Community preserving sparsification based on K-core for enhanced community detection in attributed networks

Tedy Setiadi [a,b,1,*], Mohd Ridzwan Yaakub [b,2], Azuraliza Abu Bakar [b,3]

[a] Faculty of IndustrialTechnology, Universiti Ahmad Dahlan, Yogyakarta,Indonesia
[b] Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Selangor, Malaysia
[1] tedy.setiadi@tif.uad.ac.id; [2] ridzwanyaakub@ukm.edu.my; [3] azuraliza@ukm.edu.my
* corresponding author

ARTICLE INFO

Article history
Received September 5, 2025
Revised October 5, 2025
Accepted October 17, 2025
Available online November 17, 2025

Keywords
Community detection
Attributed networks
Graph sparsification
K-Core decomposition
Scalability

ABSTRACT

Community detection is an important aspect of complex network analysis, especially in attribute networks where topological structure and attribute information both play a role in community formation. Traditional structure-based methods tend to result in topologically dense but semantically inconsistent communities, while attribute-based approaches can improve semantic coherence but face scalability constraints and high computational costs. On the other hand, graph sparsification techniques have been used to reduce the size of the network, but most focus on structural aspects alone and rarely consider attributes, so the quality of the resulting community is often degraded. This study proposes CPSK (Community Preserving Sparsification based on K-core), a sparsification framework that combines k-core decomposition with attribute-based side weighting. This approach is designed specifically for attribute networks, with the aim of maintaining a balance between structural reduction and community semantic consistency, while improving the efficiency of the detection process. Evaluation of the six datasets showed that CPSK consistently generates more stable and meaningful communities than existing attribute-based community detection methods, while maintaining an edge in computing efficiency on large-scale and heterogeneous networks.

1. Introduction

Complex networks consist of many entities interconnected by intricate relationships. They are commonly represented as graphs, where nodes denote individuals and edges represent their interactions. In reality, various types of complex networks exist, including social networks, public health networks, transportation networks, power grids, and many others. Community structure is one of the most significant characteristics of complex networks, as identifying these structures helps us better understand network organization and uncover valuable information. To this end, numerous community detection methods have been proposed [1]. The goal of community detection is to reveal groups of nodes within a complex network that exhibit strong internal connectivity and weak external connections. For instance, community detection can help identify friendship circles in social media [2], uncover high-risk groups in public health networks to prevent the spread of epidemics [3], reveal densely connected routes for optimizing transportation systems and reducing congestion [4], and user segmentation based on borrowing patterns to support more personalized collection recommendations in libraries [5].

A variety of approaches have been developed for community detection. Classical methods such as hierarchical clustering [6], [7], modularity maximization [8], [9], non-negative matrix factorization [10], and clustering-based algorithms [11] primarily rely on structural information while ignoring node attributes. To overcome this limitation, several attribute-based community detection methods have been proposed, including label propagation approaches [12], [13], mathematical programming [14], unsupervised network embedding that combines graph attention auto-encoders, modularity maximization, and self-training clustering [15]. These methods are further expanded by others, such as extensions of Louvain that incorporate homophily [16] and multi-view graph convolutional networks that integrate both structural and attribute information [17]. Although these methods are effective in leveraging attributes by calculating node similarity based on both structural and attribute information, they typically consider only local similarity. Consequently, two nodes may appear similar locally but belong to different communities in the global structure, leading to less cohesive communities overall because broader topological patterns are not captured. Furthermore, attributed community detection often suffers from scalability issues, as integrating structural and attribute information in large-scale networks substantially increases computational complexity

To address these challenges, one widely considered strategy is graph sparsification. This approach not only reduces the complexity of large-scale networks but also has the potential to improve the quality of community detection. In the context of community detection, sparsification offers advantages by lowering computational complexity and clarifying hidden community structures. A variety of methods have been proposed, including link pruning [18], [19], spanner-based sparsification [20], spectral sparsification [21], effective resistance–based approaches [22], and edge sampling [23]. However, most of these approaches are not explicitly designed to preserve community structures, often resulting in degraded detection quality. This highlights the need for sparsification methods that are more community-preservation-oriented, particularly in attributed networks, to enhance community-detection quality while maintaining efficiency at scale.

Although graph sparsification has proven effective in improving the scalability of network analysis, most existing approaches remain focused on topological aspects while neglecting node attributes. As a result, the detected communities are often structurally cohesive but semantically fragmented [23], [24]. Recent studies have attempted to incorporate attributes into the community detection process [25], [26], yet such efforts are generally separate from sparsification strategies. In fact, preserving the network's core connectivity through sparsification while simultaneously maintaining attribute homogeneity represents two essential requirements that should be addressed together. This highlights the need for a comprehensive approach that integrates both structural and attribute information within a consistent framework.

