Otto-von-Guericke University Magdeburg

Faculty of Computer Science

Master Thesis

Spectral Clustering in Citation Networks: LLM-Based Similarity Graphs vs Citation Graphs

Author:

Abhivanth Murali

February 11, 2026

Advisors:

Prof. Dr. rer. nat. habil. Gunter Saake
Department of Computer Science, Otto-von-Guericke University, Magdeburg

Prof. Dr.-Ing. Robert Heyer
Faculty of Technology, Bielefeld University, Bielefeld

Dr.-Ing. David Broneske
German Center for Higher Education Research and Science Studies (DZHW),
Database and Software Engineering Group, Otto-von-Guericke University Magdeburg

MSc. Daniel Walke
Database and Software Engineering Group, Otto-von-Guericke University, Magdeburg

Msc. Rahul Mondal
Technische Hochschule Ingolstadt

Murali, Abhivath:
Spectral Clustering in Citation Networks: LLM-Based Similarity Graphs vs Citation Graphs
Master Thesis, Otto-von-Guericke University Magdeburg, 2026.

Abstract

The thesis aims to apply spectral clustering to the arXiv OGBN Citation Network. Spectral clustering is a graph-based clustering method, making it suitable for graph-based datasets, such as the arXiv OGBN. The main steps of spectral clustering are to first construct a similarity graph, such as a KNN graph, an epsilon-neighborhood graph, or a full similarity graph. Afterwards, the Graph Laplacian Matrix, Eigenvalues, and Eigenvectors are calculated. Finally, the data is clustered by k-Means based on Eigenvectors. The proposed method aims to implement three possible approaches to constructing the similarity graphs. The similarity graphs include large-language-model (LLM) text embeddings of the paper titles and abstracts, Citation edges, and a Multi-view graph approach that combines the first two. All these approaches were implemented in a mini-batch fashion, thereby reducing the time required for spectral clustering. The hypothesis is that LLMs can capture the semantic context of a text, and that their embeddings of the text data can yield coherent clusters. The thesis compares the above-said similarity graph approaches using the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) as the metrics. The results indicate that text embedding similarity graphs yield better metrics than citation edges and Multi-View. Additionally, it was noticed that smaller values of k in the KNN graph yield better metrics. The text-based embeddings spectral clustering outperformed the baseline K-Means by 30% to 40% on ARI using the same text embeddings.

Acknowledgments

I extend my deepest gratitude to Prof. Dr. Gunter Saake and Prof. Dr.-Ing. Robert Heyer for giving me the opportunity to work on this topic. Their support and suggestions helped shape the topic.

I would like to thank Dr.-Ing. David Broneske, for the valuable comments and suggestions during the thesis.

I would also like to thank Daniel Walke and Rahul Mondal for their constant support and mentoring throughout the thesis. Their patience in explaining topics helped shape the thesis and made it possible to reach submission.

Finally, I would like to thank my friends and family in Magdeburg and India for their support during the thesis.

Contents

List of Figures

List of Tables

1. Introduction

The proposed research, comparing spectral clustering of citation networks based on similarity graphs built from text embeddings generated by a Large Language Model (LLM) with citation graphs, is motivated by the need to understand and analyse academic research. Citation networks, such as the Ogbn-arXiv dataset, are networks of scholarly articles that constitute citation relationships, formal structural relationships represented as graphs. However, spectral clustering on citation graphs relies solely on citation edges, leading to clusters that are more aligned with author-based than with semantic-similarity clusters [Han et al., 2005].

Graphs can be used to represent real-world data, like social and biological data [Newman, 2003]. Basic Graph terminology can be understood from [Raj PM et al., 2018], which states that a Graph is $G(V, E)$, where V is the set of all Vertices (that is, the nodes), and E is the set of all edges. Graphs can largely be categorised into directed and undirected graphs. A citation network is the graph representation of academic articles [Radicchi et al., 2011]. For the purpose of representation, citation networks are made undirected in the implementation of the thesis for the purpose of applying spectral clustering.

The spectral clustering algorithm itself is a concept derived from graph theoryDin [2024]. It is an unsupervised machine learning algorithm. One of the important steps of spectral clustering is to construct similarity graphs like K-NN, ϵ-neighbourhood graph, etc [Luxburg, 2007]. One motivation of the thesis is to determine how the similarity graph constructed from different text embedding techniques affects the resulting clusters. [Patil et al., 2023] discusses in detail different embedding techniques which can be used to convert text into numbers. It discusses simple Embedding techniques like Bag Of Words, which simply count word occurrences, to advanced neural network-based methods like BERT, which also capture the contextual information of words. For the scope of this thesis, large language Models (LLMs) such as Qwen[Yang et al., 2025a] and Llama[Touvron et al., 2023], which are pre-trained with different parameter settings, are used to numerically represent texts associated with citation networks. Similarity graphs constructed with the assistance of LLMs will not only give structurally coherent clusters, but also thematically consistent

clusters, which is one of the main hypotheses of the thesis. This would help facilitate a literature review and trend analysis of research.

The OGBN-ArXiv dataset [Hu et al., 2020] is used to evaluate different clustering approaches. The Ogbn-arXiv dataset, with its wide variety of class label assignments, is the right choice for testing this hypothesis because it spans a broad range of computer science fields where semantic differences can significantly impact performance. Another reason for the choice is the limited availability and accessibility of citation network datasets that contain both text and citation information.

In addition, the thesis aims to challenge the boundaries of graph clustering methodology by leveraging advancements in natural language processing. The quality of the similarity graphs directly affects the resulting clusters [Luxburg, 2007]. Generating a similarity matrix from LLM embeddings leverages the power of pretrained models trained on large text corpora and enables one to capture contextual dependencies and latent thematic parallels. This can be contrasted with citation graphs, which depict the relationships between nodes. The advantages of both methods can be combined through Multi-View Spectral clustering, which treats LLM-embedded graphs as one view and citation graphs as another.

1.1 Research scope

For the scope of the thesis, I have considered Ogbn-arXiv as the dataset to compare spectral clustering based on:

1. **LLM embedding based similarity graphs:** This approach uses LLMs to convert text in each node into vectors, which are used to create similarity graphs.
2. **Citation-based similarity graphs:** This approach constructs an adjacency matrix from nodes and edge list, which is used to create similarity graphs.
3. **Multi View Spectral Clustering:** which takes 1 and 2 each as one view to perform spectral clustering.

All the implementation will be in Python programming language, which will use SciKit-learn [Pedregosa et al., 2011] library for spectral clustering and Mvlearn [Perry et al., 2021] for Multi-view spectral clustering. Evaluation metrics used are:

1. **Adjusted Rand Index (ARI)**
2. **Normalised Mutual Information (NMI)**

1.2 Research Questions

- **RQ1: Which approach to graph construction gives higher scores on NMI and ARI metrics, LLM-based text-embedding similarity graphs or citation graphs?**

  To answer this question, two pipelines are implemented, the same set of data points is passed to both, and the proposed evaluation metrics are compared across the three approaches. The resulting clusters are also visually plotted to understand their structure.

- **RQ2: Which Large Language Model(LLaMA, Qwen) with different parameters (3B, 32B) can better embed the Arxiv Ogbn dataset compared based on NMI and ARI score?**

  To answer this question, text embeddings are generated from different models, which are then used to perform spectral clustering. The results are then compared against evaluation metrics to determine which model best encodes semantic information.

- **RQ3: Can combining LLM-based text-embedding similarity graphs and citation graphs to perform Multi-view spectral clustering yields better NMI and ARI score?**

  To answer this question, multi-view spectral clustering is done on two views. One view is the similarity graph based on embeddings, and the other is the citation graph. The results of the evaluation metrics are compared across all proposed approaches. This will show how a hybrid strategy behaves in spectral clustering of citation networks.

1.3 Structure of this Thesis

This thesis is structured as follows. Chapter 2 serves as the background and theoretical knowledge necessary for citation networks, spectral clustering, embedding techniques, and LLMs. Chapter 3 will be the review of the related research and the existing methods that will be used to contextualise the proposed approach. Chapter 4 defines the description of the methodology that has been developed, like pre-processing the dataset into nodes and edges, generating embeddings, and performing spectral clustering. Chapter 5 will present the findings of the experiment and discuss the interpretation of the findings. The limitations that were observed and the potential future directions of the research are finally presented as a conclusion of the thesis in Chapter 6.

2. Background

This chapter introduces Graphs in 2.1 and discusses Citation Networks in 2.2 and Clustering and Spectral Clustering in 2.3, and at the end gives an introduction to Large Language Models in 2.4

2.1 Graphs

Graphs are often used to model data and its relationships. It can be used in many real-world scenarios, such as social media networks, transportation networks, and the visualization of relationships. A graph has two major elements: Edges and Nodes. For example, in a real-world road transport network like the Autobahn, highways such as the A6 and A14 are the Edges, and cities connected by these roads are the Nodes. A standard representation of graphs would be $G(V, E)$ where V is the set of vertices, which are nodes, and E is the set of edges.

The mathematics of modeling and analysis of relational data is the basis of graph theory and, therefore, is significant for the analysis of the structure and dynamics of complex graphs such as citation graphs [Sporns, 2018]. It describes graphs in terms of nodes and edges, how these two concepts are connected, and offers a set of metrics for connectivity, influence, and structural organization. Other basic concepts, such as adjacency representations, degree distributions, connectivity patterns, and path structures, can empower a researcher to model local and global interactions. Centrality, clustering coefficient, and other measures also help show the flow of information within the graph and the extent to which it is a community.

In addition, the sparsity of the graph [Diestel, 2025], edge weights, and directionality influence the computational feasibility and analysis performance, particularly for spectral algorithms that require Laplacian matrices and eigenvalue decomposition. Graph theory is therefore not only the theoretical structure of graph modeling and interpretation but is also used to design algorithms, ensure scalability, and recover actual, significant clusters and patterns in large-scale scholarly graphs [Majeed and Rauf, 2020].

2.1.1 Types of Graphs

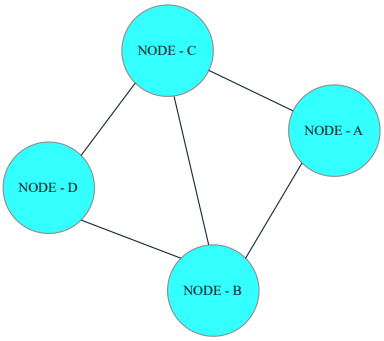
The two most common types of graph representations are ***Directed*** and ***Undirected*** graphs. This section will cover the types of graphs as defined in [Raj PM et al., 2018].

Undirected Graph

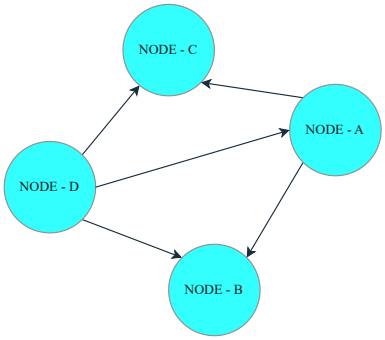
An undirected graph G has all its edges as bidirectional. For Example, an edge E between nodes A and B could be represented either way, AB or BA. Figure 2.1a represents an undirected graph with 4 nodes A, B, C, D with 5 bidirectional edges.

Directed Graph

A directed graph has all its edges with only one direction. For Example, an edge E starting from node A and ending directed towards node B, can only be represented as edge AB. Figure 2.1b represents a directed graph with 4 nodes A, B, C, D and directed edges.



(a) Undirected Graph



(b) Directed Graph

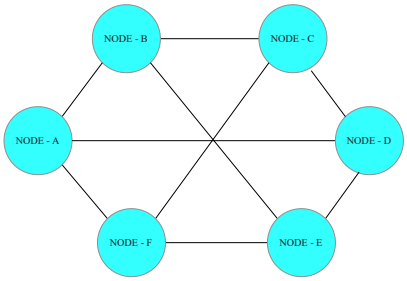
Figure 2.1

Complete Graph

A complete graph can be directed or undirected. Let us consider a graph G with n vertices V_n and maximum no of edges E_{max}. A graph is said to be a complete undirected graph if it satisfies $E_{max} = V_n(V_n - 1)/2$. It is constructed so that every unique pair of vertices has an undirected edge. Figure 2.3a represents a complete undirected graph. The graph is said to be a complete directed graph if it satisfies $E_{max} = V_n(V_n - 1)$. It is constructed so that every pair of distinct vertices has exactly two directed edges between them. Figure 2.3b represents a complete directed graph. 2.1a is also an example of a complete undirected graph.

Regular Graph

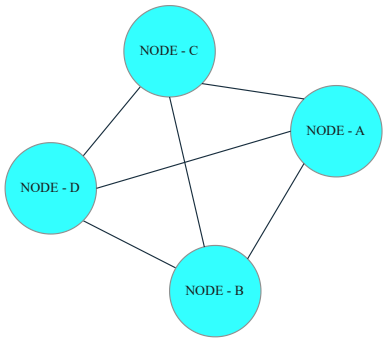
The degree of a Node in a graph is the number of edges connected to it. A complete undirected graph has all its nodes with the same degree k. Additionally, if we consider the inwards and outwards edges in directed graphs, a complete directed graph has all its nodes with equal inwards degree k_i and outwards degree k_o. A regular graph with degree k for all its nodes is called k *regular graph*.



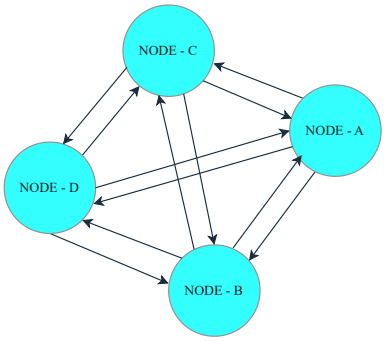
(a) Regular Graph

Bipartite Graph

The nodes of a bipartite graph can be split into two sets of N_r and N_l, where every edge links one node from N_r to one node from N_l. No node from N_r is connected to any other nodes of the same set.



(a) Complete Undirected Graph



(b) Complete Directed Graph

Figure 2.3

2.1.2 Graph Representation

Adjacency Matrix

A graph can be represented as an adjacency matrix A_ij. Each entry in the matrix represents the relation between each node and every other node in the graph. Every entry in the matrix is either 0 or 1. A 0 denotes no relation between nodes, and a 1 denotes an existing relation. Adjacency matrix of 2.1a can be represented as:

$$A_ij = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Edge List

A graph can be represented in an edge list E_l, where every edge e can be represented as $e(N_i, N_j)$. An edge list of graph 2.1a can be represented as :

$$(A, B), (B, C), (C, D), (D, B), (A, C)$$

Adjacency List

Adjacency matrices and edge lists can be effective representations of a graph, but they consume a large amount of space as the graph grows. An Adjacency List can be useful in such cases. It uses a hash table, where each node is a key, and its value is a list of adjacent nodes. This representation is more effective than an adjacency matrix and gives more information than an edge list. An adjacency list of 2.1a can be represented as:

A	B,C
B	A,C,D
C	A,B,D
D	B,C

2.1.3 Spectral Graph Theory Foundations

Spectral graph theory provides the mathematical and conceptual tools for understanding how graph structure influences connectivity, flow, and clustering. This discipline connects the combinatorial properties of graphs with smooth geometries by studying the eigenvalues and eigenvectors of graph-based matrices, such as the Laplacian [Wang et al., 2020]. The spectral properties reveal the extent of the interconnectedness among different portions of the graph, the formation of a community, and the spread of information across the graph. As a result, the spectral clustering algorithm relies on spectral graph theory, which makes it easier to project complex graphs, e.g., citation networks, social networks, and similarity networks, into low-dimensional spaces where valuable groupings are clearer and easier to recognize [Stevanović, 2018].

