ISyE 6740 - Fall 2025
Final Report

Team Member Names: Will Coughlin
Project Title: Spectral Approximation for Large Networks

1 Problem Statement

Eigendecomposition of a graph Laplacian or normalized Laplacian[1] matrix is a key step in spectral clustering. For very large graphs, actual eigendecomposition presents a major computational bottleneck, often with complexity of $O(n^3)$ for dense matrices, where n is the number of nodes.[4] This expensive operation is infeasible for graphs encountered in modern big data and network analytics settings.

This computational barrier prevents the application of powerful spectral methods to problems such as community detection in massive social networks and the processing of complex image data, where n can easily reach millions or billions.

As such, approximating the eigenvalues and eigenvectors is a necessary workaround for these large-graph use cases. This project will implement and study two methods for approximate eigendecomposition and evaluate them with respect to performance and fidelity. The central purpose is to demonstrate the feasibility of spectral clustering for problems involving very large graphs, quantifying an acceptable loss of accuracy when compared to the baseline of a true decomposition. The report will first highlight the limitations of a moment-based method for estimation of graph spectra while illustrating its analytical utility. Subsequently, a Nyström method-based technique will be applied and evaluated, demonstrating its efficiency and accuracy in scalable spectral clustering.

2 Literature Review

2.1 Moment-Based Approximation

In their work, Cohen-Steiner et al. [1] describe a sub-linear time algorithm for estimating the spectrum of a graph. They first demonstrate that the moments of a graph's random walk transition matrix can be estimated using repeated simulated random walks. The empirical distribution of these moments is then shown to be connected to an approximation of the entire spectrum of the normalized Laplacian matrix. The primary advantage of this technique is computational speed, as the time complexity scales with respect to the number of random walks, not the number of nodes. The authors report good performance with 10,000 random walks over 20 trials. However, the key limitation is that this method alone is not sufficient to generate a spectral embedding for use in clustering, as it does not return the required eigenvectors. A strategy for overcoming this constraint will be addressed in the Methodology section.

2.2 Nyström Method Approximation

The Nyström Method for kernel approximation was developed by Drineas et al. [2] and later expanded upon with applications in spectral clustering. Pourkamali-Anaraki [5] introduces an extension of this work that defines an algorithm for estimating eigenvectors of a full similarity matrix by extrapolating from the eigenvectors decomposed from a much smaller sub-matrix based on a random sample of the original data. It is known that the normalized similarity (or adjacency matrix) $D^{-1/2}KD^{-1/2}$ is related to the normalized Laplacian $L_{\text{sym}} = I - D^{-1/2}KD^{-1/2}$ by a perturbation theory approach, as described in von Luxburg's "A Tutorial on Spectral Clustering." [6] Therefore, the estimated eigenvectors of the similarity matrix are directly applicable and useful for spectral clustering tasks.

3 Data Sources

3.1 Synthetic Graphs

For this study, 30 synthetic graphs of various sizes and community interconnectivity were generated using the *LFR* algorithm.[2] The graphs range in size from $N = 500$ to $N = 5000$, incremented by 500, yielding a total of 10 different sizes. The LFR algorithm utilizes a mixing parameter, $\mu \in [0, 1]$, which measures the connectivity between nodes of different communities. This value was varied across $\mu \in \{0.3, 0.5, 0.7\}$ to test

[1] L_{sym} as defined in von Luxburg[6]

[2] Implemented in the `networkx` package for Python: `https://networkx.org/documentation/stable/reference/generated/networkx.generators.community.LFR_benchmark_graph.html`

"low," "medium," and "high" levels of community mixing. A crucial feature of the LFR algorithm is that it provides a ground truth list of true community assignments, against which the spectral cluster assignments can be rigorously compared.

3.2 Real World Graphs

Additionally, several publicly available datasets from the Stanford SNAP[3] repository were analyzed, ranging from approximately 1,000 to over 1 million nodes. These sample graphs provide a better understanding of how the approximation methods perform on real-world data in familiar domains:

1. *EU Email Core*, with 1,005 nodes and 25,571 edges;
2. *General Relativity Collaboration Network*, with 5,242 nodes and 14,496 edges;
3. *DBLP Collaboration Network*, with 317,080 nodes and 1,049,866 edges;
4. *YouTube Social Network*, with 1,134,890 nodes and 2,987,624 edges.

