**Journal of Mathematics and Scientific Computing with Applications**

E-ISSN : 2580-5754; P-ISSN : 2580-569X
Volume 7, Number 1, 2026
DOI: 10.53806/jmscowa.v7i1.1601
Page: 179–193

# CENTRALITY-BASED GRAPH PRUNING AND GAT-PSO FOR THE MAXIMUM CLIQUE PROBLEM IN PROTEIN-PROTEIN INTERACTION NETWORKS

**Gilland Fausta Putra Achyar**[1], **Annisa**[2], **Heru Cahya Rustamaji**[3],**Wisnu Ananta Kusuma**[4]

[1]School of Data Science, Mathematics and Informatics, IPB University, Bogor, 16680, Indonesia
[2]Bioinformatics Study Program, Faculty of Mathematics and Natural Sciences, IPB University, Bogor, 16680, Indonesia
[3]Tropical Biopharmaca Research Center, IPB University, Bogor, 16128, Indonesia
[4]Faculty of Industrial Engineering UPN Veteran Yogyakarta, Yogyakarta, 55283, Indonesia

**Article Info**

***Article history:***
Received : June 30, 2026
Revised : July 16, 2026
Accepted : July 30, 2026

***Keywords:***
Centrality Metrics;
Graph Attention Network;
Graph Pruning;
Maximum Clique;
Protein-Protein Interaction.

**ABSTRACT**

Finding maximum cliques in protein-protein interaction networks (PPINs) is computationally NP-hard. Large-scale PPINs typically contain dense and redundant interaction structures that exponentially increase search time. To address this computational bottleneck while preserving topological integrity, this study proposes a two-stage pruning strategy. The framework first employs K-core decomposition to filter peripheral noise, followed by a particle swarm-optimized graph attention network (GAT-PSO) that integrates four centrality metrics. This centrality-aware design explicitly captures complex structural dependencies, successfully mitigating the dense-core bias inherent in conventional statistical feature-based pruning and ensuring the retention of critical connector nodes. Evaluation across 12,535 STRING-derived PPINs demonstrated average node and edge reductions of 95.87% and 91.11%, respectively, thereby accelerating the MaxCliqueDyn (MCQD) algorithm by up to 106.73 times. Despite this extreme dimensionality reduction, the pruned networks maintained strong structural fidelity, achieving a clique-size similarity of 97.23% and a Jaccard index of 86.70%. Furthermore, functional enrichment confirmed that the retained modules align with established biological pathways. These results validate the proposed framework as a robust, scalable pre-processing solution for accelerating exact clique detection in massive PPINs.

*This is an open access article under the CC BY-SA license.*

***Corresponding Author:***
Wisnu Ananta Kusuma,
School of Data Science, Mathematics and Informatics,
IPB University
Email: ananta@apps.ipb.ac.id

## 1. INTRODUCTION

The maximum clique problem (MCP) is a combinatorial optimization problem in graph theory that is classified as NP-hard [1]. The objective of this problem is to find the largest clique [2], defined as a complete subgraph in which every vertex is directly connected to every other vertex, with the maximum number of vertices within a graph. In the field of bioinformatics, the MCP has important applications, particularly in the analysis of Protein-Protein Interaction Networks (PPINs). In a PPIN representation, vertices represent proteins, while edges indicate physical or functional interactions between those proteins [3], [4]. Searching for the maximum clique in a PPIN can be used as a computational strategy to identify candidate protein complexes or densely connected functional modules that interact closely and are fundamental to biological processes [5].

Although its application is biologically relevant, the search for cliques in large-scale PPINs is hindered by the complexity of large, sparse, noisy, and heterogeneous biological networks, which often exhibit scale-free and small-world properties [6]. To ensure the discovery of exact solution, state-of-the-art exact branch-and-bound algorithms such as MaxCliqueDyn (MCQD) are widely relied upon due to their efficiency in implementing the ColorSort algorithm and dynamic upper-bound determination [7]. However, the massive size of biological graphs inherently expands the search space of exact algorithms [7], [8]. Many nodes in large PPINs may have limited contribution to the final maximum clique search, but exact algorithms still need to evaluate or eliminate these candidates through upper-bound computation and backtracking. The accumulation of low-contributing or irrelevant candidate nodes may trigger a combinatorial explosion, drains memory, and significantly extends execution time.

To address this computational burden, graph pruning techniques serve as an essential pre-processing step to limit the search space while minimizing the loss of structurally and biologically relevant information [9]. Previous approaches, such as Learning Graph Pruning for MCE (LGP-MCE), have combined k-core decomposition and deep learning architectures to filter node candidates [10]. However, these approaches have several limitations when applied to PPIN networks. First, extreme pruning that leaves only nodes at the maximum k-core value is an aggressive step that may permanently eliminate nodes belonging to target maximum cliques or biologically meaningful dense modules [10]. Second, the architectural features extracted in that study used general network statistical metrics, such as the local clustering coefficient (LCC) and the chi-square ($\chi^2$) function, which may provide limited biological interpretability in PPIN analysis. In fact, a node with an intermediate degree may still play an important role as a bottleneck or connector node between distinct functional modules [11].

To bridge this research gap, this study proposes a novel centrality-aware pruning framework. Centrality analysis is essential for assessing the functional importance of proteins; for instance, high degree centrality often indicates fundamental cellular roles, while closeness and betweenness identify key proteins that serve as shortest-path connectors or bridges between biological clusters [12]. Algorithmically, we hypothesize that integrating four complementary topological measures, specifically degree, betweenness, closeness, and eigenvector centrality, creates a highly discriminative feature space for exact clique membership. Mathematically, we hypothesize that by leveraging a Graph Attention Network (GAT), the multi-head self-attention mechanism can dynamically compute anisotropic weights for these centrality features, enabling the model to precisely isolate true clique nodes from surrounding dense topological noise [13], [14], [15].

Driven by these hypotheses, our framework employs a two-stage topological strategy. First, rather than applying extreme pruning, the degree threshold in the initial k-core reduction is dynamically adjusted to establish a proportional lower bound, thereby retaining potential dense module candidates. Second, the extracted centrality features are evaluated by the GAT classifier. However, optimizing GAT on large-scale PPINs is computationally sensitive due to the severe class imbalance inherent in maximum clique distribution [16]. To address this, the Particle Swarm Optimization (PSO) metaheuristic is integrated to systematically navigate the hyperparameter search space [17]. By balancing exploration and exploitation, PSO strictly calibrates the network to mitigate over-smoothing and reduce false positives. Finally, the structural validity of this algorithmic reduction is empirically verified through functional enrichment analysis against Gene Ontology (GO) and KEGG Pathways to ensure the retained nodes represent actual biological complexes [18].

Overall, this study aims to develop and evaluate a centrality-guided, PSO-optimized graph pruning framework to exponentially accelerate exact MCP solvers in PPINs. The primary contribution of this research lies in a novel two-stage pruning strategy that shifts from basic statistical filtering to a centrality-aware topological evaluation, effectively avoiding the dense-core bias of prior methods. This structural evaluation is further strengthened by implementing PSO to systematically optimize the GAT architecture, successfully overcoming severe class imbalance while ensuring high classification precision. Ultimately, the entire framework is comprehensively validated to prove that massive computational speedup through search-space reduction can be achieved without sacrificing the discovery of structurally and biologically vital protein complexes.

## 2. RESEARCH METHOD

This study develops a two-stage graph pruning technique to reduce the computational complexity of the maximum clique search on PPINs. These networks, consisting of physical and functional protein associations, were extracted from the STRING database version 12.0 [19]. The process begins with data pre-processing, followed by initial network reduction using k-core decomposition. Node probabilities are then evaluated based on four centrality metrics (degree, betweenness, closeness, and eigenvector) using a graph attention network. This step aims to identify and remove irrelevant nodes while preserving structurally important candidates. Finally, the resulting pruned network is functionally validated through enrichment analysis. The overall research workflow is presented in Figure 1.