Key Concepts	Description
Graph Laplacian Formulations	Unnormalized Laplacian: puts attention on the rough connectivity changes. Normalized symmetric Laplacian: balances the node degree. Random-walk Laplacian: This is a graph diffusion or random walk model.
Eigenvalues and Eigenvectors	Eigenvalues may provide information about connectivity (e.g., number of connected components, spectral gap). The representation of community structure is provided by eigenvectors, which offer low-dimensional representations.
Theoretical Advantages and Disadvantages of Spectral Methods	Advantages: principled community detection, interpretable embeddings, connection to graph cut problems. Disadvantages: Graph construction is computationally intensive, sparsity, and noise.

Table 2.1: Key components of spectral graph theory

2.1.4 Graph Laplacian Formulations

The graph Laplacian is a tool in spectral graph theory that allows the computation of the flow of information over a graph. It can be characterized in various ways, giving the graph another perspective. The Laplacian, in its crude form, is the difference between the strength of a node's connection to its neighbors. The normalized Laplacian is symmetrized so that values are not concentrated on nodes with very high or very low degree, and thus becomes more resilient to the real-world connectivity between nodes, which is highly dissimilar. Some information about the diffusion and flow of the graph is captured by the random-walk Laplacian, which describes the movement of a random walker across the graph. They emphasize diverse structural characteristics, and one of them may influence spectral clustering outcomes, which is why it is necessary to choose the appropriate formulation to ensure an appropriate analysis. Table 2.1 describes the key components of spectral graph theory. Given below are the representations of the Laplacian matrix as described in [Mondal, 2023]:

Unnormalized Graph Laplacian Matrix : L
Normalized graph Laplacian Matrices:
- Symmetric $L_{sym} = D^{-1/2}LD^{-1/2}$
- Random Walk $L_{rw} = D^{-1}L$
Generalized Graph Laplacian matrix $L_G = D^{-1} = L_{rw}$
Relaxed Laplacian $L_\rho = L - \rho D$
Where D is the Diagonal matrix, A is the Adjacency matrix

2.1.5 Eigenvalues, Eigenvectors, and Laplacian Spectrum

The eigenvalues and eigenvectors of the Laplacian present some of the best structural properties of a graph. The number of zero eigenvalues indicates the number of

unconnected components of the graph. The second-smallest eigenvalue is a measure of the connectedness of the graphs in general, with large values indicating bottlenecks or poorly connected regions [Liu and Han, 2018]. The separation between successive eigenvalues, also known as the spectral gap, is relevant in determining whether the graph can be partitioned into useful clusters. The resulting eigenvectors are smooth signals that, in fact, represent communities or well-connected areas. Exploring the entire spectrum can reveal the eigenvalues, the dynamics of the eigenvectors, and thus whether spectral clustering will yield conclusive and correct group configurations [Cheng and Mishne, 2020].

2.1.6 Advantages and Disadvantages of Spectral Methods

Spectral methods have several strengths, including the ability to detect natural groupings, smooth structural patterns, and meaningful connectivity relations in a graph [Khan and Maji, 2019]. They provide a principled approach to reducing hard clustering problems to easier geometric ones. However, such approaches also have major weaknesses. Their performance is highly sensitive to the construction of the graph- there can be enormous differences in terms of edge weights, the definition of similarity, and the sparsity of the graph [Humbert et al., 2021]. They also entail computing eigenvectors, which can be extremely expensive for large graphs[Shi and Malik, 2000],[Ng et al., 2001]. In addition, spectral methods can yield unstable or misleading embeddings in low-noise or extreme degree-difference settings, or noisy connections in the graph, unless they are normalized with extreme care. Despite the issues, in a clean graph structure with distinct separation of the Laplacian spectrum, spectral methods can yield highly meaningful and interpretable clusters.

2.2 Citation Networks

Citation Networks are a type of graph used to model relationship between academic literature. The academic papers are the nodes in these diagrams, and directed edges reflect the citation relationships that form a multi-graphic pattern and capture the flow of knowledge between fields and through time [Radicchi et al., 2011]. Citation networks are regarded as the primary means of studying intellectual impact, topic development, collaboration patterns, and the impact of research on scientific progress. More elaborate forms of analysis, such as spectral clustering, community detection, and embedding-based modeling, may be supported by their graphical framework and become an asset to large-scale scientific research.

2.2.1 Structure of Citation Network

Citation graphs model the literature as a graph, the nodes of which are documents (research papers, journals, books), and the directed edges between documents have the form of citing one document by another, the directionality of the edge representing knowledge flowing directionally, and the age of the older document [Nishikawa and Koshiba, 2024]. In practice, the vast majority of graph-analytic methods reduce the directed citation graph to a form that maintains the same topology, or its weighted counterpart. Citation graphs are sparse and highly skewed; the majority of nodes have very few incident edges, and a small percentage of nodes (seminal papers,

surveys) are hubs with large in- and out-degree citations. The heterogeneity provided by this property has a variety of consequences for centrality, community structure [Radicchi et al., 2011].

The citation network has other structural properties. They form strong, small communities of papers on the same topic. They are time-stratified by paper publication date. They show assortative property as they form communities at multiple scales [Waltman and Van Eck, 2012]. Real-world algorithms must therefore be robust to self-citations, citation bursts, and domain-dependent citation norms. Choices about whether to preserve direction, normalize degrees, and weighted edges have a real effect on downstream clustering tasks.

2.2.2 Role of Citation Networks in Knowledge Discovery

A citation network is also an excellent substrate to diverse knowledge-discovery tasks because the topology of citation networks reflects complementary relations to textual similarity Ivanisenko et al. [2024]. Community discovery in citation networks can reveal research fronts and intellectual communities by identifying clusters of papers that cite other papers or a shared influential antecedent. Temporal analysis of citation relationships can reveal how topics evolve, how ideas spread, fuse, or fracture over time, and can be used to reconstruct the lineages of innovation paths.

The edge direction and weight are both used to model citation cascades, citation-based prestige measures like PageRank, and the diffusion of knowledge, which can be used to rank papers or to map interdisciplinary influence [della Briotta Parolo et al., 2020; Radicchi et al., 2009]. Besides pure topology, citation networks have been applied jointly with metadata and content features: citation community membership can be linked to author affiliations, venues, or keywords, enabling label clusters to be traded off in more understandable ways and helping detect editorial or domain bias Huong and Koch [2025]. citation-based clustering Radicchi et al. [2011] forms the foundation of literature recommendation systems, automated systematic review, and science of science research, although it is worth noting that citation relationships reflect both sociological and pragmatic behavior, such as reward systems, visibility, and citation conventions.

2.2.3 Challenges in Analyzing Citation Networks

Large-scale citation networks have exacerbated the problem of interdependent systems at the root of these issues. As a result, the analysis and algorithm design in these graphs have become more complicated [Nishikawa and Koshiba, 2024]. In fact, noise and incompleteness, like missing citations, OCR/text-mining errors in references, inconsistent DOI or author disambiguation, and incomplete metadata, all of which lead to the areas of spurious sparsity or false links that propagate error through the network [Van Der Vet and Nijveen, 2016]. Also, domain heterogeneity significantly influences citation practices, citation counts, and the level of detail in topics across different areas (e.g., physics vs. computer science vs. humanities).

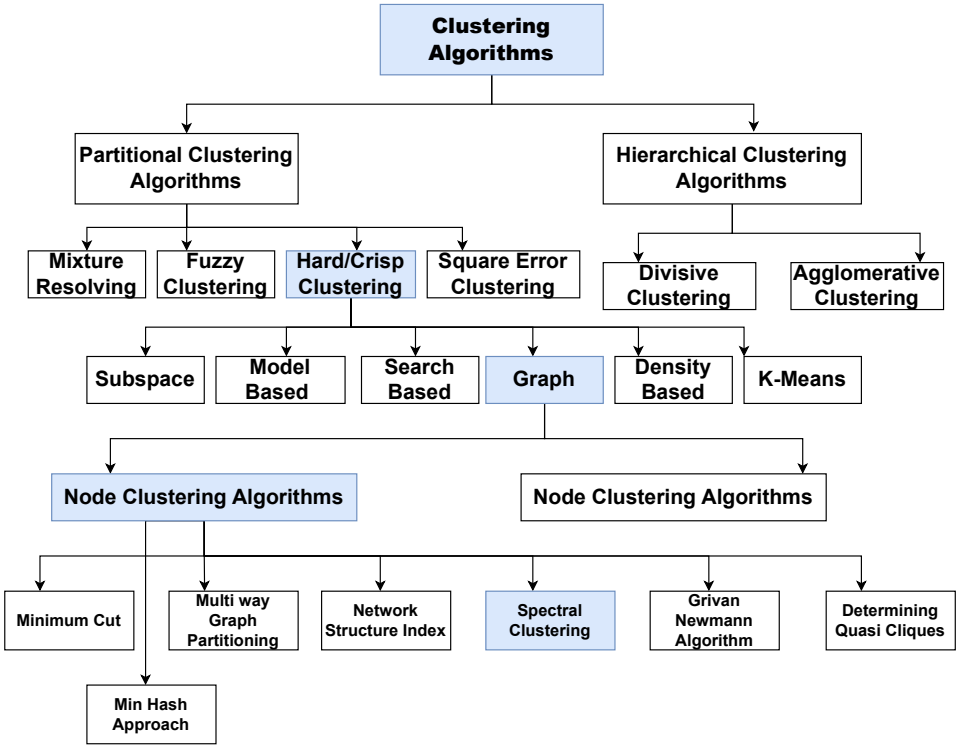


Figure 2.4: Cluster taxonomy represented by [Mondal, 2023], and from [Aggarwal and Wang, 2010] [Ezugwu et al., 2021]

2.3 Clustering

Clustering is the broad term for categorizing similar data into groups. In machine learning techniques Clustering comes under unsupervised learning. It means categorizing data without knowing its target labels or the data itself. According to [Pitafi et al., 2023], clustering algorithms can be categorized into Partitional and Hierarchical. Figure 2.4 shows the taxonomy of clustering as described from the works of [Aggarwal and Wang, 2010], [Ezugwu et al., 2021], [Mondal, 2023]. In Partitional clustering, a fixed number of clusters k is assumed at the start, and data points are iteratively assigned to these predefined clusters based on a similarity measure between each data point and every other. The main limitation of this type of clustering is the assumption of k without dataset insights, which leads to ambiguous clusters [Jain et al., 1999], [Ahmed et al., 2020]. In Hierarchical clustering, unlike the latter, each datapoint is considered as an individual cluster and iteratively combined based on similarity into final clusters. The main limitation of this clustering type is its relatively high time complexity [Müllner, 2011].

2.3.1 Spectral Clustering

Spectral clustering methods exploit the eigenstructure of graph representations to uncover useful clusters in challenging data sets. Instead of employing traditional distance-based clustering, the methods compress the data into a low-dimensional spectral space, where relationships are characterized by the connectivity patterns

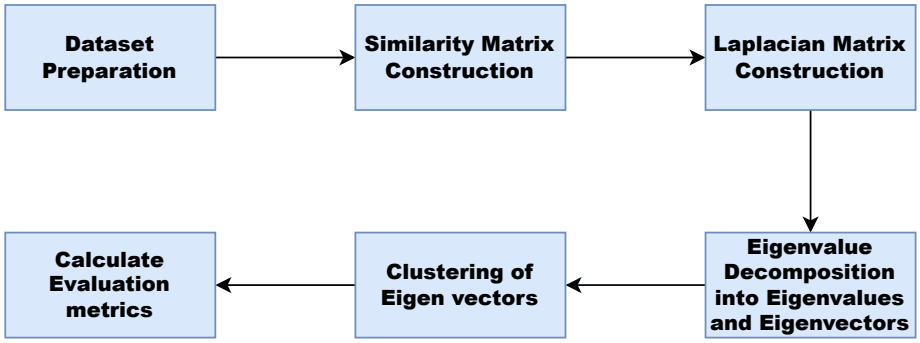


Figure 2.5: Spectral Clustering Flowchart

encoded in a graph [Luxburg, 2007]. Spectral clustering detects natural clusters (by examining the eigenvectors of the Laplacian) that are difficult to observe in the original feature space. It is especially useful in networks, similarity graphs, and high-dimensional data. Figure 2.5 shows the general flow of the spectral clustering algorithm.

2.3.2 Traditional Spectral Clustering Algorithm Workflow

The classical spectral clustering algorithm begins by creating a similarity graph in which the nodes are the data points and the edges represent the sum of the strengths of the associations between nodes [Yang et al., 2022]. According to this graph, a Laplacian matrix is constructed to capture the network's structural properties. The Laplacian is then reduced to eigenvalues and eigenvectors, and these exhibit patterns in a lower dimension that can be interpreted as the meaningful group structure in the data. Data points are then mapped into a novel geometric space using a specified number of the smallest-scale nontrivial eigenvectors, which cluster together, thereby making the clusters more disaggregated. Finally, the spectral embeddings are clustered using a conventional clustering algorithm, such as k-means, to obtain the final cluster assignments.

2.3.3 Similarity Metrics and Their Role in Clustering

The measures of similarity determine the degree to which two data points are certain to be connected in the graph, and this is very significant in the quality of spectral clustering [Luxburg, 2007]. Different domains can also be compared using similarity measures such as cosine similarity, Gaussian kernels, nearest-neighbor distances, or semantic relationships (e.g., language model embeddings). These measures determine the shape and density of the similarity graph, including the number of edges, the presence of clusters, and the distance between communities. A chosen similarity measure that provides a good representation of interesting relationships, whether structural, semantic, or statistical, results in a more interesting spectral embedding and a more refined cluster boundary.

2.3.4 Selecting the Optimal Number of Eigenvectors

The most important will be the number of eigenvectors that were successfully selected, which defines the dimensionality of spectral embedding, and the clustering result is

highly influenced by it [Tremblay and Loukas, 2019]. The number of eigenvectors used may not be sufficient to simplify the structure and merge two different communities, or may be excessive, introducing noise and obscuring cluster boundaries. In practice, the selection can be determined by analyzing the eigenvalue curve, with large gaps between eigenvalues, to propose the number of meaningful components. Domain knowledge, such as the expected number of clusters or classes in the data, can also be incorporated into the number [Zhu et al., 2021]. This is a tradeoff between the richness of representation and computational efficiency, even in large and complicated networks. Lastly, selecting the optimal set of eigenvectors ensures that the embedding captures the graph's important structure without unnecessary complexity.

2.3.5 Evaluation Metrics in Clustering

The performance of clustering in graph-based and embedding-based environments should be analyzed using a well-rounded set of metrics that capture the clustering's internal structure and its alignment with available ground truth. The correctness and consistency of the cluster assignments are measured using external clustering metrics, including the Adjusted Rand Index and the Normalized Mutual Information. For the context of the thesis, only external cluster metrics are considered, as different techniques employ different strategies to obtain eigenvalues and eigenvectors.

