University of Texas at Arlington

MavMatrix

Computer Science and Engineering Theses | Computer Science and Engineering Department

Summer 2025

Centrality Algorithms For Weighted Homogeneous Multilayer Networks

Ayomide Ayowole-Obi

Follow this and additional works at: https://mavmatrix.uta.edu/cse_theses

 Part of the Computer Sciences Commons

Recommended Citation

Ayowole-Obi, Ayomide, "Centrality Algorithms For Weighted Homogeneous Multilayer Networks" (2025). *Computer Science and Engineering Theses*. 531.
https://mavmatrix.uta.edu/cse_theses/531

This Thesis is brought to you for free and open access by the Computer Science and Engineering Department at MavMatrix. It has been accepted for inclusion in Computer Science and Engineering Theses by an authorized administrator of MavMatrix. For more information, please contact vanessa.garrett@uta.edu.

CENTRALITY ALGORITHMS FOR WEIGHTED HOMOGENEOUS MULTILAYER NETWORKS

by

AYOMIDE AYOWOLE-OBI

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2025

Copyright © by Ayomide Ayowole-Obi 2025

All Rights Reserved

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Sharma Chakravarthy, for his unwavering guidance, encouragement, and mentorship throughout this research. His insights and constructive feedback played a vital role in shaping the direction and depth of this work.

I am also sincerely thankful to Dr. Abhishek Santra, whose active involvement and thoughtful input greatly enriched this thesis. His support throughout the year, both technical and academic, was truly invaluable to the development of this work.

I would also like to thank Dr. David Levine for for serving on my committee and for providing helpful comments and feedback during the course of this work. His time and contributions are sincerely appreciated.

I would also like to thank the members of the IT Lab for their encouragement and support during the course of this research. I am very thankful to UTA for providing this opportunity and environment to pursue this research and present my work.

Finally, I am deeply grateful to my family and friends for their constant support, patience, and encouragement throughout this journey. Their belief in me has been a source of strength and motivation.

August 04, 2025

ABSTRACT

CENTRALITY ALGORITHMS FOR WEIGHTED HOMOGENEOUS MULTILAYER NETWORKS

Ayomide Ayowole-Obi, M.S.

The University of Texas at Arlington, 2025

Supervising Professor: Prof. Sharma Chakravarthy

Applications need to be modeled for analyses. With the availability of many alternate data Models, choosing one needs to be done carefully by considering the complexity of data to be modeled and its analyses requirements. Social networks and other newer applications are primarily relationship-oriented and also have multiple types of entities and relationships. Although simple and attributed graphs have been used historically for modeling these applications, recently, multilayer networks (or MLNs) have been shown to be more effective in preserving the semantics of the application better and further provide flexibility of analyses. However, choice of MLN as a data model poses additional challenges for analyses as extant simple and attributed graph algorithms cannot be used directly. A wide range of computations can be performed on graphs, including traversals, searches, path-finding, to name a few. Our interest in this thesis is in aggregate analyses on MLNs, such as community, centrality, substructure etc. For these, new algorithms are being developed for MLNs to leverage modeling advantages.

Briefly, MLNs consist of multiple graphs or layers (also called networks), each capturing a different type of relationship or interaction. MLNs can be categorized into: Homogeneous MLNs (HoMLNs), where each layer represents a relationship in the form of intra-layer edges and all layers have the same type of entity; Heterogeneous MLNs (HeMLNs), where layers represents a different relationship based on entity type in that layer, and inter-layer edges representing relationships between entity types from two layers; and Hybrid MLNs (or HyMLNs), which combine both homogeneous and heterogeneous characteristics. While many algorithms exist for analyzing simple graphs, weighted or unweighted, to the best of our knowledge, these algorithms are not available for MLNs and are being developed only recently.

Algorithms for MLNs can be developed in multiple ways. Recently, a decoupling framework has been proposed and its advantages have been demonstrated. This framework separates the analyses into two stages: independent analysis of each layer and composing results from each layer for MLN analysis. Each layer of the HoMLN is analyzed independently using any layer-specific analysis function, without using the knowledge of other layers. The partial results from two layers are then composed to derive the overall results for the HoMLN. This approach is not only efficient, thanks to the potential for parallelism during the analysis stage, but has also been shown to be efficient and scalable.

For validating MLN algorithms, typically, layers are aggregated using Boolean AND or OR operations (suited for unweighted graphs) and existing algorithms are used. This has been shown to lead to ambiguity in edge interpretation, not preserving layer characteristics, and information loss in general. The goal of MLN algorithms is to preserve accuracy avoiding information loss. Weighted graphs pose additional issues for aggregation as, in addition to Boolean operations, weights can be combined in multiple ways, such as sum, max, min, or average. Challenges for developing

decoupling-based MLN algorithms for weighted MLNs are: i) choice of aggregation method, ii) composition algorithm, iii) tradeoff on information carried over from layer analysis and used for composition, and iv) intuition-based heuristics for maximizing accuracy.

This thesis focuses on generalizing decoupling-based algorithms for weighted MLNs, specifically HoMLNs and for degree centrality metric. Although algorithms have been developed for unweighted MLNs, to the best of our knowledge, these have not been done for weighted MLNs. As the decoupling-based algorithms use independently generated analysis information from layers, accuracy is a primary challenge. To address this, the thesis proposes a set of heuristics for degree centrality that aim to maximize accuracy and efficiency while carrying over minimal metric information from layers. These heuristics are compared against both ground truth and naive baseline to evaluate their effectiveness. Proposed heuristics demonstrate comparable accuracy to the ground truth while significantly reducing computation time and outperforming the naive approach in both performance and precision.

To validate the proposed algorithms, synthetically generated HoMLN data sets of different sizes, distribution, overlap, and weight characteristics are used. In addition, real-world datasets have been utilized to establish that the heuristics developed work on real-world data sets equally well. This ensures that the heuristics are tested across graphs with diverse structures and characteristics. Jaccard similarity for accuracy and execution time are used for performance. Scalability of the algorithms is established using numerous increasingly large data sets.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Data with relationships between entities can be represented using different models, one of which is a graph. Graphs consist of nodes (vertices) representing entities and edges (links) to indicate how they relate to each other. Weights can also be associated with edges and even with nodes. Graphs are useful for representing information in ways that other models cannot, at least not without additional computation. This representation is also very easy to visualize and analyze. However, many real-world applications involve multiple types of relationships between the same entities. Modeling such complexity with simple graphs can lead to information loss or oversimplification. Multilayer Networks (MLNs) preserve these semantics by representing each relationship type in a separate layer, enabling richer and more flexible analysis [2]. Centrality analysis provides valuable information on each node with regard to its place within the graph network. Using centrality metrics within a graph, especially multilayer networks, becomes more relevant as information continues to grow.

Community detection, cluster analysis, centrality metrics, etc. are some of the different metrics that are typically used for graph analysis. Characteristics and attributes of a graph differ from graph to graph, and different vertices might have different concentrations and distributions of edges between them. Some vertices might be connected to a large number of nodes, while others may have fewer connections to other nodes. These connections or edges could have additional value termed edge weights, which are costs attributed to that connection. They can be used to measure how important or expensive a particular edge is. For example, connections indicating

the number of friends can be used to determine how influential a person is within that social group.

Centrality metrics, such as degree, closeness, and betweenness, provide a way to measure how important or influential a vertex is within the network. Different algorithms have been proposed to calculate centrality measures. One of those methods was proposed by Freeman [3] after analyzing and categorizing several published papers using communication networks. Degree, betweenness, and closeness were defined by Freeman [3].

Degree centrality shows how important a particular node is, based on the number of edges it has, within its network. In a weighted network, the importance would be based not only on the number of connections (or degree), but also on the weight associated which each node, termed the strength of a node. Using the node's strength and degree, degree centrality is used to calculate how strong that node is within the network. This thesis **focuses on undirected weighted graphs**. With directed graphs, out and in degrees are used and are extended to strengths accordingly. We believe this work can be extended to directed graphs as well.

1.1 Informal Definitions

Informally, **Degree centrality** (DC) uses the edges of a particular node within a given network to provide the relative importance of that node [4]. In a weighted network, degree centrality is adapted to account for edge weights, providing the relative strength of the node by considering both the number of edges and their weights [5]. Degree centrality is a local metric as compared to closeness and betweenness which are global metrics.

In weighted social networks, the weight on an edge typically represents the intensity, frequency, or duration of interactions between two individuals. For example,

the weight might indicate the number of years two people have known each other, how often they communicate, or how many times they've interacted. In such graphs, degree centrality can be adapted to consider weights, often referred to as node strength, which captures the total weight of connections a node has. For instance, in an interaction network, a person's strength could reflect how frequently or deeply they interact with others. Sorting individuals by strength in descending order enables the identification of the *top-k* most connected or engaged individuals, who may be the most influential within the network.

The time complexity of calculating the degree centrality on both weighted and unweighted graphs is O(E) where E is the total number of edges in the graph. This method is widely used in analyzing huge graphs and being a local metric can be computed in linear time. Computing degree centrality is different for weighted undirected and weighted directed graphs.

1.2 Multilayer network Types

With the increasing complexity of data being analyzed, Multilayer Networks (MLNs) have emerged as a way to represent multiple types of relationships between the same or different sets of entities. Multilayer networks consist of multiple single graphs which have a set of edges that connects those single graphs. With multilayer networks, there is an added flexibility of being able to analyze and process layers individually thereby reducing the size of the graph being analyzed. Different heuristics may use different amounts of partial information from layers for composition, as the overall goal is to keep that to a minimum. In this thesis, decoupling-based algorithms for degree centrality computation in a weighted homogeneous multilayer network using different heuristics for composition. We primarily use OR aggregation

for verifying correctness and weights are aggregated using some of the alternatives mentioned earlier (sum, max, min, average).

Multilayer networks can be categorized into three types, homogeneous, heterogeneous, and hybrid multilayer networks. In a **Homogeneous multilayer network (HoMLN)**, all layers have the same type of nodes and their intersection is, typically, non-empty. This can be used to solve problems where there are different relationships between the nodes. An example of this is an ant interaction network, where each layer represents interactions among the same colony of ants on a different day, one of the real-world MLNs we use for analysis in this thesis. For instance Layer 1 could represent ant-to-ant interactions on Day 1, while Layer 2 represents interactions on Day 2. The nodes (ants) remain the same across both layers, but the set of edges and associated weights, such as frequency or duration or interactions, can differ based on the day.

This allows for an analysis of behavioral changes over time. For example, on Day 1, ant A might have frequently interacted with ant B and C, but on Day 2, it may have interacted more with ant D. By analyzing such changes across layers, researchers can identify shifts in group/colony dynamics, the emergence of influential individuals, or behavioral consistency in certain ants.

In **Heterogeneous multilayer network (HeMLN)**, each layer contains a different set of nodes/entities. For example, in a movie recommendation system, one layer could represent the users, another could represent movies, and a third layer could represent genres. Intra-layer edges depict relationships within each layer, such as friendships and connections between the users, similarities between movies or themes shared between the genres. Inter-layer edges connect the different types of layers, like a user watching a movie or a movie belonging to a genre. A HeMLNs contains both intra-layer and inter-layers edges.

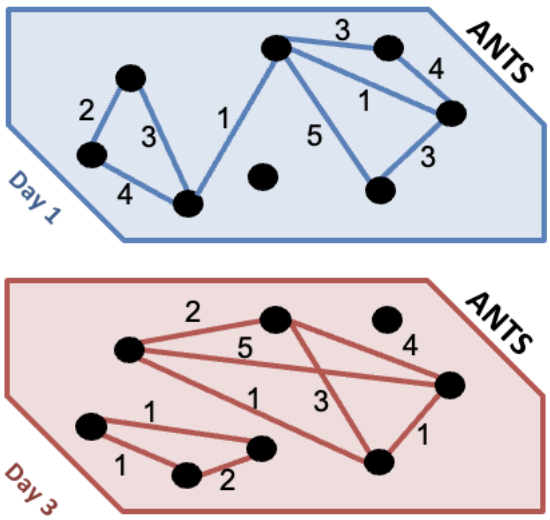


Figure 1.1: An example of Weighted HoMLN Multilayer Network - Ants dataset.

Hybrid multilayer network (HyMLN) is a combination of both Homogeneous MLNs and Heterogeneous MLNs. Using the highway example, two layers could be the highways and non-highway roads in the same state, like Texas, and the other layer would be the highways in another state, like Oklahoma. In this case, the road networks in Texas would be considered as an HoMLN and the combination of Texas and Oklahoma would be HeMLN which provides us with this Hybrid network.

This thesis mainly focuses on weighted HoMLNs.

Figure 1.1 shows an example of HoMLNs where both layers consist of the same ants as nodes. In the first layer, ants are connected if they interact during Day 1; in the second layer, they are connected if they interact during Day 2. Each interaction has an associated weight, frequency of contact, which is added as an edge attribute.

There are other applications that can be modeled as weighted HoMLNs. Centrality metrics are applicable to transportation, Internet, ant relationships, and other MLNs.

Due to lack of algorithms for MLNs, most existing methods reduce multilayer networks into a simple graph for analysis by aggregating layer graphs using Boolean OR or AND operations. This is straightforward for unweighted graphs as only presence or absence of an edge is used for the Boolean operation. While these operations provide a way to combine edges from multiple layers, they also result in the loss of meaningful distinctions between relationships. For example, an edge present in only one layer is treated the same as an edge present in all layers under a Boolean OR, whereas Boolean AND retains only overlapping edges and discards the rest. In both cases, the aggregation flattens layer-specific semantics, which can distort the results of centrality or structural analysis by ignoring the contextual or temporal significance embedded in individual layers. Important distinctions, such as differences in interaction types, time frames, or contextual relationships, are flattened into a single representation, which can distort the underlying meaning of the network. This compromises the accuracy as well as interpretation of any analysis that relies on the structural or semantic integrity of individual layers.

In homogeneous multilayer networks (HoMLNs), an approach would be to combine all the edges into a single graph through the use of Boolean operations. For weighted networks, aggregation becomes even more complex. In addition to Boolean logic, edge weights need to be combined using functions, such as sum, maximum, minimum, or average, depending on the analysis goals. While this produces a unified graph for applying centrality measures, it introduces ambiguity in how the edge weights should be interpreted. Additionally, aggregating all layers into a single graph increases the computational burden, especially for OR aggregation and eliminates opportunities for layer-wise parallel analysis. As a result, while such aggregation is

straightforward, it is often inefficient and leads to potential misrepresentation of the underlying data.

In this thesis, we explore the decoupling-based approach to analyze the multilayer networks. To ensure both accuracy, efficiency, and scalability, this thesis introduces heuristics that intelligently combine centrality results from individual layers, aiming to approximate the results that would be obtained from analyzing the entire multilayer network as a whole.

1.3 Problem Statement

Given a weighted or unweighted homogeneous multilayer network (HoMLN), the objective is to design and implement heuristic-based algorithms for a decoupling-based framework to identify influential nodes, or hubs, based on centrality measures. Unlike traditional single-layer analysis, multilayer networks pose unique challenges due to the presence of multiple layers, each capturing distinct structural information. Aggregating all layers into a single graph may obscure meaningful intra-layer patterns, while analyzing layers independently requires a principled strategy for combining partial results.

This thesis addresses the problem of identifying centrality-based hubs in HoMLNs using a decoupling-based approach. Specifically, we focus on **degree centrality (DC)** and develop heuristic algorithms that compute centrality within each layer independently, then merge the partial results using composition functions. The key challenges include preserving the semantics of individual layers while supporting meaningful cross-layer comparisons, and defining composition functions that accurately reflect the true importance of nodes across layers. Our proposed approach is evaluated on both synthetic and real-world datasets using accuracy metrics such as Jaccard Similarity, Precision, Recall, and F1-score.

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

Chapter 2 is about related work in degree centrality measures in weighted and unweighted networks.

Chapter 3 provides more details of the decoupling-based framework for multilayer networks.

Chapter 4 discusses prior work on multilayer network modeling and analysis, with a focus on approaches for aggregation, decoupling, and composition strategies.

Chapter 5 provides details on the datasets used in this thesis.

Chapter 6 describes the proposed heuristics for computing degree centrality in weighted HoMLNs, and presents the evaluation methodology and results.

Chapter 7 includes the conclusion and future work.

CHAPTER 2

RELATED WORK

As applications become more complex, their modeling and analysis demand expressive frameworks that go beyond traditional simple graphs. Multilayer networks (MLNs), which model multiple types of relationships across layers, preserve semantic distinctions while enabling rich structural analysis. This chapter surveys related work across several dimensions of MLN research, with emphasis on centrality detection in weighted homogeneous multilayer networks (HoMLNs). We particularly highlight methods incorporating weight, structure, and algorithmic scalability, and contrast them with our decoupling-based framework.

2.1 Centrality Measures in Multilayer Networks

2.1.1 Spectral and Energy-Based Methods

Huang et al. [6] proposed an improved gravity-based centrality method that incorporates Supra-Laplacian energy to capture both local and global structural information. Their approach models inter-layer connectivity in a multiplex network using spectral analysis and treats the network holistically via a supra-matrix representation. While this aligns with our motivation to account for cross-layer influence in node importance, our work differs in that we decouple layers entirely, preserving intra-layer semantics and avoiding global matrix construction. Instead of spectral energy, we rely on scalable, interpretable heuristics that combine per-layer centrality using composition functions tailored to weighted HoMLNs.

2.1.2 Classical Centrality Definition for Weighted Networks

In real-world applications, networks frequently contain weighted edges to represent interaction frequency, tie strength, or cost. Traditional (or unweighted) degree centrality measures a node's importance solely based on the number of its direct connections, without considering the strength of those connections. This limitation makes it less suitable for contexts where the magnitude of interaction matters just as much, if not more, than the mere number of connections. Additionally, while degree centrality is computationally simple and remains well-defined even in disconnected graphs, it captures only local structure and cannot identify nodes that are globally influential in sparse or fragmented networks.

To address this limitation, Opsahl et al. [5] proposed a generalized centrality measure that incorporates both the number of connections (degree) and their total weight (strength) using a tunable parameter α. The weighted degree centrality of node i is defined as

$$\text{Degree Centrality}(i) = k_i^{1-\alpha} \times s_i^{\alpha} \tag{2.1}$$

Where k_i is the degree of a node i, s_i is the strength of a node i (i.e., the sum of weights of all edges incident on i), and $\alpha \in [0, \infty)$ controls the balance between degree and strength. When $\alpha = 0$, the metric reduces to classical unweighted degree centrality, which only considers the number of connections. When $\alpha = 1$, the metric becomes node strength, which emphasizes the sum of the weights of the connections. Values of α between 0 and 1 is typically used. The formula is general and includes other possibilities by increasing α beyond 1.

This formulation provides a generalization of degree centrality, enabling flexibility in tuning the metric based on whether the quantity or intensity of connections

is more meaningful in a given context. In this thesis, we only explore $\alpha = 1$, placing full emphasis on edge weights to evaluate node importance in weighted networks.

2.1.3 Hybrid Centrality in Weighted Networks

Khansari et al. [7] explored centrality-based immunization strategies in weighted single-layer networks, introducing hybrid measures like PageRank degree and Radiality degree to identify influential nodes. Their results highlight the benefit of incorporating both local and global features along with edge weights, an idea that resonates with our approach. However, while their work focuses on single-layer graphs and other methods combined with degree centrality, our framework generalizes this idea to multiple layers, where centrality values are computed independently and combined structurally, accounting for both degree and weight variations across layers.

2.1.4 Empirical and Aggregated Models

Gu and Wang [8] investigated real-world public transportation systems by modeling metro and bus networks as layers in a weighted multiplex network. They defined node strength (weighted degree) as the sum of edge weights connected to a node, where edge weights were derived from empirical measures such as passenger flow and average travel time. Specifically, the weight w_{ij} of an edge between stations i and j was computed using a formula that combined flow and cost components to reflect real-world accessibility and usage. The weighted degree centrality of node i was then calculated as

$$\text{Degree Centrality}(i) = \sum_{j=1}^{n} w_{ij}, \tag{2.2}$$

captures the total "importance" of a station based on weighted connectivity. Their analysis showed that node strength scales sub-linearly with city population, highlighting the influence of multilayer structure and real-world weights on urban

transport efficiency. While their work presents a meaningful way to quantify accessibility in domain-specific networks, it aggregates the layers into a unified structure and applies a fixed centrality definition. In contrast, our work retains per-layer structure and introduces a flexible, tunable centrality framework that allows different combinations of degree and strength across layers. We focus on algorithmic general applicability, enabling centrality estimation in weighted homogeneous multilayer networks without reliance on domain-specific metrics or network aggregation.

Zhou et al. [9] introduced a structural analysis of metro-bus networks through a composite network constructed using the Space L model. They introduced reliability metrics such as node overlap and participation to study resilience. Their network is aggregated into a unified topology, where nodes and edges from different modes are merged for analysis. Our framework avoids such flattening by applying layer-specific centrality followed by structural aggregation. Although Zhou et al. target redundancy and robustness in transportation systems, our work generalizes structural analysis to a broader class of HoMLNs, enabling node importance estimation through tunable weight-sensitive heuristics. Another study done by Zhou et al. [10] studied multilayer transportation networks, this time incorporating land use information into node ranking. Using a multi-criteria decision framework (TOPSIS-CRITIC), they combined structural metrics like betweenness with external socioeconomic data. While their work enhances the interpretability of centrality rankings through domain-specific attributes, our approach is entirely structural and self-contained. We do not rely on external features or application-specific weighting; instead, we provide a centrality estimation method that is both generalizable and explainable across weighted HoMLNs.

2.2 Dynamics and Simulation-Based Analyses

Ma and Yuan [11] examined how information and virus propagation are affected by community heterogeneity in a two-layer network. Their study demonstrates that intra-community connectivity plays a critical role in determining propagation thresholds. While their results underscore the structural significance of layers in MLNs, their focus is on dynamic state transitions and simulations over unweighted networks. Our work, by contrast, addresses static centrality estimation using weighted edges and structural composition, aiming for scalable analysis rather than behavioral simulation. Cui et al. [12] modeled diffusion processes in weighted multilayer networks using a nonlinear adoption model influenced by psychological thresholds. Their simulations reveal how individual behavior and heterogeneous edge weights impact spread dynamics across layers. Although we also recognize the importance of weights and interlayer influence, our work does not rely on behavioral models or simulations. Instead, we compute static, weight-aware centrality scores that quantify structural importance, independent of dynamic adoption or temporal state change.

Zhang et al. [13] applied MLNs to model metro system vulnerability under extreme rainfall. They modeled stations and lines as separate layers and used Monte Carlo simulations to estimate performance degradation. Their method captures functional distinctions between layers and supports resilience planning. While this aligns with our interest in maintaining structural separation across layers, their work focuses on simulated vulnerability and efficiency loss, whereas our goal is to develop centrality measures that estimate node importance through static topological and weight-based features across decoupled layers.