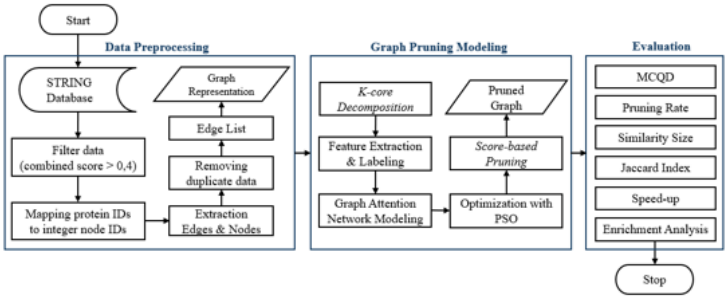



**Figure 1.** Stages of pruning for graph structure simplification

### 2.1. Data Pre-processing

The PPIN dataset was automatically acquired from the STRING database version 12.0 [19] via an API request using Python 3.10. The network was filtered using a combined score threshold of ¿0.4. This threshold was chosen based on the consideration that scores that are too high can cause fragmentation of the main network, while scores that are too low risk including invalid biological interactions [20]. Subsequent pre-processing stages involved mapping alphanumeric STRING protein IDs to contiguous zero-based integer indices to ensure compatibility with graph neural network data structures. Furthermore, the graph topology was strictly simplified by removing self-loops, as maximum clique formulations require interactions between distinct nodes. We also deduplicated multiple interaction records between the same protein pair into a single unweighted, undirected edge to prevent topological distortion. The pre-processing pipeline concludes with the compilation of a clean edge list representing the complete graph structure.

### 2.2. Centrality-Based Graph Pruning Modeling

To significantly reduce the search space, the pruning workflow is implemented through the following steps:

a Graph Representation
The process begins with the construction of an undirected protein-protein interaction (PPI) graph $G(V, E)$, where the set of vertices $V$ represents proteins and the set of edges $E$ represents biological interactions between proteins.

b K-Core Decomposition
The initial network simplification employs k-core decomposition, utilizing a degree-based "safe pruning" strategy. The fundamental principle relies on the structural property that a clique of size $S$ requires every internal node to possess a minimum connectivity degree of $S-1$ [10]. Therefore, rather than applying a static threshold or utilizing the maximum network degeneracy—which inherently carries a high risk of inadvertently eliminating true maximum clique members—this study implements a dynamic heuristic function to determine a conservative pruning threshold $k$ for each network using the formulation $k = \max(\text{mc\_size} - 1, 2)$.
To obtain the initial clique size estimate (`mc_size`) efficiently, the Bron-Kerbosch algorithm is employed as a heuristic approach, explicitly constrained by an early stopping mechanism. To prevent exhaustive computation, this initial search terminates immediately after evaluating 15 maximal clique candidates or upon discovering a clique containing a minimum of 3 proteins. The derived estimate is then substituted into the threshold function, acting as a structural safety net.
Furthermore, the absolute lower bound of 2 is mathematically established to guarantee computational efficiency. Because the biological datasets are acquired as interaction edge lists, every mapped node in the raw graph inherently possesses a minimum degree of 1; completely isolated nodes (degree 0) do not exist. Consequently, computing a 1-core pruning (e.g., if the initial heuristic only found a simple edge, yielding mc_size=2) would be a computational waste. By enforcing a strict lower bound of $k = 2$, the algorithm immediately targets and recursively eliminates pendant nodes (degree 1) and irrelevant tree structures during the very first iteration, as nodes with only a single connection have zero mathematical potential to form a maximum clique.

c Labeling, Centrality Feature Extraction, and Dataset Partitioning
To build a supervised learning model, the k-core graphs were labeled using the exact maximum clique algorithm, specifically MaxCliqueDyn (MCQD). A computational time limit of 3,600 seconds per network was set to ensure an optimal clique search to serve as the ground truth. Nodes are assigned binary labels: 1 if they belong to a maximum clique, and 0 if they do not. The dataset is then partitioned into training, validation, and test sets using a stratified split approach to ensure model generalization. To guarantee scientific reproducibility, the topological categories were explicitly defined using the overall dataset median values as numeric thresholds: graph scale (Large: $> 2677$ nodes and Small: $\leq 2677$ nodes) and connection density (Dense: $> 0.0157$ and Sparse: $\leq 0.0157$).
For each training graph, the structural characteristics of the nodes are extracted into a feature matrix $X \in \mathbb{R}^{N\times 4}$ that summarizes four complementary centrality metrics. Mathematically, these features are defined as follows for a given node $v$:

i. Degree Centrality ($C_D$): Measures the normalized local connectivity volume relative to the entire network, defined as:

$$C_D(v) = \frac{\deg(v)}{n-1} \tag{1}$$

where $\deg(v)$ is the number of direct edges connected to node $v$, and $n$ is the total number of nodes in the graph.

ii. Betweenness Centrality ($C_B$) : Quantifies structural bottlenecks by calculating the fraction of shortest paths passing through the node:

$$C_B(v) = \sum_{s\neq v\neq t} \frac{\sigma_{st}(v)}{\sigma_{st}} \tag{2}$$

where $\sigma_{st}$ is the total number of shortest paths from node $s$ to node $t$, and $\sigma_{st}(v)$ is the number of those paths that pass through $v$.

iii. Closeness Centrality ($C_C$) : Measures signal propagation capacity as the inverse of the average shortest path length to all other nodes $u$:

$$C_C(v) = \frac{1}{\sum_{u\neq v} d(v,u)} \tag{3}$$

where $d(v,u)$ is the shortest path distance between nodes $v$ and $u$.

iv. Eigenvector Centrality ($C_E$) : Computes global influence based on the centrality of neighboring nodes $M(v)$, defined via the adjacency matrix $A$ and its principal eigenvalue $\lambda$:

$$C_E(v) = \frac{1}{\lambda} \sum_{u\in M(v)} C_E(u) \tag{4}$$

Together, this integrated feature set ensures a multidimensional representation of both local density and global network positioning that a single metric cannot achieve.

d Probability prediction using GAT-PSO
The feature matrix is fed into a hybrid Graph Attention Network (GAT) architecture that combines local feature extraction from the GATConv layer (multi-head attention) and global features via global max pooling. The network parameters are optimized in a supervised manner using the Adam algorithm [21] over a maximum of 100 training epochs. To prevent overfitting and ensure an objective stopping criterion, early stopping was implemented, which terminates the training if the validation loss fails to improve for 15 consecutive epochs. Extreme class imbalance is addressed using a cost-sensitive learning framework [22], specifically implemented through the BCEWithLogitsLoss formulation. This function acts as a weighted binary cross-entropy, employing a proportional positive weight compensation, calculated mathematically as the ratio of majority negative nodes to minority positive nodes (yielding a weight of roughly 39.78), to strictly penalize minority class misclassifications [23].
To systematically navigate the hyperparameter space of Graph Neural Networks, Particle Swarm Optimization (PSO) was adopted. Recent computational studies demonstrate that PSO significantly enhances the stability and convergence efficiency of graph-based neural models. By dynamically balancing exploration and exploitation within the search space, PSO efficiently overcomes the limitations of conventional grid or random search methods [24]. To systematically tune the architecture, Particle Swarm Optimization (PSO) was initialized with a constrained search budget of 10 particles over 10 iterations to navigate the hyperparameter space. The search bounds were explicitly defined to ensure computational feasibility: hidden channels $\in [16, 128]$, attention heads $\in [1, 8]$, and learning rate $\in [0.0001, 0.01]$. The PSO objective function strictly evaluates these combinations to maximize the Area Under Curve (AUC) metric on the validation set, returning a highly calibrated final probability score ranging from 0 to 1 for each node.

e Score-Based Pruning
Based on the generated confidence scores, final-stage pruning is executed using a standard probability threshold of 0.5, which provides a mathematically balanced cutoff for binary classification without biasing either class. Nodes with scores below the threshold are identified as sparse nodes (non-clique) and removed along with all their connections

f Pruned Graph The final output is a reduced graph focused on dense cluster regions, drastically reducing memory requirements and execution time when processed by search algorithms.