2.3.6 Scalability Challenges in Spectral Clustering

The high computational cost of Laplacian eigenvectors makes spectral clustering difficult to scale on large graphs. The algorithm typically involves forming large matrices and performing eigen-decomposition, and its computational cost grows exponentially with the number of nodes and edges. This is the problem of memory limitation, which arises when storing dense similarity matrices, i.e., in high-dimensional or fully connected graphs [Ling and Strohmer, 2020]. Various solutions are used to address these issues, including sparse graphs, approximate nearest neighbors, sampling, and randomized or iterative eigen-solvers. Spectral clustering can be very costly, particularly when dealing with very large or dynamic networks; though such optimizations are possible, they require careful design choices to balance efficiency and accuracy. Scalability problems are therefore a concern in the implementation of spectral clustering on real-world data, such as citation networks or semantic similarity graphs.

2.4 Large Language Models

Large Language Models (LLMs) today are the backbone of chat assistants such as ChatGPT, Claude, and Gemini. The core concept of LLMs evolved from the Transformer model by Ashish [2017]. They solved their predecessors' problems in training efficiency and memory. The transformer model led to more optimized variants, such as [Radford et al., 2018], [Radford et al., 2019], [Brown et al., 2020], etc. Natural Language Processing(NLP) is the study of how machines process human texts into a machine-understandable form. The advent of LLMs has advanced NLP research [Zubiaga, 2024].

Before LLM techniques such as Long Short-Term Memory (LSTM) models [Hochreiter and Schmidhuber, 1997], where each word in an embedding task was processed sequentially [Kandadi and Shankarlingam, 2025]. This led to the problem of a word's semantic context, which does not take into account where a word occurred in a text. Transformers, on the other hand, introduced the **self-attention** concept [Ashish, 2017], which took each word in a text and processed it in parallel. The proposed Transformer model had encoder and decoder layers. In this self-attention concept, the pairwise similarity of all words is calculated, then normalized and projected into vectors. Finally, techniques such as mean pooling and max pooling aggregate the vectors of each word into a single vector for the entire text [Xing et al., 2024]. Since Transformer models only understand numbers, words are tokenized before being passed to them.

Based on this Transformer model, subsequent models emerged, primarily **encoder-only** or **decoder-only**. Encoder-only models like BERT [Devlin et al., 2019] give attention to their preceding and succeeding words. On the other hand, decoder-only models like GPT-3 [Brown et al., 2020] attend to the word itself and its preceding words. Since decoder-only models are scalable by avoiding succeeding words and can predict the next word in a text, they are suitable for further research. While encoder-only models have their training parameters capped at 10 billion, decoder-only models like GPT-3 have 175 billion parameters, and newer models reach up to One Trillion parameters.

Training of a Transformer model according to [Ouyang et al., 2022] consists of 3 parts. First is the Pre-training on a large text corpus, followed by the Supervised Fine-Tuning stage with input-output pairs, and finally Reinforcement Learning from Human Feedback. Supervised fine-tuning techniques like [Wang et al., 2023],[Sanh et al., 2021], [Longpre et al., 2023] summarize how Transformers are fine-tuned so that the model can distinguish between correct and wrong responses. Reinforcement learning techniques such as [Stiennon et al., 2020] and [Schulman et al., 2017] explain how the correct responses of a Transformer model are improved.

The models chosen for the thesis include **Meta LLaMA 3.1 8B**, **Meta LLaMA 3.2 3B**, and **Qwen3 32B**. According to [Dubey et al., 2024], the LLaMA 3.1 8B model has a 32-layer decoder-only transformer architecture with 8 billion parameters and a 4096-dimensional hidden state. LLaMA 3.2 3B has a 28-layer decoder-only architecture with 3 billion parameters and a 3072-dimensional hidden state [Dubey et al., 2024]. The Qwen3 32B is a 64-layer decoder-only transformer model with 32 billion parameters and a 4096-dimensional hidden state [Yang et al., 2025b].

3. Related Work

This section aims to build on relevant and related research in a similar direction. It discussed spectral clustering research on cation networks and how LLMs can be used in spectral clustering.

3.1 Spectral Clustering in Citation Networks

Spectral clustering has emerged as a powerful data mining technique for analyzing citation networks, offering a conceptually important approach to communities, research areas, and hidden network structure in vast collections of scholarly articles [Chi et al., 2009]. Citation network topology provides valuable insights into disciplinary boundaries and intellectual impact, as it reflects the flow and development of knowledge through directed connections between publications. Spectral algorithms exploit the eigenvector decomposition of Graph Laplacians and project these high-dimensional connectivity patterns into a lower-dimensional embedding in which clusters are more recognizable and clearer.

As a result, spectral clustering has been widely used to identify topical clusters, map the scholarly space, and study the formation and dynamics of the research space. However, its performance is highly dependent on the quality of the underlying graph representation, which is why the effects of citation-only graphs and semantically enhanced similarity graphs on clustering results warrant further exploration.

3.1.1 Structural Properties of Citation Networks in Clustering

There are several structural properties of the citation network that directly affect the outcome of clustering: the network is directed, sparsified, and highly degree-heterogeneous with a few and highly cited hub papers and the long tail of poorly cited papers [Luxburg, 2007]. Directionality refers to temporal, causal knowledge flow, but most clustering algorithms operate on an undirected representation, so options for summarization or edge weighting have a profound impact. Topical cohesion is demonstrated by the local structures of triangle and co-citation/co-reference structures, the global properties of assortativity, community modularity, and the

presence of a giant connected component, which define whether clusters are coherent fields of research or disrupted across subdomains [Wang et al., 2017]. The dynamics of citations over time, i.e., the accretion of citations as time progresses, are non-stationary, and clusters identified at a fixed point in time may reflect the influence of the past rather than current topicality.

3.1.2 Graph Laplacian Approaches for Community Detection

Graph Laplacians are a principled collection of approaches for identifying communities in citation networks via discrete-time methods and for divulging continuous measurements of continuous partitions. By accumulating Laplacian operators that account for node degree and edge weights, scholars use eigen-decomposition to obtain low-dimensional representations in which clusters of highly connected papers geometrically cluster around each other [Villegas et al., 2025]. The Laplacian variants and normalizations are selected to minimize the impact of hubs and degree heterogeneity, and extensive experiments are conducted to assess the effects of various Laplacian variants on the stability and interpretability of partitions. By analogy, Laplacian embeddings are often combined with an existing clustering algorithm to generate community assignments, and multi-scale Laplacians, diffusion-based operators, and multi-view Laplacian fusion are used when interacting with two or more relationship types. The Laplacian model, therefore, provides conceptual connections to graph-cut optimization and a set of flexible computational tools for excavating cohesive scholarly communities from citation data.

3.1.3 Topic Discovery and Research Field Segmentation

Citation-based spectral embeddings have found wide applications in topic discovery and topic field segmentation, where the goal is to map topic structure into an understandable network. The clusters identified using spectral methods, in most cases, reflect existing subject areas or communities within conferences or journals, or new subfields. The process tends to perform well at discovering broad, well-separated domains and can also be used to identify hierarchical relationships between topics when multiple embedding dimensions are used concurrently. A combination of textual metadata with spectral clusters, such as titles, abstracts, and keywords, increases the interpretability of the clusters, enabling automatic label assignment and verification of human-curated subject categories [Dong et al., 2016]. Spectral techniques also provide temporal trends in the separation, amalgamation, and movement of clusters, offering insight into disciplinary changes in science. Nonetheless, the solution in which clusters are sought, and connections between the conduct of citation and relative closeness will still be crucial, and, accordingly, the comparison to ground-truth taxonomies and qualitative analysis will remain important.

3.1.4 Limitations of Citation-Only Graphs in Clustering

Citation-only graphs are indispensable for measuring scholarly influence; however, relying solely on citations imposes significant constraints on clustering. For one, citation links are very much influenced by social and cultural factors (self-citation, citation bias, visibility effects) and, therefore, they do not directly correspond to

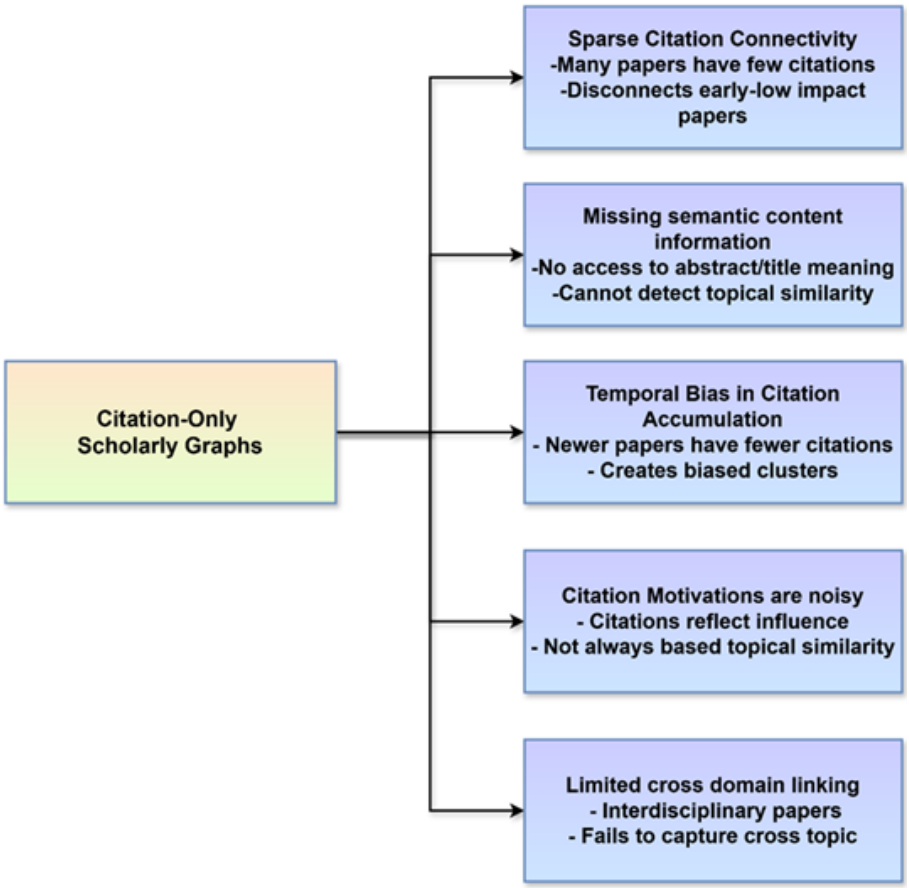


Figure 3.1: Limitations Citations Only Graph

topical similarity [Trower et al., 2025]. The second is that there can be a substantial number of papers in development or cross-disciplinary areas that may be only weakly linked or even separate nodes, leading to clusters that are either fragmented or misleading. The third point is that citation sparsity in the most recent publications and field-dependent citation norms may introduce biases in the results of citation-based clustering. Moreover, citation graphs emphasize past linkages and can be less accurate than conceptual ones naturally present in the text; thus, clusters based solely on citations can only recognize topical coherence that can be revealed by text embeddings. The major limitations of citation-only graphs are visually represented in 3.1

3.1.5 LLM-Based Semantic Similarity Graphs for Citation Networks

Big data produced by Large Language Models is another new paradigm for constructing semantic similarity graphs of scholarly data, in that the results of citation linking cannot provide the same deep, context-sensitive knowledge of research papers that giant data does. Unlike the classical similarity of bags of words or even of keywords, LLM embeddings capture subtle meaning in titles, abstracts, and full text and can discover topical similarities when there are no, or only a few, direct citation links between two documents. Such embeddings can be used to compute the similarity score between pairs of papers, which are then visualized as weighted

graphs, where edges indicate the semantic distance between the papers rather than citation behavior. These graphs possess many advantages: they reveal latent or emergent themes that have not yet encoded in the citation patterns, reduce biases between regions caused by citation standards, and provide more useful information in order to cluster the results using the combination of conceptual relatedness and linguistic knowledge

The GATFELPA method of [Tang et al., 2025] combines a graph attention network with the label propagation algorithm DPCELPA to assign final clusters. The evaluations are performed on well-known citation networks, Cora, CiteSeer, PubMed, and ogbn-arxiv. A Jaccard index of node similarities is passed to the attention network, and the embeddings are passed through neural network layers to optimize the similarity before passing it to the label propagation algorithm.

Meanwhile, classification techniques like graph neural networks may treat these spectral embeddings as a potential source of initialization or additional features that can facilitate generalization and robustness. However, scalability remains the most important issue: the need for efficient approximations, sparsification methods, and distributed computation strategies cannot be avoided when calculating eigenvectors for extremely large graphs. On top of highlighting the potentials of spectral methods, previous work on spectral clustering and classification at benchmark datasets has also been pointing out their limitations in large-scale scenarios [Shi and Mishne, 2025] and thus, providing the significant insights that pave the way for the emergence of hybrid models which integrate structural, semantic, and learned representations that aligns with the objectives of the proposed research.

4. Methodology

In this chapter, the implementation of experiments and the dataset structure and preparation are explained. The first section explains details of the dataset 4.1. The following section 4.2 explains the experimental setup along with the preprocessing step. The Section 4.3 explains all the proposed Spectral Clustering setups.

4.1 Arxiv Obgn Dataset

The Open Graph Benchmark (OGB) [Hu et al., 2021] is a set of large, heterogeneous, and well-calibrated graph datasets directly applicable to the evaluation of machine learning models on more realistic network learning problems. The latter is the OGBN: Open Graph Benchmark - Node Property Prediction, which includes a small number of graph-structured datasets from networks, e-commerce, biological interactions, citations, and heterogeneous academic knowledge. These datasets possess odd structural properties, feature descriptions, and predictive objectives that enable researchers to investigate the generalizability and robustness of graph learning and clustering algorithms.

OGBN-products of the OGBN is an Amazon co-purchasing network, modeled as a graph with products as nodes and items that are often bought together as edges. The node features are obtained as a bag-of-words compressed description of product features, and the classification task, using a realistic, popularity-based split, predicts the top-level product category. Similarly, the OGBN-Protein dataset of biologically relevant protein-protein interactions is an undirected, weighted graph with 8 species with 8-dimensional confidence features on the edges. The prediction is multi-label, and the model is expected to predict 112 possible protein activities. The data are divided by species to assess inter-species generalization.

The OGBN suite also includes large-scale citation networks that reflect the actual patterns of communication in academia. The citation links between Microsoft Academic Graph (MAG) computer science papers are represented in the OGBN-arXiv dataset, where each paper is represented as the 128-dimensional average word embeddings of its title and abstract. The classification exercise is to find the primary

Scale	Name	Package	#Nodes	#Edges*
Medium	ogbn-products	>=1.1.1	2,449,029	61,859,140
Medium	ogbn-proteins	>=1.1.1	132,534	39,561,252
Small	ogbn-arxiv	>=1.1.1	169,343	1,166,243
Large	ogbn-papers100M	>=1.2.0	111,059,956	1,615,685,872
Medium	ogbn-mag	>=1.2.1	1,939,743	21,111,007

Figure 4.1: OGBN Node Property Prediction Datasets Overview, Borrowed from [Stanford, 2025]

subject category in 40 arXiv fields, using a chronological split: training on older papers and testing on new ones. For the thesis experiments, I have chosen OGBN-arXiv and performed spectral clustering, which is an unsupervised learning method. The reason for choosing this is due to the readily available citation information and text data.