2.3 Recent Trends in MLN Applications

The reviewed studies showcase the growing interest in MLNs across infrastructure, social, biological, and economic domains. However, many rely on:

- Layer aggregation into a single graph, often losing semantic separation,
- External, domain-specific metrics (e.g., socioeconomic factors), or
- Simulation-based dynamic processes (e.g., adoption, failure).

In contrast, this thesis introduces a decoupling-based framework that:

1. Preserves the topological and weight-specific properties of each layer,
2. Computes static centrality per layer using scalable heuristics,
3. Integrates results through interpretable composition functions.

This structure-aware, layer-independent design supports generalization across HoMLNs without sacrificing analytical clarity or performance.

2.4 Prior Work on Degree Centrality in MLNs

Traditional centrality measures like degree have been extended to MLNs through naive aggregation methods, such as Boolean OR and AND, which often distort inter-layer semantics. To address this limitation, Pavel et al. [14] introduced a heuristic framework for computing centrality in HoMLNs using a decoupling-based approach.

The method proposed by [14] defines degree centrality per layer and explores multiple composition heuristics to estimate a ground truth ranking derived from the OR-aggregated graph. They analyze the trade-off between computational cost and approximation accuracy and validate their methods across synthetic and real datasets using metrics like precision and Jaccard similarity. Their formal observations and lemmas provide theoretical grounding for heuristic performance.

This thesis builds upon that foundation by extending the decoupling-based approach to **weighted** HoMLNs. We retain the semantic and structural fidelity of each layer while incorporating edge weights into the centrality computation. Our framework introduces weight-sensitive heuristics and composition functions that balance strength and connectivity in estimating node importance efficiently.

CHAPTER 3

MULTILAYER NETWORK ANALYSIS

Multilayer networks (MLNs), also known as multiplex networks, consist of multiple layers of weighted or unweighted graphs, where each layer captures a distinct type of interaction among a set of nodes. In homogeneous MLNs (HoMLNs), these layers are structurally aligned: nodes are consistent across all layers, and edges within each layer represent a specific relationship (e.g., co-authorship during a particular year or interactions on different days).

MLNs provide a powerful data model for capturing complex systems in which multiple dimensions of interaction must be preserved. Applications of MLNs span a wide range of domains, including modeling connectivity patterns in the brain, analyzing transportation networks, and studying dynamic behavior in social systems.

Advantages of modeling using MLNs: The main advantage of using HoMLNs is their ability to represent multiple views or time slices of a system without conflating them into a single graph. Unlike single-layer networks that force aggregation, HoMLNs allow each relationship or interaction type to be analyzed separately while still capturing their collective behavior. For example, in a co-authorship network, different layers might represent collaborations in different years; in biological studies, interactions may be recorded under varying experimental conditions. By preserving this layer-wise distinction, HoMLNs retain richer structural and insights [15].

Disadvantages: Analyzing MLNs can pose several challenges. Traditional graph algorithms, like those for centrality or community, are designed for single-layer net-

works and cannot be directly applied to multilayer networks. A common workaround is to aggregate all layers into a simple or attribute graph.

Graph representation: A homogeneous multilayer network (HoMLN) can be represented using an adjacency matrix or an edge list, where each edge is defined by a source and destination node ID. For weighted networks, edge weights can be included as an additional attribute, and layer-specific files can be used to maintain the separation between interactions across layers.

3.1 Analyzing Multilayer Networks

Several approaches have been proposed for computing centrality and other network metrics in multilayer networks (MLNs), including homogeneous (HoMLNs), heterogeneous (HeMLNs), and hybrid (HyMLNs). These approaches vary in how they process the layered structure of MLNs and the extent to which they preserve layer-level semantics. Figure 3.1 illustrates three main alternatives explored in the literature. We describe two of the most commonly used approaches below and elaborate on the decoupled approach which we use in this thesis.

- Aggregate MLN layers into a simple graph and use pre-existing algorithms: This approach, as shown in figure 3.1(a) flattens the MLN into a single-layer graph by aggregating edges from all layers, typically using Boolean operators (e.g., AND, OR) and, in addition, combine weights if layers are weighted. The result is a weighted simple graph for which standard graph algorithms can be applied. Aggregation strategies such as type-independent [16] and projection-based [17] methods are commonly used. Though simple, this approach discards layer-specific semantics and may distort connectivity patterns [14, 18].
- Develop new algorithms on the entire MLN to compute desired metrics: Algorithms like random walk-based methods operate directly over the entire MLN structure,

preserving semantics and inter-layer interactions. However, they are computationally intensive and inflexible, requiring recomputation even for partial-layer analysis [14].

Compared to the above alternatives, the decoupling-based approach (shown in Figure 3.1(b)) offers a promising balance between preserving structure and ensuring computational efficiency. This topic is discussed in detail in the next section.

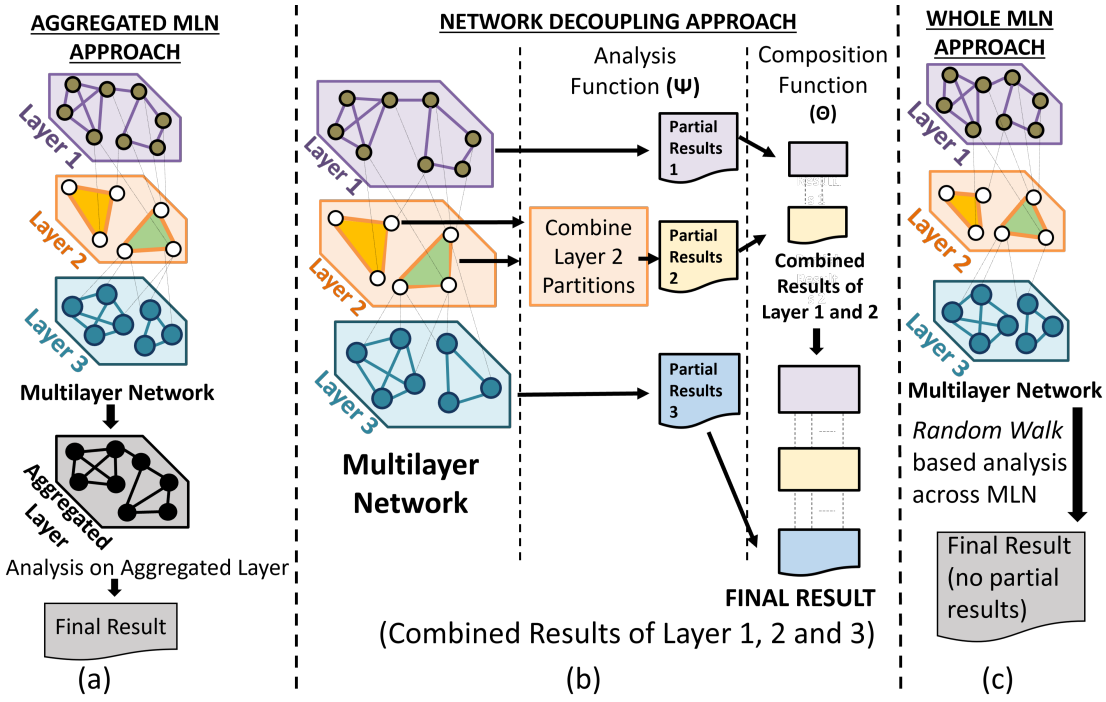


Figure 3.1: (a) Lossy, (b) Decoupling, (c) Whole MLN approaches [1]

3.2 The Decoupling Approach

To address the shortcomings of traditional aggregation in multilayer networks, the decoupling-based approach was introduced for computing centrality MLNs [2]. Rather than flattening all layers into a single graph, this method analyzes each layer independently and then composes the results, preserving the structure of the MLN

as well as the semantics of each layer. The decoupling approach was initially developed for unweighted MLNs and for ground truth (GT), only Boolean AND and OR aggregations were considered. This ground truth needed to be extended for weighted MLNs where, in addition to Boolean operators, the choice of including weights for aggregation is also important. The alternatives for this are shown earlier [2].

Presence of weights and the alternative choice for aggregation introduces additional complexity for the composition part of the algorithm as well as the information to be carried over from each layer for composition. Heuristics need to take into account both Boolean operator used for Gt and the way in which weights are aggregated. However, the principles and functions of he decoupling approach are the same: (i) layer-wise centrality computation and (ii) heuristic-based composition. The figure below shows the architecture of this approach. Overlap of edges across layers need special attention as compared to the unweighted MLNs using only Boolean operators. Hence, another dimension that includes percentage of edge overlaps (all the way from no-overlap to complete overlap) are consided for weighted MLNs.

Figure 3.2 illustrates the decoupling framework. In the first step, each layer of the HoMLN is analyzed independently as a separate graph using the designated analysis function. Since layers in a HoMLN share the same node subset but differ in edge sets and weights, this approach preserves the semantic integrity of each layer. The result of this function is a partial result from each layer. In the second step, the partial results from any two layers are combined using a composition function, which integrates the outcomes to produce results that reflect the multilayer structure as a whole.

This decoupling approach is the same as in unweighted approach and the challenge is to identify (minimal) information to carry over from each layer and developing heuristics based on intuition for the aggregation method used.

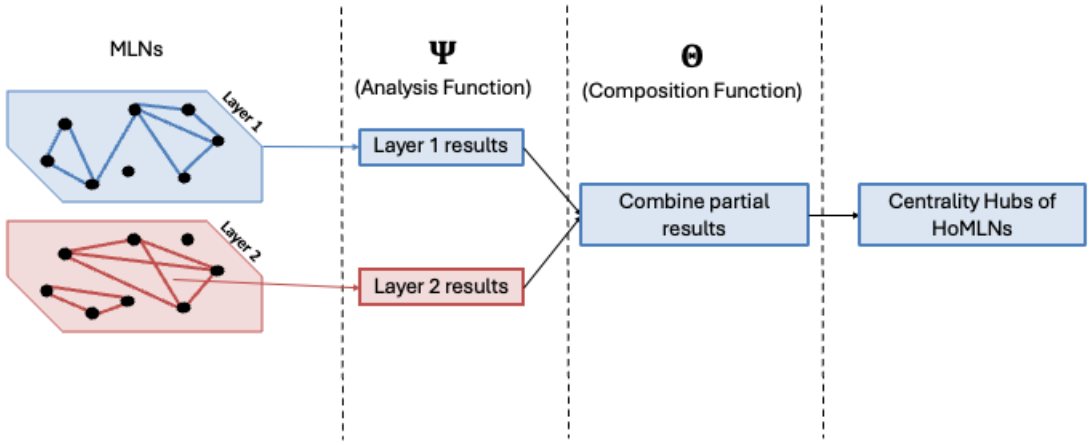


Figure 3.2: Decoupling approach

3.2.1 Advantages of the Decoupling Approach

- Enables parallelism, as layers are processed independently, improving computational efficiency.
- Preserves layer semantics by avoiding information distortion introduced by flattening.
- Utilizes existing algorithms, since each layer can be processed using standard single-layer graph techniques.

3.2.2 Challenges of the Decoupling Approach

- High accuracy is harder to achieve compared to results obtained from single-layer network approaches.
- Requires new algorithm for composition using heuristics.

- Lacks inter-layer awareness, making it challenging to retain potentially relevant global information for the MLN.

3.3 Related work at ITLab

The decoupling-based framework has been actively explored at the ITLab for a number of various tasks on multilayer networks, some of which includes community detection in both homogeneous and heterogeneous multilayer networks (HoMLNs and HeMLNs) [2, 19–25] as well as substructure discovery in HoMLNs [26–30]. These studies have demonstrated the effectiveness of the decoupling approach in enabling efficient analysis while preserving the structural integrity of individual layers.

3.3.1 Community Detection

A community refers to a group of nodes that are densely connected within a network. Well-established algorithms such as Louvain and Infomap are commonly used for community detection. In the context of multilayer networks (MLNs), community detection methods based on the decoupling approach have been proposed and have demonstrated effective results.

Community detection in HoMLNs: Using the network decoupling approach, communities are first detected independently in each layer through the analysis function. For an AND composition, final communities can be derived by intersecting nodes or edges from the communities identified in each layer, retaining only those that are consistently grouped across layers. In contrast, an OR composition aggregates communities by taking the union of nodes or edges from all layers, capturing any node that appears in a community in at least one layer.

3.3.2 Substructure Discovery

A decoupling-based approach is employed for substructure discovery in Homogeneous Multilayer Networks (HoMLNs), leveraging the Map/Reduce framework to enable parallel processing across layers. In this method, each layer is processed independently to identify substructures of size k, which are then expanded iteratively by adding one edge at a time. After each expansion step, a composition function is applied to combine substructures across layers. This process continues until a specified termination condition is met. The final output consists of substructures discovered across the entire HoMLN without collapsing the layers [26].

CHAPTER 4

PRELIMINARIES

Foundational concepts such as centrality measures in classical graph models serve as a basis for exploring the same algorithms for more complex network structures including, edge-weighted and unweighted multilayer networks. These advanced structures enable richer modeling of real-world systems but also introduce new analytical challenges, particularly in how relationships are aggregated and defined across layers. Recognizing the limitations of conventional aggregation methods on such complex networks, the decoupling-based approach is presented as a scalable and interpretable alternative for analyzing centrality in weighted and unweighted Homogeneous Multilayer Networks (HoMLNs).

4.1 Graph-Based Modeling and Centrality

Graphs are fundamental data structure for representing relationships between entities. A graph G(V, E) consists of a set of nodes (or vertices) V and a set of edges E, where each edge connects a pair of nodes in V. Graphs may be directed or undirected, depending on whether the direction of interaction matters. They can also be either weighted or unweighted depending on whether the edges carry numerical values that reflect properties of the relationship, such as frequency, strength, or cost. In graph-based analysis, a wide variety of tasks can be performed, such as traversal, path finding, connectivity analysis, and node ranking based on centrality metrics.

One of the analysis performed on graphs – identifying influential nodes – is particularly useful for understanding the importance of a node and how it can be lever-

aged for propagating information in a network. This is achieved through centrality measures, which quantify the various ways in which a node is important based on their position in the network. The concept of centrality was first introduced by Bavelas [31] in the study of communication patterns, and later formalized by Freeman [3], who categorized the most widely used metrics: degree, closeness, and betweenness. Degree centrality (DC) measures the number of edges incident on a node; in unweighted graphs, it corresponds to the size of a node's immediate neighborhood. Closeness centrality (CC) is inversely related to the sum of the shortest path distances from a node to all other reachable nodes, reflecting how quickly a node can reach others. Betweenness centrality (BC) captures the frequency with which a node lies on the shortest paths between other node pairs, highlighting its role as a bridge or broker in the network. Each of these metrics offers a distinct perspective for interpreting influence, accessibility, and structural importance.

4.2 HoMLNs: Extending the Scope of Graph Models

HoMLNs are well-suited for scenarios where the relationships among a fixed group of entities vary. For example, temporal social networks, where each layer could represent interactions between the same individuals on different days, or biological networks, where the same proteins might interact differently under varying conditions, with each condition as a layer. The uniformity of nodes across layers makes HoMLNs a natural choice for analytical methods that aim to isolate or compare behavioral patterns across different relational settings. Non-temporal snapshots of different interactions among the same people as in Facebook, LinkedIn, and Twitter can also be meaningfully formulated as HoMLNs. If a strength measure is available (e,g., liking on a scale in match-making networks) that can also be included and treated as weighted layers or weighted MLNs.

4.3 Unweighted to Weighted HoMLNs

Most of the early works on MLNs has focused on unweighted graphs, where edge presence alone defines relationships. However, real-world data often includes weights representing interaction strength, frequency, or cost. Although incorporating weights into MLNs allows for more expressive modeling, it introduces challenges for developing analysis algorithms.

In weighted MLNs, it becomes necessary to decide how to interpret or aggregate edge weights across layers, whether to take the sum, minimum, maximum, average, or apply layer-specific importance. Centrality measures and other structural analyses need to be extended to handle the added complexity, particularly in HoMLNs. This thesis addresses these challenges by focusing on weighted HoMLNs and proposing scalable, layer-aware heuristics for degree centrality analysis.

Graph Type	**Logical Operation**	**Aggregation Method**	**Description**
Unweighted	AND	Boolean AND	Edge exists only if present in all layers
	OR	Boolean OR	Edge exists if present in at least one layer
Weighted	AND	Minimum	Minimum weight for edges in all layers where edge exists
	AND	Maximum	Maximum weight for edges in all layers where edge exists
	AND	Sum	Sum of the weights of edges present in all layers
	AND	Average	Average of weights of edges in all layers
	OR	Minimum	Minimum weight for edges in any layer where edge exists
	OR	Maximum	Maximum weight for edges in any layer where edge exists
	OR	Sum	Sum of the weights of edges present in any layers
	OR	Average	Average of the weights of edges in any layer

Table 4.1: Logical operations and aggregation methods for edge construction in multilayer networks.

Table 4.1 outlines the possible aggregation strategies depending on whether the multilayer network (MLN) is weighted or unweighted. In this work, we focus on weighted homogeneous multilayer networks (HoMLNs), where each layer encodes a distinct yet structurally aligned relationship among a fixed set of nodes. For centrality computation, we adopt Boolean OR as the logical operation for aggregating inter-layer connections, as it captures the existence of interaction in at least one layer and provides a comprehensive representation of the network's connectivity.

Within this Boolean framework, we explore two aggregation strategies for combining edge weights: sum and max. The sum operator emphasizes cumulative influence by rewarding repeated interactions across layers, while the max operator highlights the strongest observed interaction between node pairs. These aggregation methods are both analytically interpretable and computationally tractable, making them suitable for estimating node strength and for serving as ground truth against which decoupling-based centrality heuristics are evaluated in later chapters.

CHAPTER 5

GENERATION OF WEIGHTED MLN DATASETS

Synthetic and real-world datasets are used in this work to validate the accuracy, efficiency, and scalability of algorithms proposed for both degree centrality of weighted MLNs. They are also used with diverse graph/MLN characteristics (for synthetic datasets) to make sure heuristics-based algorithms perform well across diverse graph/MLN characteristics. We do not have such control over real-world datasets. Hence, both real-world and MLNs with diverse graph characteristics are tested to make sure our proposed algorithms works across varied datasets. Synthetic datasets were generated from base graphs generated using **subgen** [32] (from the AI Lab at Washington State University) and **PaRMAT** [33] as described in the work by Pavel [14] and Mukunda [34].

These datasets were extended to add weights randomly in a given range to generate weighted graphs. These weighted graphs were then converted into layers by specifying: number of layers, edge split or distribution among layers, and overlap as a percentage across layers. Note that all layers contain the same set of nodes. This approach allows fine-grained control over structural properties such as edge distribution between layers, percentage of edge overlap, and weight ranges. This makes them suitable for benchmarking under a variety of multilayer configurations. Real-world datasets were obtained from domain-specific studies, where each layer represents observations from different time points.

Once the layers are generated or identified (in case of real-world datasets), ground truth graph need to be generated using the layers, specified Boolean operator,

and the function for computing the strength of the edge in the ground truth. We have chosen sum and max function for this thesis. Other possible functions are minimum, average, and custom combining of weights.

Analysis on the ground truth graph is performed using existing algorithms available for weighted simple graphs. In contrast, analysis output (and any additional carryover information appropriate for the heuristics being used) of the layers is combined in the composition function using chosen heuristics for generating the analysis for MLNs.

We first outline the two methods used for generating synthetic HoMLN datasets and also describes the real-world datasets used in the evaluation.

Two general strategies were considered for generating synthetic homogeneous multilayer networks:

- Bottom-up: Combining two existing graphs into a simple graph using a Boolean operator (AND or OR) and a weight function (one of sum, minimum, maximum, average.)
- Top-down: Start from a single graph and partition it into layers as needed. This method was used to generate all synthetic datasets in this work.

5.1 Synthetic Dataset Generation Using Top-down

The reason for this choice is that it affords much better control in the generation of layers with diverse characteristics. For our purpose, a number of layer characteristics are important to make sure that the heuristics developed for composition works well across layer graph characteristics. Subgen [32] is used to generate the base graph from which layers are generated. Parameters used for layer generation include: edge distribution (e.g., 90-10, 70-30, and 50-50), weight range (e.g., randomly chosen in

the interval [1, 10]), and edge overlap across layers (e.g., 0%, 25%, 50% 75%, and 100%) to make sure a wide range of MLN characteristics are tested.

Our layer generation starts with a single unweighted base graph and generates a k-layer HoMLN by the distributing the edges across layers using parameters indicated above and given in a configuration file. The following are user defined parameters that control the generation of an MLN from the base graph:

- Number of layers: Specifies how many layers the base graph is to be split into. For this work, we have used two.
- Edge Distribution: Determines the percentage of total edges allocated to each layer (e.g., 50-50, 70-30, 90-10). This is to simulate different graph densities among layers and explore their impact on the performance of the heuristic.
- Aggregation Method: Specifies the function to be used when computing the ground truth from the two layers (e.g. sum, max, min, average). This ensures consistency in the weights when the layers are later recombined.
- Weight Range: Specifies the minimum and maximum edge weights to assign during generation. For weighted graphs, edge weights are randomly chosen from the specified range (min, max). In this thesis, we use the range (1,10). For unweighted graphs, all edge weights are set to 1 by default.
- Random seed: A random seed is used to ensures reproducibility of MLNs from the base graph.

It is important to understand that once the layers are generated from the base graph, we still need to generate the aggregated/flattened ground truth graph **after** the generation of layers as the initial base graph does not have any weights. For this work, each generated HoMLN consists of two layers with shared node sets but different edge distributions and weights. These layers are then aggregated using the

selected aggregate function (e.g., sum, max, or min) and the Boolean operator (OR in our case) for conflating layer graph to generate ground truth graph for evaluation.

5.1.1 Characteristics of synthetic datasets used

The datasets were generated from larger unweighted base graphs using the top-down approach. For each base graph, the total edge set is split into two layers using different edge distribution configurations (90-10,70-30, and 50-50), with edge overlap percentages of 0%, 25%, 50%, and 75% being applied. In tables 5.1, 5.2, 5.3:

- The **dataset** column indicates the dataset name.
- **Edges (l_1)** and **Edges (l_2)** specify the number of edges in layer 1 and layer 2, respectively.
- **Max Deg** gives maximum node degrees across both layers. The minimum degree for all nodes shown in the datasets are 0.
- **Min Wt** and **Max Wt** indicate the range of edge weights assigned randomly within the specified interval.
- **% Overlap** represents the percentage of edges shared across both layers.
- **% Disjoint** indicates the percentage of edges that are unique to each layer.