### 2.3. Performance Evaluation and Preservation Metrics

This stage aims to measure the effectiveness of graph simplification techniques (K-Core and GAT-PSO) and their impact on the computational performance of the exact maximum clique search algorithm.

#### 2.3.1 Performance Evaluation of the GAT-PSO Classification Model

Classification performance is evaluated using a Confusion Matrix that maps the model's predictions against actual labels into True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) [25]. Given that standard accuracy is prone to bias in highly imbalanced data such as PPI networks, this study specifically uses Balanced Accuracy [26] along with other classification robustness metrics [25] via the following formulation:

a Balanced Accuracy (B-Acc)
Calculates the average of sensitivity (True Positive Rate/TPR) and specificity (True Negative Rate/TNR). This metric provides a fair and objective representation of performance without being biased by the dominance of the majority class [26] and is calculated using Equation 5.

$$B - Acc = \frac{\mathrm{TPR} + \mathrm{TNR}}{2} \tag{5}$$

where specificity or $\mathrm{TNR} = \frac{\mathrm{TN}}{TN+FP}$

b Precision
Measures the accuracy of the model's positive predictions. This value represents the proportion of nodes predicted to be in a clique that are actually confirmed to be correct [25], calculated using Equation 6.

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{FP} + \mathrm{TP}} \tag{6}$$

c Recall
Evaluates the model's capacity and sensitivity in identifying and extracting all actual positive samples (the actual target clique members) without any being omitted [25], calculated using Equation 7.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{7}$$

d F1-Score
Calculates the harmonic mean between precision and recall. This metric provides an excellent single evaluation indicator to assess the model's balance in simultaneously minimizing false positives and false negatives [25], calculated using Equation 8.

$$\mathrm{F1-score} = \frac{2\mathrm{PPV} \times \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} \tag{8}$$

#### 2.3.2 Performance Evaluation of Graph Pruning

a Pruning Rate
This metric measures the percentage of nodes and edges that were successfully removed from the original graph. The higher this metric's value, the smaller the remaining search space for the algorithm to explore [10]. He reduction percentages for nodes ($N_p^R$) and edges ($E_p^R$) are calculated using Equations 9 and 10.

$$\mathrm{N_p^R} = \frac{|\mathrm{V}| - |\mathrm{V_p}|}{|\mathrm{V}|} \cdot 100 \tag{9}$$

$$\mathrm{E_p^R} = \frac{|\mathrm{E}| - |\mathrm{E_p}|}{|\mathrm{E}|} \cdot 100 \tag{10}$$

where V and E are the total number of vertices and edges in the original network, while $|V_p|$ and $|E_p|$ are the number of remaining vertices and edges in the post-pruning network.

b Structure Preservation Metrics
Since the maximum clique in a PPIN represents a biological functional module, pruning must not compromise the integrity of that structure. Evaluation is performed using two metrics. First, clique size similarity is used to ensure that the algorithm on the pruned graph can find a clique with the same size capacity as the original graph. This clique size ratio evaluation approach is adopted from the concept of clique size accuracy by [27], which is calculated using Equation 11.

$$\text{similarity} = \frac{\text{MCp}}{\text{MC}} \times 100 \tag{11}$$

where $|MC_p|$ is the size of the maximum clique in the pruned graph, and MC is the size in the original graph. Second, the Jaccard Index is introduced to precisely measure the degree of overlap or protein membership identity between the two cliques [28], as shown in Equation 12.

$$\text{J}\left(\text{MC}, \text{MC}_\text{p}\right) = \frac{|\text{MC} \cap \text{MC}_\text{p}|}{|\text{MC} \cup \text{MC}_\text{p}|} \tag{12}$$

The Jaccard Index equals 1 if the protein composition in both cliques is perfectly identical.

c Computational Speed-up
The computational speed-up metric is used to empirically demonstrate the significance of the graph reduction process on computational time efficiency in problem-solving. This ratio is calculated by dividing the execution time of the exact MCQD algorithm on the original graph by the execution time of MCQD on the reduced graph [29]. To ensure a balanced and fair computational comparison, evaluations under both graph conditions are strictly executed using the same threshold parameter, namely $T_{limit} = 0.025$. The use of this default parameter follows the empirical recommendation of the MCQD algorithm's creator, who demonstrated that this value represents an optimal point for balancing computational load and efficiency across various graph density levels [7]. This metric is formulated in Equation 13.

$$\text{Speed up} = \frac{\text{Original graph time}}{\text{Pruned graph time}} \tag{13}$$

Furthermore, to provide a global representation of the computational performance of the proposed method across various topological variations, the average speed-up is used. This value is calculated from the entire test dataset of size $N$ [29], using Equation 14.

$$\text{Average speed up} = \frac{\sum \text{speed up}_{(\text{i})}}{\text{N}} \tag{14}$$

### 2.4. Enrichment Analysis

The final stage of the methodology involves medical and biological validation of the pruned graph using functional enrichment analysis. The predicted maximum clique structures were evaluated using the Metascape platform to map functional ontologies. The analysis focuses on four main annotation categories [18]:

a GO Biological Processes: Identification of macro-level biological processes driven by protein clusters.

b GO Molecular Functions: Validation of the biochemical and enzymatic activities of proteins within the clusters.

c GO Cellular Components: Identification of the subcellular locations of proteins.

d KEGG Pathway: Structural involvement of clusters in specific metabolic pathways or pathogenic infection mechanisms.

This step ensures that the pruning algorithm does not merely reduce the graph mathematically, but rather precisely extracts and isolates real biological molecular machines within the cell.

## 3. RESULT AND ANALYSIS

### 3.1. Characteristics of the PPI Network Dataset

Data acquisition via the STRING API, using a combined score $> 0.4$, successfully compiled protein-protein interaction networks from 12,535 organisms. This threshold of 0.4 is explicitly defined by the STRING database (version 12.0) as the standard "Medium Confidence" cut-off [19], which provides an optimal balance for capturing biologically relevant functional interactions without over-filtering the topological structure. Processing was carried out in 26 batches to maintain memory stability. The pre-processing results are presented in Table 1, showing extreme scale variations. The smallest graph has 73 nodes and 749 edges, with a density of 0.00214, while the largest graph has 82,364 nodes and 26,384,736 edges, with a density of 0.4658.