There is also OGBN-papers100M, which triples this structure to over 111 million scientific publications and citations, making it one of the largest publicly available graph datasets. The arXiv subset itself is not a true graph and only includes the associated label data for 172 subject areas, but the rest of the dataset includes contextual citation links, and models must rely heavily on the graph structure to make informed predictions. Figure 4.1 shows the overview of available datasets in OGBN

4.2 Experimental Setup and Preprocessing

All experiments are conducted on the university server, which has a processor **Intel(R) Xeon(R) Gold 6338 CPU with 2.00GHz**, **256 GB** of RAM, and an Nvidia RTX A6000 **graphics card**. The whole experiment can be divided into stages to perform spectral clustering on the dataset. A Python environment with **Miniconda** was set up with Python version **3.12.10**. A table of libraries and their used versions can be found at A.1. Preprocessing stages are explained individually in the sections below.

4.2.1 Citation Dataset Builder

The first stage of the experiment is to load the dataset and to perform pre-processing. Figure 4.2 shows the flow of data loading and pre-processing it to nodes and edges, which represent the citation network. The **ogb** library provides the methods needed for importing the dataset. The library downloads the necessary dataset files when loading; in our case, the OGBN arXiv files. First edge index pairs with source and target indices are loaded. The node indices with label id are then loaded. The downloaded files also include supplementary files that map each node to relevant information, such as titles, abstracts, and labels. Using these files, we map the indices to the nodes and edges CSV files.

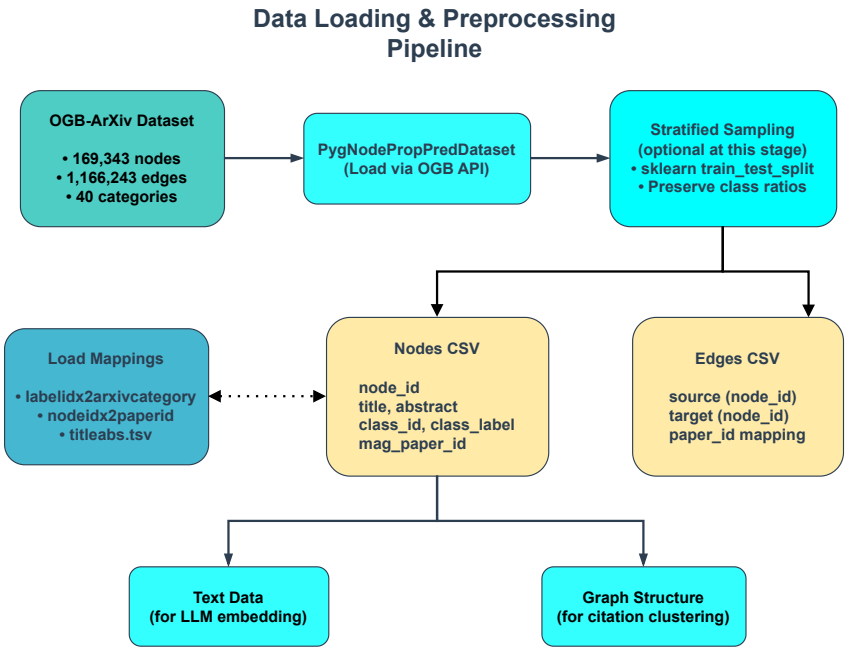


Figure 4.2: OGBN arXiv Dataset Loading and Pre-Processing

There is also a stratification of the dataset, which can be done at this stage. Stratification means loading only a subset of the dataset, while preserving the class distribution of the full set. It is Optional here to load a small dataset to jump-start experiments and get quick results. To get the stratified set pass the desired number of nodes to the loader, when not needed, pass the total available number of nodes.

4.2.2 Embeddings Generation Pipeline

After building the citation network, we end up with nodes and edges. From the Nodes, we can now generate embeddings by using LLMs. Figure 4.3 shows the workflow for embedding generation. We start by tokenizing the texts, which are the titles and abstracts of each paper, in each node. This converts each word into a token, which is a number. The tokenizer also adds Paddings for empty spaces in the text. These Paddings can be later backtracked using Attention Masks, which are also generated by the tokenizer. The tokenizer used is the **AutoTokenizer** from the **HuggingFace Transformers** library. This will enable us to load the correct tokenizer for each model. The Tokenizer can be imagined as a dictionary that maps each vocabulary item to a number. After tokenization, the tokens for each paper are passed into the LLM's transformer model, and embeddings are generated. Gradient tracking is disabled for embeddings since we are not training the model. The LLM at the end gives a single vector for each token in the paper. To obtain a single vector for a paper that captures its semantic concept, we compute a weighted average of all the paper's vectors. This weighted average is called the **Mean Pooling**. The Mean Pooling formula is:

$$e_i = \frac{\sum(v_t.m_t)}{\sum(m_t)}$$

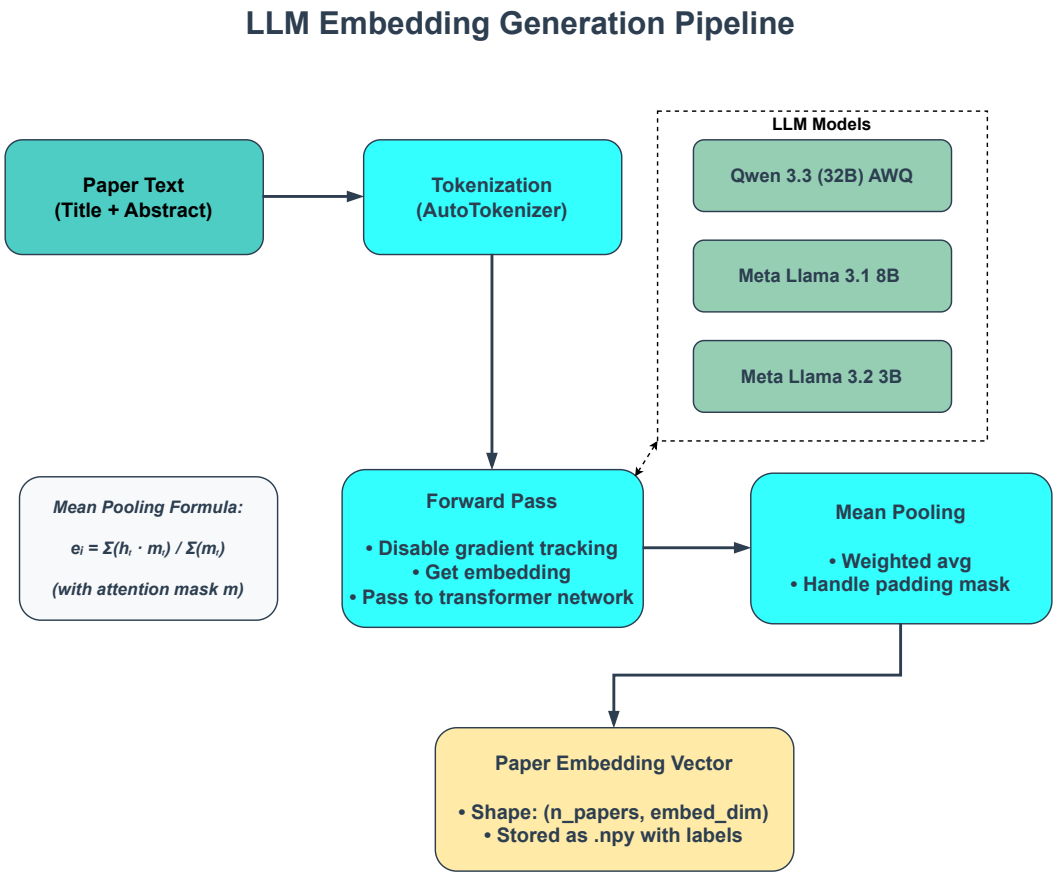


Figure 4.3: Embedding Generation Pipeline

Here, the embedding for the paper i is obtained by the summation of the dot product between the vector value of a token v_t and the mask value of the corresponding token m_t, and dividing it by the summation of the mask values. The embeddings are then stored in a **.npy** file along with the class id and node id of the corresponding paper. This file format is a **NumPy** library file format for storing arrays.

4.3 Spectral Clustering Setup

After the initial setup, the actual clustering pipelines can be set up. The thesis's clustering setup has two stages. The first one is the Eigen Gap Analysis, and the second is the actual Clustering. The clustering setup is divided into three. They are as follows:

1. Citation only Clustering
2. Embeddings only Clustering
3. Multi-view Clustering

All the methods are executed in mini-batches. It means that clustering is first performed on a small subset of the entire dataset. The remaining nodes are then

assigned to the pre-existing clusters. Each method is explained in the following sections along with the Eigen Gap Analysis.

4.3.1 Eigen Gap Analysis

Eigenvectors and Eigenvalues calculated as part of spectral clustering can be used to estimate the optimal number of clusters we can predict based on our data points (i.e., nodes). While spectral clustering can be used as a classification tool and compared with other methods, there exists an optimal number of clusters k in the calculated spectral embedding. It is obtained by sorting eigenvalues in ascending order; the initial values are small, and as we move further away, the values tend to increase. Small values indicate strong connectivity. The index of the values at which the largest gap occurs indicates that beyond these values, there are no meaningful clusters. The obtained K value can be used as input in the spectral clustering as the desired number of clusters. As the thesis presents 3 different approaches to spectral clustering, there is a separate eigen-gap calculation for each approach. This is due to approaches that use different inputs to cluster the same data.

4.3.2 Citation Only Clustering

In this setup, as the name suggests, the papers' citations are used as the basis for clustering. Figure 4.4 explains the overall flow of this setup. As the first step, the node and edge files are provided as input. A stratified split of the nodes is performed. This split is done to maintain the class distribution as it is in the whole dataset. The assumption behind the split is that it will allow us to approximate the entire dataset using a small subset, thereby enabling faster processing. Here, there is an option to add all nodes connected to the split nodes, thus expanding the subset. This option is included in the experiments. It looks for outgoing or incoming edges directly connecting to the split set and adds them to the subset. The subset is now used to construct a **Citation Sub Graph** and stored as a **Sparse Adjacency Matrix**. Now, spectral clustering is performed on the subgraph to obtain initial clusters. This is done through the **Scikit-Learn** library. The adjacency matrix is passed into the library's **SpectralClustering** method. First, the **Graph Laplacian** is calculated, and then the **Eigenvectors** are calculated. Finally, **K-Means Clustering** is done on the calculated vectors. With this, the first Spectral clusters and embeddings are obtained for the subgraph.

Next, the remaining nodes, which were not part of the initial subgraph clustering is clustered in batches. First citation vectors are constructed based on their connectivity to the previous stratified nodes. Next, the cosine similarity is calculated between the remaining node and each of the previous stratified nodes. Then the remaining nodes are now ready to be processed in mini-batches. In each batch, the K most similar stratified neighbors are selected. Each node in the batch is projected into the spectral embedding space using a **K-NN weighted average**. The similarity score between the k most similar nodes and the selected node in the batch is normalized, and the normalized score is pushed into the spectral space with spectral embedding of the k most similar nodes. Now the Node is ready to be assigned to its cluster, as it is already in the spectral space. The node's nearest cluster centroid is its new assignment.

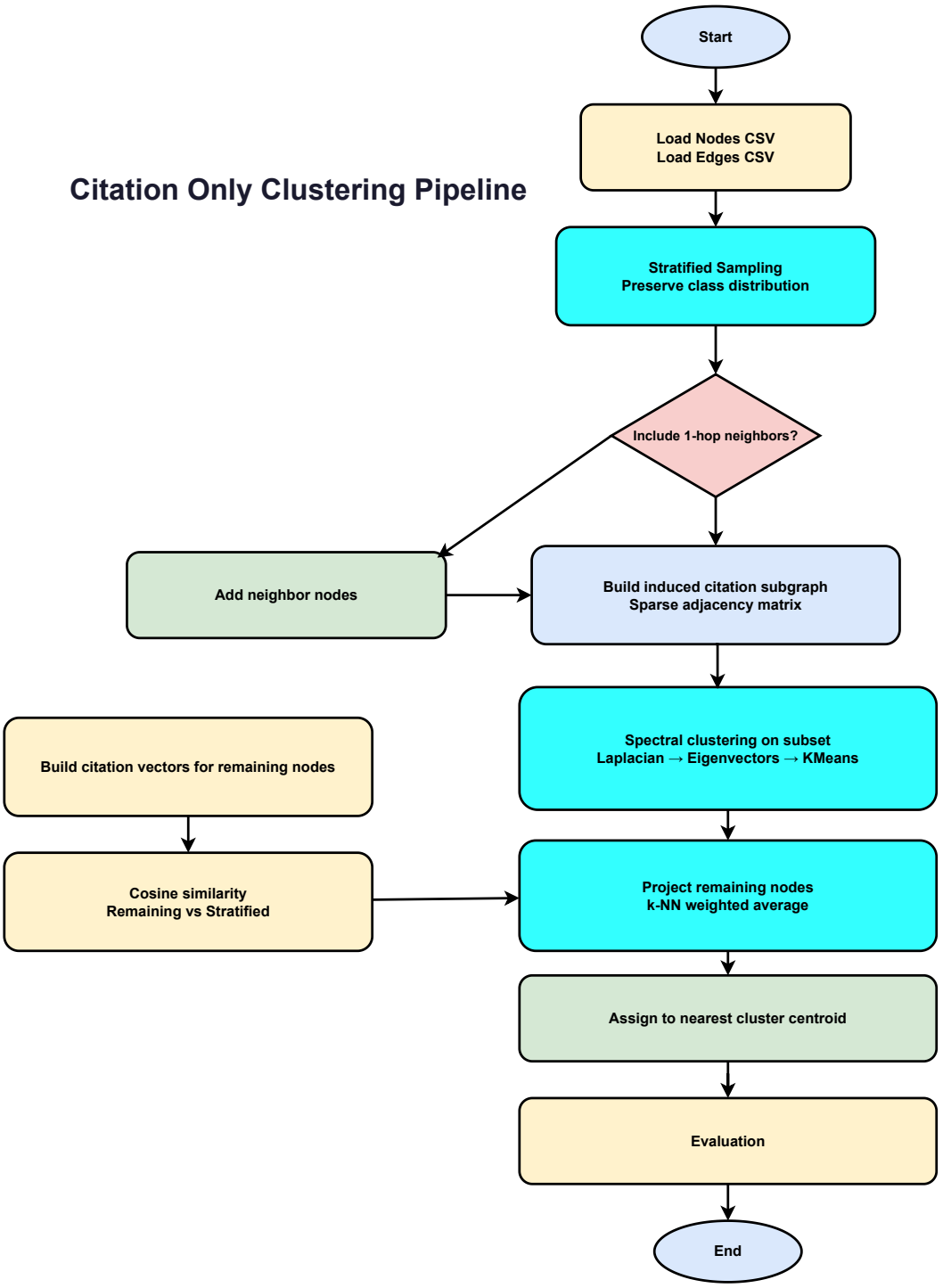


Figure 4.4: Citation Only Clustering Pipeline

4.3.3 Embeddings Only Clustering

In this setup, the embeddings generated for each paper at each node are primarily used for clustering. Figure 4.5 shows the overall flow of the embeddings-only clustering pipeline. As the first step, precomputed embeddings saved in the .npy file are loaded. Each row contains the embedding of the respective paper, along with the *nodeId* and *classId*. Next, a stratified split of all available nodes is taken. This is again, as in the citation-only pipeline, to maintain the dataset's actual class distribution. Afterwards, Cosine similarity is calculated for all stratified papers, and the kNN-similarity graph is obtained. This step is done by using the **NearestNeighbors** method in the **Scikit-Learn** library. After this, spectral clustering is done on the stratified kNN similarity graph using **SpectralClustering** method in the library. Afterwards, the centroids of each cluster are calculated and used later to assign the remaining nodes to their respective clusters.

