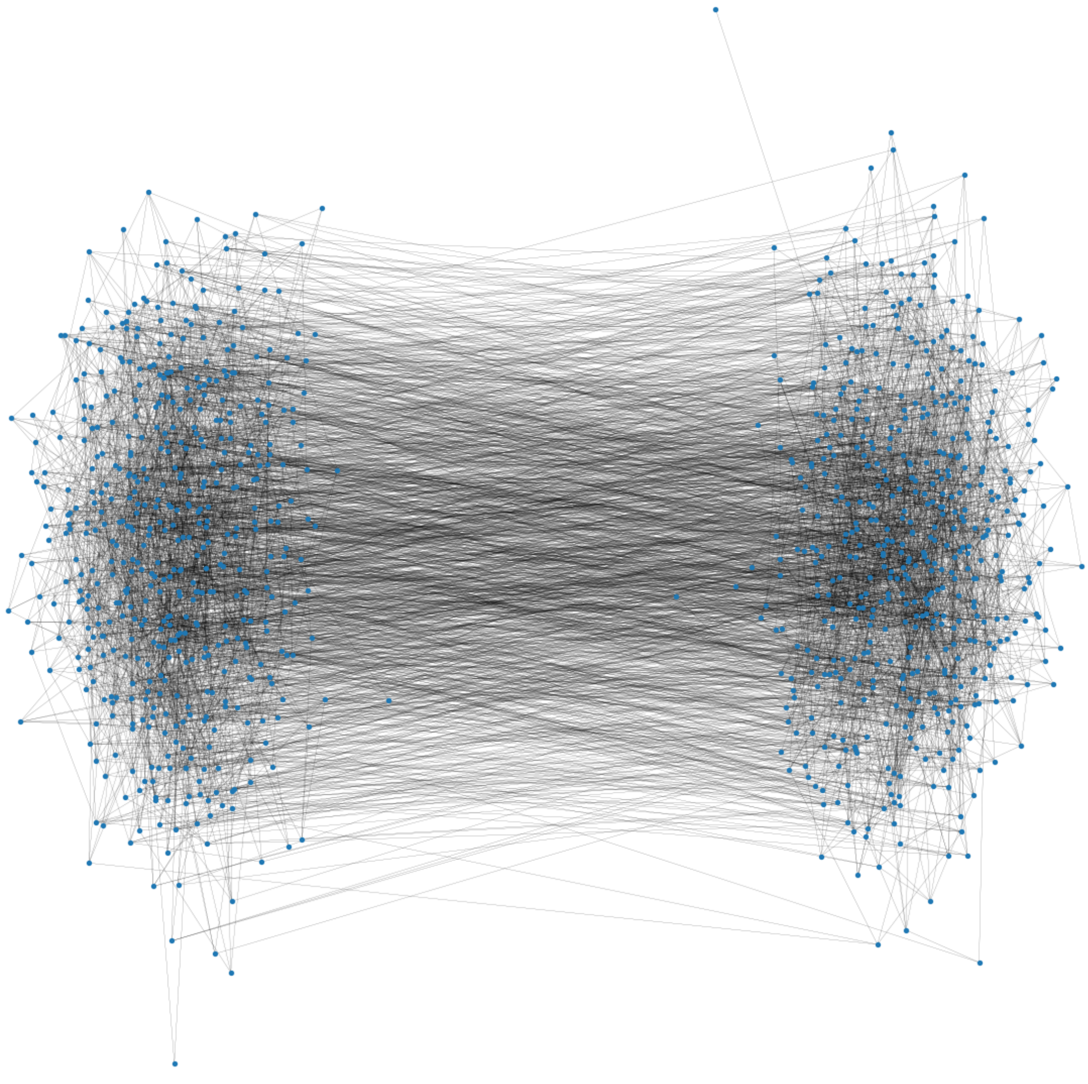
## 3.

If p=q, then the probability of developing an edge within a community is equal to the probability of developing an edge between the two communities. Hence, the two communities is effectively one community as the intra and inter relationship between vertices are essentially the same.