Table 1: Characteristics of protein-protein interaction networks

| Dataset | Min | | | Max | | |
|---|---|---|---|---|---|---|
| | $N$ | $E$ | Density | $N$ | $E$ | Density |
| STRING1–500 | 112 | 2,050 | 0.0031 | 42,994 | 4,834,155 | 0.3297 |
| STRING501–1000 | 535 | 6,888 | 0.0045 | 10,457 | 1,416,079 | 0.2730 |
| STRING1001–1500 | 140 | 4,533 | 0.0034 | 30,086 | 4,858,953 | 0.4658 |
| STRING1501–2000 | 174 | 2,605 | 0.0032 | 18,314 | 1,409,535 | 0.2603 |
| STRING2001–2500 | 172 | 3,155 | 0.0030 | 18,009 | 1,268,569 | 0.2145 |
| ... | ... | ... | ... | ... | ... | ... |
| STRING10501–11000 | 167 | 2,544 | 0.0035 | 48,459 | 13,228,300 | 0.2315 |
| STRING11001–11500 | 246 | 5,431 | 0.0031 | 24,375 | 2,752,955 | 0.1845 |
| STRING11501–12000 | 206 | 5,078 | 0.0024 | 44,525 | 16,750,875 | 0.2404 |
| STRING12001–12500 | 373 | 1,744 | 0.0021 | 22,888 | 5,699,991 | 0.1198 |
| STRING12501–12535 | 1,863 | 30,060 | 0.0032 | 20,224 | 2,467,700 | 0.0200 |
| ALL | 73 | 749 | 0.0021 | 82,364 | 26,384,736 | 0.4658 |

The density distribution and average degree are visualized in Figure 2. The majority of networks are highly sparse and concentrated around low values, which validates the scale-free nature of biological networks. Conversely, a small number of graphs at the long tail of the distribution have densities approaching 0.46 and extreme average connection numbers reaching up to 1,482. It is these characteristics of high-density graphs that trigger an exponential explosion in the MCP algorithm's search space, thereby justifying the need for search space reduction.

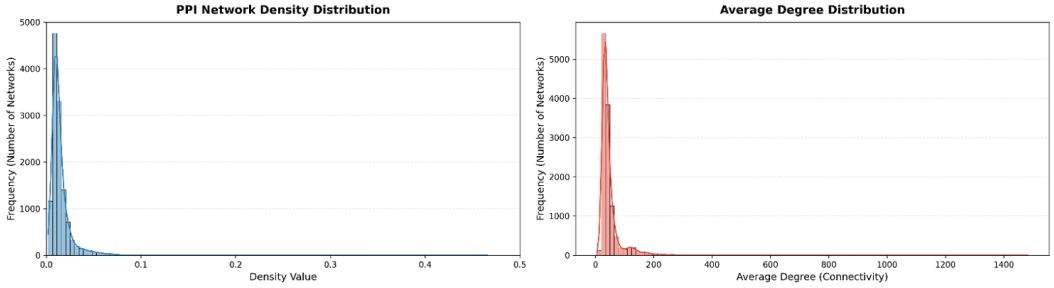



**Figure 2.** Density and Average Degree Distribution of PPI Networks

### 3.2. K-Core Decomposition

To safely prune the search space, the degree-based pruning approach using K-Core is applied before feature extraction. The threshold value ($k$) is dynamically determined based on a heuristic estimate of the initial clique size $k = \max(mc\_size - 1, 2)$. The lower bound $k = 2$ ensures the elimination of all isolated nodes (degree 0) and branch/pendant nodes (degree 1) in the first iteration. The results of the node and edge reduction are presented in Table 2.

Table 2: Results of Node and Edge Reduction in the K-Core Decomposition Stage

| No | Network ID | $k$-threshold | Initial Nodes | Pruned Nodes | Node Reduction (%) | Initial Edges | Pruned Edges | Edge Reduction (%) | Time (s) |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1004 | 5 | 6,568 | 5,128 | 21.92 | 137,278 | 134,271 | 2.19 | 0.9915 |
| 2 | 1007103 | 3 | 7,327 | 6,507 | 11.19 | 143,034 | 141,942 | 0.76 | 1.0897 |
| 3 | 1008305 | 7 | 1,816 | 1,427 | 21.42 | 31,971 | 30,663 | 4.09 | 0.1986 |
| 4 | 1030533 | 2 | 7,457 | 7,128 | 4.41 | 206,257 | 205,961 | 0.14 | 1.7899 |
| 5 | 1033744 | 4 | 1,704 | 1,511 | 11.33 | 28,893 | 28,493 | 1.38 | 0.1812 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 12531 | 444157 | 2 | 1,925 | 1,800 | 6.49 | 23,649 | 23,548 | 0.43 | 1.1626 |
| 12532 | 428406 | 3 | 4,505 | 4,017 | 10.83 | 67,700 | 67,028 | 0.99 | 0.4343 |
| 12533 | 460 | 2 | 3,361 | 3,022 | 10.09 | 66,277 | 65,978 | 0.45 | 0.7949 |
| 12534 | 471852 | 2 | 4,760 | 4,439 | 6.74 | 69,337 | 69,055 | 0.41 | 0.5355 |
| 12535 | 477 | 3 | 2,371 | 2,019 | 14.85 | 30,352 | 29,847 | 1.66 | 0.1906 |

A total of 31.92% of the networks were pruned with a lower bound of $k = 2$, while 6.99% of the extreme networks triggered aggressive pruning ($k \geq 10$), as shown in Figure 3. Globally, the pruning of 12,535 networks successfully reduced a per-graph average of 14.52% of the nodes and 2.41% of the edges. The low percentage of

edge reduction confirms that the removed nodes were sparse nodes, thus preserving the dense cluster architecture. It is this $k$-core graph representation that is passed on to the feature extraction stage.

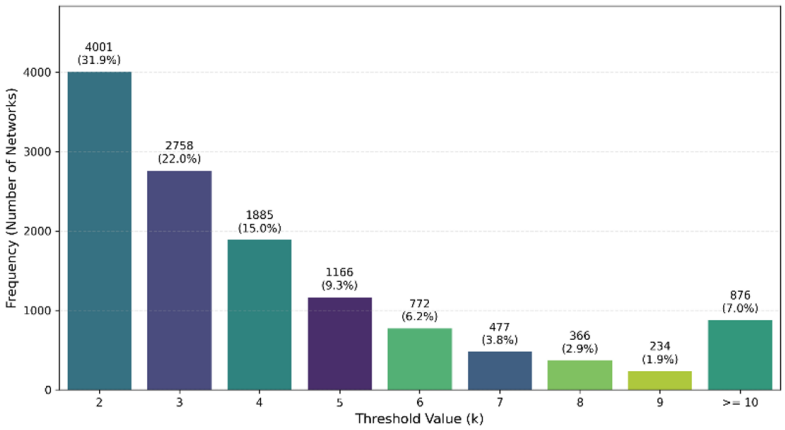



**Figure 3.** Threshold Value (k) Distribution in the Pruning Stage

### 3.3. Centrality Feature Extraction and Attention-based Pruning (GAT-PSO)

To prune complexes not captured by $k$-core, a GAT model was applied. The initial stage involved ground-truth labeling using an exact algorithm with a computation timeout of 3,600 seconds per network. Of the total 12,535 graphs, 1,405 were excluded because the exact algorithm failed to complete the search (converge) within the time limit due to the extreme NP-Hard complexity of the highly dense network topology. This process effectively left 11,130 valid labeled networks. It must be explicitly acknowledged that excluding these 1,405 extreme-density instances introduces a limitation to this study. The current GAT-PSO model is trained and evaluated on a subset that inherently omits the absolute hardest combinatorial cases. Addressing these ultra-dense networks remains an open challenge for future research, potentially requiring distributed computing frameworks or more relaxed exact-search heuristics.

Next, four centrality metrics were extracted from the remaining networks as topological features: degree, betweenness, closeness, and eigenvector centrality. Distribution analysis on a representative random sample of 500 networks reveals significant statistical differences between the maximum clique (Label 1) and non-clique (Label 0) classes, as shown in Figure 4. The most striking structural separation is observed in degree and eigenvector centrality, where the quartile range of the class far exceeds the maximum value of the non-clique class, providing a strong discriminative feature foundation for GAT.