Dataset	**Nodes**	**Edges** (l_1)	**Edges** (l_2)	**Max Deg**	**Min Wt**	**Max Wt**	**% Overlap**	**% Disjoint**
25KV100KE	**25000**	90039	9961	408	1	10	0.0	100.0
		92398	32561	428	1	10	24.959	75.041
		94898	55102	439	1	10	50.0	50.0
		97444	77387	456	1	10	74.831	25.169
10KV200KE	**10000**	180105	19895	1004	1	10	0.0	100.0
		184866	65006	1050	1	10	24.936	75.064
		189945	110009	1083	1	10	49.977	50.023
		194912	154886	1114	1	10	74.899	25.101
100KV1ME	**100000**	1370037	151817	35892	1	10	0.0	100.0
		1407562	494715	37019	1	10	24.997	75.003
		1445557	837224	37873	1	10	50.0	50.0
		1483822	1179423	38995	1	10	75.0	25.0
325KV1ME	**325557**	1577761	175005	9717	1	10	0.0	100.0
		1620942	570016	9953	1	10	25.0	75.0
		1664967	963981	10226	1	10	49.989	50.011
		1708864	1358261	10494	1	10	74.988	25.012
862KV10ME	**862664**	9039100	1004825	58006	1	10	0.0	100.0
		9290897	3262680	59643	1	10	24.987	75.013
		9540868	5524794	61252	1	10	49.998	50.002
		9792402	7784467	62824	1	10	75.0	25.0
1MV8ME	**1382908**	7618756	846557	10090	1	10	0.0	100.0
		7829858	2750582	10394	1	10	24.986	75.014
		8040864	4655471	10644	1	10	49.981	50.019
		8253255	6561043	10920	1	10	75.0	25.0

Table 5.1: Characteristics of Synthetic Graphs for a 90-10 Distribution Split

Dataset	**Nodes**	**Edges** (l_1)	**Edges** (l_2)	**Max Deg**	**Min Wt**	**Max Wt**	**% Overlap**	**% Disjoint**
25KV100KE	**25000**	70224	29776	312	1	10	0	100
		77489	47470	365	1	10	24.959	75.041
		84848	65152	401	1	10	50	50
		92528	82303	436	1	10	74.831	25.169
10KV200KE	**10000**	140379	59621	776	1	10	0	100
		154815	95057	895	1	10	24.936	75.064
		169785	130169	995	1	10	49.977	50.023
		184850	164948	1065	1	10	74.899	25.101
100KV1ME	**100000**	1066208	455646	27968	1	10	0	100
		1179803	722474	31076	1	10	24.997	75.003
		1293882	988899	33951	1	10	50	50
		1408278	1254967	37078	1	10	75	25
325KV1ME	**325557**	1227476	525290	7590	1	10	0	100
		1359139	831819	8367	1	10	25	75
		1489917	1139031	9212	1	10	49.989	50.011
		1621635	1445490	9949	1	10	74.988	25.012
862KV10ME	**862664**	7028451	3015474	45155	1	10	0	100
		7783578	4769999	49906	1	10	24.987	75.013
		8536813	6528849	54769	1	10	49.998	50.002
		9291293	8285576	59685	1	10	75	25
1MV8ME	**1382908**	5923827	2541486	7857	1	10	0	100
		6559199	4021241	8695	1	10	24.986	75.014
		7194477	5501858	9522	1	10	49.981	50.019
		7831080	6983218	10367	1	10	75	25

Table 5.2: Characteristics of Synthetic Graphs for a 70-30 Distribution Split

Dataset	**Nodes**	**Edges** (l_1)	**Edges** (l_2)	**Max Deg**	**Min Wt**	**Max Wt**	**% Overlap**	**% Disjoint**
25KV100KE	**25000**	50068	49932	233	1	10	0	100
		62471	62488	293	1	10	24.959	75.041
		74815	75185	357	1	10	50	50
		87526	87305	410	1	10	74.831	25.169
10KV200KE	**10000**	100035	99965	570	1	10	0	100
		124885	124987	713	1	10	24.936	75.064
		149537	150417	888	1	10	49.977	50.023
		174858	174940	1015	1	10	74.899	25.101
100KV1ME	**100000**	761644	760210	20021	1	10	0	100
		951498	950779	25162	1	10	24.997	75.003
		1141111	1141670	29985	1	10	50	50
		1332199	1331046	35037	1	10	75	25
325KV1ME	**325557**	876761	876005	5451	1	10	0	100
		1096128	1094830	6794	1	10	25	75
		1313920	1315028	8151	1	10	49.989	50.011
		1533929	1533196	9454	1	10	74.988	25.012
862KV10ME	**862664**	5019703	5024222	32333	1	10	0	100
		6276767	6276810	40361	1	10	24.987	75.013
		7530699	7534963	48390	1	10	49.998	50.002
		8789019	8787850	56503	1	10	75	25
1MV8ME	**1382908**	4230583	4234730	5619	1	10	0	100
		5289206	5291234	7049	1	10	24.986	75.014
		6346357	6349978	8454	1	10	49.981	50.019
		7407691	7406607	9814	1	10	75	25

Table 5.3: Characteristics of Synthetic Graphs for a 50-50 Distribution Split

5.2 Ground Truth Graph Construction

Once the synthetic HoMLNs are generated, the ground truth network must be constructed to serve as a reference when evaluating the accuracy and performance of the proposed heuristics. The ground truth represents an aggregated view of the multilayer network structure and is computed by applying a Boolean operator along with weight aggregation.

For example: Given two layers, $G_x = (V, E_x)$ and $G_y = (V, E_y)$, where each layer has the same set of nodes but potentially different edge sets and weights, a ground truth $G_{\text{GT}} = (V, E_{\text{GT}})$ is created using the following steps:

- **Boolean Aggregation:** Edges from the two layers are aggregated structurally using either Boolean OR or Boolean AND:
  - **Boolean AND Aggregation:** Only edges that exist in both layers are included in the ground truth network. Formally, $E_{\text{GT}} = E_x \cap E_y$. This places emphasis on commonality between layers and might result in sparser graphs.
  - **Boolean OR Aggregation:** All edges present in either layer are included in the ground truth. Formally, $E_{\text{GT}} = E_x \cup E_y$. This ensures that no edge information is lost, which makes it suitable for cases where completeness is prioritized. **This thesis focuses on Boolean OR-aggregation.**
- **Weight Aggregation:** For unweighted graph, there is no additional aggregation of edge weights performed apart from Boolean operator semantics for edges (presence or absence). However, if the graph edges are weighted, then for each edge $(u, v) \in E_{\text{GT}}$, its weight is computed using one of the following methods. w_x and w_y are weights of edges belonging to layer graphs G_x and G_y respectively.
  - **Sum:** $w_{\text{GT}}(u, v) = w_x(u, v) + w_y(u, v)$

- **Max:** $w_{\mathrm{GT}}(u,v) = \max(w_x(u,v),\ w_y(u,v))$
- **Min:** $w_{\mathrm{GT}}(u,v) = \min(w_x(u,v),\ w_y(u,v))$
- **Average:** $w_{\mathrm{GT}}(u,v) = \frac{w_x(u,v)+w_y(u,v)}{2}$

If an edge exists in only one layer, (under Boolean OR), its weight from that layer is retained directly in G_{GT}.

In our implementation, Boolean operation and weight aggregation are performed together to generate the final ground truth graph G_{GT}, which captures both the structural and weighted properties of the two-layer HoMLN.

5.3 Real-world Datasets

To complement the synthetic datasets, real-world homogeneous multilayer network (HoMLNs) have ben used to validate the effectiveness of the proposed heuristics for computing weighted degree centrality. These datasets capture real interactions from two different domains and provide layered views of the same set of entities over time.

- **Co-Authorship Network:** This dataset is derived from the Open Graph Benchmark's ogbl-collab dataset [35]. This network represents a subset of collaborations between authors (Co-authorship) which was indexed by Microsoft Academic Graph (MAG). Each node represents an author, and an edge between two authors indicates a co-authored paper. Each edge is timestamped by year and weighted according to the number of joint publications for a particular year. For this thesis, two layers of this dataset were extracted to form a two-layer HoMLN:
  - Layer 1: Co-authorship's from the year 2004
  - Layer 2: Co-authorship's from the year 2005

The nodes remain consistent across both layers, while the edge sets and weights differ depending on the collaboration activity for each year. The edge weights were retained as-is from the dataset to reflect the intensity of the interaction. Table 5.4 shows the characteristics of the each layer for the Co-Authorship networks

Property	**Co-authorship 2004**	**Co-authorship 2005**
Number of Nodes	235,868	235,868
Number of Edges	32,880	42,257
Network Density	1.18×10^{-6}	1.52×10^{-6}
Number of Connected Components	222,422	219,924
Minimum node degree	0	0
Maximum node degree	47	56
Minimum edge weight	1	1
Maximum edge weight	7	7
Percentage of overlapping edges	6.58%	6.58%
Percentage of disjoint edges	93.42%	93.42%

Table 5.4: Structural properties of the Co-Authorship dataset across two layers (2004 and 2005)

- **Ant Interaction Network:** This dataset is drawn from a curated repository of animal social networks that compiles over 1,000 networks from diverse species [36]. One of the networks from this repository captures interactions in an ant colony over multiple days. Each node represents an individual ant, and edges reflect observed interactions between the ants. The edges are defined by spatial proximity, meaning a pair of ants is said to have interacted when the front end of one ant enters the trapezoidal proximity zone of another ant. For this thesis, two observations days were selected:
  - Layer 1: Interactions observed on Day 1
  - Layer 2: Interactions observed on Day 3

Edge weights indicate the frequency of interactions during the respective day. Since the colony is the same, the node set remains the same across both layers. Table 5.5 shows the characteristics of the each layer for the Ant Interaction networks

Property	**Ants Day 01**	**Ants Day 03**
Number of nodes	164	164
Number of edges	10,731	10,090
Network density	0.8029	0.7549
Number of connected components	1	1
Minimum node degree	41	20
Maximum node degree	160	158
Minimum edge weight	1	1
Maximum edge weight	229	108
Percentage of overlapping edges	74.92%	74.92%
Percentage of disjoint edges	25.07%	25.07%

Table 5.5: Structural properties of the Ants interaction dataset across two layers (Days 1 and 3)

CHAPTER 6

COMPUTING DEGREE CENTRALITY FOR WEIGHTED-HoMLNs

In weighted graphs, degree centrality is commonly extended to not only account for the number of edges incident on a node but also for the strength of those connections. This is particularly important in real-world networks, where edge weights represent interaction intensity, frequency, or cost. However, when such graphs are extended into HoMLNs, computing degree centrality becomes significantly more complex.

Existing work on degree centrality in multilayer networks has largely focused on unweighted graphs, using techniques such as Boolean aggregation (e.g., OR or AND operations) or heuristic-based composition. As discussed in Chapter 2, Opsahl et al. [5] proposed a generalized degree centrality measure that incorporates both the degree and the strength of a node using a tunable parameter α. In this thesis, as we want to place full emphasis on edge weights and node strengths, so we set $\alpha = 1$ for the formula shown in 2.1 which reduces to formula shown in 6.1, where s_i is the sum of the weights of the edges incident at node i.

$$DC_i = s_i \tag{6.1}$$

Ground Truth for Weighted Degree Centrality: To evaluate the accuracy and efficiency of the proposed heuristics, the ground truth must be established. In this thesis, for weighted-degree centrality, the ground truth is constructed using Boolean-OR aggregation of all layers, meaning that an edge is included in the aggregated network if it appears in at least one of the layers. For edges that appear in multiple

layers, weights are aggregated using one of the following strategies: sum, maximum, minimum, or average. Among these strategies, this thesis on weighted-degree centrality focuses on sum and maximum aggregation methods:

- Sum aggregation captures cumulative interaction strength across layers.
- Max aggregation preserves the strongest observed interaction between node pairs.

The aggregation graph produced by Boolean-OR combined with sum or max weights is then used to compute degree hubs for the GT weighted graph. Nodes with strengths greater than or equal to the graph average are identified as degree hubs and are used as the ground truth for evaluating heuristic performance. Jaccard coefficient, precision, and recall are used for comparing the results with the ground truth.

All heuristics proposed in this thesis are tested on two-layer HoMLNs, although the methodology can be generalized to more than two layers. For the decoupling approach, each layer is independently analyzed as outlined earlier in Chapter 3. Each layer is passed into an analysis function, which computes node strength using weighted degree centrality. The output includes global statistics, such as maximum and minimum strengths, node IDs along with their corresponding strengths, and the total number of nodes, etc. These outputs from all layers are passed as inputs into the composition function, which combines information from layers using heuristics to estimate the hubs for the entire HoMLN.

Heuristic Design and Strategy: For the sum and max aggregation method, we propose one heuristic each. The goal of these heuristics is to retain minimal information from each layer and obtain high accuracy in estimating or computing the degree hubs of the HoMLN. These heuristics are evaluated based on accuracy and computation time.

Naïve Accuracy: As a baseline for measuring improvement in accuracy and performance of our heuristics, we define a naïve approach for the decoupling-based framework that uses minimal information from each layer. For each layer, we identify the degree hubs in each individual layer, and take the union (as our aggregation Boolean operator is OR) of the resulting hub sets as the estimated set of hubs for the HoMLN. This method does not utilize any additional information beyond the identified layer hubs, which is the minimal information, and applies a simple heuristic. Naïve degree hub computation for HoMLN also involves a minimal amount of computation as part of the composition function. While this method is computationally inexpensive, the naïve approach is generally not accurate and is unlikely to match the ground truth results except in pathological cases [14]. Therefore, this method establishes a minimum benchmark for accuracy, which can be used to further evaluate the performance of the proposed heuristics.

6.1 Weighted Degree Centrality Using Sum Aggregation

As outlined earlier, the decoupling approach separates the computation into two stages: an analysis function, which operates on individual layers only without using any other layer or global information, and a composition function, which merges the partial results from each layer using heuristics. This modularization results in flexibility of using a subset of layers, efficiency of composition, and scalability due to reduced layer sizes. It also facilitates the design and fine-tuning of heuristics to use partial layer-wise information.

Analysis Function: The input to the analysis function is a single layer of the HoMLN, which includes nodes and weighted edges. The function computes the strength (weighted degree) of each node in the layer by summing the weights of

its incident edges. Analysis results, which are degree hubs, and any other layer-only information, are generated and passed on to the composition stage.

Composition Function: This function receives the outputs of the analysis function from the two layers, which are denoted as Layer_x and Layer_y. For each node u, the estimated strength in the aggregated graph is:

$$\hat{s}(u) = s_x(u) + s_y(u) \tag{6.2}$$

where $s_x(u)$ and $s_y(u)$ are the node strength values from Layer_x and Layer_y, respectively. The global average strength can be computed using —V— as the number of nodes:

$$\bar{s} = \frac{1}{|V|} \sum_{u \in V} (s_x(u) + s_y(u)) \tag{6.3}$$

Any node for which $\hat{s}(u) > \bar{s}$ is marked as a degree hub.

However, if the layer strength sums are carried over, the above can be computed by adding them and dividing by the number of nodes, which remains the same for the GT graph.

To better understand the behavior of the decoupling-based framework on weighted homogeneous multilayer networks (HoMLNs), consider a small HoMLN with five nodes $V = \{a, b, c, d, e\}$. Figures 6.1 and 6.2 show two layers ($Layer_x$ and $Layer_y$) and their corresponding ground truth graph, constructed using Boolean-OR for edge structure and sum aggregation for weights. As introduced earlier, the decoupling framework separates the analysis of each layer from its eventual composition. Node strengths for each layer and the ground truth are shown in Table 6.1.

Lemma 1. *A node that is a hub in only one layer of a weighted homogeneous multilayer network (HoMLN) may not remain a hub in the ground truth network constructed using Boolean-OR aggregation with sum aggregation.*

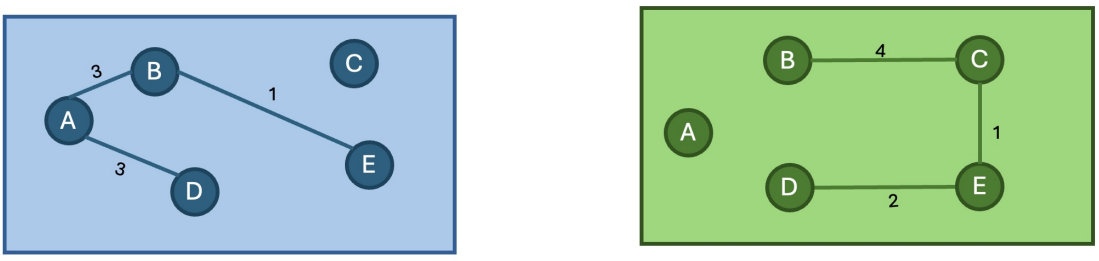


Figure 6.1: Layer-wise weighted edges in the multilayer network.

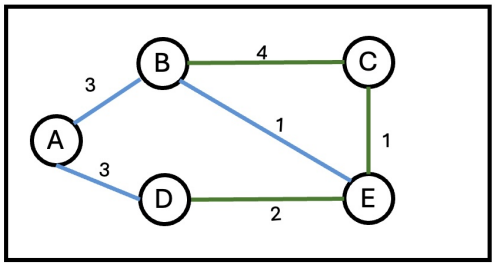


Figure 6.2: Ground truth network (G_{GT}) using Boolean-OR and sum aggregation.

Proof. Consider the two-layer HoMLN shown in Figure 6.1, with the corresponding ground truth network shown in Figure 6.2. Table 6.1 lists the strengths of each node in both individual layers and the aggregated graph.

In $Layer_x$, the average strength is 2.8, and nodes a, b, and d are identified as hubs (i.e., $s(u) > 2.8$). In $Layer_y$, nodes b, c, and e also exceed this threshold and

Node	**Strength in** $Layer_x$	**Strength in** $Layer_y$	**Strength in Ground Truth**
a	6	0	6
b	4	4	8
c	0	5	5
d	3	2	5
e	1	3	4
average	2.8	2.8	5.6

Table 6.1: Node strengths in $Layer_x$, $Layer_y$, and the ground truth GT

are marked as hubs. The naïve approach, which takes the union of hubs from both layers, therefore identifies the full set $\{a, b, c, d, e\}$ as hubs.

However, when the layers are aggregated using Boolean-OR with sum aggregation, the ground truth average rises to 5.6, and only nodes a and b qualify as hubs.

While node b is a hub in both layers and remains a hub in the ground truth, nodes c, d, and e, each of which was a hub in only one of the two layers, do not meet the global average strength threshold and are excluded from the final hub set.

This example shows that being a hub in a single layer is not sufficient to ensure hub status in the aggregated network. Sum aggregation increases the overall strength baseline, which can reduce the relative centrality of nodes that are only locally prominent. This also highlights how the naïve method can significantly overestimate the true hub set. □

To ensure correctness, the composition algorithm 1 computes each node's total strength across all layers and compares it against the global average to determine hub status:

Algorithm 1 Composition algorithm For Weighted Degree Centrality For Sum Aggregation

Input:

- $Layer_x = \{(u, s_x(u)) \mid u \in V\}$ *Nodes & Strengths from* $Layer_x$
- $Layer_y = \{(u, s_y(u)) \mid u \in V\}$ *Nodes & Strengths from* $Layer_y$
- $n = |V|$ *Number of nodes*
- $\texttt{sum}_x = \sum_{u \in V} s_x(u)$
- $\texttt{sum}_y = \sum_{u \in V} s_y(u)$

$\bar{s} \leftarrow (\texttt{sum\_x} + \texttt{sum\_y})/n$
Hubs $\leftarrow [\,]$
for each node $u \in V$ **do**
    $\hat{s}(u) \leftarrow s_x(u) + s_y(u)$
    if $\hat{s}(u) > \bar{s}$ **then**
        Hubs.append(u)
    end if
end for
return *Hubs*

This full-node evaluation guarantees accurate computation of ground truth hubs by evaluating the strength of every node using the sum of its strengths across all layers. However, this exact approach can be computationally intensive in large-scale networks. As a baseline for comparison, a naïve approach is considered. This naïve method composes the final hub set by simply taking the union of hubs identified independently from each layer, thereby avoiding global strength computation. While computationally efficient, it may overlook true hubs that emerge from cumulative strength contributions distributed across layers.

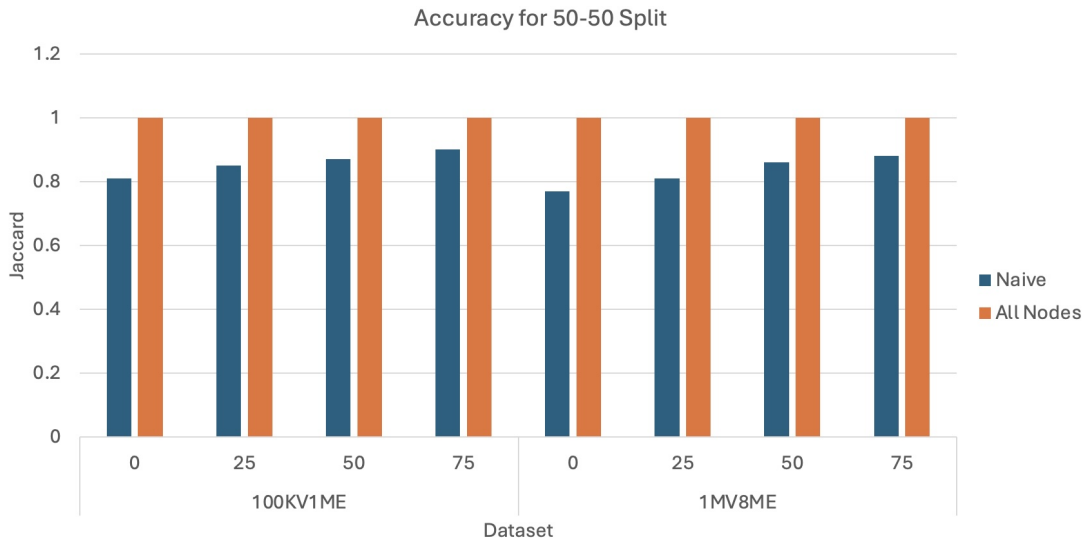


Figure 6.3: Accuracy for Sum Aggregation Using All Nodes vs. Naïve Approach (50-50 Distribution)

The following figures compare the accuracy of this naïve approach against keeping and using all the nodes and their strengths ("All Nodes") under the different edge distribution splits. The Jaccard coefficient is used to assess the accuracy of each method by comparing its hub set to the ground truth.

Figures 6.3 through 6.5 present the accuracy of hub detection across different datasets and edge distribution splits. The X-axis shows different configurations of edge overlap (0%, 25%, 50%, 75%) grouped by distribution splits (50-50, 70-30, 90-10). The Y-axis reports the Jaccard coefficient, which measures similarity between the estimated and ground truth hub sets. The results demonstrate that using all nodes in the composition step yields perfect accuracy (Jaccard coefficient of 1.0) across all distribution splits. In contrast, the accuracy of the naïve approach varies considerably depending on the dataset characteristics.