3.3 Exploratory Data Analysis

Simple preliminary EDA was performed to help guide the eventual experiment configuration.

3.3.1 Modularity Analysis

For each graph with known true community labels, modularity[4] was computed. Modularity is a measure of the strength of community structure within a graph, quantifying observed connections within communities relative to what would be expected by chance. High modularity reflects a strong, distinct community structure or high-quality cluster assignments. For illustrative purposes, the three smallest $N = 500$ LFR graphs were visualized as heatmaps. Figure 1 shows high density of connections along the diagonal for low μ graphs, representing high density of within-community connections. For the middle and high end of the μ scale, the connection density is less concentrated within specific communities.

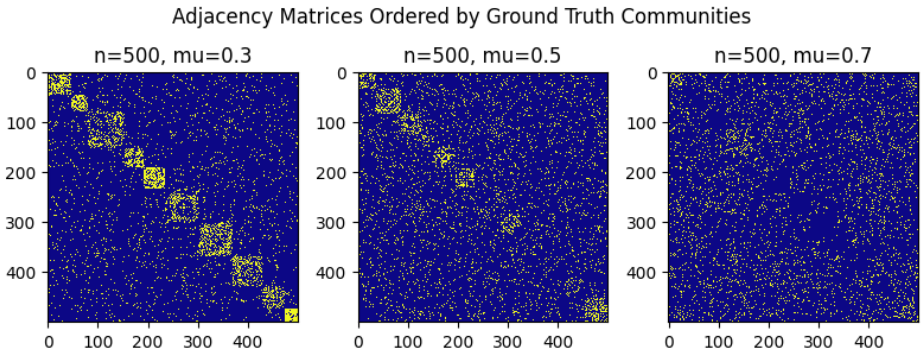

Figure 1: Adjacency heatmaps for three 500-node LFR graphs.

Of particular note for the LFR graphs, is that modularity decreases as the parameter μ increases. This is intuitive, because as high μ increases the number of connections between nodes in different groups, the communities become more mixed, or less insular. Figure 2 illustrates how average modularity for graphs of various sizes decreases as the μ value increases.

[3]Stanford Large Network Dataset Collection: `http://snap.stanford.edu/data`

[4]See `networkx` documentation for formula and implementation details: `https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.community.quality.modularity.html`

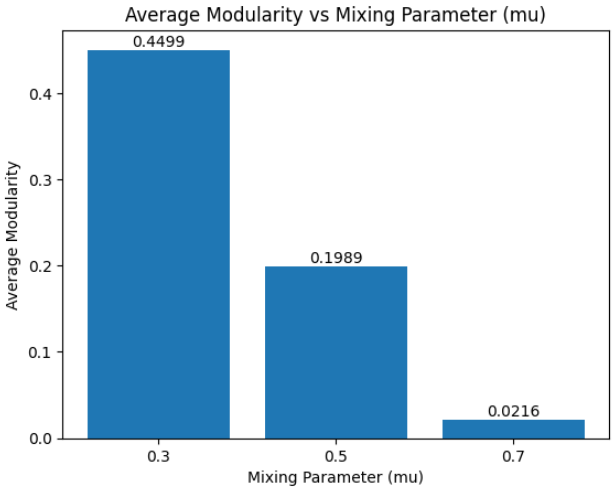

Figure 2: Average modularity measures for graphs of various mu values.

The modularity of real-world graphs with available ground truths were also calculated and recorded in Table 1. These results show that the LFR graphs with high μ are more challenging for meaningful quality comparison against a ground truth due to their low inherent community structure.

Name	N	μ	**Modularity**
LFR_0.3	(multiple)	0.3	0.4499
LFR_0.5	(multiple)	0.5	0.1989
LFR_0.7	(multiple)	0.7	0.0216
EU Email	1005	n/a	0.3138

Table 1: Summary listing of graph modularity.

4 Methodology

4.1 Baseline: Exact Spectral Decomposition

Where computationally feasible, the exact eigendecomposition was performed on the normalized Laplacian of each graph. This decomposition serves as the ground truth for evaluating the accuracy of the approximation methods. For graphs with known community labels, the number of unique communities was used to define the dimensionality of the spectral embedding. For graphs without ground truth, the embedding dimensionality was determined heuristically by generating a scree (or elbow) plot of the resulting eigenvalues. The chosen leading eigenvectors were then used to form the spectral embedding for subsequent K-Means clustering.