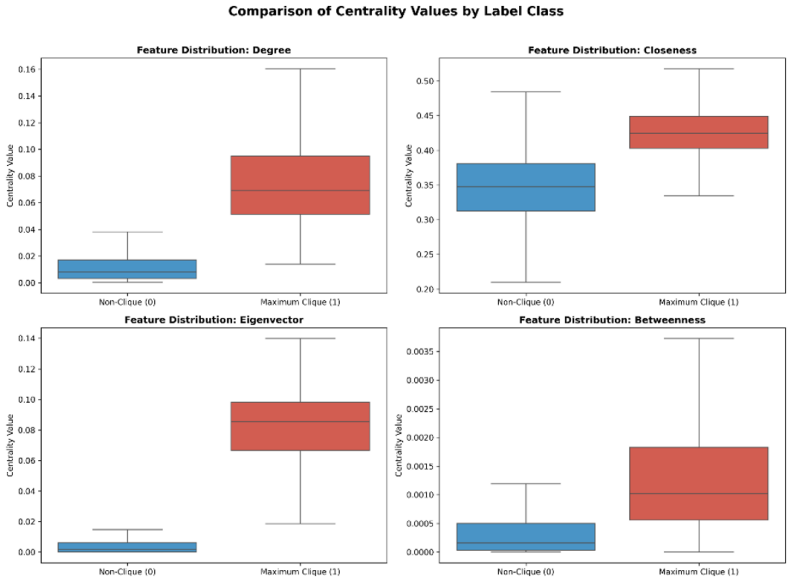



**Figure 4.** Centrality Value Distributions between Maximum Clique (1) and Non-Clique (0)

The entire dataset of 11,130 graphs was split using a stratified split, as shown in Table 3. The severe class imbalance challenge in the training data, with a ratio of 1 to 40 comprising 23,054,029 Label 0 nodes and 579,462 Label 1 nodes, was addressed by adding a positive compensation weight of 39.78 to the BCEWithLogitsLoss function.

Table 3: Dataset Split Distribution

| Topology Category | Training Set | Validation Set | Test Set | Total |
|---|---|---|---|---|
| Large-Dense | 739 | 99 | 149 | 987 |
| Large-Sparse | 3,427 | 457 | 693 | 4,577 |
| Small-Dense | 3,434 | 458 | 686 | 4,578 |
| Small-Sparse | 741 | 99 | 148 | 988 |
| Total | 8,341 | 1,113 | 1,676 | 11,130 |

The baseline evaluation phase was conducted using a heuristic parameter configuration consisting of a batch size of 4, a learning rate of 0.001, 64 hidden channels, and 4 attention heads. The baseline GAT training converged at epoch 72 with an AUC of 0.9973. However, excessive sensitivity due to the positive weight led to an over-prediction of positive classes, resulting in a low validation precision of 0.5143.

To address this calibration deficit, PSO was applied using 10 particles over 10 iterations to search for the optimal model architecture. The PSO algorithm exhibited ideal convergence behavior, achieving absolute convergence quickly by the third iteration with a peak fitness value (AUC) of 0.9986. This optimization definitively locked in the best hyperparameter configuration presented in Table 4.

Table 4: Optimal GAT Hyperparameters Discovered by PSO

| **Optimization Parameter** | **Global Best** | **Parameter Description** |
|---|---|---|
| Hidden Channels | 41 | Feature space dimension in the convolutional layer |
| Attention Heads | 7 | Number of parallel attention mechanisms |
| Learning Rate | 0.003441 | Weight update rate in the optimizer |

To empirically justify the constrained search budget of 10 particles over 10 iterations, the convergence dynamics of the swarm were plotted (see Figure 5). The optimization targeted the Area Under Curve (AUC) metric, which is robust against the extreme class imbalance of the PPIN dataset. As illustrated in the convergence curve, while the global best fitness rapidly reached a plateau of 0.9986 by the third iteration, the swarm average AUC continued to fluctuate actively throughout the remaining iterations. This dynamic confirms that the particles maintained a continuous exploration of the search space rather than stagnating. Given the relatively low-dimensional search space (optimizing only three architectural parameters) and the massive computational cost of training a GNN over 11,130 large-scale networks per evaluation, this 100-evaluation budget was proven sufficient to establish an optimal equilibrium without incurring premature convergence to a local optimum.

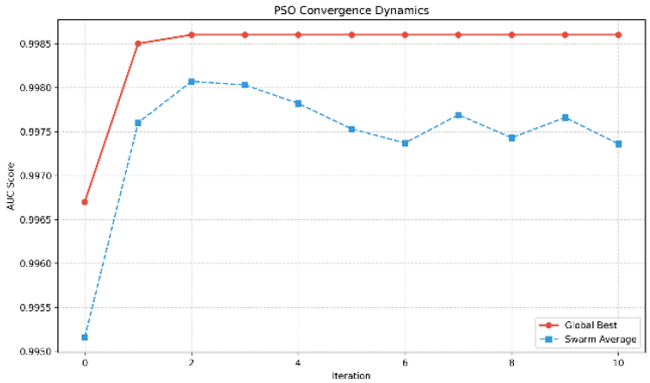



**Figure 5.** PSO Convergence Dynamics (Hyperparameter Optimization)

Integrating PSO had a significant impact on the model's predictive quality, as evidenced by a comprehensive comparison of performance metrics and convergence dynamics summarized in Table 5 and Figure 6, respectively. Table 5 presents the performance comparison evaluated on the unseen test set using a statistical bootstrapping approach (1,000 iterations), reporting the mean values alongside their standard deviations. The results demonstrate that GAT-PSO significantly improves Precision and F1-Score compared to the baseline, confirming its effectiveness as a regularizer. The inclusion of standard deviation values rigorously accounts for statistical uncertainty, validating that the observed improvements are robust and numerically significant.

Table 5: Comparison of GAT and GAT-PSO Performance

| Evaluation Metrics | GAT | GAT-PSO | Improvement |
|---|---|---|---|
| AUC | $0.9948 \pm 0.0001$ | $0.9942 \pm 0.0001$ | $-0.06\%$ |
| Precision | $0.5570 \pm 0.0011$ | $0.6461 \pm 0.0012$ | $+8.91\%$ |
| Recall | $0.9732 \pm 0.0005$ | $0.9412 \pm 0.0007$ | $-3.20\%$ |
| F1-Score | $0.7085 \pm 0.0009$ | $0.7662 \pm 0.0009$ | $+5.77\%$ |
| Balanced Accuracy | $0.9769 \pm 0.0002$ | $0.9641 \pm 0.0004$ | $-1.28\%$ |

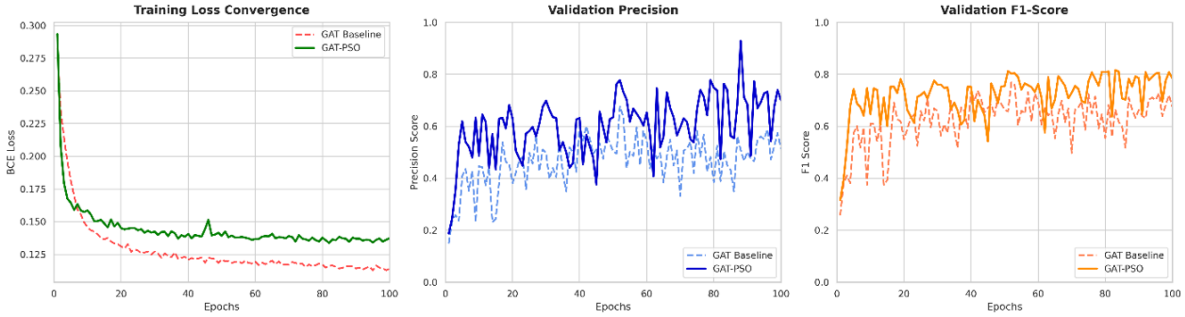



**Figure 6.** Training Curves Between the GAT Baseline and GAT-PSO Over 100 Epochs

Observing the dynamics of the training curves in Figure 6, PSO acts as a robust regularizer. The GAT-PSO model successfully reduced false-positive predictions by eliminating a substantial amount of noise nodes, directly boosting Precision by 8.91% and the F1-Score by 5.77%. However, this strict calibration introduces an inevitable trade-off, resulting in a 3.20% drop in Recall and a 1.28% decrease in Balanced Accuracy. From a structural validity perspective, this drop indicates that the model occasionally misses peripheral or boundary nodes of the true maximum cliques, risking the identification of slightly sub-optimal (smaller) dense modules. Nevertheless, within the context of massive PPIN reduction, prioritizing Precision ensures that the retained sub-graphs are highly reliable and devoid of topological noise, which is essential for accelerating downstream exact search algorithms.