One key factor influencing the naïve accuracy is the extent of the edge overlap. As overlap increases, the Jaccard coefficient of the naïve approach increases across all

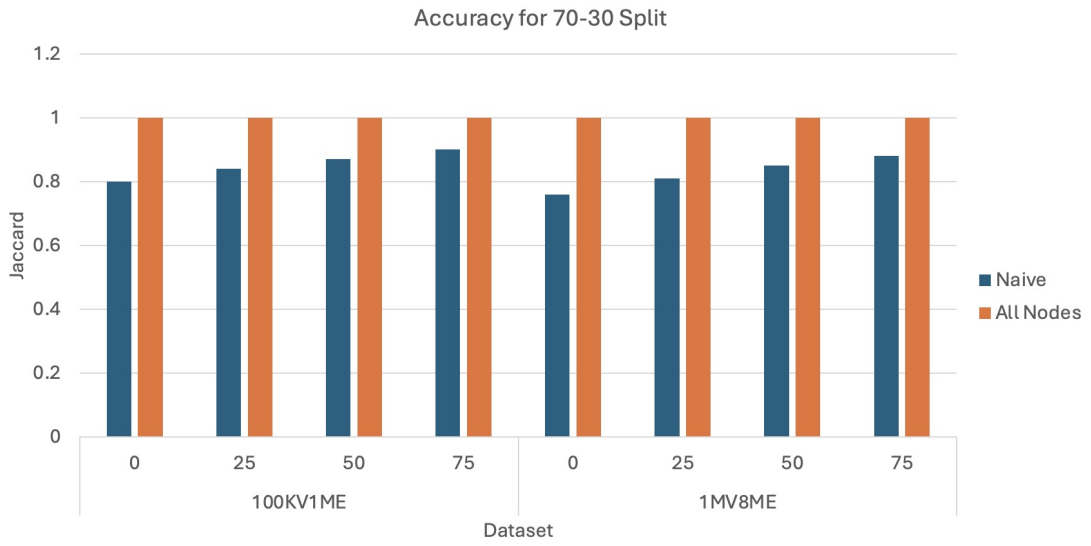


Figure 6.4: Accuracy for Sum Aggregation Using All Nodes vs. Naïve Approach (70-30 Distribution)

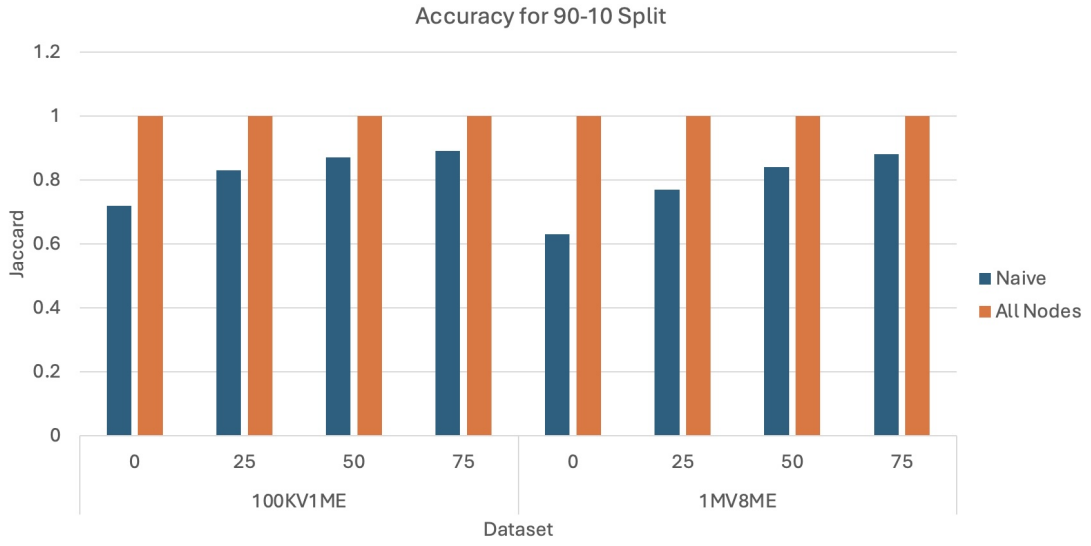


Figure 6.5: Accuracy for Sum Aggregation Using All Nodes vs. Naïve Approach (90-10 Distribution)

distribution splits. This is because the higher the overlap, the higher the likelihood that the same edges appear in both layers. When edges are shared across layers, the strength contributions they provide are retained in both individual layer computations. Consequently, the individual hub sets derived from each layer become more similar to the hub set that would be obtained from aggregating the layers. Since the naïve approach relies on the union of these individual hub sets, its output becomes increasingly aligned with the ground truth as overlap increases. The redundancy in overlap reduces the likelihood of missing important nodes while taking the union in the naïve approach, leading to higher overall accuracy.

Another important factor is the edge distribution split across layers. The naïve method performs noticeably worse under unbalanced splits such as 90-10, especially when the overlap is low. This decline in accuracy is attributed to the sparsity of one layer, which reduces the chances of correctly identifying hubs based solely on that layer. When strength is unevenly distributed across layers, many nodes may fail to exceed the average strength in either layer individually, even if their combined strength would qualify them as hubs in the aggregated network.

Figure 6.6 shows the accuracy of the full-node and naïve composition approaches on real-world datasets. The X-axis shows different composition strategies: All Nodes and Naïve. The Y-axis shows the Jaccard coefficient comparing each result with the ground truth hub set. Similar to trends observed in the synthetic evaluations, the full-node strategy achieves perfect agreement with the ground truth (Jaccard score of 1.0) for both datasets. However, the naïve approach exhibits a sharp decline in accuracy for the Ant Interaction network, achieving a Jaccard score below 0.85. This decline is likely due to the small size and dense connectivity of the ant network, where subtle weight differences can significantly impact hub designation. In contrast, for the co-authorship network, the naïve approach performs comparably to the full-

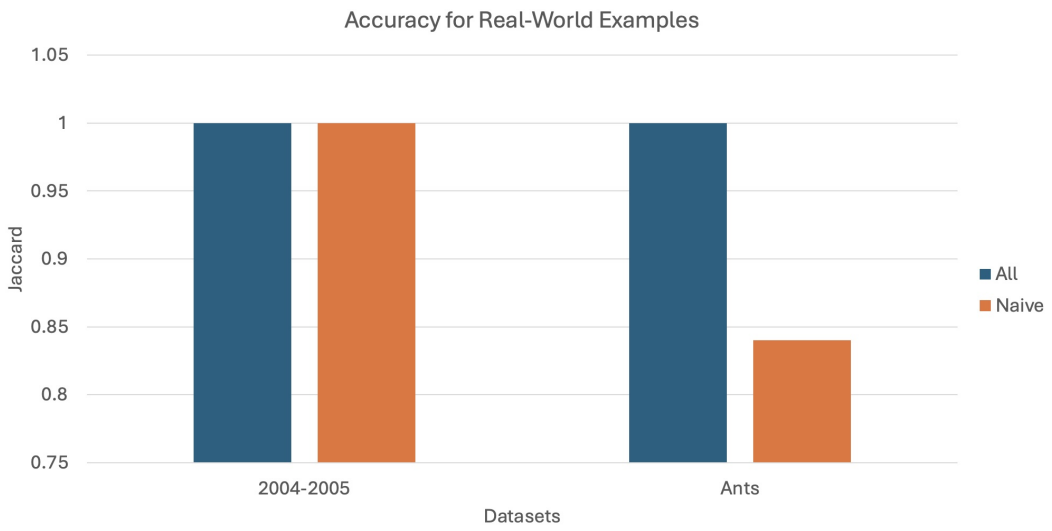


Figure 6.6: Accuracy for Real-World Datasets: All Nodes vs. Naïve Composition

node method. This is likely because the co-authorship graph has a large number of nodes, and the top hubs tend to appear in both individual layers, making the union of per-layer hubs sufficient for high accuracy.

Interestingly, while the synthetic dataset 1MV8ME also achieves high accuracy at higher overlaps, its behavior differs from that of the co-authorship network. The co-authorship layers are inherently correlated, with many influential nodes maintaining their prominence across years, resulting in strong alignment between per-layer hubs and the ground truth. In contrast, 1MV8ME uses probabilistic edge generation, which introduces variation in node strengths across layers, particularly at lower overlaps. As a result, the naïve accuracy in 1MV8ME starts lower and improves gradually with increased overlap. This contrast highlights that even when the naïve approach achieves high accuracy, the underlying cause varies: in real-world co-authorship networks, it stems from the temporal consistency of influential nodes, whereas in synthetic graphs, it results from deliberate structural redundancy due to high edge overlap.

This contrast also underscores a fundamental limitation of the naïve approach: it fails to capture nodes whose importance arises only when strengths are aggregated across layers.

This motivates the next observation: nodes that are not hubs in either individual layer may still become ground truth hubs due to the additive effects of sum aggregation.

Lemma 2. *A node that is not a hub in one of the layers of a weighted homogeneous multilayer network (HoMLN) may become a hub in the ground truth network constructed using Boolean-OR aggregation with sum aggregation for the weights.*

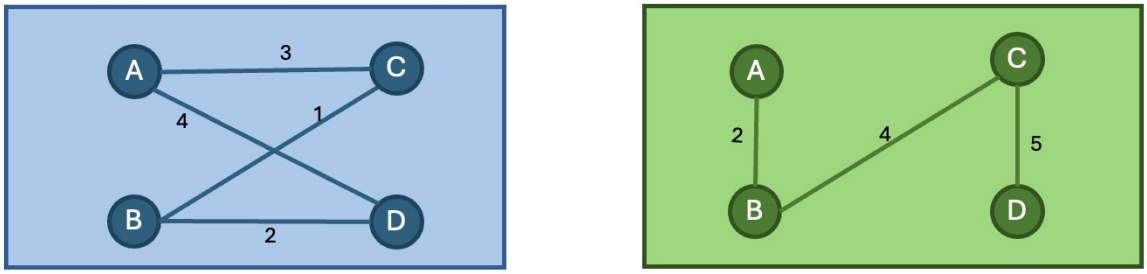
Proof. Consider again the multilayer network in Figures 6.1 and 6.2, with node strengths provided in Table 6.1.

Node a has strength 6 in Layer x, but 0 in Layer y, meaning it is not a hub in Layer y. However, its cumulative strength in the ground truth remains 6, which exceeds the global average of 5.6, making it a hub in the aggregated network.

This shows that even partial contributions from a single layer can elevate a node's aggregated strength above the threshold. Hence, relying solely on layer-local hub identification risks missing important global hubs in the GT network. □

The following lemma formalizes this limitation and highlights a fundamental shortcoming in such heuristics, namely that evaluating only layer-local hubs can overlook nodes whose combined strength across layers qualifies them as hubs in the ground truth network.

Lemma 3. *Heuristics that evaluate only those nodes identified as hubs in one or more input layers do not guarantee recovery of all true hubs in the ground truth network constructed using Boolean-OR aggregation with sum aggregation for the weights.*



Layer$_1$ **Layer$_2$**

Figure 6.7: Layer-wise weighted edges in the multilayer network.

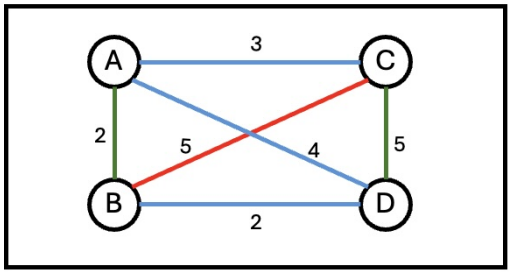


Figure 6.8: Ground truth network (G_{GT}) using Boolean-OR and sum aggregation.

Node	**Strength in** $Layer_1$	**Strength in** $Layer_2$	**Strength in Ground Truth**
A	7	2	9
B	3	6	9
C	4	9	13
D	6	5	11
average	5.0	5.5	10.5

Table 6.2: Node strengths in $Layer_1$, $Layer_2$, and the ground truth GT

Proof. Consider the two-layer HoMLN shown in Figure 6.7, with the corresponding ground truth network shown in Figure 6.8. Table 6.2 lists the strengths of each node in both individual layers and in the aggregated graph.

In $Layer_1$, the average strength is 5.0, and nodes A and D are identified as hubs (i.e., $s(u) > 5.0$). In $Layer_2$, the average strength is 5.5, and nodes B and C are identified as hubs. Under the Hubs-Only heuristic, the GT strength for each node is computed using *only* the layer(s) in which that node is a hub. For example, if a node

is a hub in $Layer_1$ but not in $Layer_2$, its GT estimate is taken solely from $Layer_1$, ignoring its contribution from $Layer_2$.

When the layers are aggregated using Boolean-OR with sum aggregation, the ground truth strengths are:

$$s_{GT}(A) = 9, \quad s_{GT}(B) = 9, \quad s_{GT}(C) = 13, \quad s_{GT}(D) = 11,$$

With a global average $\bar{s}_{GT} = 10.5$. The resulting GT hub set is $\{C, D\}$.

While node C is correctly identified as a hub by both the heuristic and the GT computation, node D illustrates the failure of the heuristic: in $Layer_1$, D is a hub with strength 6; in $Layer_2$, D is not a hub and its strength 5 is ignored by the heuristic. This yields an estimated strength $\hat{s}(D) = 6 \leq 10.5$, causing the heuristic to exclude D. In reality, D's combined strength in the GT is $11 > 10.5$, making it a true GT hub.

This example shows that by ignoring contributions from layers where a node is not locally a hub, the Hubs-Only heuristic can fail to identify all true hubs in the aggregated network.

□

Consequently, the heuristic cannot guarantee complete recovery of the GT hub set. To balance efficiency and accuracy, one possible refinement is to expand the candidate set to include not only the layer-local hubs but also additional high-strength nodes, such as the top-k nodes by strength in each layer. The parameter k can be varied to observe how the inclusion of additional nodes affects the completeness of the recovered hub set. This hybrid strategy, formalized in Algorithm 2, increases the likelihood of identifying the ground truth hubs without analyzing the entire node set in the composition step.

Figures 6.9 through 6.11 show that using only layer-local hubs in the composition function yields slightly better results than the naïve approach but still falls short of the accuracy achieved with all-node evaluation. In particular, the Jaccard score improves over the naïve union but does not reach the perfect score obtained through full composition. As the percentage of overlap between layers increases, the Jaccard score also improves, indicating that layer similarity influences heuristic accuracy. In each figure, the X-axis shows edge overlap percentages (0%, 25%, 50%, 75%) grouped by edge distribution splits (50-50, 70-30, 90-10), while the Y-axis reports the Jaccard coefficient measuring similarity with the ground truth hub set.

Figure 6.12 presents the accuracy of three composition strategies, using all nodes, using only layer-local hubs, and the naïve approach, applied to real-world datasets. The all-node strategy achieves perfect accuracy, recovering the exact set of ground truth hubs. The hubs-only method performs nearly as well in the co-authorship network but shows reduced accuracy for the Ant Interaction network, likely due to ground truth hubs not identified as hubs in either individual layer. The naïve strategy, which relies on the union of layer-local hubs without strength evaluation, consistently underperforms, especially on smaller, denser networks. These results validate Lemma 3 and motivate the hybrid heuristic of Algorithm 2, which augments hubs with high-strength non-hub candidates to achieve higher accuracy while avoiding full all-node evaluation.

These results reinforce the limitations outlined in Lemma 3, where nodes that are not hubs in individual layers may still qualify as hubs in the aggregated network. To improve accuracy without incurring the full computational cost of all-node evaluation, we explore extended heuristics that expand the candidate set while retaining efficiency.

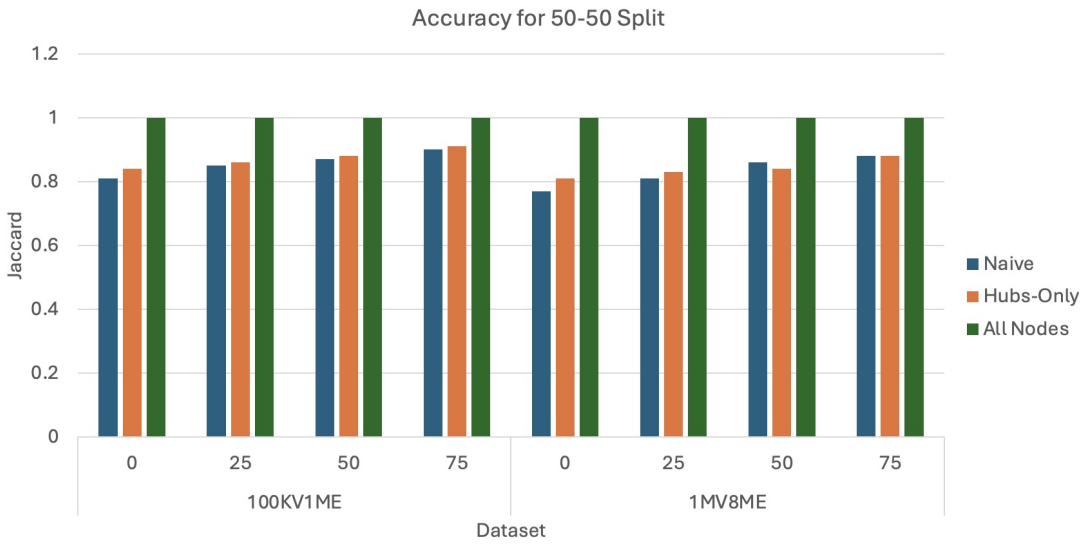


Figure 6.9: Accuracy for Sum Aggregation Using Hubs-Only Heuristic (50-50 Distribution)

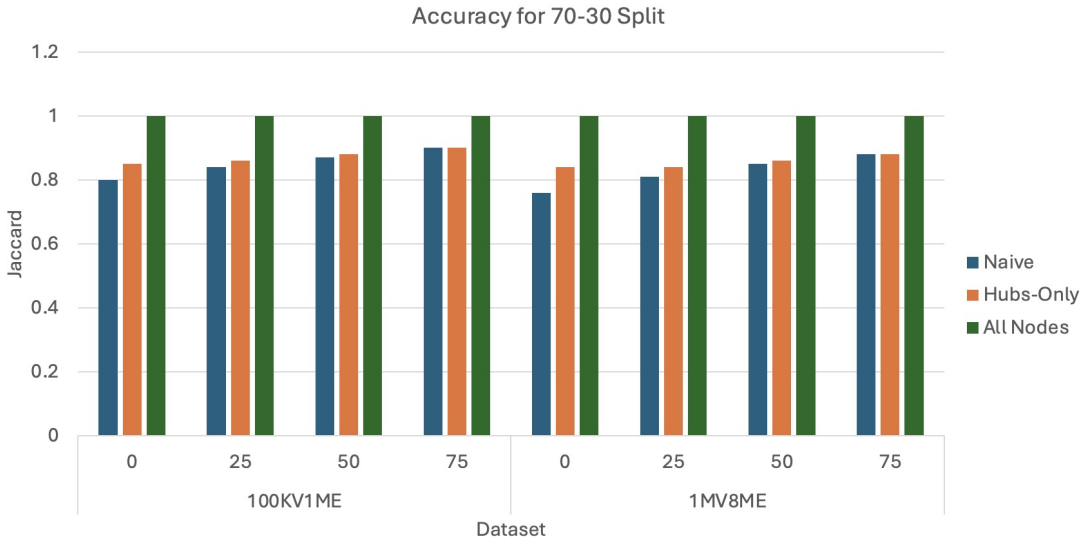


Figure 6.10: Accuracy for Sum Aggregation Using Hubs-Only Heuristic (70-30 Distribution)

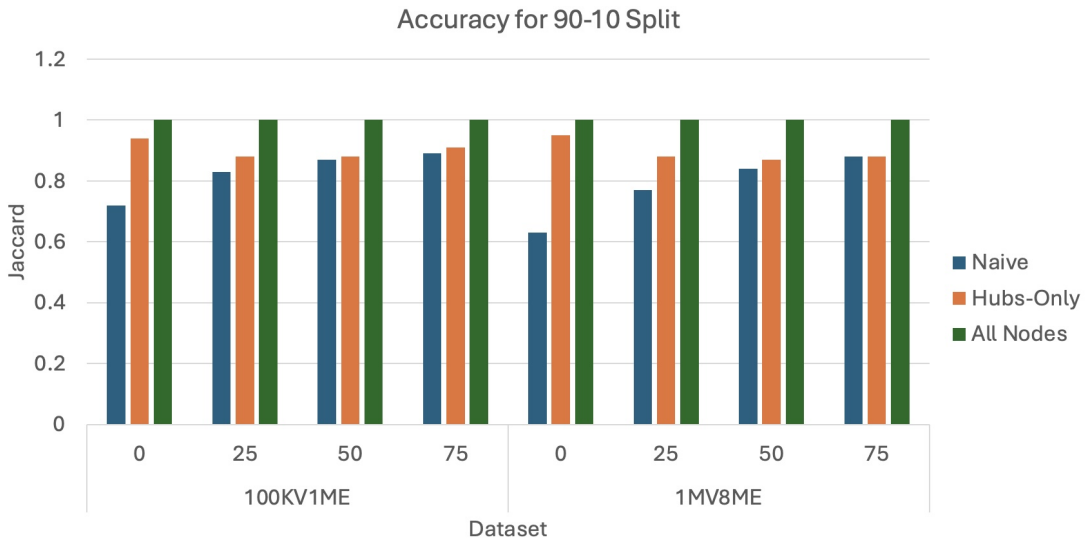


Figure 6.11: Accuracy for Sum Aggregation Using Hubs-Only Heuristic (90-10 Distribution)

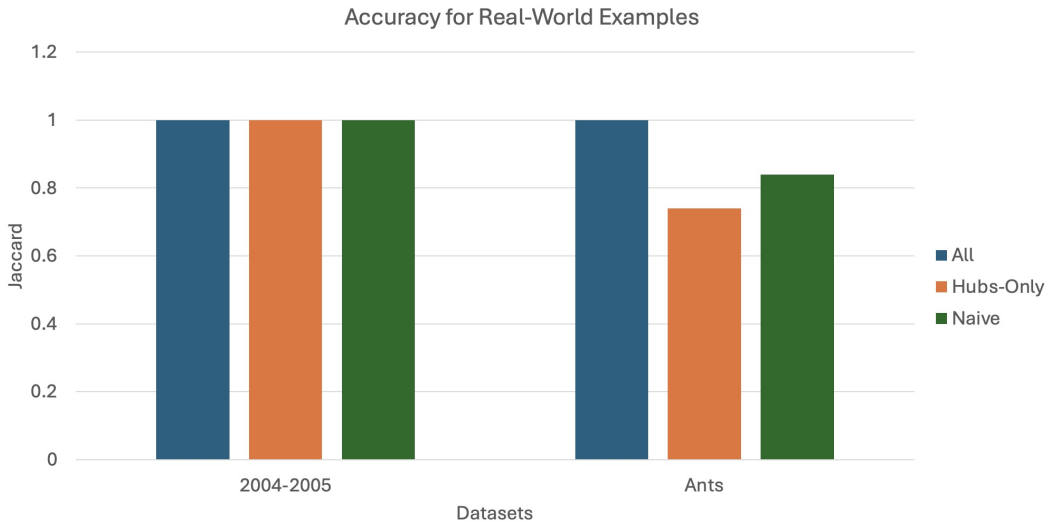


Figure 6.12: Accuracy on Real-World Datasets for All Nodes, Hubs-Only, and Naïve Strategies