4.2 Moment-Based Approximation for Spectral Analysis

The moment-based method directly estimates the spectral density of the graph, yielding only eigenvalues and not the corresponding eigenvectors. The initial approach was to use these estimated eigenvalues to initialize the Inverse Iteration algorithm for eigenvector computation[3]. However, empirical testing determined that the spectral estimates lacked the precision required to compute viable eigenvectors for accurate clustering, rendering the Inverse Iteration approach infeasible.

Therefore, the utility of the moment-based approximation was constrained to its primary strength: Spectral Analysis. To evaluate this revised method, estimates of the graph spectra were produced on synthetic LFR graphs of varying size and modularity. Results were assessed based on execution time and the accuracy

of the eigenvalue distribution compared to the true spectrum using the Wasserstein/Earth Mover's distance (EMD).

4.3 Nyström Method Approximation for Spectral Embedding

The primary advantage of the Nyström Method is its ability to produce a full spectral embedding, allowing it to directly feed into an algorithm (e.g., K-Means) for spectral clustering.

For graphs small enough to compute a true eigendecomposition, a full pipeline was established to evaluate the Nyström approximation:

1. True eigendecomposition and selection of embedding dimensionality (k) via the scree plot heuristic or known community labels.
2. Clustering using the true eigenvectors (Baseline).
3. Clustering using the Nyström-approximated eigenvectors.

These results will be expressed in terms of runtime comparison and cluster assignment similarity using the Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI) scores, where ground truth labels are available. In cases where no true decomposition can be performed, the quality of the Nyström-based clustering is evaluated by intrinsic metrics and interpreted in isolation.

5 Results

5.1 Baseline: Eigendecomposition and Spectral Clustering Performance

5.1.1 LFR Graphs

First, the runtime complexity and speed-accuracy relationship of clustering with true eigendecomposition was evaluated. The runtime analysis confirms that this method exhibits cubic time complexity, which is prohibitive for large networks. Table 2 and Figure 3 show this rapid scaling, with a computation time increase from approximately 0.13s for $n = 500$ to 28s for $n = 5000$.

N	$\mu = 0.3$ **Time (s)**	$\mu = 0.5$ **Time (s)**	$\mu = 0.7$ **Time (s)**	**Average Time (s)**
500	0.0893	0.0818	0.2043	0.1251
1000	0.8024	0.9330	0.6666	0.8007
1500	1.8074	1.6668	1.1394	1.5379
2000	2.4958	2.4914	2.5129	2.5000
2500	4.3976	4.5378	4.8092	4.5815
3000	7.7091	9.3912	10.6283	9.2429
3500	12.5825	10.6211	10.6000	11.2679
4000	16.8404	17.2511	17.5425	17.2113
4500	23.0093	22.2285	21.0851	22.1076
5000	29.1169	30.0292	25.1017	28.0826

Table 2: Eigendecomposition runtime for LFR graphs of all N and μ.

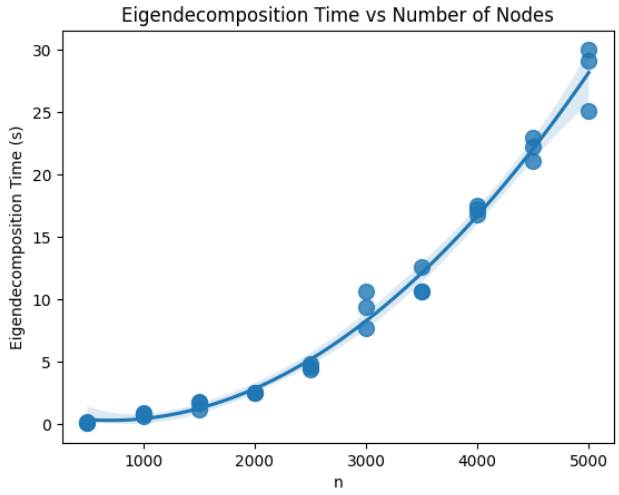

Figure 3: Cubic scaling of runtime for eigendecomposition of LFR graphs with regard to size N.