### 3.4. Comparative Evaluation of Feature Selection: Statistical and Centrality

To objectively evaluate the effectiveness of the centrality-based approach, an ablation study was conducted to compare its performance against the statistical network features utilized in previous studies. It is crucial to note that this comparison involves two completely distinct models trained independently from scratch. The first model was trained exclusively on the statistical features extracted from the dataset, while the second model was trained on the proposed centrality features. To ensure a fair and rigorous methodology, both distinct models were independently optimized using the PSO algorithm to achieve their respective peak architectural configurations.

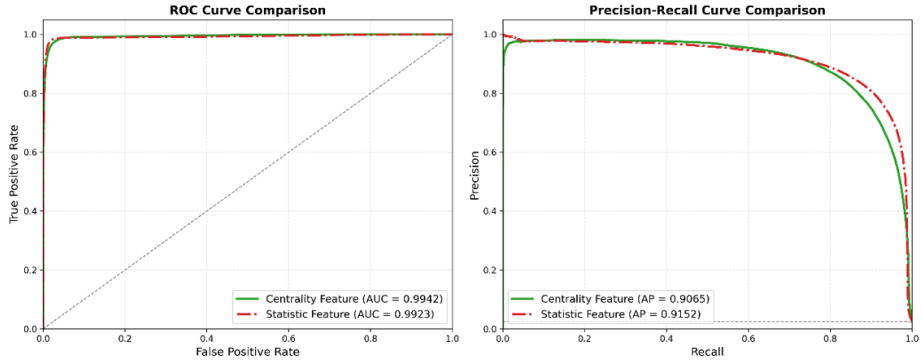



**Figure 7.** Comparison of ROC and Precision-Recall Curves Between Models Trained on Statistical and Centrality Features

In the evaluation of threshold-independent classification metrics as shown in Figure 7, the model trained on statistical features demonstrated a slightly higher Average Precision (AP = 0.9152) compared to the centrality model (AP = 0.9065). However, to determine whether this metric advantage translates effectively to downstream tasks, both independent models were tested directly in the graph pruning execution phase, as shown in Table 6.

Table 6: Pruning Execution Evaluation Results

| Evaluation Category | Measurement Metric | Statistical Feature | Centrality Feature |
|---|---|---|---|
| Topology Reduction | Node Reduction | 97.00% | 95.87% |
| | Edge Reduction | 94.24% | 91.11% |
| Clique Preservation | Jaccard Index | 79.02% | 86.70% |
| | Similarity Size | 89.53% | 97.23% |
| Computational Speedup | Average Original Time | 0.017969 s/graph | 0.017969 s/graph |
| | Average Pruned Time | 0.000188 s/graph | 0.000449 s/graph |
| | Average Speedup Ratio* | 455.59x Faster | 106.73x Faster |

*Calculated as the average of individual speedup ratios per graph, rather than the ratio of the global average times shown.

The downstream evaluation results in Table 6 expose a critical limitation of evaluating topological models solely through global classification metrics. Although the model trained on statistical features generated a highly aggressive search space reduction (achieving a mean per-graph acceleration of 455.59x), this pruning was inherently destructive. It permanently damaged the integrity of the target cliques, causing the Jaccard Index to plummet to 79.02%.

The slightly higher AP and AUC scores observed in the statistical model are, in fact, byproducts of a dense-core bias. Global classification metrics indiscriminately reward the correct prediction of highly dense, easy-to-classify core nodes but fail to penalize the misclassification of critical intermediate-degree bottleneck nodes. In contrast, the GAT-PSO model trained on centrality features successfully captured these topological bottlenecks. While it sacrificed a marginal amount of mathematical classification precision and pruning speed, it maintained the structural integrity of the biological modules (achieving a robust Jaccard Index of 86.70% and a Similarity Size of 97.23%). In the context of exact maximum clique search, preserving true biological complexes fundamentally outweighs minor gains in threshold-independent classification metrics.

### 3.5. Analysis of Clique Multiplicity and Structural Preservation

The gap between Similarity Size (97.23%) and the Jaccard Index (86.70%) highlights the distinction between algorithmic clique preservation and exact membership recovery. Mathematically, the MCQD algorithm on the pruned graph successfully identified cliques of a size equivalent to the ground truth, but with slightly varying protein compositions. This topological divergence can be explained by several structural factors:

a. Search Space Modification: The elimination of specific edges during GAT-PSO pruning alters the recursive search tree of the MCQD solver, occasionally directing it toward an alternative set of nodes that form a complete clique of equal size.

b. Maximum Clique Multiplicity: Graphically, large networks often contain multiple maximum cliques of the exact same size but with different node compositions [30]. This mathematical multiplicity explains how Similarity Size can reach 100% while the Jaccard Index remains below 1.

c. Peripheral Subunit Substitution: There is extreme membership overlap ($> 90\%$) in the majority of alternative cliques. For example, in graph 1001585, 95.65% of the membership consists of static core proteins. The decrease in the Jaccard Index is primarily driven by the algorithmic rotation of peripheral nodes at the edges of the dense cluster.

d. Disjoint Twin Clusters: In rare cases, the network maintains completely disjoint clusters with identical maximum density. For instance, in graph 1007099, the algorithm identified an alternative 63-node cluster sharing 0% of the original core nodes, indicating the discovery of a topologically equivalent, yet distinct, dense module.

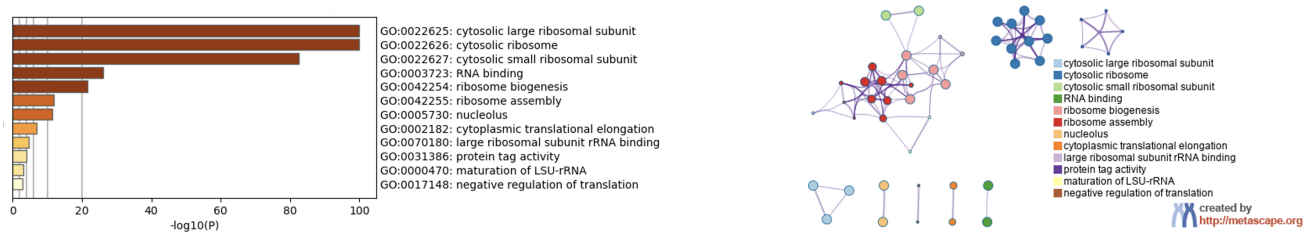
e. Topological Pleiotropy: This shift in exact membership reflects structural overlap rather than algorithmic failure. In biological contexts, this topological behavior frequently mirrors pleiotropy, where a single protein is functionally involved in multiple molecular machineries simultaneously [30],[31].