Now the remaining nodes can be assigned clusters by calculating the kNN similarity between each remaining node and the stratified nodes. Afterwards, a weighted average of those neighbours' spectral embeddings is calculated, and the node is pushed into the spectral space. Finally, the nodes are assigned to their nearest cluster centroid. The evaluation score is calculated for the final clusters.

4.3.4 Multi View Clustering

In the previous sections 4.3.3 and 4.3.2 explained the Clustering using either the use of Citation or Text Embeddings. In this approach, both Citations and Text Embeddings will be used as separate views to cluster the data. The multi-view approach needs feature vectors for each data point as views. Text embeddings view is normalized and used directly. For the citation view, the graph is constructed from stratified samples, the graph Laplacian is calculated, and eigenvectors are obtained. From the eigen vectors, the k smallest vectors are used as the feature vector of the node. Then, multi-view clustering is performed using **MultiviewSpectralClustering** of the **Mvlearn** library. As seen in previous approaches, the resultant cluster centroid is computed.

The remaining nodes are now ready to be clustered. For each remaining node, its embedding similarity to the text centroid, which is the avg of the text embeddings in that cluster, is calculated. Also, the similarity between the spectral embeddings of the remaining nodes, calculated in batches, is used to determine their spectral positions, as in previous approaches. Then, the spectral similarity between the remaining node and the spectral centroid is calculated. By now, we have both the spectral similarity from the citation view and the text similarity from the text view. Finally, we calculate a combined similarity and assign clusters.

4.3.5 K-Means Base Line clustering

To compare the results with a standard baseline method, a **k-Means pipeline** is set up. The dataset 4.1 comes with a standard 128-dimensional feature vector obtained from a skip-gram model Mikolov et al. [2013]. In this setup, both the standard feature vector and LLM embeddings can be used as input. The metrics calculated from this setup will help direct comparison between the proposed embedding-based spectral clustering and a standard K-Means approach.

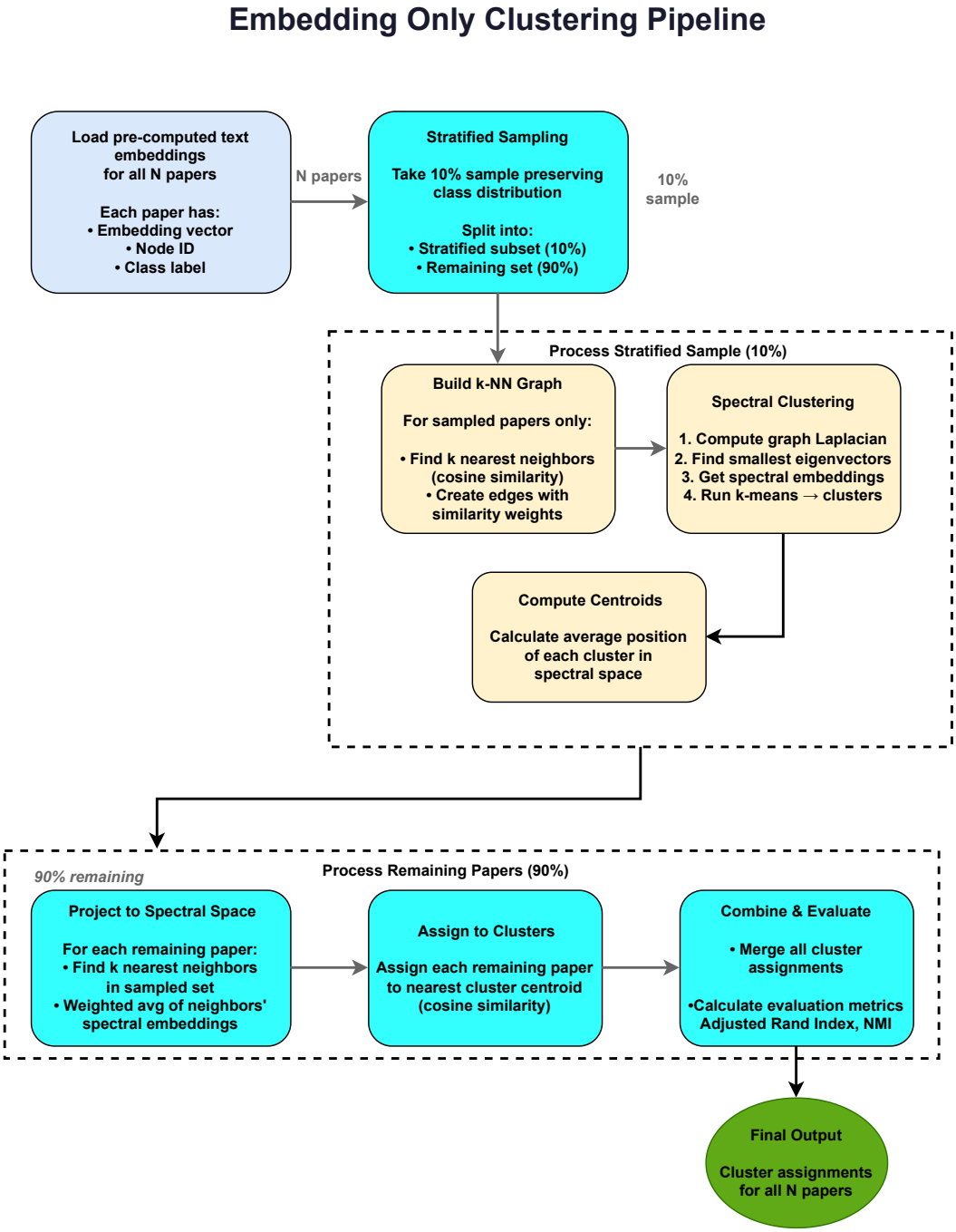


Figure 4.5: Embedding Only Clustering Pipeline

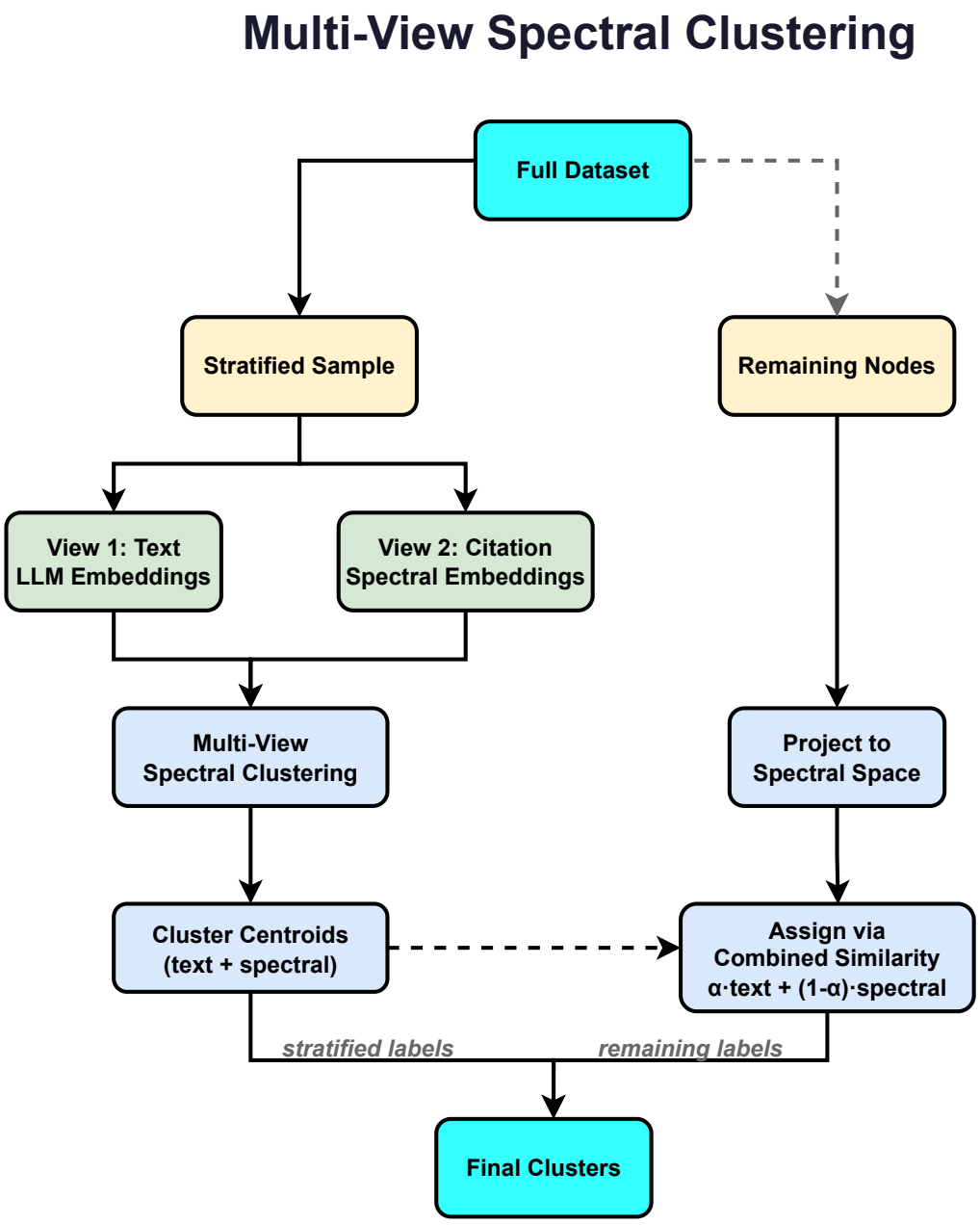


Figure 4.6: Multi view Clustering

5. Evaluation

In this chapter will evaluate the proposed spectral clustering approaches. Section 5.1 shows how the eigen gap analysis looks for different approaches. Section 5.2 discusses the actual results of the experiment with Adjusted Rand Index(ARI) and Normalized Mutual Information(NMI).

5.1 Eigen Gap Analysis

The eigen-gap analysis is the preprocessing step that enables us to calculate eigen-gaps between the eigenvalues of each of our approaches to determine the best number of clusters for each approach. Table 5.1 shows the top number of clusters N recommendations for each method for varying values of K for KNN graph calculation. As we can see, for a low value of k (3), we obtain more clusters, and even with a

Method	**K-Value (KNN)**	**Top Cluster Recommendations (N)**
Embeddings Only		
Qwen3 32B	3	20, 24, 33
	5	1, 5, 12
Llama3.2 3B	3	29, 30, 31
	5	1, 3, 2
Llama3.1 8B	3	33, 32, 34
	5	2, 4, 3
Citation Only		
Citation Edges	–	1, 2, 3
Multi-View		
Llama3.2 3B + Citations	3	2, 1, 5
	5	2, 1, 14
Llama3.1 8B + Citations	3	2, 7, 10
	5	1, 3, 2

Table 5.1: Eigen Gap Analysis Results

gradual increase to $k = 5$, the number of clusters decreases. This can be due to the KNN graph taking the most similar neighbors, leading to more isolated groups and, finally, more clusters. The *Embeddings Only* approach gives the maximum number of clusters, while the other two approaches yield a low number of clusters. The Citation edges, which do not account for the paper's semantic context, and even a few weak citations between two dissimilar topics, will merge into a larger cluster.

5.2 Spectral Clustering Experiment Results

The spectral clustering Experiment results are summarized in Table 5.2. The results shown in the table are split by approach, and the *Adjusted Rand Index* and *Normalized Mutual Information* are shown for each. The *KNN-k* refers to the K value used to calculate the KNN Graph for the initial stratified sample. The *Proj-K* refers to the k value used in KNN for the projection of remaining nodes into the spectral space.

Method	Model	Clusters	ARI	NMI	KNN-K	Proj-K
Embedding-only						
Qwen3 32B AWQ		20	**0.3009** ↑	0.4357	3	10
		40	0.2214	0.4428	3	3
Llama3.2 3B		29	0.2856	0.4869	3	10
		40	0.2390	0.4797	3	3
Llama3.1 8B		33	0.2848	**0.5017** ↑	3	10
		40	0.2376	0.4820	3	3
Citation-only						
Citation Network	–	2	**0.003** ↓	**0.001** ↓	–	–
Multi-view						
Llama3.2 3B + Citations		2	0.0105	0.0277	3	10
Llama3.1 8B + Citations		2	0.0098	0.0305	3	10
Qwen3 32B + Citations		7	0.0336	0.0984	3	10
		40	0.0153	0.0888	3	10
K-means Baseline						
OGB Embeddings (128-dim)		40	0.0687	0.2212	–	–
Qwen3 32B Embeddings		40	0.1397	0.3350	–	–
Llama3.2 3B Embeddings		40	0.1928	0.4654	–	–
Llama3.1 8B Embeddings		40	0.2086	0.4709	–	–

Table 5.2: Spectral Clustering Results: OGBN-Arxiv Dataset

As seen previously in section 5.1, each method's best possible number of clusters, as obtained from the eigen-gap analysis, is used as N, the number of clusters. In addition to our proposed approaches, the K-means baseline is computed for standard embeddings provided with the dataset, as well as for LLM embeddings. The best NMI was observed in the **Llama 3.1 8B** model with **33 clusters**, and the worst performance was observed in the **Citation Only** approach with **2 clusters**. When taking into account both the ARI and the NMI scores The **QWen3 32B AWQ** model performs better with **20 clusters**. The following sections explain each approach individually and compare it with the baseline.