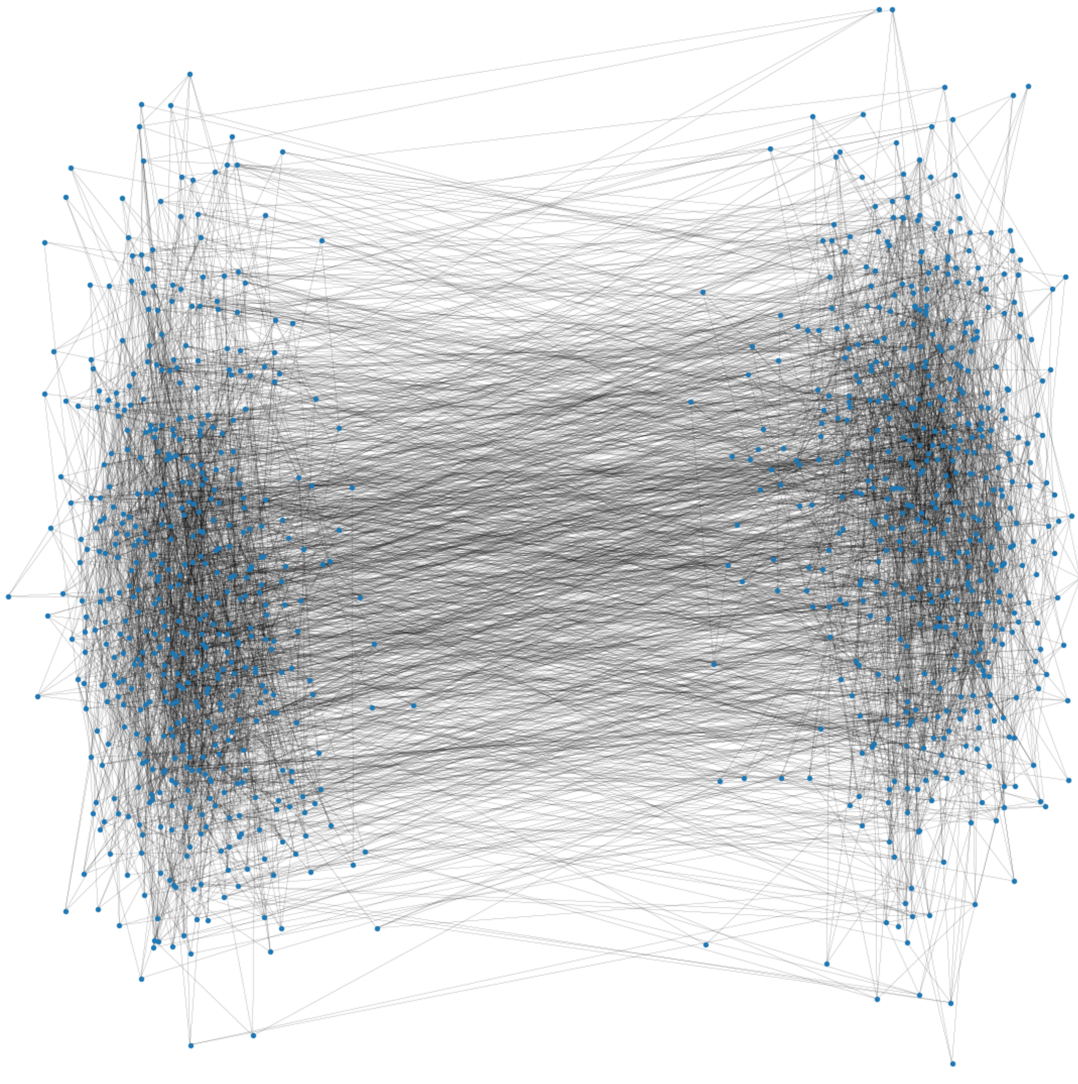
**4.**

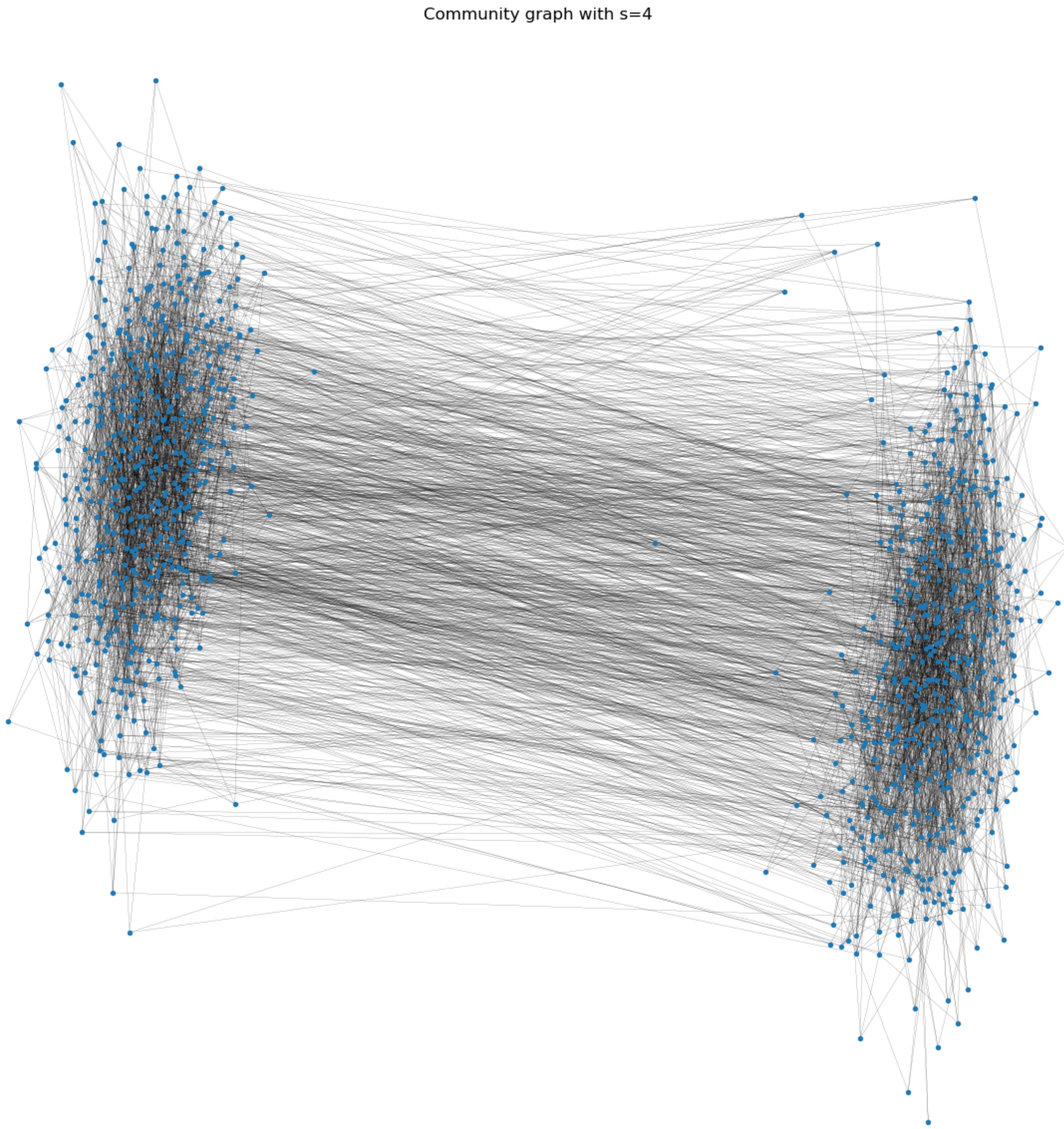


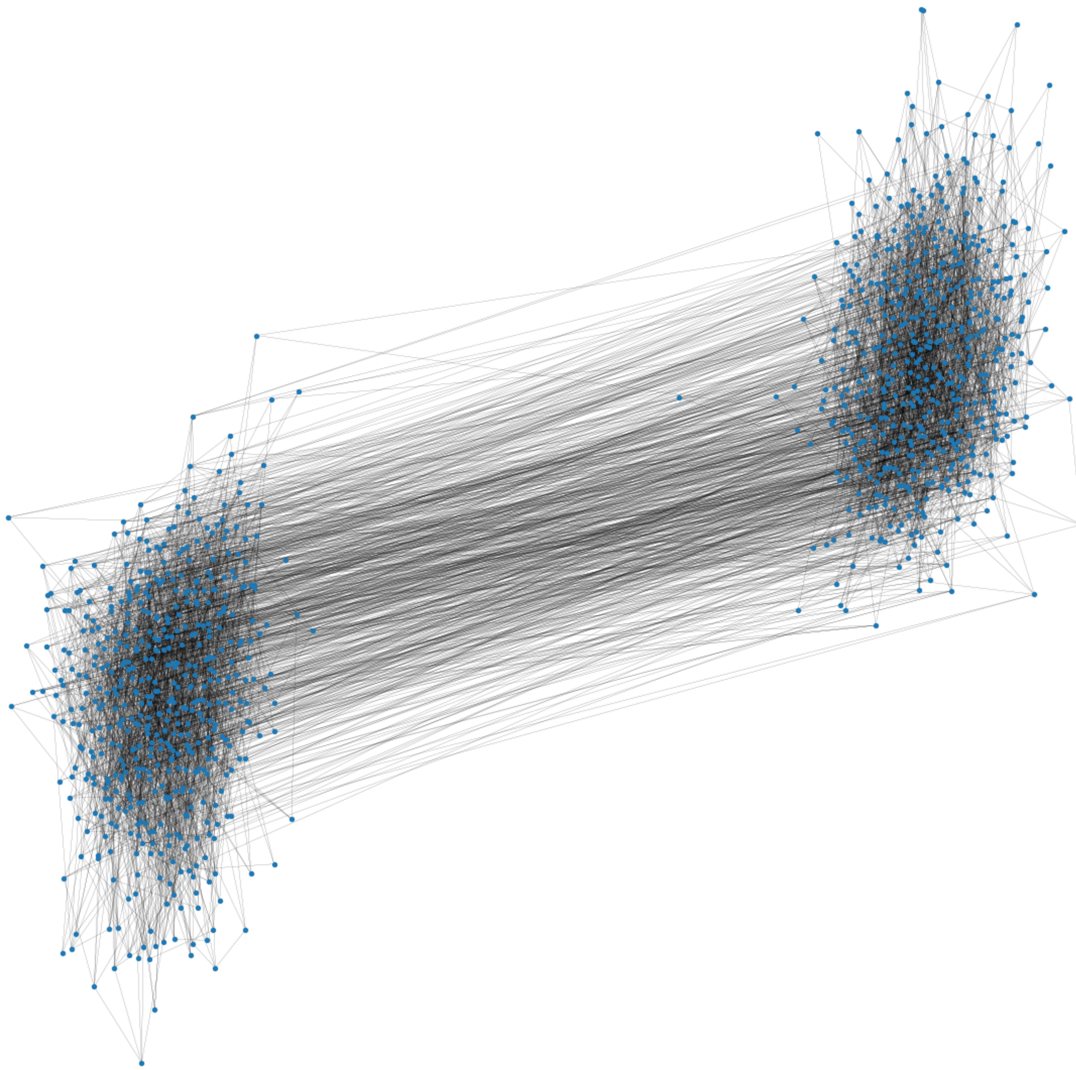
Community graph with s=2



Community graph with s=3




Community graph with s=4




**5.**

Since the task is clustering the nodes with the same label together, let us first assume our predictor is optimal, i.e., $\forall y_i = 1, \hat{y}_i = 1$ or $\forall y_i = 1, \hat{y}_i = -1$. Then we have $max\{\cdot\} = |V| \implies$ Overlap $= 1$.

Then let us assume we have the worst predictor, it is essentially labeling each node uniformly in $[-1, 1]$ since then the performance increases if it labels nodes with bias. In worst case, half of our predictions are correct therefore we have $max\{\cdot\} = \frac{1}{2}|V| \implies$ Overlap = 0. Hence, Overlap is bounded between 0 and 1. It is essentially a measure that reflects how much is our prediction better than the random guesses.