With respect to accuracy, even the baseline of true eigendecomposition struggled with LFR graphs exhibiting subtle community structure (high μ). This reflects the low modularity previously discussed, reinforcing the ambiguity of the clustering problem for these specific graph types. Table 3 shows these results averaged for all μ values.

μ	**Average Modularity**	**Average ARI**	**Average NMI**
0.3	0.4499	0.9997	0.9997
0.5	0.1989	0.6530	0.7312
0.7	0.0216	0.0245	0.0636

Table 3: Baseline clustering accuracy for LFR graphs averaged by μ.

5.1.2 SNAP Graphs Feasibility

Then, runtime and accuracy metrics were collected for the real-world graphs from SNAP, shown in Table 4. Only *EU Email* and *General Relativity* were small enough to perform true eigendecomposition. Since *EU Email* provides ground truth, modularity and accuracy were able to be computed. The computed modularity falls between the observed average modularity for the LFR graphs of $\mu = 0.3$ and $\mu = 0.5$, indicating low to moderate mixing of the community structures. However, based on the ARI and NMI scores, the accuracy is lower than what was observed for the LFR graphs with those μ values.

Name	N	**Runtime (s)**	**Modularity**	**ARI**	**NMI**
EU Email	1005	0.3646	0.3138	0.5016	0.6904
General Relativity	5242	50.6538	*n/a*	*n/a*	*n/a*

Table 4: Runtime and clustering metrics on SNAP real-world datasets. Only *EU Email* is small enough and has true labels so that all metrics can be captured.

5.2 Moment-Based Approximation: Eigenvalue Estimation and Shortcomings

The Cohen-Steiner moment-based method was primarily evaluated for its speed in estimating the spectral distribution. While the method successfully estimated the density of spectra for massive networks with computation time independent of size N, comparison with the ground truth on smaller networks showed mediocre accuracy. Cohen-Steiner et al. reported EMD of under 0.03 for all experiments; however, the results

for all LFR networks in this study were consistently around 0.1, with mixed results for the SNAP networks (Table 5). Furthermore, eigenvectors derived from these estimations by inverse iteration completely failed for downstream spectral clustering (ARI and NMI < 0.05). The failure suggests that the estimated eigenvalues from this implementation lack the precision required for accurate eigenvector estimation, rendering this approach impractical for spectral clustering tasks.

Name	N	**Baseline Decomp. Time (s)**	**Eigenvalue Approx. Time (s)**	**EMD**
EU Email	1005	0.3646	12.3000	0.1062
General Relativity	5242	50.6538	22.7400	0.0229
DBLP	317000	*n/a*	46.9001	*n/a*
Youtube	1134890	*n/a*	43.1202	*n/a*

Table 5: Moment-based spectral approximation metrics for SNAP graphs. DBLP and YouTube datasets are too large to compute baselines.

5.3 Nystrom Method Approximation: Spectral Clustering Performance

5.3.1 LFR Graphs Runtime and Speedup

The results for the LFR graphs confirm the improvement in runtime complexity for the Nyström method clustering. Runtime reductions were as high as approximately 29.4× when sampling 20% of nodes, and approximately 3.8× when sampling 60%. As expected, sampling more nodes leads to a less drastic speedup. However, the Nyström approximation's quadratic scaling, compared to the baseline's cubic scaling, offers massive improvements and makes formerly infeasible calculations possible. Figure 4 below shows how runtime and speedup change as graph size N increases. On the left, the baseline curve is fitted with an order 3 regression curve, while the Nyström curves are fitted with order 3. On the right, observe the diminishing returns as N increases, with the speedup rate tapering off. A summarized form of this data is also given in Table 6.