Figures 6.13 through 6.15 demonstrate the effectiveness of the Hubs + Top-k heuristic in recovering the ground truth hubs across varying edge distribution splits (50-50, 70-30, and 90-10). In each figure, the X-axis indicates edge overlap levels (0%,

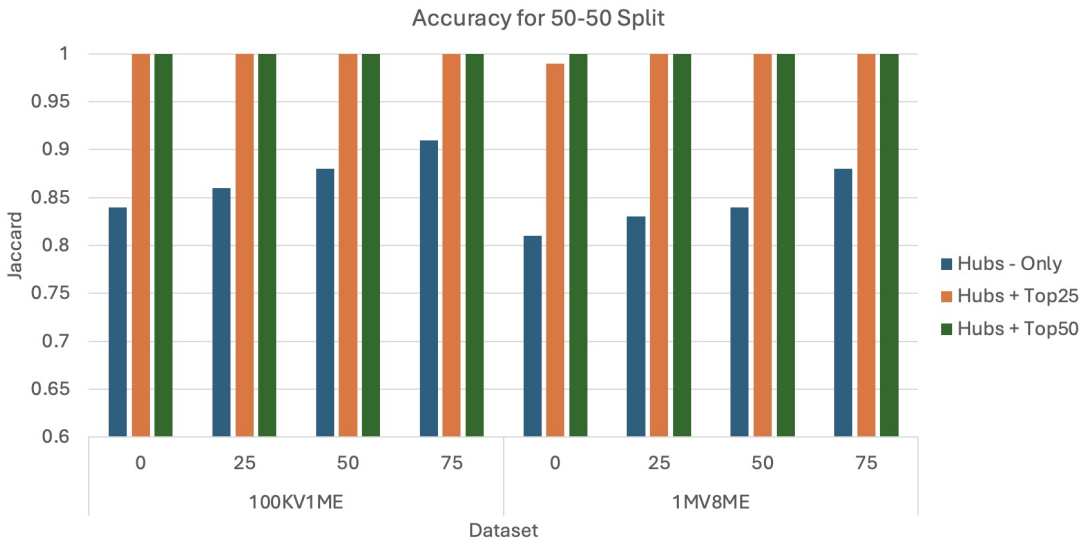


Figure 6.13: Accuracy for Sum Aggregation Using Hubs + Top-k Heuristic (50-50 Distribution)

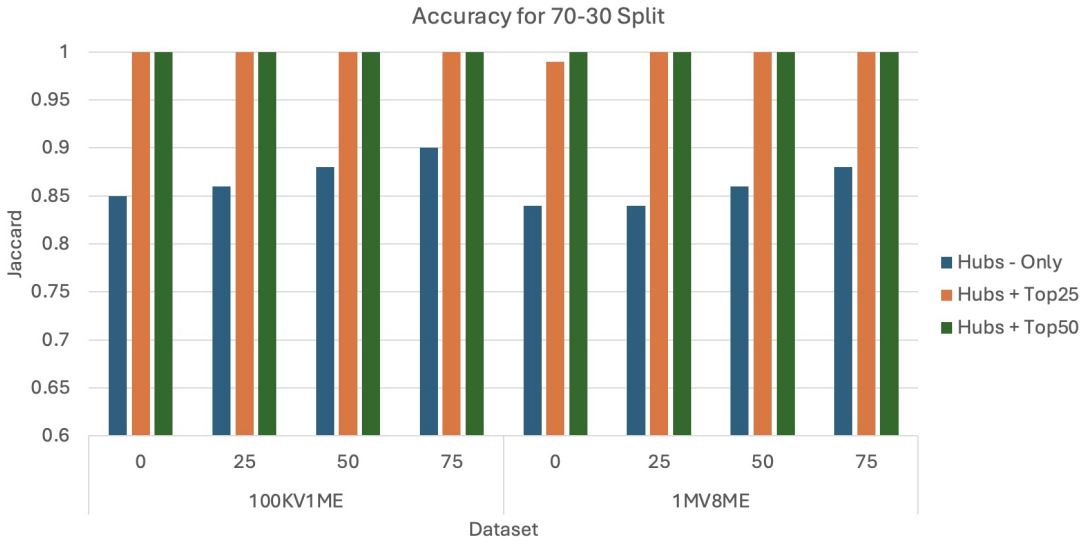


Figure 6.14: Accuracy for Sum Aggregation Using Hubs + Top-k Heuristic (70-30 Distribution)

Algorithm 2 Layer Analysis Function for Weighted HoMLNS - Hubs + Top-k (excluding hubs)

Input:

- $G = (V, E, w)$ *Layer graph with weighted edges*
- k = number of top-ranked non-hub nodes to include

Output: Set of Hubs and Top-k non-hub nodes

Initialize strength map: $s(u) \leftarrow 0$ for all $u \in V$
for each edge $(u, v) \in E$ **do**
    $s(u) \leftarrow s(u) + w(u, v)$
    $s(v) \leftarrow s(v) + w(u, v)$
end for
Compute average strength: $\bar{s} \leftarrow \frac{1}{|V|} \sum_{u \in V} s(u)$
$Hubs \leftarrow \{u \in V \mid s(u) > \bar{s}\}$
$Remaining \leftarrow V \setminus Hubs$
Sort nodes in $Remaining$ by $s(u)$ in descending order
$TopK \leftarrow$ first k nodes from sorted $Remaining$
return $(Hubs, TopK)$

25%, 50%, 75%) grouped by edge distribution split, and the Y-axis reports the Jaccard coefficient. As k increases, the Jaccard score improves steadily, reaching a perfect score of 1.0 in most cases. This shows that augmenting the layer-local hub set with even a modest number of high-strength non-hub nodes is often sufficient to recover the true hubs. Intuitively, nodes with moderate strength in each layer may fail to exceed the local hub threshold but surpass the global average when their strengths are aggregated. Including top-ranked non-hubs captures these borderline cases without requiring full-node evaluation. Higher layer overlap increases the likelihood of cap-

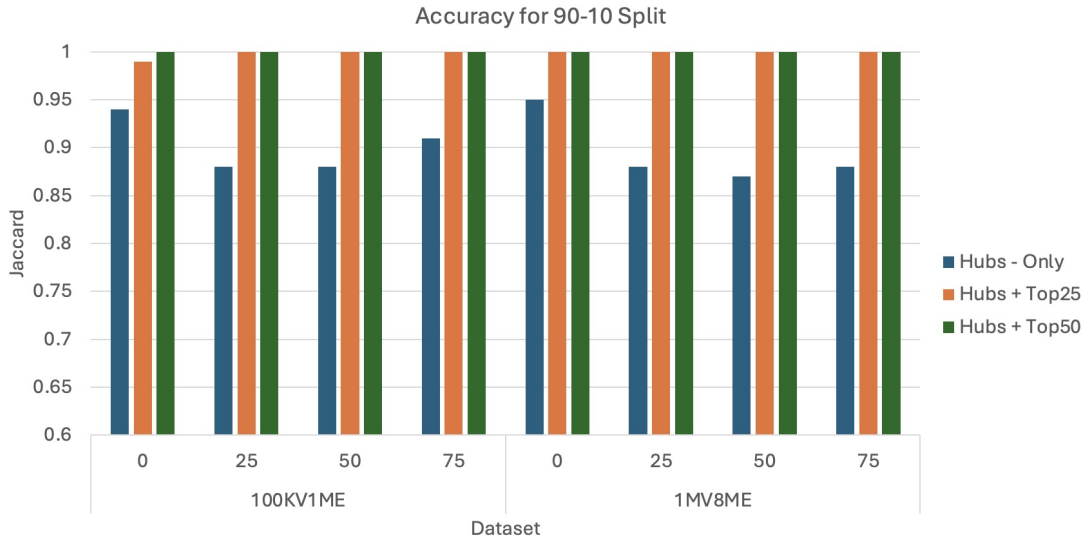


Figure 6.15: Accuracy for Sum Aggregation Using Hubs + Top-k Heuristic (90-10 Distribution)

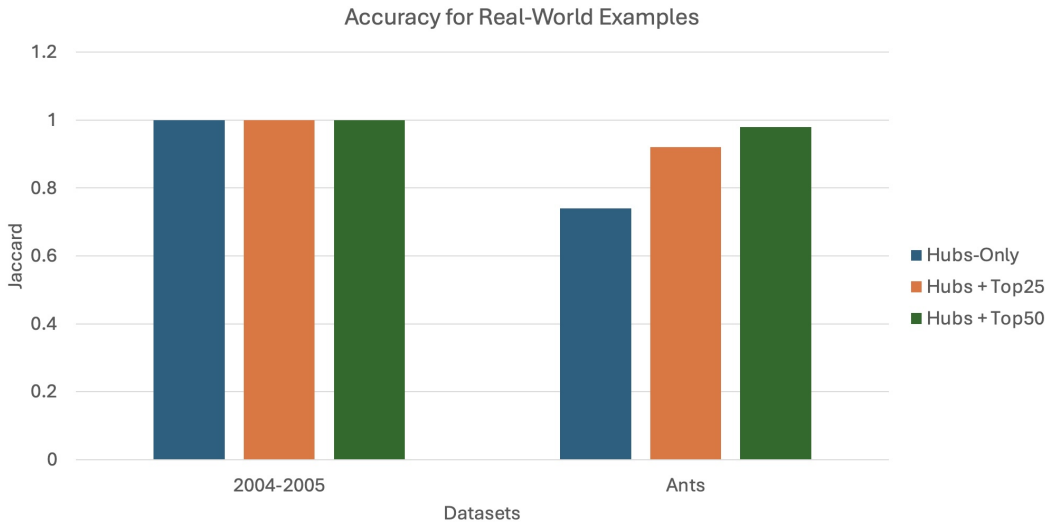


Figure 6.16: Accuracy on Real-World Datasets Using Hubs + Top-k Heuristic

turing such cumulative effects with fewer added nodes, enabling faster convergence to perfect accuracy. Figure 6.16 highlights the heuristic's performance on real-world datasets. For the co-authorship network, all three strategies, Hubs-Only, Hubs + Top-25, and Hubs + Top-50, achieve perfect accuracy, indicating that the most influential

nodes are already captured within the per-layer hubs or lie close in rank. In contrast, the ant interaction network benefits significantly from the inclusion of top-k non-hub nodes. While Hubs-Only underperforms, Hubs + Top-25 and Hubs + Top-50 show substantial gains, with the latter approaching perfect Jaccard similarity. This suggests that in smaller, denser graphs, where node influence is more evenly distributed, additional candidates are necessary to capture cumulative multilayer effects. These real-world results align with trends observed in synthetic networks: perfect recovery is often achieved with k=25 when layer overlap is non-zero, whereas datasets with 0% overlap may require k=50 to match full-node accuracy. This indicates that greater overlap between layers makes true hubs easier to identify with fewer additional candidates. Overall, the Hubs + Top-k approach strikes an effective balance between the Hubs-Only and full composition methods, achieving near-perfect accuracy with significantly fewer node evaluations. While Algorithm 2 narrows the candidate set considerably, the composition step still evaluates every node in this expanded set. In large-scale networks, especially with large k or dense hub sets, this can remain computationally costly. These refinements, from full-node composition to Hubs-Only, to Hubs + Top-k, form a progression of increasingly efficient heuristics for hub detection in weighted homogeneous multilayer networks. The next section presents a detailed runtime evaluation of these approaches, comparing their efficiency across synthetic datasets with varying sizes and edge distribution characteristics.

6.1.1 Runtime Evaluations for Aggregation Heuristic

Aggregating and analyzing the ground truth requires combining the layers into a single graph before computing the weighted degree centrality. This process involves summing the edge weights of overlapping edges across all layers, computing node

strengths, and the global average strength in order to identify hubs. This method, as mentioned earlier, is used as the baseline to compare against the proposed solutions.

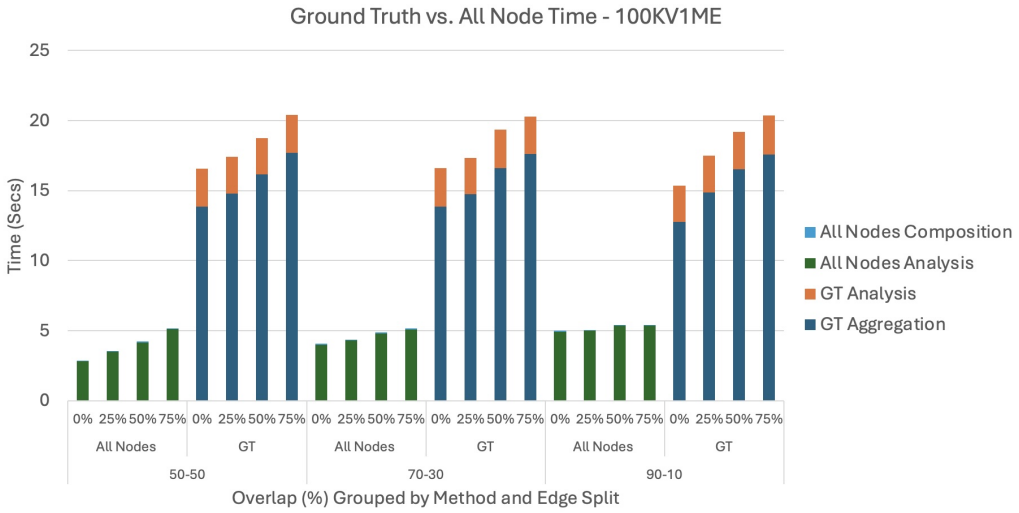


Figure 6.17: Ground Truth vs. All-Nodes Runtime for 100KV1ME Across Overlaps and Edge Splits

Figures 6.17 and 6.18 compare the runtime of the ground truth (GT) method against the All Nodes approach for synthetic datasets of different sizes. The X-axis represents the percentage of edge overlap between the two layers (0%, 25%, 50%, and 75%), while the Y-axis shows runtime in seconds. Across all configurations, GT consistently requires substantially more time than All Nodes, on the 1MV8ME dataset, GT takes roughly 4x longer, while on the smaller 100KV1ME dataset, the gap is smaller but still significant. This difference stems from GT's additional aggregation step, which merges edge sets before analysis, in contrast to All Nodes, which skips this step and directly computes strengths.

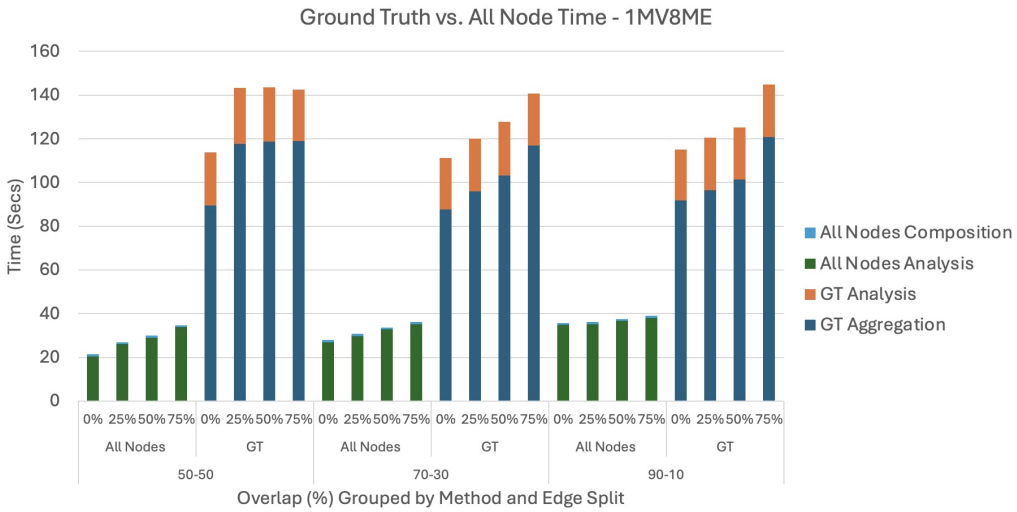


Figure 6.18: Ground Truth vs. All-Nodes Runtime for 1MV8ME Across Overlaps and Edge Splits

Dataset size has a clear impact: both methods scale up considerably from 100KV1ME to 1MV8ME. For example, GT aggregation alone increases from under 20 seconds for 100KV1ME to over 120 seconds for 1MV8ME. Overlap further amplifies GT time, at 0% overlap, aggregation is faster because the Boolean-OR operation is a simple union, while non-zero overlap requires identifying and merging duplicate edges, adding computational overhead.

The results in Table 6.7 show that while the real-world datasets are significantly smaller than the synthetic ones, the aggregation and analysis functions still incur measurable computational costs. For the co-authorship network, the aggregation step takes over one second due to the relatively large number of nodes and edges, whereas the Ant Interaction network requires only a fraction of a second for both aggregation and analysis. Notably, the analysis time for the co-authorship network is substantially higher than that of the ant network, despite both undergoing the same

computation steps. This increase may be attributed to the disconnected nature of the co-authorship graph, which can introduce overhead in computing node strengths or centrality, especially when the algorithm must iterate over many disconnected components. These findings further support the need for scalable heuristics that can reduce computation time without sacrificing accuracy, particularly when applied to larger real-world networks.

Dataset	**Aggregation (s)**	**Analysis (s)**
Co-authorship (2004–2005)	1.2017	2.3016
Ant Colony (Day 1 and 3)	0.0685	0.0067

Table 6.3: Runtime (in seconds) for Sum Ground Truth Computation on Real-World Datasets

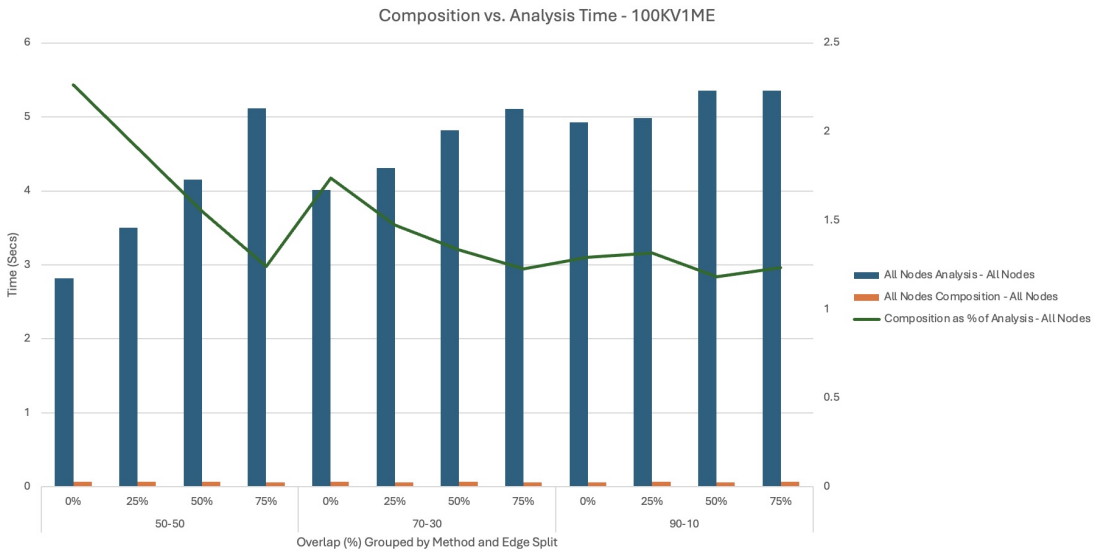


Figure 6.19: Composition vs. analysis time for the All-Nodes approach on the 100KV1ME dataset across overlaps and edge splits

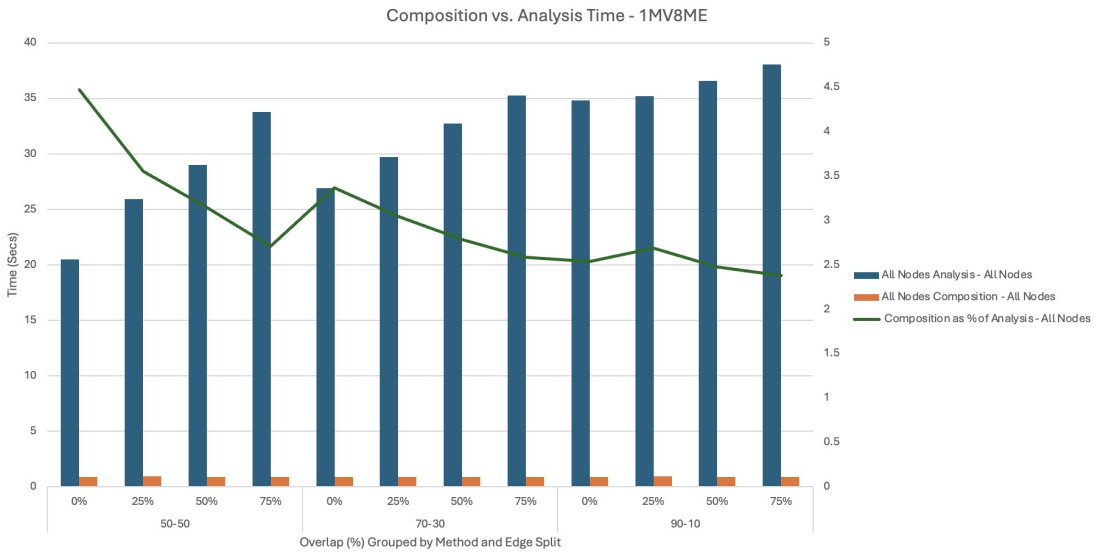


Figure 6.20: Composition vs. analysis time for the All-Nodes approach on the 1MV8ME dataset across overlaps and edge splits

Figures 6.19 and 6.20 break down the All Nodes runtime into analysis and composition functions. In all cases, composition time remains below 5% of the analysis time, well under one second for 100KV1ME and under two seconds for 1MV8ME, making it negligible in comparison to the total computation time. This indicates that for the All Nodes approach, the analysis function dominates overall runtime, and any efficiency improvements should focus on reducing analysis time rather than composition. Notably, composition time is relatively stable across different overlap percentages and edge distribution splits, with only minor fluctuations. This stability reflects the lightweight nature of the composition function for the All Nodes method: it involves a direct strength computation without the costly edge aggregation step required in GT. However, even though composition time is already small in the All Nodes baseline, further reductions can be valuable when working with extremely

large networks. These results motivate the exploration of more efficient heuristics. Figures 6.21 and 6.22 compare composition time for three strategies:

1. All Nodes – evaluates composition for every node in the network.
2. Hub-Only – restricts composition to nodes identified as hubs during layer analysis.
3. Naïve – takes the union of layer-specific hub sets without additional filtering.

The X-axis shows edge overlap grouped by distribution split; the Y-axis represents composition time in seconds. The naïve approach achieves the lowest runtime across all distributions and overlap levels by operating on a very small candidate set, but, as shown in the accuracy evaluation, this efficiency comes at the cost of substantially reduced accuracy, making it unsuitable when high-quality results are required. The All-Nodes method is the most computationally expensive but guarantees maximum accuracy. The hub-only heuristic offers a strong middle ground, significantly reducing composition time while retaining much of the accuracy, making it a practical, scalable alternative to exhaustive computation in large-scale multilayer networks.