Therefore, while the GAT-PSO pruning fundamentally preserves the algorithmic structural properties of the network, these alternative topological clusters often correlate with functionally relevant modules rather than representing random network artifacts.

### 3.6. Enrichment Analysis

The Metascape platform was utilized to evaluate the functional associations of the dense clusters retained post-pruning across different kingdoms. It is important to note that while these retained cliques are mathematically rigid structures, enrichment analysis serves to estimate their potential biological relevance rather than definitively proving in vivo complex formation:

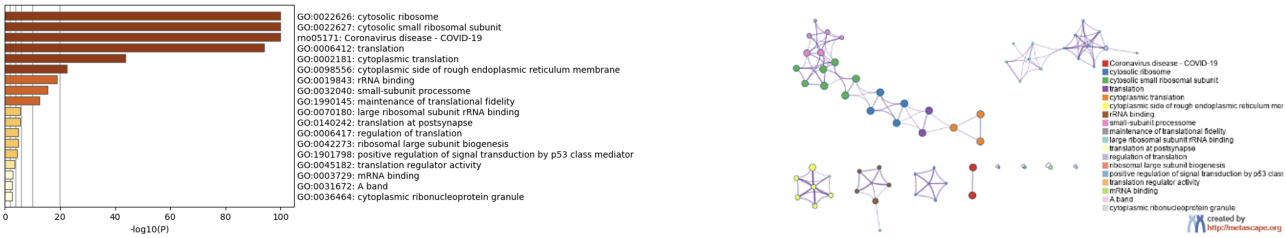
a. For *Schizosaccharomyces pombe*, the enrichment analysis shows strong association with translation processes. Figure 8a demonstrates that 140 unique genes were mapped with high statistical significance to the Cytosolic Ribosome (GO:0022626) and KEGG Ribosome (spo03010) pathways. Topologically, the enrichment network (Figure 8b) visualizes this cohesive structure, highlighting highly interconnected modules such as ribosome assembly (red cluster) and cytosolic ribosomes (dark blue cluster).




**(a)** **(b)**

**Figure 8.** Enrichment analysis for *Schizosaccharomyces pombe*. (a) Top GO/KEGG pathways by significance. (b) Functional network showing ribosomal translation.

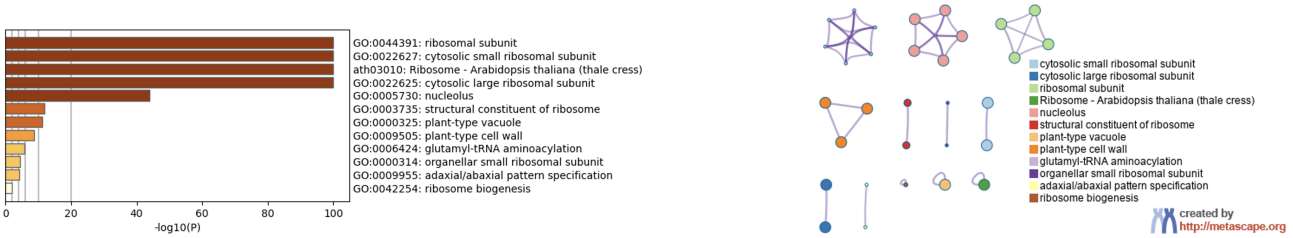
b. For *Rattus norvegicus*, the analysis indicates an intersection between basic cellular machinery and pathogenic pathways. Figure 9a shows 91 proteins (78.45%) significantly mapped to the COVID-19 pathway (rno05171), which may reflect known mechanisms of viral interaction with host ribosomes. Topologically, the network plot (Figure 9b) illustrates this COVID-19 pathway (red cluster) operating alongside densely interconnected ribosomal machinery (blue and green clusters).




**(a)** **(b)**

**Figure 9.** Enrichment analysis for *Rattus norvegicus*. (a) Top GO/KEGG pathways by significance. (b) Functional network showing ribosomal and COVID-19 clusters..

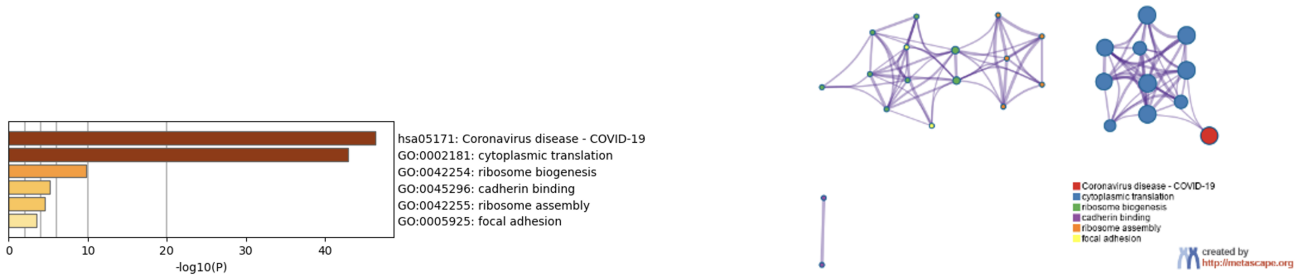
c. For *Arabidopsis thaliana*, the results highlight conserved cellular machinery alongside kingdom-specific structural associations. Figure 10a shows significant mapping to plant-specific structures, including plant vacuoles, cell walls, and the adaxial/abaxial pattern specification process (GO:0009955). The network plot (Figure 10b) reflects this diversity, displaying distinct functional clusters for plant-type cell walls (orange cluster) operating alongside conserved nucleolar subunits.




**(a)** **(b)**

**Figure 10.** Enrichment analysis for *Arabidopsis thaliana*. (a) Top GO/KEGG pathways by significance. (b) Functional network showing plant-specific structural modules.

d. For Homo sapiens, the model demonstrated significant search space reduction (99.87%), retaining a core of 26 genes. As shown in Figure 11a, 23 of these genes (92%) were highly enriched for the COVID-19 pathway (hsa05171). The functional network (Figure 11b) visually reinforces this topological relationship, showing the COVID-19 module (red node) structurally anchored to the human cytoplasmic translation network (blue cluster).




**(a)** **(b)**

**Figure 11.** Enrichment analysis for Homo sapiens. (a) Top GO/KEGG pathways ranked by significance. (b) Functional network highlighting the human translation machinery. .

## 4. CONCLUSION

This study introduces a two-stage graph pruning framework that mitigates the NP-hard computational complexity of the Maximum Clique Problem (MCP) in Protein-Protein Interaction Networks (PPINs). Mathematically, the framework contributes a safe topological reduction strategy by combining a dynamic $k$-core lower bound with a Particle Swarm Optimized Graph Attention Network (GAT-PSO). By computing anisotropic attention weights across four complementary centrality metrics, the model effectively isolates true clique members and avoids the dense-core bias inherent in pure statistical filtering.

Evaluated on 11,130 valid networks, the proposed strategy reduced the topological search space by an average of 95.87% for nodes and 91.11% for edges, accelerating the exact MaxCliqueDyn solver by up to 106.73 times. Crucially, this substantial dimensionality reduction was achieved while preserving topological integrity. The pruned networks maintained a clique size similarity of 97.23% and a Jaccard Index of 86.70%. Furthermore, functional enrichment analysis confirmed that the retained modules align with established biological pathways, proving the framework's viability as a structurally aware filter for complex network analysis.

Despite these results, the study is bounded by the computational timeout of the exact solver on 1,405 ultra-dense networks, leaving the most extreme combinatorial cases unresolved. Additionally, the computational cost limited the PSO hyperparameter search budget. Future research should explore distributed computing frameworks to scale the model for these extreme topologies and extend the centrality-aware GAT-PSO architecture to evaluate weighted and dynamic biological networks.