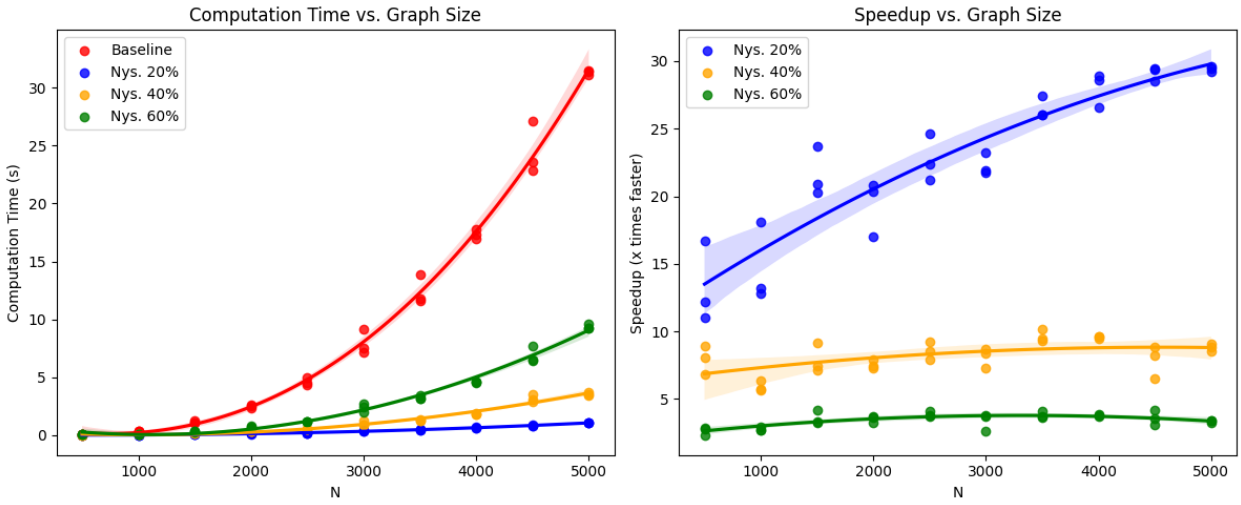

Figure 4: Computation time and speedup by graph size for baseline and each sample ratio.

N	Baseline (s)	20% Sample Time (Speedup)	40% Sample Time (Speedup)	60% Sample Time (Speedup)
500	0.0962	0.0075 (13.3151)	0.0123 (7.9284)	0.0367 (2.6439)
1000	0.3214	0.0223 (14.7017)	0.0545 (5.9017)	0.1154 (2.7854)
1500	1.1123	0.0514 (21.6240)	0.1403 (7.9432)	0.3108 (3.5949)
2000	2.4942	0.1301 (19.3823)	0.3305 (7.5587)	0.7115 (3.5257)
2500	4.6043	0.2025 (22.7492)	0.5378 (8.5667)	1.1919 (3.8624)
3000	7.9558	0.3575 (22.3060)	0.9912 (8.1369)	2.3907 (3.3756)
3500	12.4112	0.4674 (26.5148)	1.2837 (9.6520)	3.2539 (3.8080)
4000	17.3374	0.6199 (28.0191)	1.8157 (9.5484)	4.5617 (3.8007)
4500	24.5469	0.8423 (29.1281)	3.1610 (7.8341)	6.9062 (3.5848)
5000	31.3310	1.0652 (29.4141)	3.5478 (8.8371)	9.3426 (3.3553)

Table 6: Average computation time and speedup by graph size for baseline and each sample ratio.

5.3.2 LFR Graphs Accuracy and Trade-offs

With respect to accuracy, results show a non-monotonic trend across μ values. Figure 5 illustrates that ARI and NMI loss are relatively moderate in graphs with strong communities ($\mu = 0.3$), exhibit high spread in the $\mu = 0.5$ test graphs, and are minimal in graphs with weak community structure ($\mu = 0.7$). With ARI, the experiment produced loss results as high as 0.40 for $\mu = 0.5$ and 20% sampling, to as low as 0.0001 with $\mu = 0.3$ and 60% sampling. Another key takeaway of this visualization is that the difference in results between sampling percentages is greatest in the middle tier of community structure, and is the least in the high tier. As shown in Table 7, the low loss results for $\mu = 0.7$ graphs are attributable to the fact that the baseline values are low to begin with, once again reinforcing the difficulty in producing meaningful results in datasets with weak structure. In fact, average ARI loss for weakly structured graphs with 60% sampling is *negative*, meaning the Nyström approximation actually improves upon the baseline. Given the small magnitude of the baseline ARI, however, this could be attributable to noise.