Table 6.4 presents the composition time for the three heuristic strategies evaluated on the real-world datasets. As expected, the naïve method achieves the lowest runtime by simply taking the union of hub sets without filtering or additional computation. The hub-based approach offers a favorable balance, significantly reducing composition time compared to using all nodes, while still maintaining better accuracy than the naïve method. Interestingly, the differences in composition times are more pronounced in the co-authorship network, where the larger node set amplifies the computational savings of more selective strategies. For the small and densely connected Ant Interaction network, the runtime differences across heuristics are minimal,

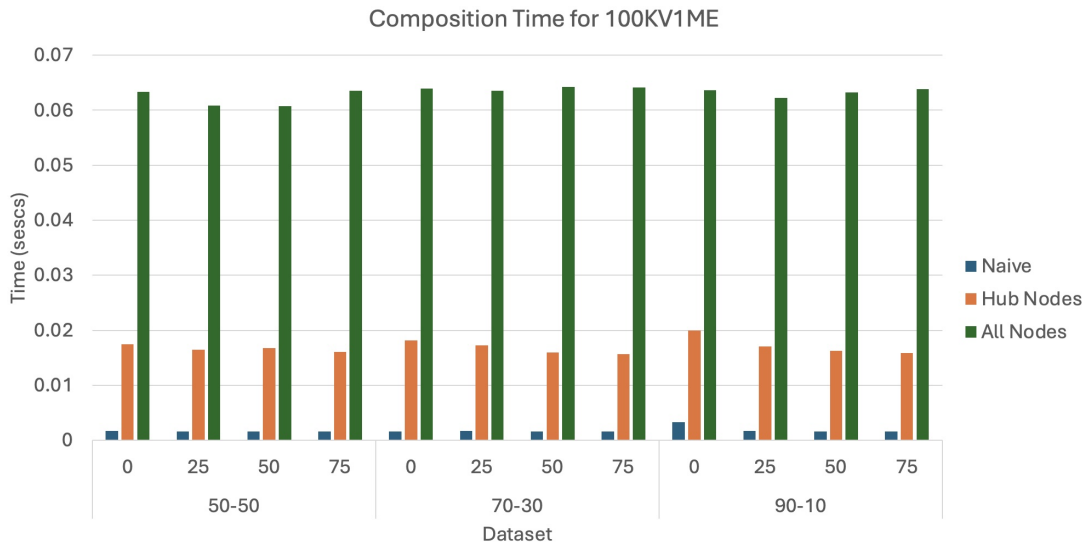


Figure 6.21: Composition Time Comparison for 100KV1ME Dataset: Naïve, Hub Nodes, and All Nodes

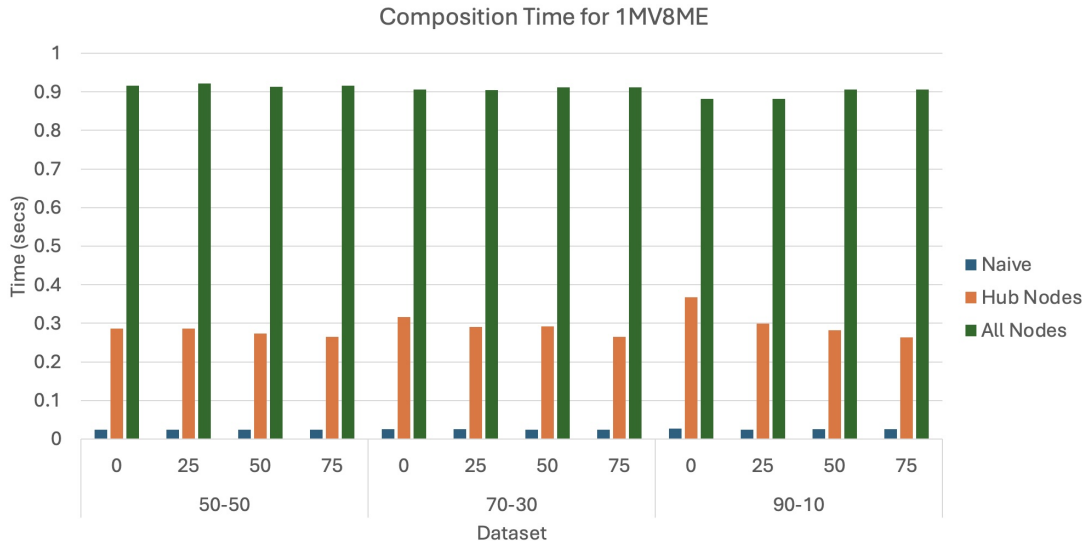


Figure 6.22: Composition Time Comparison for 1MV8ME Dataset: Naïve, Hub Nodes, and All Nodes

but remain consistent in trend. These results further reinforce the value of pruning candidate nodes before composition, especially in large or sparse real-world networks.

Dataset	All Nodes	Hub Nodes	Naïve
Co-authorship (2004–2005)	0.1547	0.0201	0.0025
Ant Colony (Day 1 and 3)	0.000331	0.000081	0.000007

Table 6.4: Composition Time (in seconds) for Sum Heuristic Strategies on Real-World Datasets

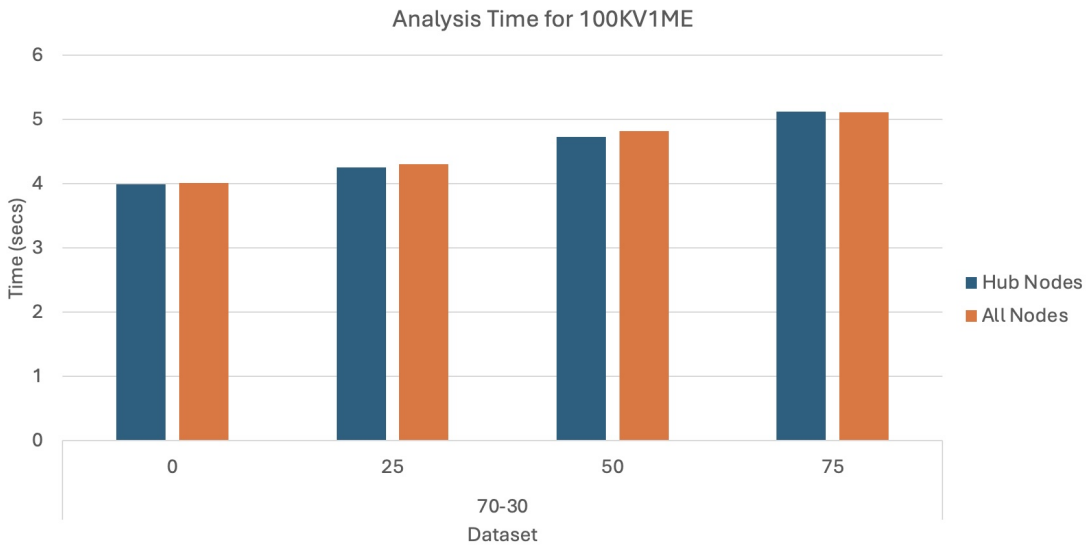


Figure 6.23: Analysis Time Comparison for 100KV1ME Dataset for 70-30 Split: Hubs and All Nodes Strategies

Figures 6.23 and 6.24 show the runtime performance of the analysis function for the 100KV1ME and 1MV8ME datasets, respectively, comparing two strategies: evaluating all nodes versus evaluating only the nodes identified as hubs in individual layers. The X-axis indicates edge overlap percentages (0%, 25%, 50%, 75%) under a fixed 70-30 edge distribution, while the Y-axis represents the total analysis time in seconds.

Because the analysis function must evaluate all nodes in order to compute their layer-specific strengths, the runtime differences between the All-Nodes and Hub-Only strategies are minimal. However, the Hub-Only strategy shows a small but consistent runtime benefit over the All-Nodes baseline. This gap is more pronounced

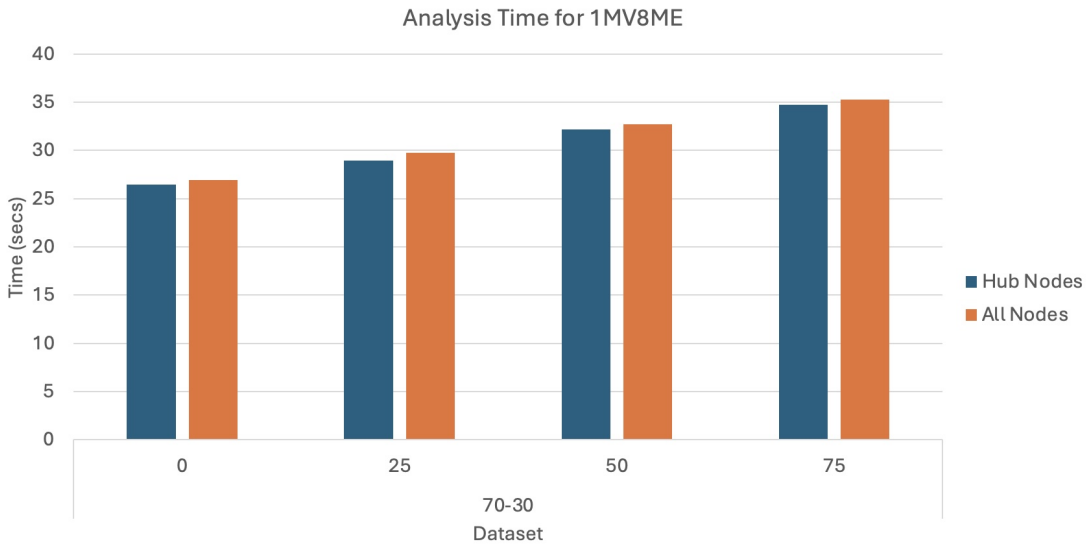


Figure 6.24: Analysis Time Comparison for 1MV8ME Dataset for 70-30 Split: Hubs and All Nodes Strategies

in the larger 1MV8ME dataset (Figure 6.24), where the analysis time increases from approximately 26 seconds at 0% overlap to over 35 seconds at 75% overlap. The All-Nodes approach incurs a similar increase, with slightly higher values throughout. In the smaller 100KV1ME dataset (Figure 6.23), analysis time ranges from roughly 4 to 5.1 seconds as overlap increases.

A key trend observed in both graphs is that the analysis time increases steadily with edge overlap, regardless of which node set is used. This is because higher overlap between the layers leads to more nodes gaining non-zero strength contributions from both layers, resulting in a larger active node set during analysis.

Because analysis time inevitably grows with overlap and all nodes must be processed, opportunities for optimizations at this stage are limited. This further motivated the need to find strategies to reduce the size of the node set that is carried forward into the composition function, where filtering can yield significant runtime savings.

To address this, we extend the Hubs-Only approach with the Hubs + Top-k strategy, which augments the hub set with the top-k highest-strength non-hub nodes for the layer. This method aims to enhance coverage while maintaining a bounded set of nodes for composition, thereby improving the trade-off between runtime and accuracy.

Figures 6.25 and 6.26 present the runtime results for three variants: Hubs only, Hubs + Top-25, and Hubs + Top-50. As expected, runtime increases slightly with the size of the augmented node set ($k = 25$ vs. $k = 50$), but both remain significantly more efficient than the All-Nodes approach. The X-axis shows edge overlap grouped by distribution split. The Y-axis indicates runtime in seconds.

Interestingly, even at $k = 50$, the runtime remains lower than the full-node composition while achieving accuracy nearly identical to the ground truth (as shown in the accuracy graphs). This highlights that a modest extension of the hub set can yield considerable gains in effectiveness without incurring high computational costs.

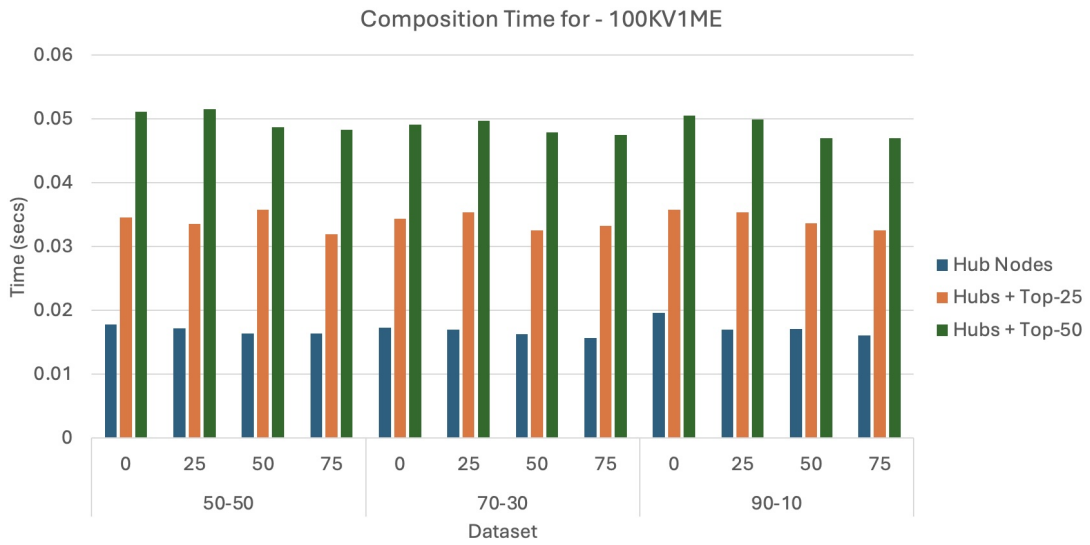


Figure 6.25: Composition Time Comparison for 100KV1ME Dataset: Hubs + Top-k Strategies

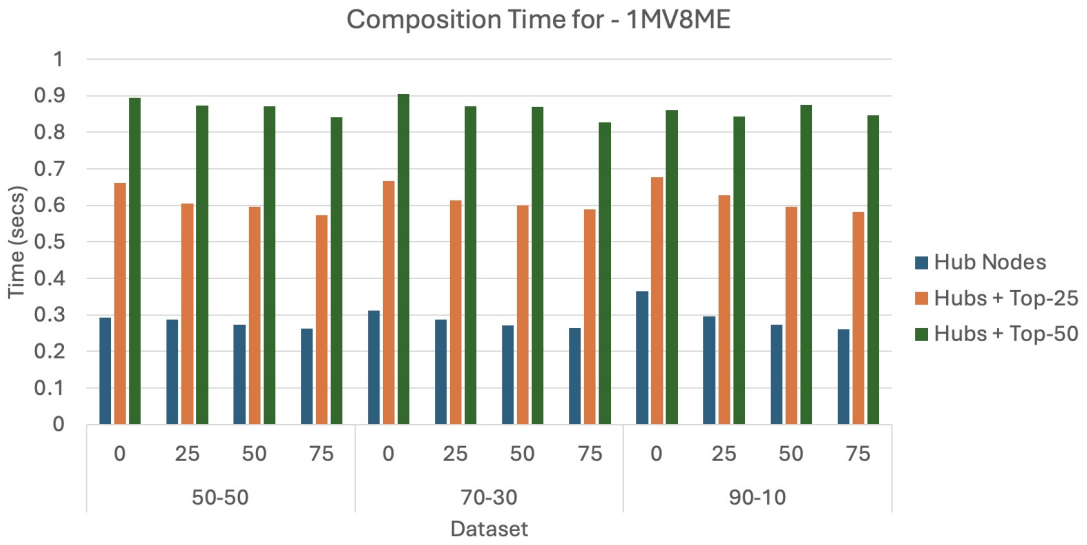


Figure 6.26: Composition Time Comparison for 1MV8ME Dataset: Hubs + Top-k Strategies

Dataset	**Hub Nodes**	**Hub + Top-25**	**Hub + Top-50**
Co-authorship (2004–2005)	0.0205	0.0563	0.0789
Ant Colony (Day 1 and 3)	0.00023	0.00024	0.00025

Table 6.5: Composition Time (in seconds) for Hubs + Top-k Strategies on Real-World Datasets

Table 6.5 shows the runtime performance of the Hubs + Top-k strategies, where a fixed number of top-ranked non-hub nodes are added to the hub set before composition. As expected, adding more non-hub nodes to the candidate set increases the composition time. For the co-authorship dataset, the inclusion of 25 and 50 top-ranked non-hubs more than doubles and triples the runtime compared to using only hubs. However, the runtime remains significantly lower than using all nodes. In the ant interaction dataset, the runtime differences are much smaller due to the limited node set, but the overall trend remains. These results indicate that while the Top-k

strategy introduces some additional overhead, it remains efficient in practice and offers a tunable trade-off between runtime and result quality, particularly when more context-aware nodes are desired beyond the core hubs.

Overall, the Hubs + Top-k strategy demonstrates a clear advantage in runtime scalability. It avoids the inefficiencies of evaluating the entire node set while compensating for the accuracy limitations of hub-only methods. As the dataset size increases (from 100KV1ME to 1MV8ME), these advantages become even more pronounced, confirming the utility of the proposed heuristic for large-scale multilayer networks.

The runtime evaluations reveal clear trade-offs between accuracy and efficiency across various hub detection strategies in weighted homogeneous multilayer networks. While full-node composition ensures perfect accuracy, it incurs high computational costs, especially as dataset size and layer complexity increase. Heuristic-based approaches, such as the naïve method or hubs-only composition, reduce runtime significantly but at the expense of accuracy. The Hubs + Top-k strategy offers an effective middle ground, achieving near-optimal accuracy with substantially reduced computation time, even for large networks.

These results collectively demonstrate that scalable and accurate hub detection is possible without exhaustive computation, provided that the heuristic is carefully designed and implemented.

In summary, the evaluation of sum aggregation heuristics demonstrates that carefully pruning the candidate node set can achieve near-perfect hub recovery while significantly reducing computation time compared to the All-Nodes baseline. Strategies such as Hubs-Only and Hubs augmented with top-ranked nodes strike an effective balance between accuracy and runtime in the additive sum setting. The next section turns to the more complex max aggregation method, where edge overlap introduces additional uncertainty and poses greater challenges for accurate strength estimation.

6.2 Weighted Degree Centrality Using Max Aggregation

Another aggregation method discussed earlier is the *max aggregation*, where the weight of each edge in the ground truth network is determined by taking the maximum weight among all corresponding edges across layers. Specifically, if an edge appears in both layers, only the highest weight is retained in the aggregated network. This contrasts with sum aggregation, where weights are accumulated regardless of overlap.

As before, we adopt a decoupling-based framework that separates the computation into two stages: an *analysis function*, which processes each individual layer independently, and a *composition function*, which estimates the global network properties by merging the outputs of the analysis function using heuristics.

Analysis Function: The analysis function is applied to a single layer of the HoMLN and operates solely on the nodes and weighted edges present in that layer. For each node, it computes its strength by summing the weights of all incident edges. These per-layer strength values serve as inputs to the composition stage.

Composition Function: Unlike sum aggregation, max aggregation poses unique challenges when analyzing multilayer networks in a decoupled manner. One of the central challenges in max aggregation is the uncertainty introduced by edge overlap across layers. When the same edge appears in multiple layers, the aggregated weight must be determined by taking the maximum of the edge weights across those layers. As a result, the final strength of a node depends not only on the magnitude of edge weights but also on the degree of edge overlap across layers.

However, the decoupling-based framework deliberately avoids accessing inter-layer edge relationships during analysis. Each layer is processed independently, and no information is retained about whether specific edges are shared across layers or are

unique to one layer. Consequently, it is not possible to compute the exact max-based strength of a node without reconstructing the full aggregated graph.

To address this limitation, we estimate a node's strength in the ground truth using upper and lower bounds. The *lower bound* assumes complete edge overlap: all of a node's edges appear in both layers and connect the same neighbors. Under this conservative assumption, the ground truth strength is estimated as the maximum of its per-layer strengths:

$$\hat{s}_{lower}(u) = \max\left(s_x(u),\ s_y(u)\right) \tag{6.4}$$

In contrast, the *upper bound* assumes there is no edge overlap between the layers, i.e., all edges in one layer are entirely distinct from those in the other. This assumption yields the most optimistic estimate, treating every edge from both layers as contributing independently to the node's total strength:

$$\hat{s}_{upper}(u) = s_x(u) + s_y(u) \tag{6.5}$$

Since the true level of edge overlap is unknown and lies somewhere between these two extremes, the actual strength of a node in the aggregated graph will fall within this range:

$$\hat{s}_{lower}(u) \leq s_{GT}(u) \leq \hat{s}_{upper}(u)$$

In this work, we focus on the **upper bound** as it provides a conservative estimate that ensures nodes with potentially high strength are included in the analysis. While it may overestimate the true strength when overlap exists, this approach avoids missing important nodes due to underestimation. The use of the upper bound thereby preserves the benefits of scalability and modularity offered by the decoupling framework, while still offering a meaningful approximation for max-based strength estimation.

While the upper and lower bounds provide practical approximations, it is important to recognize that neither can recover the true strength of a node unless edge-level overlap information is available. This limitation arises directly from the independence of the per-layer analysis in the decoupling framework. The following lemma formalizes this inherent restriction on exact strength recovery under max aggregation.

Lemma 4. *In a decoupling-based framework with max aggregation, exact reconstruction of a node's true strength in the ground truth graph is not guaranteed unless edge-level overlap across layers is known.*

Proof. Let $G = \{\text{Layer}_x, \text{Layer}_y\}$ be a homogeneous multilayer network consisting of two layers defined over the same node set V. Each layer contains a set of weighted edges. Let $u \in V$ be a node in the network. For each layer:

- $E_x(u)$ and $E_y(u)$ denote the sets of edges incident to node u in Layer x and Layer y, respectively.
- $w_x(u, v)$ and $w_y(u, v)$ represent the weights of the edge (u, v) in Layer x and Layer y, respectively.
- $N(u)$ denotes the set of neighbors of u in the aggregated graph (i.e., nodes connected to u in either layer).

In the ground truth network constructed using max aggregation, the weight of each edge (u, v) is given by:

$$w_{GT}(u, v) = \max\left(w_x(u, v),\ w_y(u, v)\right) \tag{6.6}$$

Accordingly, the true strength of node u in the ground truth graph is the sum of the maximum weights across all its neighbors:

$$s_{GT}(u) = \sum_{v \in N(u)} w_{GT}(u, v) = \sum_{v \in N(u)} \max\left(w_x(u, v),\ w_y(u, v)\right) \tag{6.7}$$

To compute $s_{GT}(u)$ exactly, it is necessary to:

1. Determine whether each edge (u, v) appears in both layers (i.e., identify overlapping edges), and
2. Compare the weights from both layers to select the maximum.

However, in the decoupling-based approach, each layer is analyzed independently. The analysis function:

- Computes and stores the total strength of each node per layer (e.g., $s_x(u) = \sum_{v \in E_x(u)} w_x(u, v)$), and
- Does not retain information about which specific edges are present, or whether they overlap with edges in the other layer.

As a result, the composition function lacks the necessary edge-level data to evaluate Equation (6.7) directly. Without this information, any estimation (such as upper or lower bounds) can only approximate the true strength.