## ACKNOWLEDGEMENT

This research was funded by the Indonesia Endowment Fund for Education under the Ministry of Finance of the Republic of Indonesia (ID 202405111702776). The author would like to express gratitude to Prof. Dr.Eng. Wisnu Ananta Kusuma, S.T., M.T., for facilitating and supporting the use of the NVIDIA DGX H100 computing server. The author also extends appreciation to the Computer Science Study Program, IPB University, for providing access to the NVIDIA GeForce RTX 3070 computing server used to conduct the experiments in this study.

## REFERENCES


[1] Q. Wu and J.-K. Hao, "A review on algorithms for maximum clique problems," *European Journal of Operational Research*, vol. 242, no. 3, pp. 693–709, May 2015, https://doi.org/10.1016/j.ejor.2014.09.064.

[2] M. Zhou, Q. Zeng, and P. Guo, "Smc-brb: an algorithm for the maximum clique problem over large and sparse graphs with the upper bound via s+-index," *The Journal of Supercomputing*, vol. 79, no. 7, pp. 8026–8047, May 2023, https://doi.org/10.1007/s11227-022-04982-7.

[3] K. H. Rosen, *Discrete Mathematics and Its Applications*, 8th ed. New York, NY: McGraw Hill, 2018.

[4] S. Rasti and C. Vogiatzis, "A survey of computational methods in protein–protein interaction networks," *Annals of Operations Research*, vol. 276, no. 1–2, pp. 35–87, May 2019, https://doi.org/10.1007/s10479-018-2956-2.

[5] S. Wang, H. Cui, Y. Qu, and Y. Zhang, "Multi-source biological knowledge-guided hypergraph spatiotemporal subnetwork embedding for protein complex identification," *Briefings in Bioinformatics*, vol. 26, no. 1, 2025, https://doi.org/10.1093/bib/bbae718.

[6] R. H. Ramos, C. de Oliveira Lage Ferreira, and A. Simao, "Human protein–protein interaction networks: A topological comparison review," *Heliyon*, vol. 10, no. 5, p. e27278, 2024, https://doi.org/10.1016/j.heliyon.2024.e27278.

[7] J. Konc and D. Janežič, "An improved branch and bound algorithm for the maximum clique problem," *MATCH Communications in Mathematical and in Computer Chemistry*, vol. 58, pp. 569–590, 2007.

[8] P. S. Segundo, A. Lopez, and P. M. Pardalos, "A new exact maximum clique algorithm for large and massive sparse graphs," *Computers & Operations Research*, vol. 66, pp. 81–94, 2016, https://doi.org/10.1016/j.cor.2015.07.013.

[9] L. F. R. Ribeiro, P. H. P. Saverese, and D. R. Figueiredo, "struc2vec," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 385–394, https://doi.org/10.1145/3097983.3098061.

[10] V. Carchiolo, M. Grassia, M. Malgeri, and G. Mangioni, "Geometric deep learning sub-network extraction for maximum clique enumeration," *PLoS ONE*, vol. 19, no. 1, p. e0296185, 2024, https://doi.org/10.1371/journal.pone.0296185.

[11] S. Agarwal, C. M. Deane, M. A. Porter, and N. S. Jones, "Revisiting date and party hubs: Novel approaches to role assignment in protein interaction networks," *PLoS Computational Biology*, vol. 6, no. 6, pp. 1–12, 2010, https://doi.org/10.1371/journal.pcbi.1000817.

[12] K. Erciyes, "Graph-theoretical analysis of biological networks: A survey," *Computation*, vol. 11, no. 10, p. 188, 2023, https://doi.org/10.3390/computation11100188.

[13] M. Saberi and S. Aref, "Evaluating global measures of network centralization: Axiomatic and numerical assessments," *PLoS ONE*, vol. 21, no. 2, p. e0341630, 2026, https://doi.org/10.1371/journal.pone.0341630.

[14] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[15] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018, https://openreview.net/forum?id=rJXMpikCZ.

[16] P. K. Yadalam, S. Ayyachamy, P. M. Natarajan, and C. M. Ardila, "Graph attention networks-based prediction of microrna-disease causality in head and neck neoplasms," *Scientific Reports*, vol. 15, no. 1, p. 40237, 2025, https://doi.org/10.1038/s41598-025-24130-4.

[17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, 1995, pp. 1942–1948, https://doi.org/10.1109/ICNN.1995.488968.

[18] Y. Zhou *et al.*, "Metascape provides a biologist-oriented resource for the analysis of systems-level datasets," *Nature Communications*, vol. 10, no. 1, 2019, https://doi.org/10.1038/s41467-019-09234-6.

[19] D. Szklarczyk *et al.*, "The string database in 2023: protein–protein association networks and functional enrichment analyses for any sequenced genome of interest," *Nucleic Acids Research*, vol. 51, no. D1, pp. D638–D646, 2023, https://doi.org/10.1093/nar/gkac1000.

[20] L. V. Bozhilova, A. V. Whitmore, J. Wray, G. Reinert, and C. M. Deane, "Measuring rank robustness in scored protein interaction networks," *BMC Bioinformatics*, vol. 20, no. 1, p. 446, 2019, https://doi.org/10.1186/s12859-019-3036-6.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR Conference Track Proceedings*, 2015, http://arxiv.org/abs/1412.6980.

[22] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009, https://doi.org/10.1109/TKDE.2008.239.

[23] S. Jadon, "A survey of loss functions for semantic segmentation," in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 2020, pp. 1–7, https://doi.org/10.1109/CIBCB48159.2020.9277638.

[24] X. Fang, J. Li, M. Wang, A. Chen, S. Shao, and Q. Liu, "A universal urban flood risk model based on particle-swarm-optimization-enhanced spiking graph convolutional networks," *Sustainability*, vol. 17, no. 22, p. 9973, 2025, https://doi.org/10.3390/su17229973.

[25] A. Tharwat, "Classification assessment methods," *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2018, https://doi.org/10.1016/j.aci.2018.08.003.

[26] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 3121–3124, https://doi.org/10.1109/ICPR.2010.764.

[27] B. Pattabiraman, M. M. A. Patwary, A. H. Gebremedhin, W. Liao, and A. Choudhary, "Fast algorithms for the maximum clique problem on massive sparse graphs," in *Internet Mathematics*, vol. 11, no. 4, 2013, pp. 156–169, https://doi.org/10.1007/978-3-319-03536-9_13.

[28] C. W. Arachchi and N. Tatti, "Jaccard-constrained dense subgraph discovery," *Machine Learning*, vol. 113, no. 9, pp. 7103–7125, 2024, https://doi.org/10.1007/s10994-024-06595-y.

[29] W. Deng, W. Zheng, and H. Cheng, "Accelerating maximal clique enumeration via graph reduction," *Proceedings of the VLDB Endowment*, vol. 17, no. 10, pp. 2419–2431, 2024, https://doi.org/10.14778/3675034.3675036.

[30] J. D. Eblen, C. A. Phillips, G. L. Rogers, and M. A. Langston, "The maximum clique enumeration problem: algorithms, applications, and implementations," *BMC Bioinformatics*, vol. 13, no. S10, p. S5, 2012, https://doi.org/10.1186/1471-2105-13-S10-S5.

[31] E. M. Hanna, N. Zaki, and A. Amin, "Detecting protein complexes in protein interaction networks modeled as gene expression biclusters," *PLoS ONE*, vol. 10, no. 12, p. e0144163, 2015, https://doi.org/10.1371/journal.pone.0144163.