5.2.1 Embeddings Only Clustering

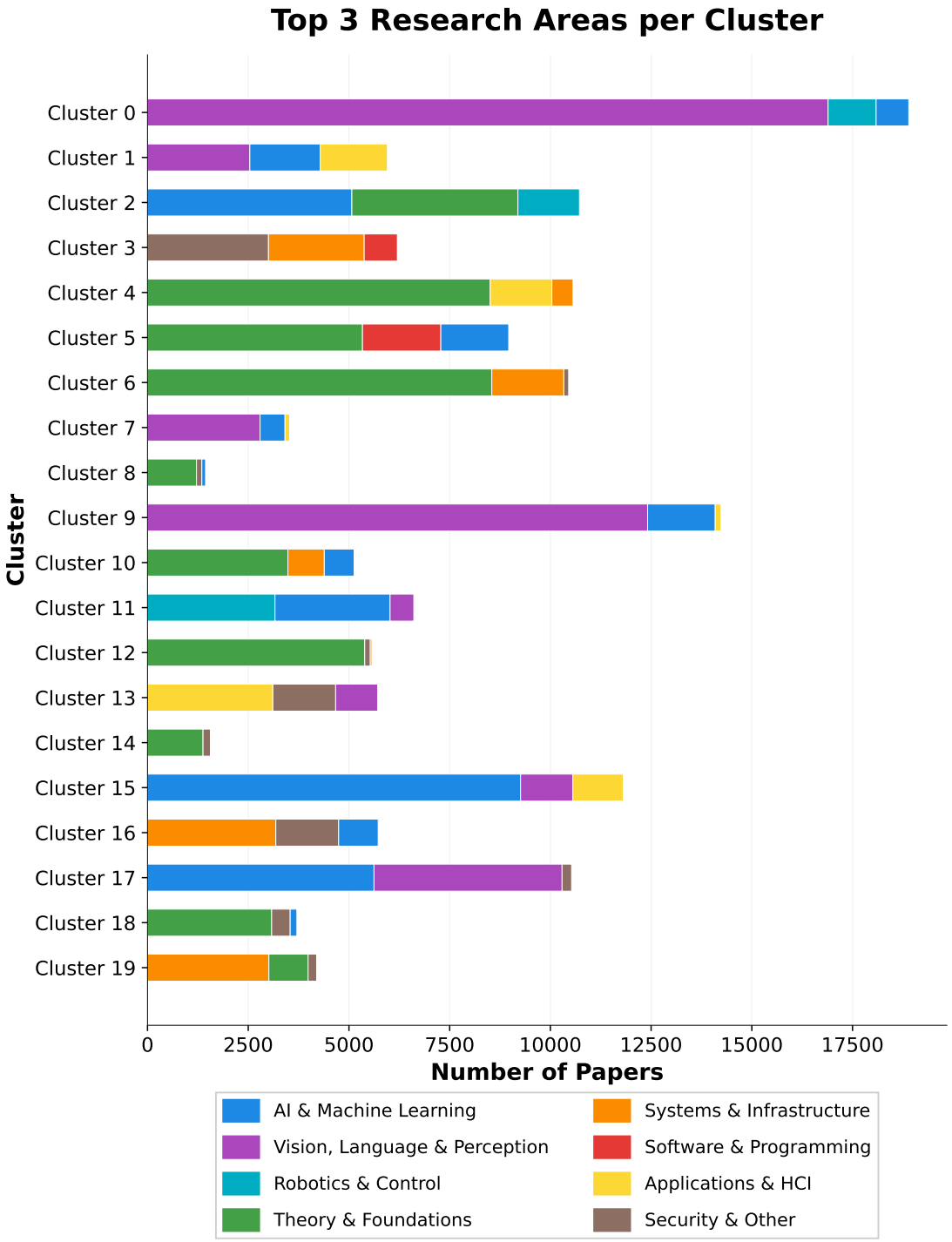


Figure 5.1: Final Clusters for Qwen3 32B AWQ Model Embeddings

To summarize the embeddings-only approach, spectral clustering is applied to the LLM embeddings. In the embeddings-only approach, the results were comparable to those of other proposed approaches. Figure 5.1 shows the final clusters of **Qwen3 32B Awq** embeddings, with the top 3 categories in each cluster marked. Here,

similar arXiv labels are merged for visualization. The **ARI** and **NMI** scores are **0.3009** and **0.4357**, respectively. The best ARI score on 20 clusters, half the number of ground truth clusters, shows that this model's embedding approach yields the most coherent clusters. When comparing the overall embeddings approach with the k-Means baseline clustering, the LLM embeddings perform better than the dataset's standard embeddings. Also, when comparing each spectral clustering with LLM embeddings to their K-Means counterparts, the Spectral clustering approach yields better results. All the k-means were calculated for 40 clusters and compared.

5.2.2 Citations Only Clustering

To summarize, the citations-only approach performs spectral clustering on the citation information alone. The citation-only clustering approach yielded the lowest scores of all approaches. Figure 5.2 shows the final clustering of the top 3 categories in each cluster, where similar arXiv labels are merged for visualization. The ARI and NMI are **0.003** and **0.001** respectively. The obtained score is lower than the K-Means baseline with standard embeddings. This shows that pure citation edges do not capture the semantic context of papers. The cross-domain citations introduce bias into this approach. For example, when two sub-clusters representing different topics have cross-domain citations, such as an AI paper citing a Data Structure and Algorithm paper, the result is larger clusters. This will result in large clusters, as shown in the result chart.

5.2.3 Multi-View Clustering

To summarize, the Multi-View approach performs spectral clustering on the citation information and LLM embeddings. In Multi-view clustering, the evaluation score indicates that the proposed method is better than the citation-only method but performs worse than the K-Means baseline and the Embeddings-only method. The influence of citations reduces the advantage of embeddings. Figure 5.3 shows the final clusters of the Multi-View approach, with the top 3 categories in each cluster of **Qwen3 32B model** with **Citations**. Here, similar arXiv labels are merged for visualization. All but one cluster shows cross-domain papers grouped together. The mentioned approach obtained ARI and NMI scores of **0.0336** and **0.0984**, respectively. When both views are given equal weight in projecting the remaining nodes onto the existing clusters of the stratified subset, the citations view negatively affects the scores. Compared with the baseline K-Means, the baselines of standard embeddings and LLM embeddings achieve higher scores than the Multi-view clustering of the respective models.

5.2.4 K-Means Embeddings Baseline Clustering

The K-means Baseline was evaluated across all selected LLMs' embeddings and the dataset's standard embeddings. Figure 5.4 shows the K-Means evaluation on **Llama3.1 8B model**, which obtained an ARI and NMI scores of **0.2086** and **0.4709**. Here, similar arXiv labels are merged for visualization. In all cases, the K-means baseline with embeddings-only performed better.

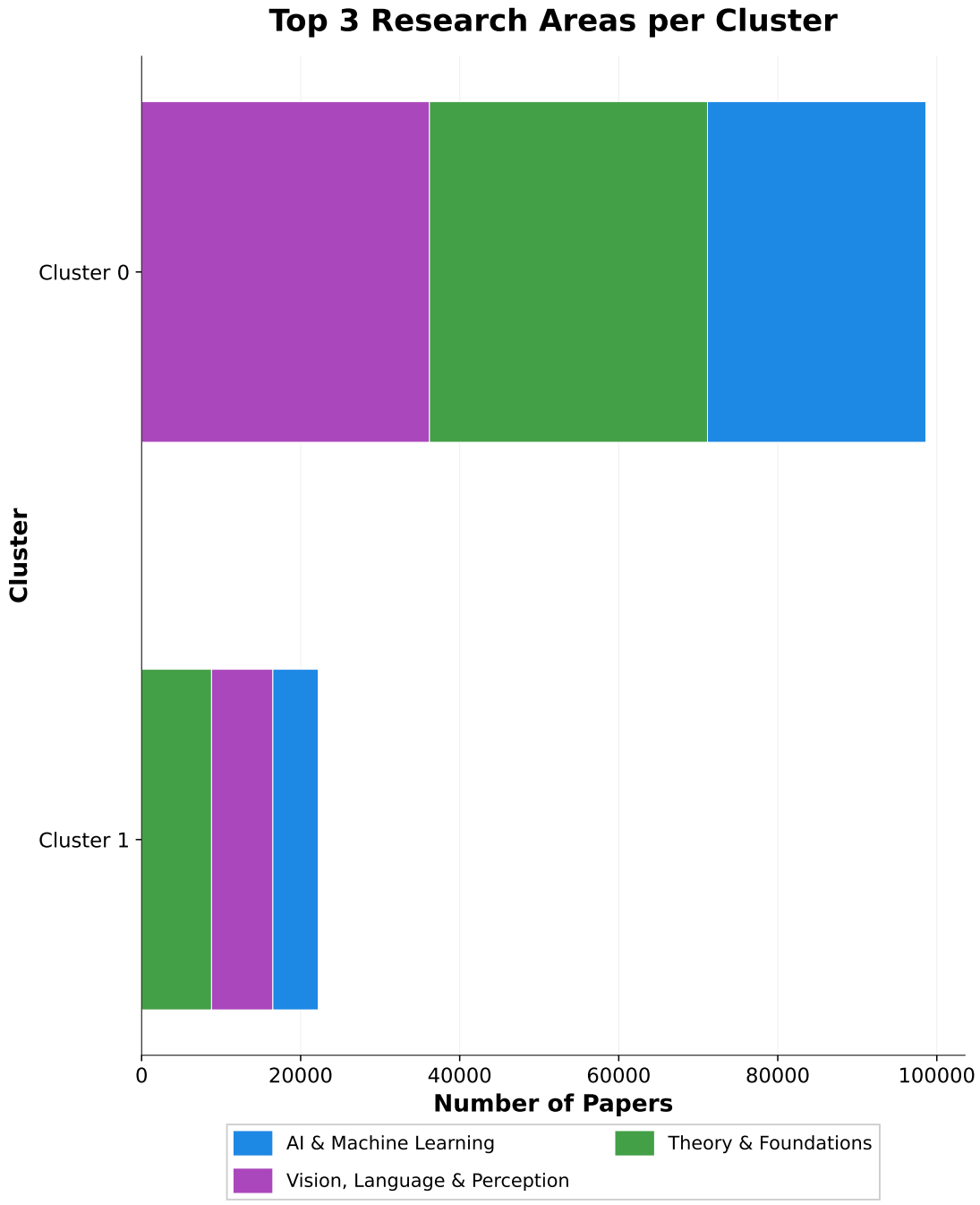


Figure 5.2: Citations Only Clustering

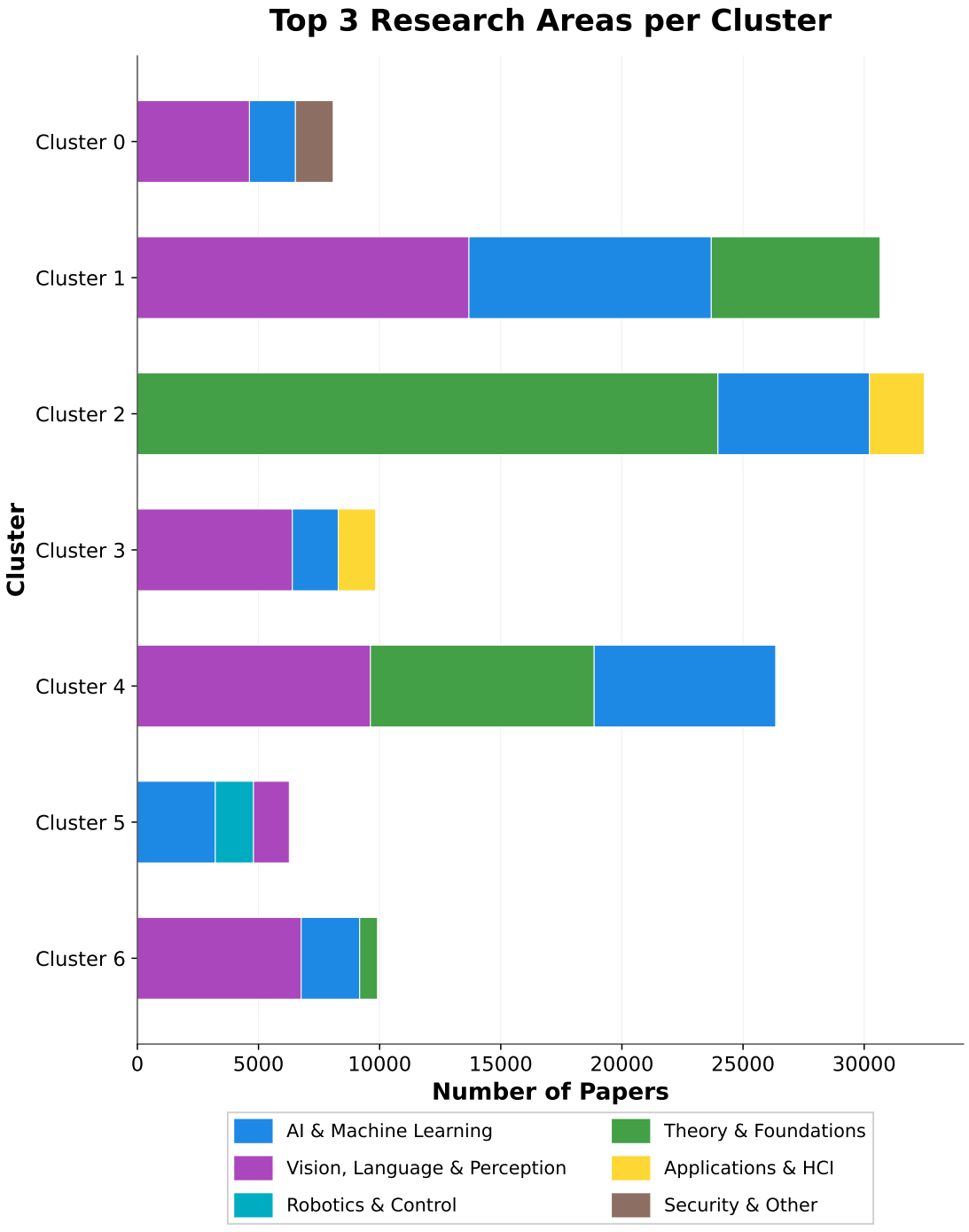


Figure 5.3: Multi View Spectral Clustering : Qwen3 32B and Citations

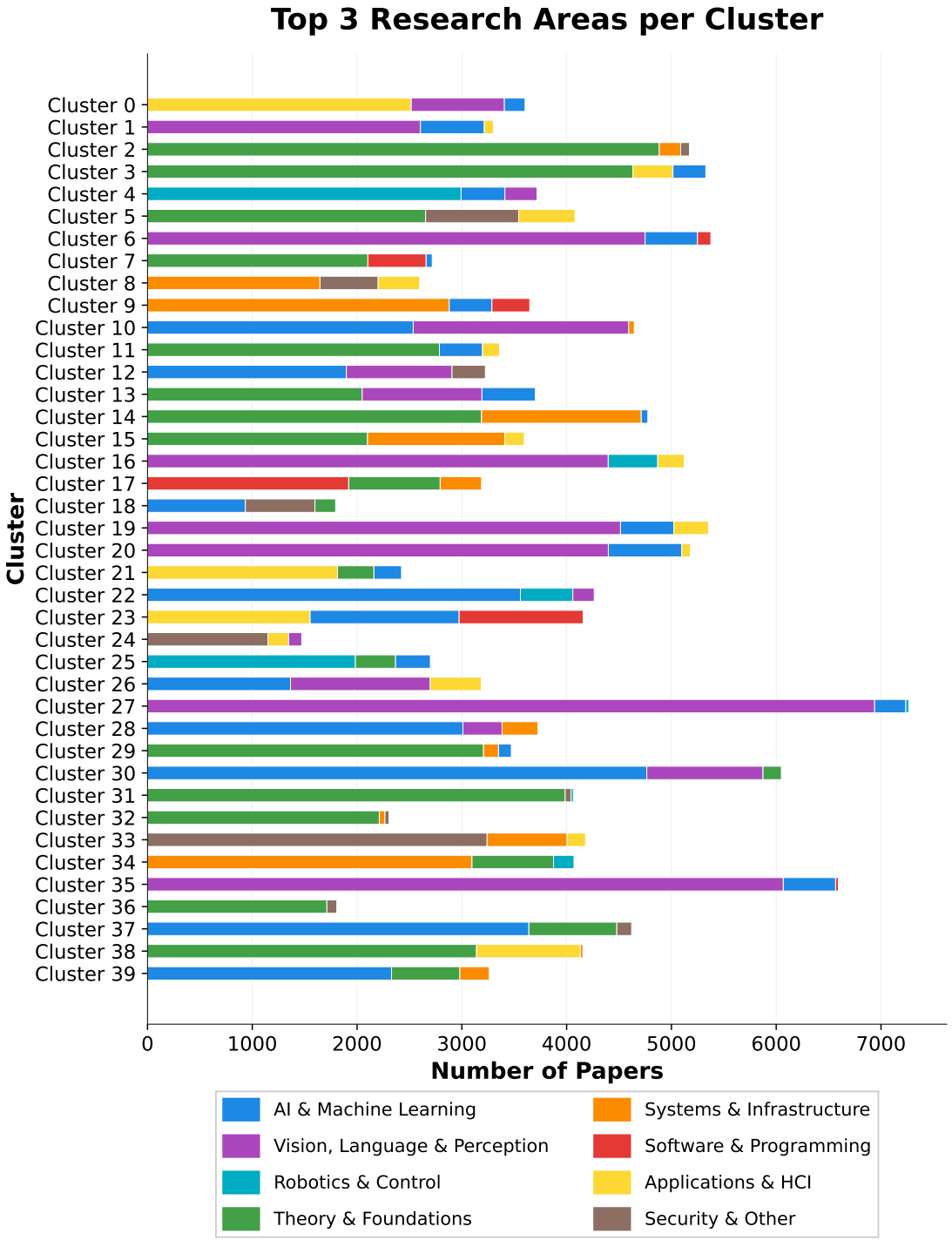


Figure 5.4: K-Means Clustering on Llama3.1 8B Model

6. Conclusion

Overall, the thesis is a multidimensional study of spectral clustering in Citation networks: arXiv OGBN. The thesis demonstrates how different representations, such as embeddings, citations, and multi-view representations, impact the quality and interpretation of cluster assignments by comparing baseline K-means clustering with LLM-based spectral clustering, Citation-network spectral clustering, and multi-view spectral clustering.