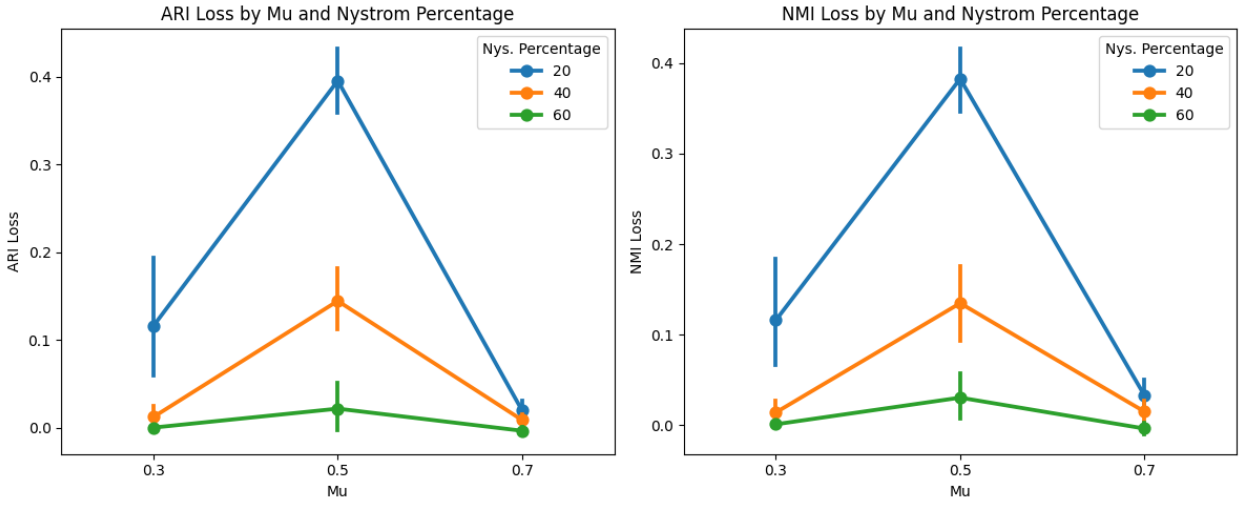

Figure 5: Average ARI and NMI loss against baseline for each LFR μ and sample ratio.

μ	Baseline ARI	20% Sample ARI (Loss)	40% Sample ARI (Loss)	60% Sample ARI (Loss)
0.3	0.9990	0.8829 (0.1161)	0.9868 (0.0122)	0.9989 (0.0001)
0.5	0.6790	0.2834 (0.3956)	0.5346 (0.1444)	0.6574 (0.0216)
0.7	0.0267	0.0069 (0.0198)	0.0179 (0.0088)	0.0302 (-0.0036)

Table 7: Nyström clustering accuracy (ARI) and corresponding loss relative to the baseline, averaged by μ.

5.3.3 SNAP Graphs Scalability

Experimental results on the SNAP graphs show runtime results in line with what was expected after experimenting on the LFR graphs. The graphs with computationally feasible baselines ran significantly faster using Nyström approximation than in true eigendecomposition, as shown in Table 8.

Name	N	Baseline (s)	20% Sample Time (Speedup)	40% Sample Time (Speedup)	60% Sample Time (Speedup)
EU Email	1005	0.3283	0.0161 (20.3913)	0.0626 (5.2444)	0.1451 (2.2626)
General Relativity	5242	68.2075	0.9736 (70.0570)	3.3192 (20.5494)	8.2928 (7.6700)

Table 8: Computation time and speedup for small SNAP graphs.

For the largest graphs, using the same sample ratios would still result in incalculable eigendecompositions. These large graphs were timed with much smaller ratios, as shown in Table 9. While the chosen sample ratios are small, they illustrate the method's ability to transform a problem involving a very large graph into one that is computationally approachable in a matter of seconds.

Name	N	1% Sample Time (s)	2% Sample Time (s)	5% Sample Time (s)	10% Sample Time (s)
DBLP	317080	0.0843	0.0986	0.1500	0.2005
YouTube	1134890	0.4155	0.5207	1.1172	1.9249

Table 9: Computation time for large SNAP graphs without baseline.

Accuracy was unable to be assessed on any dataset without a ground truth. Only the *EU Email* graph was tested with a ground truth label assignment. The results observed show a baseline ARI of 0.5016, putting it nearest to the $\mu = 0.5$ LFR graphs in terms of performance. The specific results for this graph, however, exhibit a novel pattern, wherein the 40% sample ratio clustering produced the best results, with an ARI loss of 0.1378, suggesting non-monotonic accuracy with respect to the sampling ratio.