Therefore, exact reconstruction of a node's strength under max aggregation is not achievable using the decoupling framework alone. □

Given this limitation, we adopt the upper bound approximation as a practical and conservative strategy for estimating node strength under max aggregation. By summing the strength of each node across both layers, we ensure that nodes with potentially high influence are not excluded due to unknown edge overlap.

The following algorithm outlines the composition step using this upper bound heuristic. It computes the estimated strength for each node by combining its values from both layers, determines the global average strength using precomputed totals from the layer headers, and returns all nodes whose estimated strength exceeds this global average.

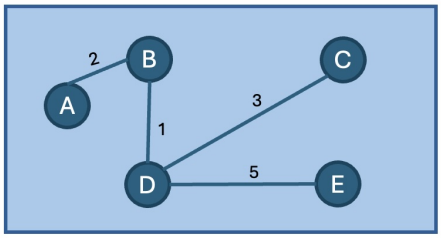
Algorithm 3 Composition Algorithm for Weighted Degree Centrality Using Max Aggregation (Upper Bound)

- $Layer_x = \{(u, s_x(u)) \mid u \in V\}$ *Nodes & Strengths from* $Layer_x$
- $Layer_y = \{(u, s_y(u)) \mid u \in V\}$ *Nodes & Strengths from* $Layer_y$
- $n = |V|$ *Number of nodes*
- $\texttt{sum}_x = \sum_{u \in V} s_x(u)$
- $\texttt{sum}_y = \sum_{u \in V} s_y(u)$

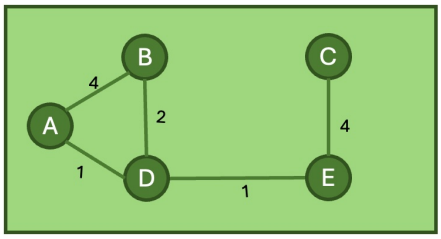
$\bar{s}_{\text{UB}} \leftarrow (\texttt{sum\_x} + \texttt{sum\_y})/n$ ▷ Upper bound on GT average under max
Hubs $\leftarrow [\,]$
for each node $u \in V$ **do**
    $\hat{s}(u) \leftarrow s_x(u) + s_y(u)$ ▷ Upper bound since $\max\{s_x, s_y\} \leq s_x + s_y$
    if $\hat{s}(u) > \bar{s}_{\text{UB}}$ **then**
        Hubs.append(u)
    end if
end for
return *Hubs*

To illustrate how max aggregation and its heuristics behave in practice, we consider the following example based on real ant interaction data collected over two days.

Consider a two-layer weighted HoMLN, shown in figure 6.27, where each layer represents ant interactions on two different days. Each layer is independently analyzed using the decoupling approach, and the weighted-degree hubs are computed based on the node strengths within that layer.



Day 1 (G_1)



Day 2 (G_2)

Figure 6.27: Layer-wise weighted edges in the multilayer network.

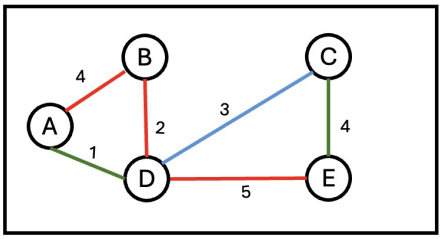


Figure 6.28: Ground truth network (G_{GT}) using Boolean-OR and Max aggregation. (Red edges denote overlapping edges that appear in both layers.)

Node	Day 1 (Layer 1)	Day 2 (Layer 2)	Ground Truth (Max)	UB Estimate
A	2	5	5	7
B	3	6	6	9
C	3	4	7	7
D	9	4	11	13
E	5	5	9	10
Avg	4.4	4.8	7.6	9.2

Table 6.6: Node Strengths in Day 1, Day 2, Max Aggregated Ground Truth, and Max Heuristic Estimate

Table 6.6 summarizes the node strengths in Day 1 and Day 2, the ground truth using max aggregation, and the estimated strengths under the upper-bound heuristic. In Day 1, nodes **D** and **E** are identified as hubs, as their strengths exceed the average of 4.4. In Day 2, nodes **A**, **B**, and **E** are identified as hubs, based on an average of

4.8. The **naïve composition**, which merges the hubs from both layers, yields the set {**A**, **B**, **D**, **E**}.

By contrast, the ground truth computed using max aggregation (Figure 6.28) identifies the hub set {**D** and **E**}, based on a global strength average of 7.6. The **upper-bound heuristic**, which estimates node strength by summing values from both layers, identifies the same set {**D**, **E**} using a slightly higher average of 9.2. This comparison highlights two key observations: (1) the naïve method includes nodes that do not qualify as hubs in the aggregated network (e.g., **A**, **B**), and (2) the upper-bound heuristic successfully recovers all ground truth hubs, despite operating without edge-level overlap information. This demonstrates that the heuristic provides a more precise and scalable alternative to naïve hub merging.

To validate these observations more broadly, we evaluate the upper-bound heuristic using the same set of synthetic and real-world multilayer networks previously used for sum aggregation. These results examine how well the upper-bound heuristic approximates the ground truth in practice, and how its performance varies with network characteristics such as edge distribution, overlap, and size.

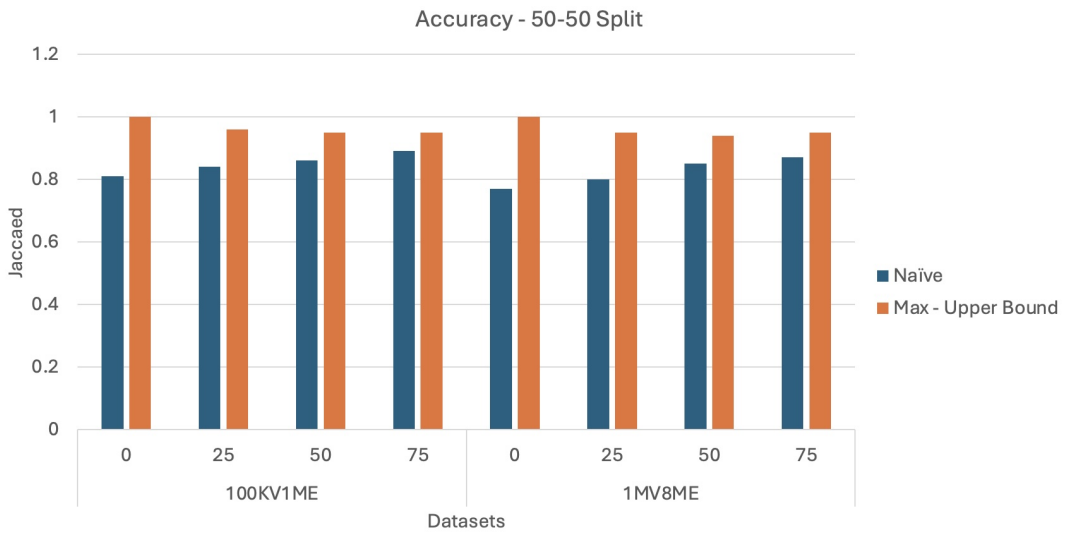


Figure 6.29: Accuracy of Naïve and Upper-Bound Heuristic under Max Aggregation (50-50 Split)

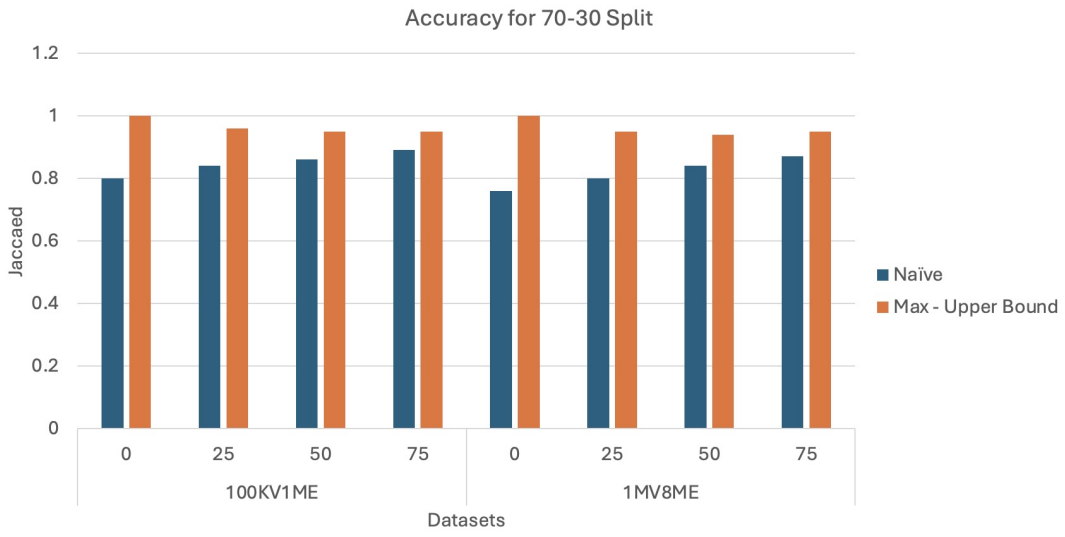


Figure 6.30: Accuracy of Naïve and Upper-Bound Heuristic under Max Aggregation (70-30 Split)

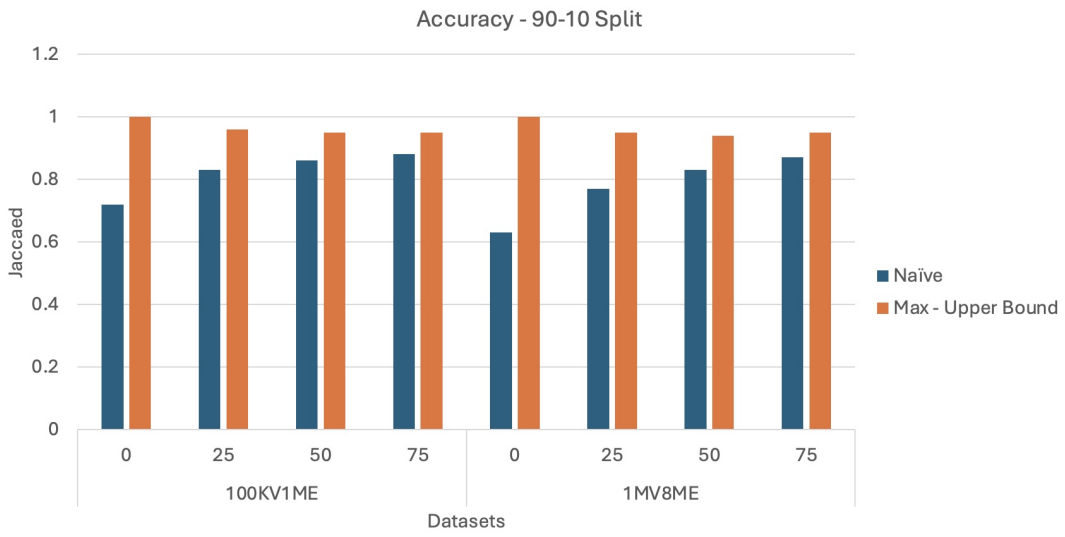


Figure 6.31: Accuracy of Naïve and Upper-Bound Heuristic under Max Aggregation (90-10 Split)

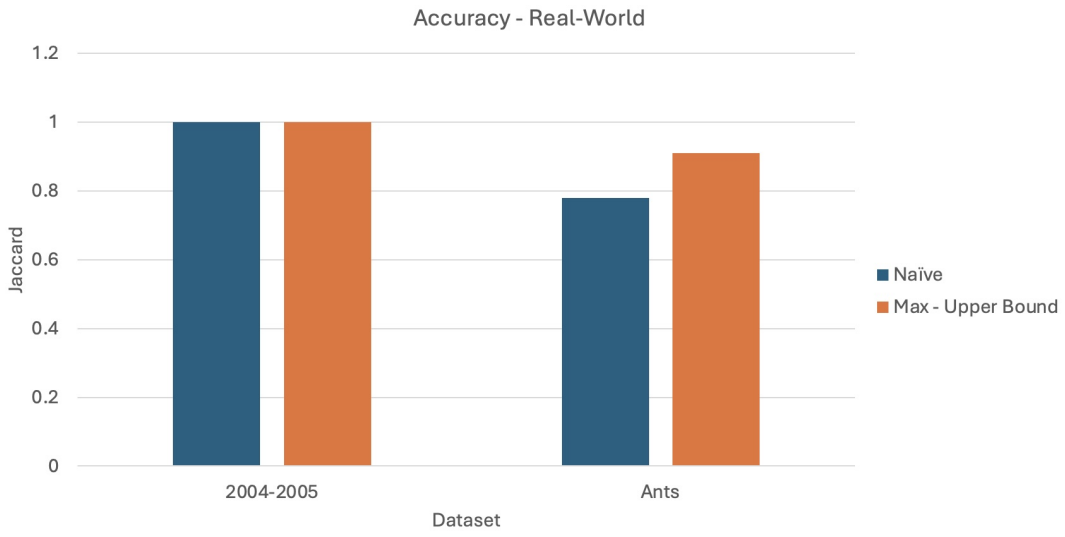


Figure 6.32: Accuracy of Naïve and Upper-Bound Heuristic under Max Aggregation (Real-World)

Figures 6.29, 6.30, and 6.31 present the Jaccard accuracy of the naïve and upper-bound heuristic approaches under max aggregation across synthetic datasets with edge distribution splits of 50-50, 70-30, and 90-10. Each figure includes results for four levels of edge overlap (0%, 25%, 50%, and 75%) across two datasets: `100KV1ME` and `1MV8ME`. The X-axis shows edge overlap percentages (0%, 25%, 50%, 75%) grouped by edge distribution split. The Y-axis shows the Jaccard coefficient comparing the heuristic hub set to the ground truth hub set.

Across all configurations, the upper-bound heuristic consistently outperforms the naïve approach, achieving the highest accuracy and more closely approximating the ground truth. Its accuracy is perfect or near-perfect at 0% overlap, where its assumption of disjoint edge sets holds exactly, and only declines slightly as overlap increases due to overestimation caused by repeated edge contributions.

In contrast, the naïve method performs relatively poorly when overlap is minimal, as it merges hubs from each layer without accounting for their actual combined strength in the aggregated graph. However, its accuracy improves as overlap increases, since nodes that are identified as hubs in both layers are increasingly likely to also qualify as hubs in the max-aggregated ground truth. Despite this improvement, the naïve method consistently lags behind the upper-bound heuristic, never matching its level of accuracy, even at 75% overlap.

These patterns are further confirmed by the real-world datasets, shown in Figure 6.32. In the co-authorship network (2004–2005), both the naïve and upper-bound methods achieve perfect accuracy, likely due to strong semantic alignment and substantial hub overlap across years. However, in the ant colony interaction dataset, the naïve method performs notably worse than the upper bound, suggesting structural divergence between the layers. The upper-bound heuristic, by conservatively aggregating strength across layers, remains more faithful to the true set of hubs. The

X-axis shows the datasets (Co-authorship and Ant Colony) and heuristic type (Naïve and Upper-Bound). The Y-axis is the Jaccard coefficient comparing the predicted hubs to the ground truth.

Taken together, these results demonstrate that while edge overlap may partially mitigate the limitations of the naïve approach, the upper-bound heuristic consistently provides a more accurate and robust estimation of node strength under max aggregation. This advantage holds across synthetic and real-world datasets, making it a preferable choice for decoupled multilayer network analysis.

6.2.1 Runtime Evaluation for Max Aggregation Heuristics

Aggregating and analyzing the ground truth under max aggregation requires combining the layers into a single graph by retaining the maximum weight for each overlapping edge. Once aggregated, node strengths are computed based on the resulting edge set, followed by the calculation of the global average strength to identify hubs. This full aggregation process serves as the baseline against which the runtimes of the proposed heuristic approaches are compared.

Figures 6.33 and 6.34 compare the runtime of the ground truth (GT) method under max aggregation with the Max - Upper Bound (Sum of Strengths) approach for the `100KV1ME` and `1MV8ME` datasets. The X-axis shows the percentage of edge overlap between layers (0%, 25%, 50%, 75%), grouped by edge distribution splits (50-50, 70-30, 90-10), while the Y-axis reports runtime in seconds.

Across all configurations, GT consistently takes substantially longer than the Max - Upper Bound (Sum of Strengths) approach. On the `1MV8ME` dataset, GT is roughly 4–5$\times$ slower, while on the smaller `100KV1ME` dataset, the gap is smaller but still significant. This difference arises from GT's additional aggregation step, which

reconciles overlapping edges by selecting maximum weights, in contrast to the Max – Upper Bound (Sum of Strengths) method that avoids this costly operation.

Dataset size has a strong effect: GT aggregation alone rises from under 20 seconds for `100KV1ME` to over 120 seconds for `1MV8ME`. Overlap further amplifies runtime, at 0% overlap, aggregation is faster because edges can be merged directly via a union, while non-zero overlap requires identifying and resolving duplicate edges, adding significant computational overhead.

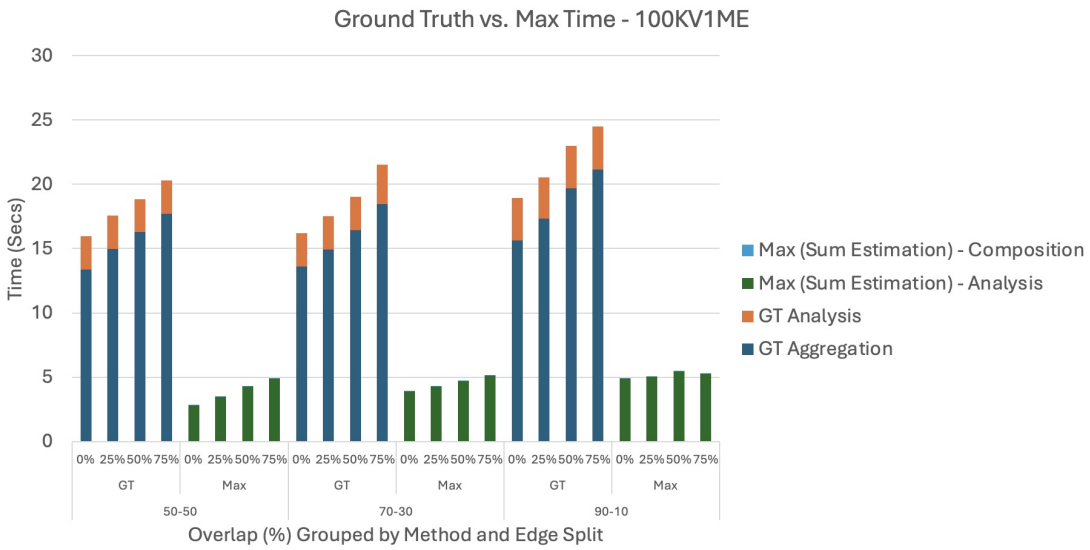


Figure 6.33: Ground Truth Runtime for Max Aggregation - 100KV1ME Dataset Across Distribution Splits and Overlap Percentages

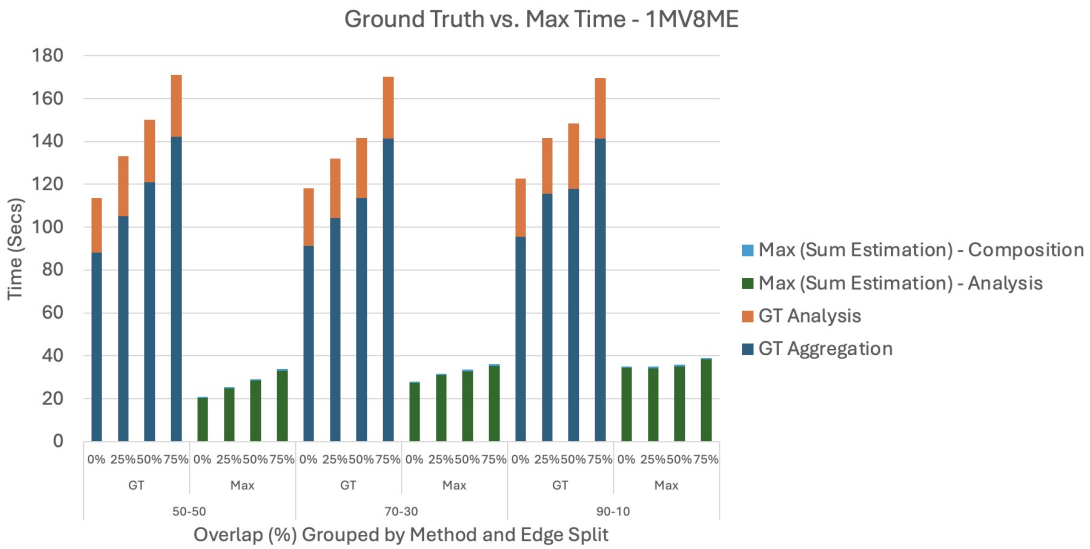


Figure 6.34: Ground Truth Runtime for Max Aggregation - 1MV8ME Dataset Across Distribution Splits and Overlap Percentages

To complement the synthetic dataset results, Table 6.7 presents the runtime breakdown for ground truth computation using max aggregation on two real-world datasets: the co-authorship network (2004–2005) and the ant colony interaction network (Day 1 and Day 3). Similar to the synthetic datasets, layer aggregation accounts for a substantial portion of the total runtime, about 34% in the co-authorship dataset. However, the absolute runtimes are significantly smaller, reflecting the much lower graph size and sparsity in the real-world networks compared to the synthetic benchmarks.

In the ant colony dataset, the aggregation step is extremely lightweight (under 0.1 seconds), and the analysis function is nearly negligible. This contrast highlights how dataset scale and density critically affect the cost of exact ground truth computation under max aggregation, further motivating the need for efficient heuristic approximations in large-scale settings.