The overall performance indicates that the embeddings-only approach with mini-batch spectral clustering yields the best scores. When we compare pure citations approach with embeddings only approach the later has better scores and clusters. Based on the evaluation, the research questions 1.2 can be answered as follows.

- **RQ1:** The **LLM-based text embedding similarity graph** construction gives **higher ARI and NMI scores** compared to citation graphs from citation network edges. This can be interpreted as the semantic context captured by LLMs better encodes the relationship between papers.
- **RQ2:** The **Llama3.1 8B** model performed better based on **NMI** and the **Qwen3 32B AWQ** performed better on **ARI**.
- **RQ3:** The Multi-View approach of combining two views of text embeddings, graph, and citation graphs does not improve the ARI and NMI metrics. This can be interpreted as follows: when combining the two views, the citation graphs negatively impact clustering by reducing the advantage of the embedding graph.

6.1 Future Work

The thesis's future perspective offers multiple directions for enhancing the usefulness of the spectral clustering method for academic data, both methodologically and practically. Even though the proposed methods were successfully designed to integrate semantic and topical relationships, there is much that can be done to make them more advantageous by introducing additional graph modalities, improved embedding

structures, and scalable spectral solvers capable of handling large numbers of academic networks. The quality and interpretability of clusters might be improved through learning-based fusion strategies and domain-adaptive embeddings. It would also enhance the practicality of the work within academic knowledge systems that could be extended to consider real-world applications, such as literature recommendation, topic evolution analysis, or research trend forecasting.

- Multi-Graph Integration: Append the data of the co-authorship, keyword networks, topic hierarchies, and temporal citation graphs into the hybrid spectral model.
- More sophisticated Embeddings: Train domain-centric LLM, graph-based Transformer to encode more semantics.
- Scalability: Approximate spectral algorithms can operate on large datasets such as **OGBN-Papers100M** with the help of graph sparsification and distributed computing.
- Adaptive Fusion Models: explore attention-dependent or learnable weighting functions to combine semantic and citation-based similarity automatically.
- Temporal Modeling: Add publication dynamics so as to capture drift of a topic, bursts of citation, and evolution of a scientific field.
- Semi-Supervised Extension: Clustering with partial label information is used to give superior predictions of categories and filter topics.
- Application to Downstream tasks: use the results of clustering to suggest systems, identify anomalies, map topics, and automatically summarize research.

Appendix

Table A.1: Software Requirements for Spectral Clustering Pipeline

Category	Library	Version	Purpose
Runtime	python	3.12.10	Python interpreter
Embedding Generation	transformers	4.51.3	LLM-based text embeddings (LLaMA, Qwen, DeepSeek)
	huggingface_hub	0.30.2	Model downloading and management
	tokenizers	0.21.1	Fast tokenization for transformers
	accelerate	1.8.1	Efficient model loading and inference
	pytorch	2.5.1	Deep learning backend
	safetensors	0.5.3	Safe model weight loading
Graph & Clustering	scikit-learn	1.6.1	K-means, spectral clustering, metrics (ARI, NMI, silhouette)
	mvlearn	0.5.0	Multi-view spectral clustering
	networkx	3.4.2	Graph construction and manipulation
	pytorch_geometric	2.6.1	Graph neural networks and graph utilities
	scipy	1.11.4	Sparse matrices, eigenvalue computation
OGB Dataset	ogb	1.3.6	Open Graph Benchmark dataset loading
	torch (pytorch)	2.5.1	Required backend for OGB
Data Processing	numpy	1.26.4	Array operations, embeddings storage
	pandas	2.2.3	Data manipulation, CSV handling
	pyarrow	20.0.0	Efficient data serialization
Visualization	matplotlib	3.10.3	Plotting (bar charts, t-SNE plots)
	seaborn	0.13.2	Statistical visualizations
Utilities	tqdm	4.67.1	Progress bars
	pyyaml	6.0.2	Config file handling
	python-dotenv	1.1.0	Environment variables

Result clusters without merging them into groups:

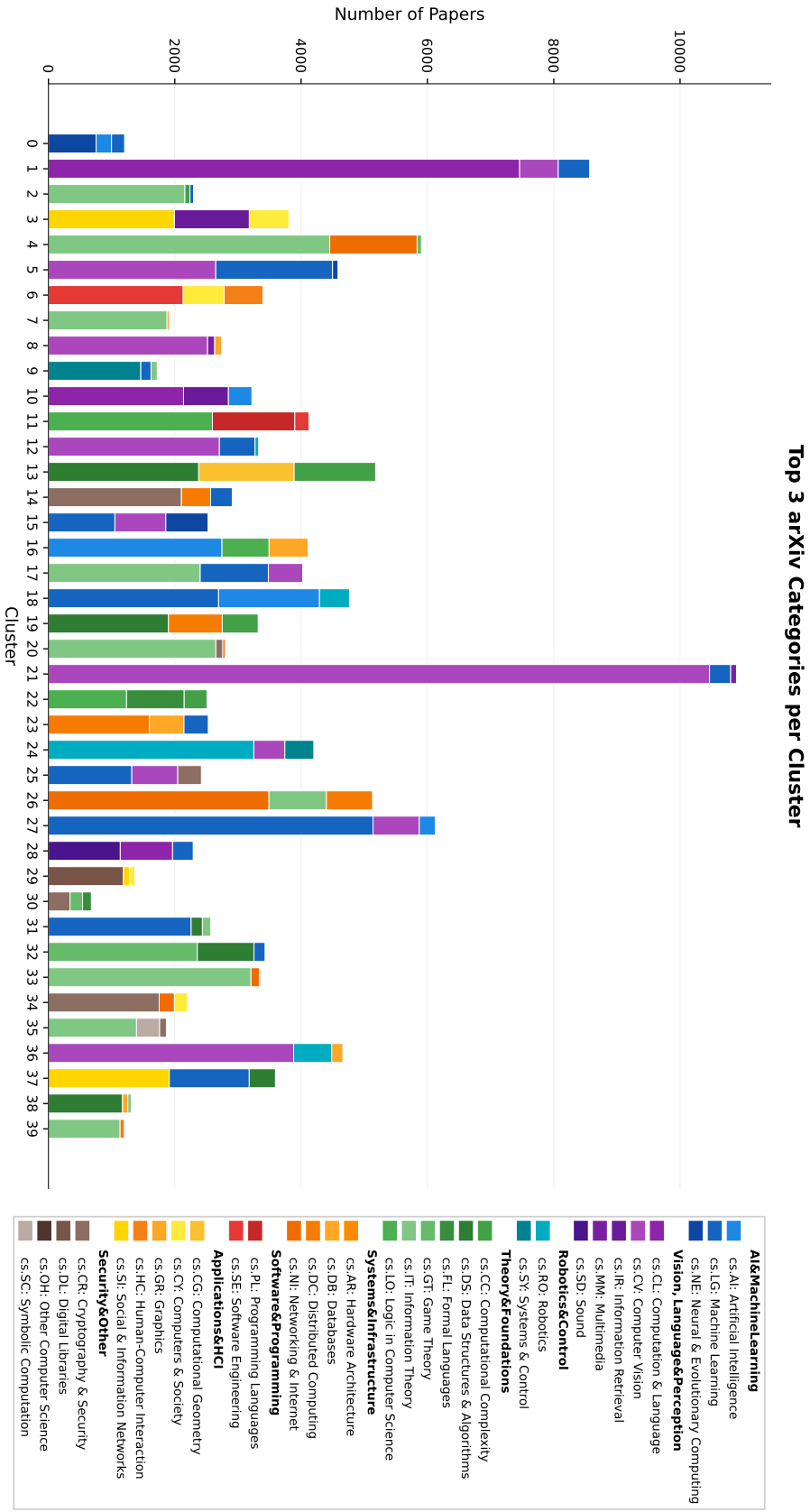


Figure A.1: Embeddings only clustering of Llama 3.1 8B model

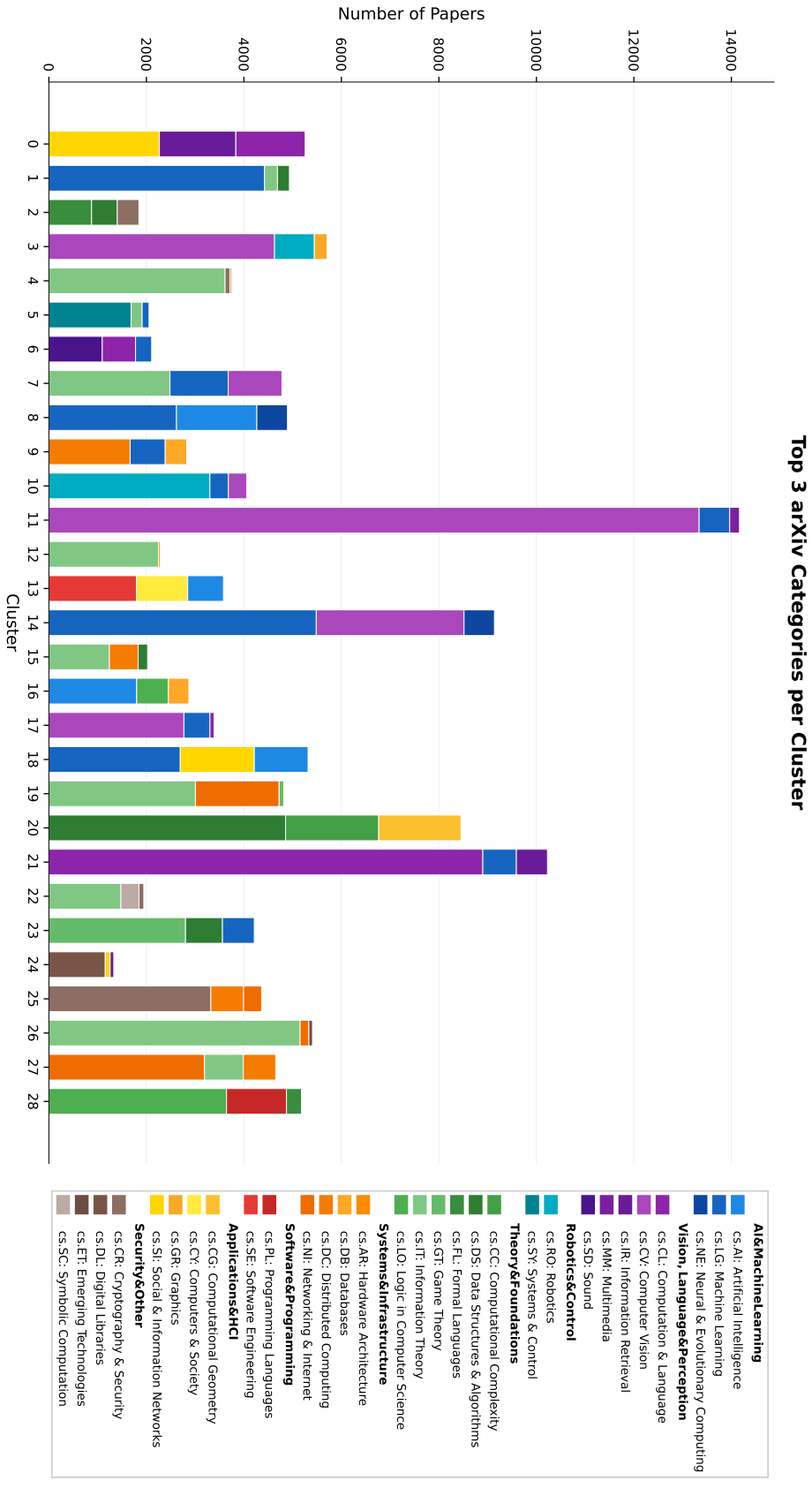


Figure A.2: Embeddings only clustering of Llama 3.2 3B model

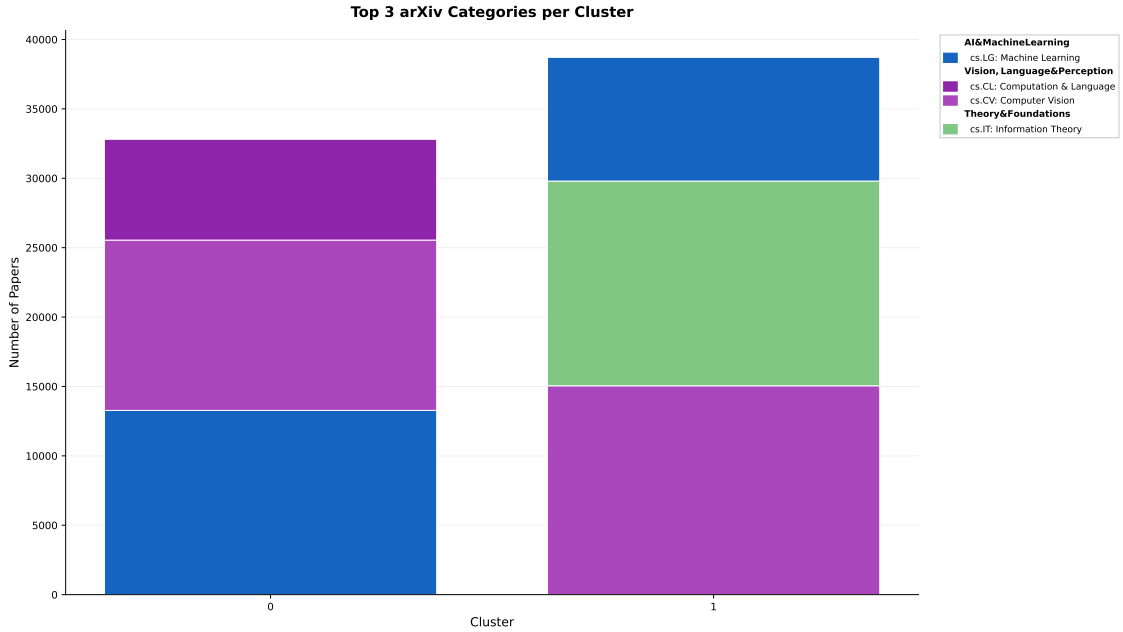


Figure A.3: Multi-View clustering with Llama 3.2 3B model and Citations

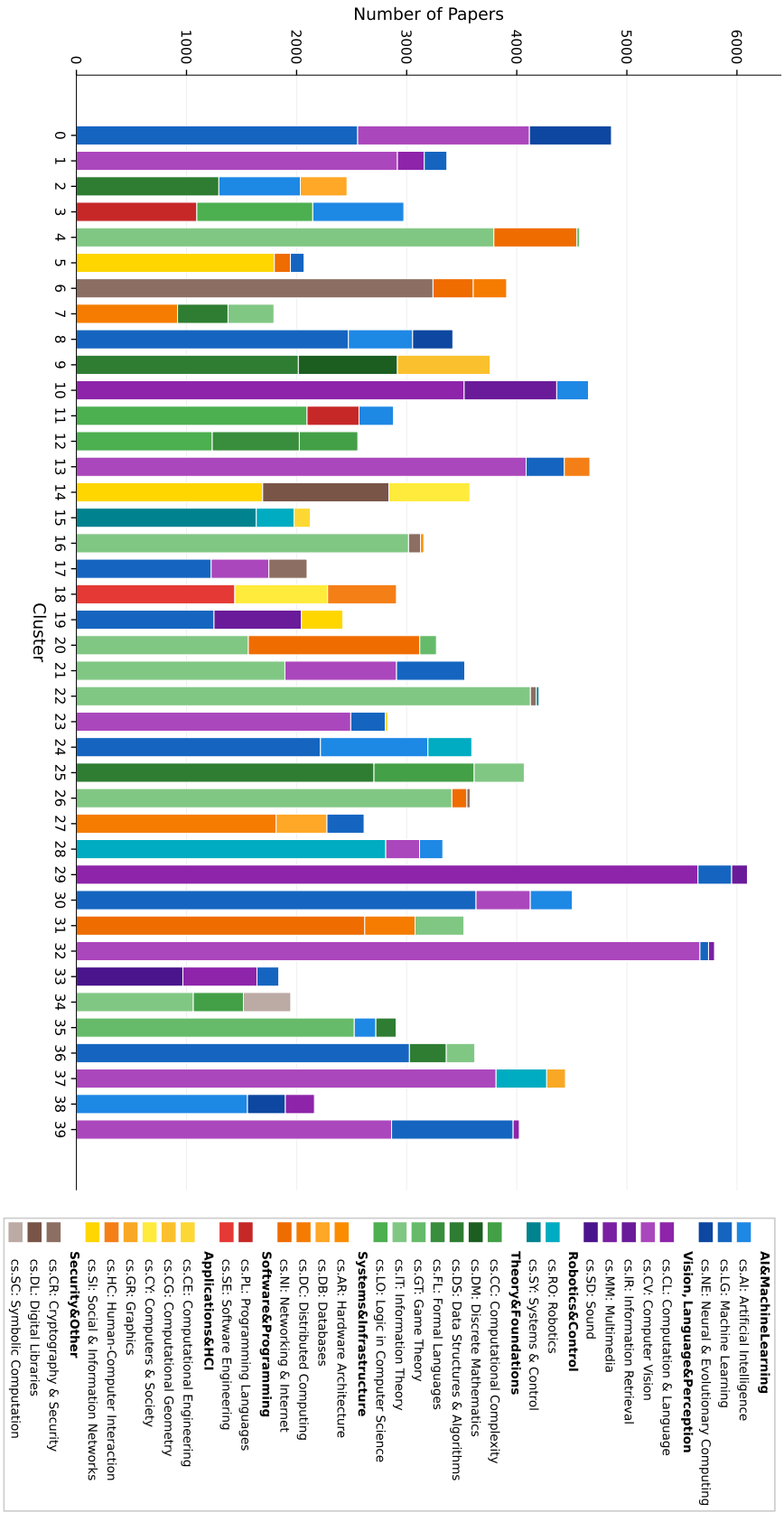


Figure A.4: K-Means baseline clustering with llama3.2 3B model

AI Declaration

I hereby declare that AI assistance (Claude by Anthropic) was used during the completion of this thesis. Specifically, Claude was employed for:

- Code generation for boilerplate code, utility functions, and class structures
- Code refactoring
- Debugging and troubleshooting of implementation errors

All research contributions, including the formulation of research questions, experimental design, methodology, data analysis, interpretation of results, and scientific conclusions, are entirely my own work.

Bibliography

Survey of spectral clustering based on graph theory. *Pattern Recognition*, 151: 110366, 2024. ISSN 00313203. doi: 10.1016/j.patcog.2024.110366. URL https://doi.org/10.1016/j.patcog.2024.110366. (cited on Page 1)

Charu C Aggarwal and Haixun Wang. A survey of clustering algorithms for graph data. In *Managing and mining graph data*, pages 275–301. Springer, 2010. (cited on Page ix and 12)

Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8): 1295, 2020. (cited on Page 12)

Vaswani Ashish. Attention is all you need. *Advances in neural information processing systems*, 30:I, 2017. (cited on Page 14 and 15)

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. (cited on Page 14 and 15)

Xiuyuan Cheng and Gal Mishne. Spectral embedding norm: Looking deep into the spectrum of the graph laplacian. *SIAM journal on imaging sciences*, 13(2): 1015–1048, 2020. (cited on Page 10)

Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. On evolutionary spectral clustering. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 3(4):1–30, 2009. (cited on Page 17)

Pietro della Briotta Parolo, Rainer Kujala, Kimmo Kaski, and Mikko Kivelä. Tracking the cumulative knowledge spreading in a comprehensive citation network. *Physical Review Research*, 2(1):013181, 2020. (cited on Page 11)

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. (cited on Page 15)

Reinhard Diestel. *Graph theory*. Springer Nature, 2025. (cited on Page 5)

Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. Learning laplacian matrix in smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23):6160–6173, 2016. (cited on Page 18)

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024. (cited on Page 15)

Absalom E Ezugwu, Amit K Shukla, Moyinoluwa B Agbaje, Olaide N Oyelade, Adán José-García, and Jeffery O Agushaka. Automatic clustering algorithms: a systematic review and bibliometric analysis of relevant literature. *Neural Computing and Applications*, 33(11):6247–6306, 2021. (cited on Page ix and 12)

Hui Han, Hongyuan Zha, and C Lee Giles. Name disambiguation in author citations using a k-way spectral clustering method. In *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pages 334–343, 2005. (cited on Page 1)

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. (cited on Page 15)

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020. (cited on Page 2)

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs, 2021. URL https://arxiv.org/abs/2005.00687. (cited on Page 21)

Pierre Humbert, Batiste Le Bars, Laurent Oudre, Argyris Kalogeratos, and Nicolas Vayatis. Learning laplacian matrix from graph signals with sparse spectral representation. *Journal of Machine Learning Research*, 22(195):1–47, 2021. (cited on Page 10)

Vu Thi Huong and Thorsten Koch. Clustering scientific publications: lessons learned through experiments with a real citation network. *arXiv preprint arXiv:2505.18180*, 2025. (cited on Page 11)

Timofey V Ivanisenko, Pavel S Demenkov, and Vladimir A Ivanisenko. An accurate and efficient approach to knowledge extraction from scientific publications using structured ontology models, graph neural networks, and large language models. *International Journal of Molecular Sciences*, 25(21):11811, 2024. (cited on Page 11)

Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999. (cited on Page 12)

Thirupathi Kandadi and G Shankarlingam. Drawbacks of lstm algorithm: A case study. *Available at SSRN 5080605*, 2025. (cited on Page 15)

Aparajita Khan and Pradipta Maji. Approximate graph laplacians for multimodal data clustering. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):798–813, 2019. (cited on Page 10)

Shuyang Ling and Thomas Strohmer. Certifying global optimality of graph cuts via semidefinite relaxation: A performance guarantee for spectral clustering. *Foundations of Computational Mathematics*, 20(3):367–421, 2020. (cited on Page 14)

Jialu Liu and Jiawei Han. Spectral clustering. In *Data clustering*, pages 177–200. Chapman and Hall/CRC, 2018. (cited on Page 10)

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. The flan collection: Designing data and methods for effective instruction tuning. In *International Conference on Machine Learning*, pages 22631–22648. PMLR, 2023. (cited on Page 15)

U. V. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17: 395–416, 2007. doi: 10.1007/s11222-007-9033-z. URL https://www.semanticscholar.org/paper/eda90bd43f4256986688e525b45b833a3addab97. (cited on Page 1, 2, 13, and 17)

Abdul Majeed and Ibtisam Rauf. Graph theory: A comprehensive survey about graph theory applications in computer science and social networks. *Inventions*, 5 (1):10, 2020. (cited on Page 5)

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013. (cited on Page 27)

Rahul Mondal. Learning from the eigenvalues of metaproteomic abundance data-spectral clustering and classification. Master's thesis, OVGU, 2023. (cited on Page ix, 9, and 12)

Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011. (cited on Page 12)

Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003. (cited on Page 1)

Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001. (cited on Page 10)

Kai Nishikawa and Hitoshi Koshiba. Exploring the applicability of large language models to citation context analysis. *Scientometrics*, 129(11):6751–6777, 2024. (cited on Page 10 and 11)

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. (cited on Page 15)

Rajvardhan Patil, Sorio Boit, Venkat Gudivada, and Jagadeesh Nandigam. A survey of text representation and embedding techniques in nlp. *IEEE Access*, 11:36120–36146, 2023. ISSN 21693536. doi: 10.1109/ACCESS.2023.3266377. (cited on Page 1)

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. (cited on Page 2)

Ronan Perry, Gavin Mischler, Richard Guo, Theodore Lee, Alexander Chang, Arman Koul, Cameron Franz, Hugo Richard, Iain Carmichael, Pierre Ablin, Alexandre Gramfort, and Joshua T. Vogelstein. mvlearn: Multiview machine learning in python. *Journal of Machine Learning Research*, 22(109):1–7, 2021. (cited on Page 2)

Shahneela Pitafi, Toni Anwar, and Zubair Sharif. A taxonomy of machine learning clustering algorithms, challenges, and future realms. *Applied sciences*, 13(6):3529, 2023. (cited on Page 12)

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. (cited on Page 14)

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. (cited on Page 14)

Filippo Radicchi, Santo Fortunato, Benjamin Markines, and Alessandro Vespignani. Diffusion of scientific credits and the ranking of scientists. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 80(5):056103, 2009. (cited on Page 11)

Filippo Radicchi, Santo Fortunato, and Alessandro Vespignani. Citation networks. *Models of science dynamics: Encounters between complexity theory and information sciences*, pages 233–257, 2011. (cited on Page 1, 10, and 11)

Krishna Raj PM, Ankith Mohan, and KG Srinivasa. Basics of graph theory. In *Practical Social Network Analysis with Python*, pages 1–23. Springer, 2018. (cited on Page 1 and 6)

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*, 2021. (cited on Page 15)

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. (cited on Page 15)

Changhao Shi and Gal Mishne. Graph laplacian learning with exponential family noise. *IEEE Transactions on Signal and Information Processing over Networks*, 2025. (cited on Page 20)

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000. (cited on Page 10)

Olaf Sporns. Graph theory methods: applications in brain networks. *Dialogues in clinical neuroscience*, 20(2):111–121, 2018. (cited on Page 5)

Ogb Stanford. Ogbn node property prediction datasets, 2025. URL https://ogb.stanford.edu/docs/nodeprop/. (cited on Page ix and 22)

Dragan Stevanović. Spectral radius of graphs. In *Combinatorial Matrix Theory*, pages 83–130. Springer, 2018. (cited on Page 8)

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020. (cited on Page 15)

Feiyi Tang, Junxian Li, Xi Liu, Chao Chang, and Luyao Teng. Gatfelpa integrates graph attention networks and enhanced label propagation for robust community detection. *Scientific Reports*, 15(1):3952, 2025. (cited on Page 20)

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. (cited on Page 1)

Nicolas Tremblay and Andreas Loukas. Approximating spectral clustering via sampling: a review. *Sampling techniques for supervised or unsupervised tasks*, pages 129–183, 2019. (cited on Page 14)

Maia Trower, Natasa Djurdjevac Conrad, and Stefan Klus. Clustering time-evolving networks using the spatiotemporal graph laplacian. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 35(1), 2025. (cited on Page 19)

Paul E Van Der Vet and Harm Nijveen. Propagation of errors in citation networks: a study involving the entire citation network of a widely cited paper published in, and later retracted from, the journal nature. *Research integrity and peer review*, 1 (1):3, 2016. (cited on Page 11)

Pablo Villegas, Andrea Gabrielli, Anna Poggialini, and Tommaso Gili. Multi-scale laplacian community detection in heterogeneous networks. *Physical Review Research*, 7(1):013065, 2025. (cited on Page 18)

Ludo Waltman and Nees Jan Van Eck. A new methodology for constructing a publication-level classification system of science. *Journal of the American Society for Information Science and Technology*, 63(12):2378–2392, 2012. (cited on Page 11)

Rui Wang, Duc Duy Nguyen, and Guo-Wei Wei. Persistent spectral graph. *International journal for numerical methods in biomedical engineering*, 36(9):e3376, 2020. (cited on Page 8)

Tzy-Shiah Wang, Hui-Tang Lin, and Ping Wang. Weighted-spectral clustering algorithm for detecting community structures in complex networks. *Artificial Intelligence Review*, 47(4):463–483, 2017. (cited on Page 18)

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 13484–13508, 2023. (cited on Page 15)

Jinming Xing, Dongwen Luo, Chang Xue, and Ruilin Xing. Comparative analysis of pooling mechanisms in llms: A sentiment analysis perspective. *arXiv preprint arXiv:2411.14654*, 2024. (cited on Page 15)

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a. (cited on Page 1)

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025b. (cited on Page 15)

Haizhou Yang, Quanxue Gao, Wei Xia, Ming Yang, and Xinbo Gao. Multiview spectral clustering with bipartite graph. *IEEE Transactions on Image Processing*, 31:3591–3605, 2022. (cited on Page 13)

Jinting Zhu, Julian Jang-Jaccard, Tong Liu, and Jukai Zhou. Joint spectral clustering based on optimal graph and feature selection. *Neural Processing Letters*, 53(1): 257–273, 2021. (cited on Page 14)

Arkaitz Zubiaga. Natural language processing in the era of large language models, 2024. (cited on Page 14)

I herewith assure that I have written the present thesis independently, that the thesis has not been partially or fully submitted as graded academic work and that I have used no other means than the ones indicated. I have indicated all parts of the work in which sources are used according to their wording or according to their meaning.

I am aware of the fact that violations of copyright can lead to injunctive relief and claims for damages of the author as well as a penalty by the law enforcement agency.

Furthermore, I confirm that I am aware that the use of content (including but not limited to text, figures, images, and code) generated by artificial intelligence (AI) in the thesis must be disclosed. In those cases, I have specified the AI system used, I have marked the specific sections of the thesis where AI-generated content was used, and I have provided a brief explanation of the level of detail at which the AI system was used to generate the content. I also stated the reason for using the tools.

I assure that even if a generative AI system has been used, the scientific contribution has been made entirely by myself.

Magdeburg, 11th February 2026