6 Evaluation

6.1 Computational Feasibility and Scalability

The baseline analysis of true eigendecomposition established the prohibitive computational cost of spectral clustering, confirming the $O(N^3)$ complexity. The runtime for LFR graphs scaled rapidly, rising from under 1 second for $N = 1000$ to over 28 seconds for $N = 5000$. More critically, this bottleneck rendered exact spectral clustering infeasible for large real-world networks like DBLP ($N \approx 317,000$) and YouTube ($N \approx 1.1$ Million).

The Nyström method successfully addresses this limitation. For the largest SNAP graphs, Nyström approximation demonstrated the ability to run in under 2 seconds even with 10% sampling, where the resulting submatrix still had a high dimension of $m \approx 114.5k$. This result validates the Nyström method as a powerful approach for scaling spectral clustering to massive networks. For smaller, feasible graphs like

General Relativity ($N = 5242$), the speedup was quantified at up to 70.1× (20% sample), transforming a 68-second computation into a sub-second task.

6.2 Nyström Method Speed-Accuracy Trade-off Analysis

6.2.1 Performance on Synthetic Data (LFR Graphs)

On the graphs with strong community structure ($\mu = 0.3$), the Nyström approximation proved highly reliable. A 60% sample virtually matched the baseline, while even a 40% sample provided a strong result (ARI loss of 0.0122) with a 5.9× average speedup. This demonstrates that for well-defined communities, computational cost can be drastically reduced with minimal accuracy sacrifice. For weakly structured graphs ($\mu = 0.7$), the Nyström method accurately reflected the poor performance of the baseline. The small (or negative) ARI loss observed suggests that the approximation mirrors the inherent ambiguity of the problem rather than introducing significant error itself.

6.2.2 Performance on Real-World Data (SNAP Graphs)

The trade-off on real-world graphs proved more complex. For the *EU Email* dataset, the non-monotonic relationship between sampling ratio and accuracy suggests that simply increasing the sample size does not guarantee a better result. The 40% sample provided the optimal balance, achieving a useful speedup with a better fidelity than the larger 60% sample. This highlights that the quality of the resulting approximated kernel matrix is highly sensitive to the chosen sample and may require optimization beyond simple percentage increase.

6.3 Shortcomings of the Moment-Based Approach

While the moment-based method proved successful in its primary objective—rapidly estimating the spectral density in a manner that did not scale with N (linear time complexity)—its use for spectral clustering was not practical. The stable Earth Mover's Distance (EMD) across the synthetic LFR graphs, while decent, was insufficient to support accurate eigenvector estimation via inverse iteration. This confirms that the moment-based approximation, in this implementation, is suitable only for analytical tasks related to the spectral distribution itself, but not for direct use in spectral clustering.

7 Conclusion

The Nyström method overwhelmingly fulfills the primary goal of this project by enabling spectral clustering where exact eigendecomposition proves computationally infeasible on large networks. This investigation demonstrated the scalability of the Nyström approximation, achieving speedups of up to 70.1× on real-world graphs and unlocking efficient clustering on graphs exceeding a million nodes. Additionally, this report reveals that the effective use of this algorithm is highly dependent on the underlying graph's community structure. Ultimately, this shifts the central challenge of community detection from overcoming cubic time complexity to optimizing the selection of the subsample ratio, spectral embedding dimension, and number of clusters to assign. Overall, this successful investigation into the Nyström method shows it to be an effective and scalable tool for large network analysis.

References

[1] David Cohen-Steiner, Weihao Kong, Christian Sohler, and Gregory Valiant. Approximating the spectrum of a graph, 2017.

[2] Petros Drineas and Michael W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *J. Mach. Learn. Res.*, 6:2153–2175, December 2005.

[3] Tobin A Driscoll and Richard J. Braun. *Fundamentals of Numerical Computation*. SIAM, 8 2022.

[4] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the nystrom method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.

[5] Farhad Pourkamali-Anaraki. Scalable spectral clustering with nyström approximation: Practical and theoretical aspects. *IEEE Open Journal of Signal Processing*, 1:242–256, 2020.

[6] Ulrike von Luxburg. A tutorial on spectral clustering, 2007.