## 6.

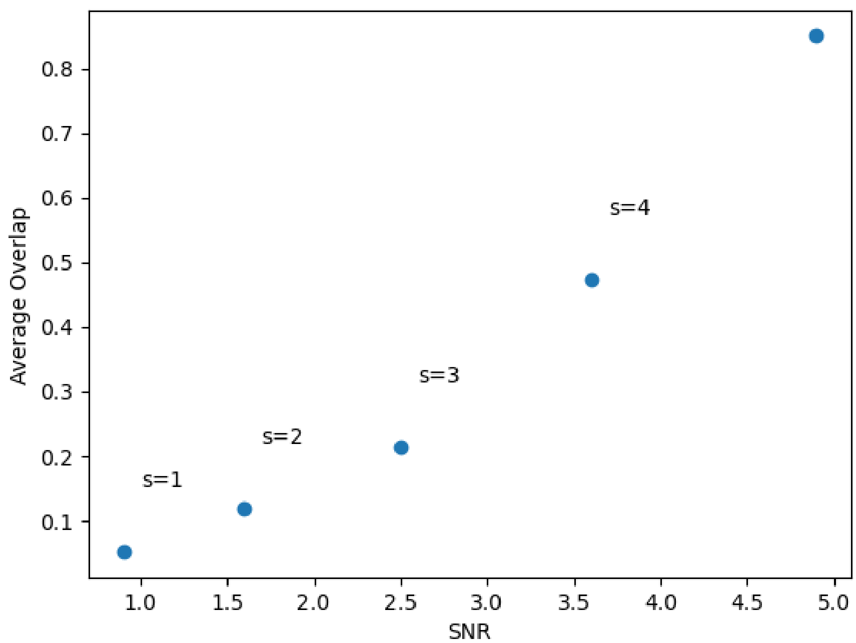



Figure 1: Plot of GCN Overlap with lr=0.06, K=15

The plot shows that as s increases from 1 to 5, the Signal-Noise-Ratio also increases because edges are more likely to be signal instead of noise in terms of intra-community relationships. Also, as I am tuning the number of layers, increasing the number of layers from a few layers to a dozens yields increase in Overlap. The training is done using Adam as optimizer, ReLU as non-linearity, and InstanceNorm1d as normalization.

**7.**

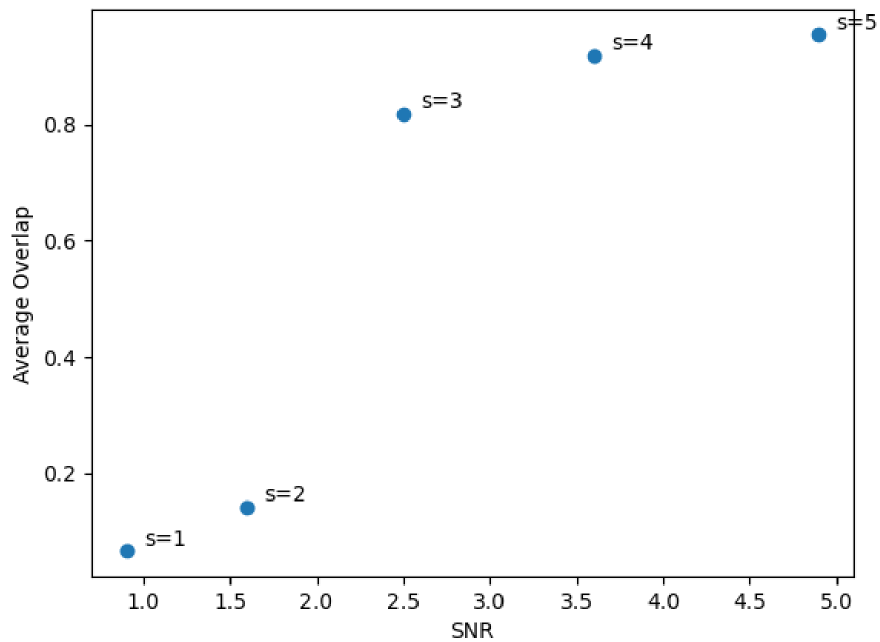



Figure 2: Plot of Spectral Clustering

Looking at the graph from Spectral Clustering (SC), we could see that its performance leapt greatly from s=2 to s=3. Generally, the baseline was able to increase the overlap when s is large, owing to the clearer distinction between the two communities.

Comparing the two overlap graphs from GCN and SC, we could see that GCN performs badly when s=3. When s=1/2, GCN and baseline showed relatively comparable performance; When s=4/5, baseline outperforms GCN by a larger extent, with the margin shrinking as s goes larger. Given a larger s, or clearer distinction between p and q, GCN might be able to shrink the gap even further, achieving comparable performance.