Dataset	Aggregation (s)	Analysis (s)
Co-authorship (2004–2005)	1.1779	2.2863
Ant Colony (Day 1 and 3)	0.0703	0.0063

Table 6.7: Runtime (in seconds) for Max Ground Truth Computation on Real-World Datasets

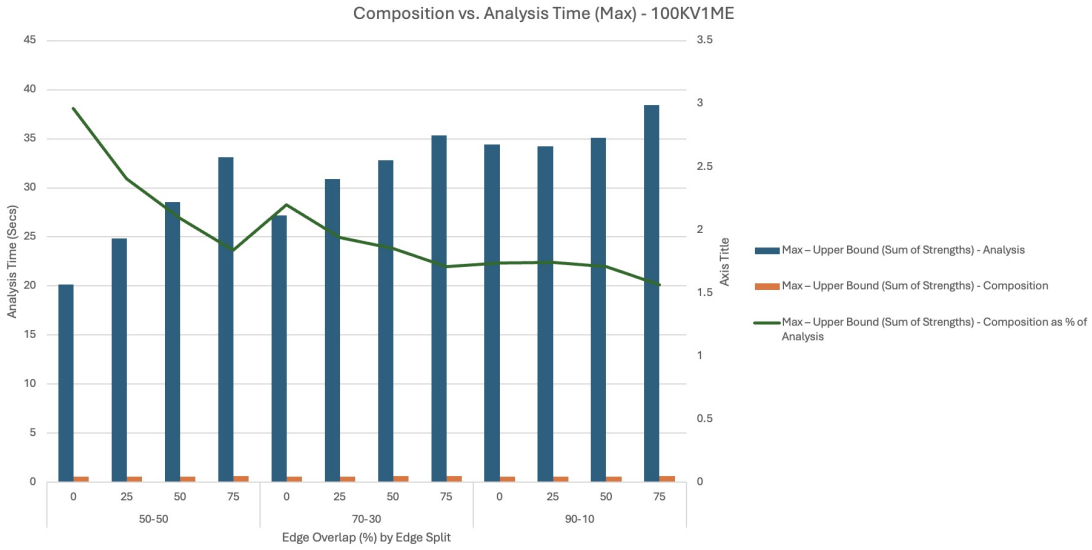


Figure 6.35: Composition vs. analysis time for the Max – Upper Bound (Sum of Strengths) approach on the 100KV1ME dataset across overlaps and edge splits

Figures 6.35 and 6.36 break down the runtime of the Max – Upper Bound (Sum of Strengths) method into its analysis and composition functions for the `100KV1ME` and `1MV8ME` datasets. In all cases, composition time remains extremely small, consistently below 2% of the analysis time and well under one second for `100KV1ME` and under two seconds for `1MV8ME`.

Composition time is stable across overlap percentages and edge distribution splits, with only minimal variation. This consistency reflects the lightweight nature of the composition function for the Max – Upper Bound (Sum of Strengths) method, which only merges per-layer strength values without performing the expensive edge aggregation step required in GT. Although already negligible, further reductions in

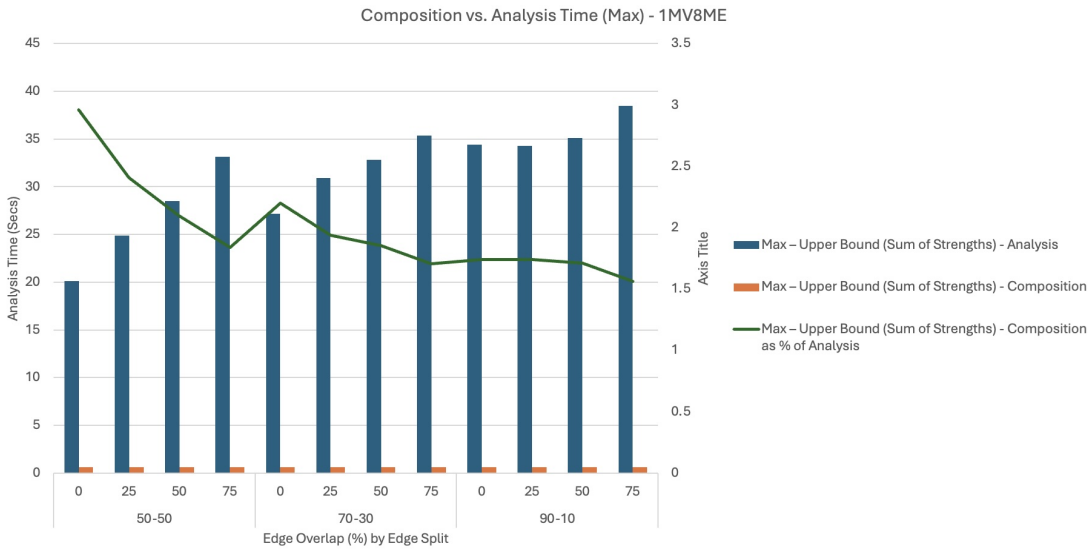


Figure 6.36: Composition vs. analysis time for the Max - Upper Bound (Sum of Strengths) approach on the 1MV8ME dataset across overlaps and edge splits

composition time may be useful in extremely large-scale scenarios where even sub-second operations accumulate.

Figures 6.37 and 6.38 report the composition runtime for the naïve and upper-bound heuristics under max aggregation for the `100KV1ME` and `1MV8ME` datasets. The X-axis reflects different dataset configurations, denoted by edge distribution ratios (50-50, 70-30, 90-10) and edge overlap percentages (0%, 25%, 50%, 75%). The Y-axis shows the time in seconds required to execute the composition step. Across all configurations, the upper-bound heuristic requires noticeably more time than the naïve strategy. This is expected, as the upper-bound method processes all nodes and performs strength evaluations, while the naïve method only combines hubs detected in individual layers without additional computation.

Despite its higher runtime, the upper-bound heuristic remains computationally efficient, completing in under 0.7 seconds even for the largest dataset (`1MV8ME`) across all edge distributions and overlap settings. This modest increase in runtime is justified

by the substantial accuracy improvements observed earlier. In contrast, the naïve strategy consistently completes in under 0.03 seconds for all configurations, with its highest runtime observed in the `100KV1ME` dataset (90-10 split, 0% overlap), yet still remaining low.

A comparative analysis across datasets reveals that the composition time, particularly for the naïve approach, is directly influenced by the size of the hub union set. For example, in the `100KV1ME` dataset with a 90-10 edge distribution and 0% overlap, the naïve method identified 24,063 and 20,977 hubs from the two layers, resulting in a large union of 29,024 nodes and incurring the highest runtime for this method. In contrast, another configuration with similar layer-local hub counts (21,050 and 21,506) but higher overlap produced a smaller union of 24,141 nodes, reducing the time required for composition. This trend confirms that the runtime is influenced not only by the number of hubs in each layer but also by their degree of overlap.

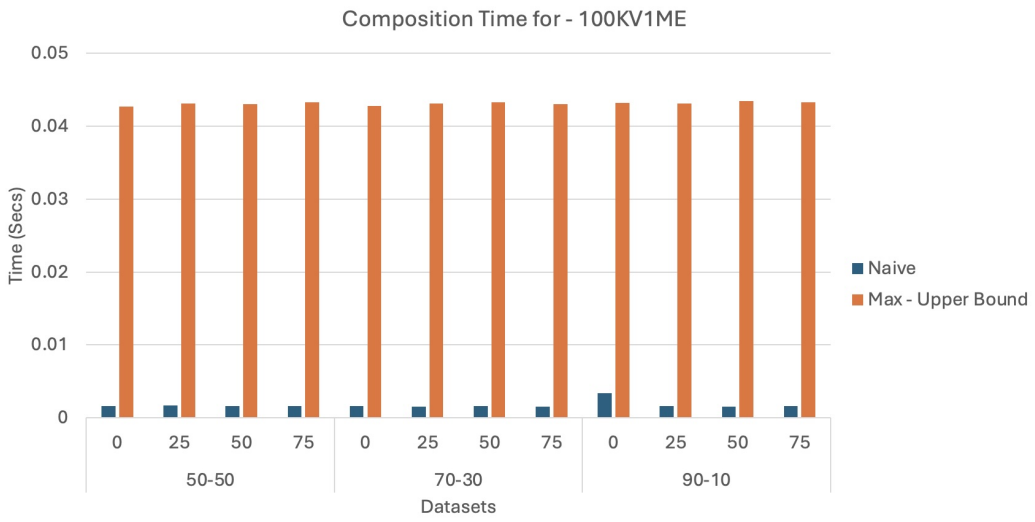


Figure 6.37: Composition Time for Naïve and Max Upper-Bound Heuristics - `100KV1ME`

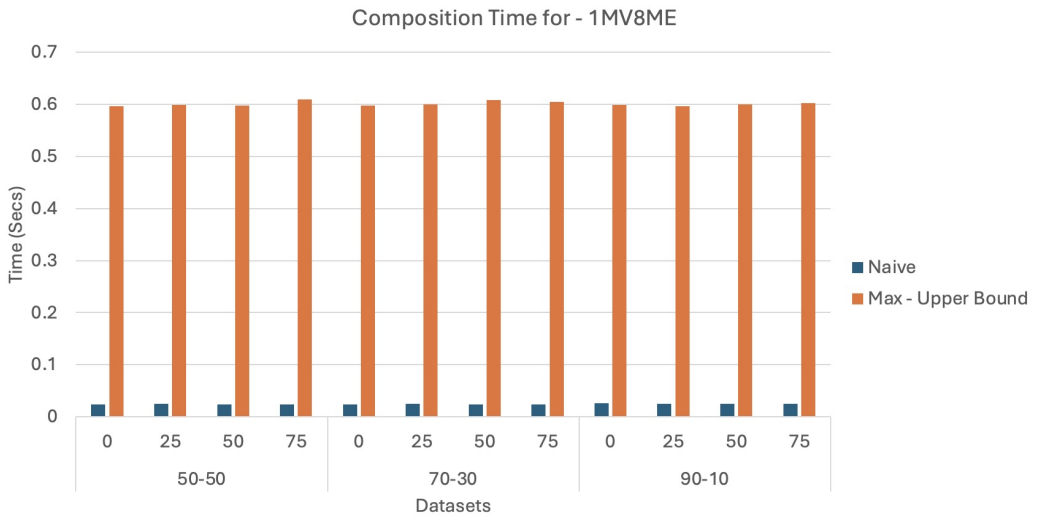


Figure 6.38: Composition Time for Naïve and Max Upper-Bound Heuristics - `1MV8ME`

Dataset	**Naïve**	**Max - Upper Bound**
Co-authorship (2004–2005)	0.00296	0.09983
Ant Colony (Day 1 and 3)	0.000025	0.00027

Table 6.8: Composition Time (in seconds) for Max Heuristic Strategies on Real-World Datasets

These trends are mirrored in the real-world datasets (Table 6.8). The co-authorship network, being larger, requires 0.0998 seconds for the upper-bound heuristic and only 0.003 seconds for the naïve approach. Similarly, for the smaller ant colony dataset, the upper-bound composition takes just 0.00027 seconds compared to 0.000025 seconds for the naïve variant.

These results reinforce that while the upper-bound method incurs a modest increase in runtime, it remains highly scalable and practical even for large graphs. This slight computational overhead is justified by the substantial accuracy gains demonstrated earlier. Moreover, both heuristics are orders of magnitude faster than ground

truth computation, making them viable for real-time or large-scale multilayer network analysis.

Taken together, the results from both the sum and max aggregation strategies underscore the effectiveness of the decoupling-based framework in approximating ground truth centrality measures in multilayer networks. By leveraging layer-local analysis and lightweight composition heuristics, this approach enables scalable and accurate identification of influential nodes without requiring full network reconstruction. While the sum-based method benefits from its additive structure, making accurate strength estimation straightforward, the max-based variant poses additional challenges due to edge overlap uncertainty. Nevertheless, the proposed upper-bound heuristic delivers strong performance across both synthetic and real-world datasets, balancing accuracy and runtime even in large-scale scenarios. These insights highlight the broader utility of decoupled heuristics for centrality computation in multilayer settings and motivate further exploration of their applicability to additional measures and tasks.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

This thesis presents a decoupling-based framework for efficient and scalable computation of weighted degree centrality in homogeneous multilayer networks (HoMLNs). Several heuristics were developed to estimate node strength without requiring full network aggregation. The most effective of these, based on selective node evaluation strategies such as top-ranked nodes, hubs, and early stopping, were evaluated extensively on both synthetic and real-world datasets. Under both sum and max aggregation models, the proposed heuristics achieve high accuracy while significantly reducing computation time relative to the baseline ground truth method. In particular, we show that with increased retention of partial layer-level information, near-perfect and even perfect accuracy (**up to 100%**) can be achieved across a range of datasets. The observed runtime savings are substantial, depending on dataset characteristics, edge distribution across layers, and inter-layer overlap.

While prior work has applied the decoupling framework to closeness and betweenness centrality in unweighted multilayer networks, extending these techniques to the weighted setting remains an important direction. Weighted variants introduce additional challenges due to their reliance on shortest path computations, but adapting the decoupling framework for such measures holds significant promise.

Future work may also explore extending this approach to **directed multilayer networks**, which are common in real-world domains such as transportation systems, biological pathways, and communication infrastructures. Additionally, applying the decoupling-based framework to other structural metrics, such as eigenvector centrality

and PageRank, offers promising opportunities for advancing the analysis of multilayer networks.

REFERENCES

[1] A. Santra, K. S. Komar, S. Bhowmick, and S. Chakravarthy, "From base data to knowledge discovery - *A life cycle approach* - using multilayer networks," *Data Knowl. Eng.*, vol. 141, p. 102058, 2022.

[2] A. Santra, "Analysis of Complex Data Sets Using Multilayer Networks: a Decoupling-Based Framework," Ph.D. dissertation, The University of Texas at Arlington, August 2020. [Online]. Available: https://itlab.uta.edu/students/alumni/PhD/Abhishek_Santra/ASantra_PhD2020.pdf

[3] L. C. Freeman, "Centrality in social networks conceptual clarification," *Social networks*, vol. 1, no. 3, pp. 215–239, 1978.

[4] P. Bródka, K. Skibicki, P. Kazienko, and K. Musiał, "A degree centrality in multi-layered social network," in *2011 International Conference on Computational Aspects of Social Networks (CASoN)*, pp. 237–242.

[5] T. Opsahl, F. Agneessens, and J. Skvoretz, "Node centrality in weighted networks: Generalizing degree and shortest paths," *Social Networks*, vol. 32, no. 3, pp. 245 – 251, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378873310000183

[6] Y. Huang, L. Lv, Y. Liu, and X. Wang, "An improved gravity model based on supra-laplacian energy for identifying key nodes in multilayer networks," *Physics Letters A*, vol. 551, p. 130647, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S037596012500427X

[7] M. Khansari, A. Kaveh, Z. Heshmati, and M. A. Motlaq, "Centrality measures for immunization of weighted networks," *Network Biology*, vol. 6, no. 1, pp. 12–27, 2016.

[8] Y. Gu and Y. Wang, "Using weighted multilayer networks to uncover scaling of public transport system," *Environment and Planning B: Urban Analytics and City Science*, vol. 49, no. 6, pp. 1631–1645, 2022. [Online]. Available: https://doi.org/10.1177/23998083211062905

[9] J. Zhou, B. Shuai, Y. Zhang, and W. Huang, "Structural characteristics and reliability analysis of multilayer transportation network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 6, pp. 7476–7485, 2025.

[10] Y. Zhou, M. Zhang, S. Deng, S. Hu, S. Zheng, and Y. Chen, "Node importance calculation in bus-metro composite network considering land use," *Tunnelling and Underground Space Technology*, vol. 163, p. 106723, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S088677982500361X

[11] J. Ma and Z. Yuan, "The impact of community heterogeneity on information propagation and virus transmission in multi-layer networks," *Physics Letters A*, vol. 549, p. 130582, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0375960125003627

[12] Y. Cui, R. Wei, Y. Tian, H. Tian, and X. Zhu, "Information propagation influenced by individual fashion-passion trend on multi-layer weighted network," *Chaos, Solitons & Fractals*, vol. 160, p. 112200, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0960077922004106

[13] D.-M. Zhang, H. Bai, C.-Z. Zheng, H.-W. Huang, B. M. Ayyub, and W.-J. Cao, "Extreme rainfall induced risk mapping for metro transit systems: Shanghai metro network as a case," *Reliability Engineering &*

System Safety, vol. 262, p. 111234, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0951832025004351

[14] H. R. Pavel, A. Santra, and S. Chakravarthy, "Degree centrality algorithms for homogeneous multilayer networks," in *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2022, Volume 1: KDIR, Valletta, Malta, October 24-26, 2022*, F. Coenen, A. L. N. Fred, and J. Filipe, Eds. SCITEPRESS, 2022, pp. 51–62. [Online]. Available: https://doi.org/10.5220/0011528900003335

[15] S. Chakravarthy, A. Santra, and K. S. Komar, "Why multilayer networks instead of simple graphs? modeling effectiveness and analysis flexibility and efficiency!" in *Big Data Analytics - 7th International Conference, BDA 2019, Ahmedabad, India, December 17-20, 2019, Proceedings*, ser. Lecture Notes in Computer Science, S. Madria, P. Fournier-Viger, S. Chaudhary, and P. K. Reddy, Eds., vol. 11932. Springer, 2019, pp. 227–244. [Online]. Available: https://doi.org/10.1007/978-3-030-37188-3_14

[16] M. D. Domenico, V. Nicosia, A. Arenas, and V. Latora, "Layer aggregation and reducibility of multilayer interconnected networks," *CoRR*, vol. abs/1405.0425, 2014.

[17] A. Berenstein, M. P. Magarinos, A. Chernomoretz, and F. Aguero, "A multilayer network approach for guiding drug repositioning in neglected diseases," *PLOS*, vol. 10, no. 1, pp. 1–33, 2016.

[18] K. Mukunda, A. Roy, A. Santra, and S. Chakravarthy, "Stress centrality in heterogeneous multilayer networks: Heuristics-based detection," in *IEEE Ninth International Conference on Big Data Computing Service and Applications, BigDataService 2023, Athens, Greece, July 17-20, 2023*. IEEE, 2023, pp.

103–110. [Online]. Available: https://doi.org/10.1109/BigDataService58306.2023.00021

[19] A. Santra, S. Bhowmick, and S. Chakravarthy, "Efficient community re-creation in multilayer networks using boolean operations," in *International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland*, ser. Procedia Computer Science, P. Koumoutsakos, M. Lees, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds., vol. 108. Elsevier, 2017, pp. 58–67. [Online]. Available: https://doi.org/10.1016/j.procs.2017.05.246

[20] K. Samant, E. Memeti, A. Santra, E. Karim, and S. Chakravarthy, "Cowiz: Interactive covid-19 visualization based on multilayer network analysis," in *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Crete, Greece, April 19-22, 2021*, https://itlab.uta.edu/cowiz/ and https://www.youtube.com/watch?v=4vJ56FYBSCg.

[21] A. Santra, S. Bhowmick, and S. Chakravarthy, "Hubify: Efficient estimation of central entities across multiplex layer compositions," in *2017 IEEE International Conference on Data Mining Workshops, ICDM Workshops 2017, New Orleans, LA, USA, November 18-21, 2017*, R. Gottumukkala, X. Ning, G. Dong, V. Raghavan, S. Aluru, G. Karypis, L. Miele, and X. Wu, Eds. IEEE Computer Society, 2017, pp. 142–149. [Online]. Available: https://doi.org/10.1109/ICDMW.2017.24

[22] K. S. Komar, A. Santra, S. Bhowmick, and S. Chakravarthy, "EER→MLN: EER Approach for Modeling, Mapping, and Analyzing Complex Data Using Multilayer Networks (MLNs)," in *Conceptual Modeling - 39th International Conference, ER 2020, Vienna, Austria, November 3-6, 2020, Proceedings*, ser. Lecture Notes in Computer Science, G. Dobbie, U. Frank, G. Kappel, S. W.

Liddle, and H. C. Mayr, Eds., vol. 12400. Springer, 2020, pp. 555–572. [Online]. Available: https://doi.org/10.1007/978-3-030-62522-1_41

[23] K. Komar, "Data-driven modeling of heterogeneous multilayer networks and their community-based analysis using bipartite graphs," Master's thesis, The University of Texas at Arlington, August 2019. [Online]. Available: http://itlab.uta.edu/students/alumni/MS/Kanthi_Sannappa_Komar/KanthiK_MS2019.pdf

[24] A. Santra, F. A. Irany, K. Madduri, S. Chakravarthy, and S. Bhowmick, "Efficient community detection in multilayer networks using boolean compositions," *Frontiers Big Data*, vol. 6, 2023. [Online]. Available: https://doi.org/10.3389/fdata.2023.1144793

[25] A. Santra, S. Bhowmick, and S. Chakravarthy, "Efficient community detection in boolean composed multiplex networks," *University of Texas at Arlington*, vol. June, 2019. [Online]. Available: http://itlab.uta.edu/research/current/Multi%20Source%20Data%20Analysis/ArXiv2019-HoMLN-Final.pdf

[26] A. Rai, A. Roy, A. Santra, and S. Chakravarthy, "Mln-subdue: Substructure discovery in homogeneous multilayer networks," in *Proceedings of the 16th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2024, Volume 1: KDIR, Porto, Portugal, November 17-19, 2024*, F. Coenen, A. Fred, and J. Bernardino, Eds. SCITEPRESS, 2024, pp. 36–47. [Online]. Available: https://doi.org/10.5220/0012906200003838

[27] K. Bolaj, A. Santra, and S. Chakravarthy, "Substructure discovery in heterogeneous multilayer networks," in *IEEE International Conference on Knowledge Graph, ICKG 2023, Shanghai, China, December 1-2, 2023*, V. S. Sheng, C. Hicks, C. Ling, V. Raghavan, and X. Wu, Eds. IEEE, 2024, pp. 1–8. [Online]. Available: https://doi.org/10.1109/ICKG63256.2024.00009

[28] K. Bolaj, "HeMLN-SD: Substructure discovery in heterogeneous multilayer networks," Master's thesis, The University of Texas at Arlington, December 2023. [Online]. Available: https://itlab.uta.edu/students/alumni/MS/Kiran_Bolaj/KBolaj_MS2023.pdf

[29] A. Singh, "HoMLN-SD: Substructure discovery in homogeneous multilayer networks," Master's thesis, The University of Texas at Arlington, December 2023. [Online]. Available: https://itlab.uta.edu/students/alumni/MS/Arshdeep_Singh/ASingh_MS2023.pdf

[30] A. Rai, "MLN-SUBDUE: decoupling approach-based substructure discovery in multilayer networks (MLNs)," Master's thesis, The University of Texas at Arlington, May 2020. [Online]. Available: https://itlab.uta.edu/students/alumni/MS/Anish_Rai/ARai_MS2020.pdf

[31] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.

[32] "AI@WSU," http://ailab.wsu.edu/subdue/download.htm, [Online].

[33] F. Khorasani, R. Gupta, and L. N. Bhuyan, "Scalable simd-efficient graph processing on gpus," in *Proceedings of the 24th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '15, pp. 39–50.

[34] K. Mukunda, "Decoupling-based approach to centrality detection in heterogeneous multilayer networks," Master's thesis, The University of Texas at Arlington, August 2021. [Online]. Available: https://itlab.uta.edu/students/alumni/MS/Harshit_A_Modi/HAModi_MS2020.pdf

[35] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *arXiv preprint arXiv:2005.00687*, 2020.

[36] M. Collier, N. Wijayatilake, P. Sah, S. Ali, J. Mendez, M. Silk, and S. Bansal, "Animal social network repository 3.0," Nov. 2024. [Online]. Available: https://doi.org/10.5281/zenodo.14168300

BIOGRAPHICAL STATEMENT

Ayomide Ayowole-Obi was born in Osun State, Nigeria, and emigrated to the United States in 2012, where she continued her education. She earned her Bachelor's degree in Computer Science from the College of Science and Engineering at Texas Christian University in May 2023. Following graduation, she began her Master's program at the University of Texas at Arlington, which she pursued from August 2023 to August 2025. She enjoys working in the areas of databases and data analysis.