Motivated by these observations, we propose Community Preserving Sparsification based on k-core (CPSK), a framework that integrates scalability, preservation of core structures, and semantic cohesion of communities. CPSK is designed to overcome the limitations of conventional sparsification methods by maintaining the core backbone of communities through k-core decomposition while simultaneously preserving attribute homogeneity across nodes. The contributions of this study are threefold:(i) we introduce the CPSK framework, which employs k-core–based sparsification to preserve the structural backbone of communities while maintaining overall network connectivity;(ii) we integrate node attribute information into the sparsification process through edge weighting and preliminary community mapping to enhance semantic consistency; and (iii) we present an extensive evaluation of the framework's effectiveness on various real-world datasets, considering both community quality and computational efficiency. The novelty of CPSK lies in integrating structural and attribute information directly within the sparsification step rather than applying sparsification and attribute-aware detection as separate stages. Finally, we show empirically that this integration improves scalability and yields robust results across parameter settings.

2. Related Works

Community detection has evolved along two major directions: structure-driven and attribute-aware methods. Classical approaches such as Louvain [8], Label Propagation [27], spectral algorithms [28], and Stochastic Block Models [29] have been refined over time through improvements like Leiden [30], overlapping detection via deep learning [31], and multi-resolution strategies [32], extending their applicability to large and dynamic networks [33]. These methods scale efficiently but rely exclusively on topological cues, often overlooking semantic consistency. In contrast, attribute-aware approaches explicitly integrate node features, such as SAC [34], EVA [16], MAGCN [17], CDSSA [35], and embedding-based frameworks [36]. While these methods achieve stronger semantic alignment, they introduce heavy computational overhead and face scalability challenges on large heterogeneous graphs. This contrast reveals a persistent trade-off between scalability and semantic coherence, motivating the exploration of graph sparsification as a graph-processing strategy to improve efficiency while preserving essential structural properties.

Various graph sparsification studies to find community structure, including similarity-based edge sparsification approaches [37], which rank and retain edges likely to belong to the same cluster but may compromise community integrity. More advanced approaches leveraged k-core decomposition to retain dense subgraphs by removing low-degree nodes [38]–[40], while edge-pruning techniques based on clustering coefficients, triangle participation [41], or betweenness centrality [42] sought to eliminate noisy or redundant links selectively. Spectral sparsification methods aimed to preserve Laplacian properties critical for modularity optimization [43], and more recent innovations include curvature-based sparsification [44] and motif-based filtering [45], both of which demonstrate improved preservation of mesoscopic structures. Despite recent progress, most sparsification methods remain structure-oriented, overlooking node attributes and peripheral connectivity [46]. While k-core decomposition effectively preserves the backbone of communities [43], excessive focus on dense cores often neglects the role of low-degree nodes, which are essential in core–periphery structures to prevent fragmentation [47].

Hybrid approaches have been proposed to combine the efficiency of graph sparsification with the semantic consistency of attribute-aware methods, representing early attempts toward a unified framework. For instance, KnnA constructs k-nearest-neighbor subgraphs based on attribute similarity to reduce redundancy [48], while SAS-LP modifies label propagation by incorporating both structural and attribute weights [25]. Other studies integrate spectral sparsification with attribute weighting to preserve modularity while filtering irrelevant links [49], and embedding-based methods have also been used to fuse structural and attribute spaces into simplified representations [26]. These approaches reflect a shift toward integration, yet most still treat structure and attributes separately or rely on global filtering. Consequently, they add computational overhead, reduce stability, and neglect low-degree nodes, highlighting the need for a more comprehensive solution.

Recent advances in attributed graph sparsification and scalable community detection have increasingly leveraged neural network-based approaches to balance efficiency and semantic consistency. Supervised sparsifiers, such as SGS-GNN, reduce computational overhead while maintaining high task performance [50], and models such as ConNet employ cross-attention to capture structural and attribute interactions for scalable community search [51]. Hybrid approaches like GATFELPA integrate graph attention with enhanced label propagation to improve community separation and mitigate over-smoothing [52]. In contrast, FS-GNN simultaneously enhances fairness and efficiency through feature and structure-aware sparsification [53]. These trends underscore the relevance of the proposed CPSK, which integrates k-core-based sparsification with attribute weighting and explicit core–periphery preservation to achieve both scalability and semantic cohesion.

In summary, graph sparsification methods enhance scalability but overlook semantic attributes, while attribute-aware approaches improve semantic consistency yet often face efficiency challenges. Hybrid strategies attempt to combine both aspects, but they still fall short in jointly preserving core structures, attribute homogeneity, and global connectivity. These limitations point to the need for a framework that begins with attribute weighting and then applies k-core-based sparsification to integrate structural and semantic preservation within a unified approach. To better highlight the novelty of our approach, Table 1. provides a comparative summary between CPSK and several representative hybrid methods.

Table 1. Comparison of CPSK with representative hybrid sparsification methods

Method	Attribute-aware?	Sparsification / Integration Strategy	Advantages	Relative performance notes
SAC	Yes	Attribute clustering + modularity maximization	Strong semantic cohesion	High purity, but poor scalability; fails on very large graphs
EVA	Yes, homophily integration	Louvain extension with attribute weighting	Balances structure + homophily	Good entropy/purity; weaker on modularity in large heterogeneous networks
MAGCN	Yes, GCN embedding	Multi-view graph convolutional network	Learns deep embeddings; strong semantic alignment	High computational cost; less scalable to large graphs
KnnA	Yes, Knn from attributes	Build Knn subgraph based on attribute similarity	Strong local attribute homogeneity	Performs well under attribute homophily; weak at backbone preservation
SAS-LP	Yes, structural + attribute weights	Modified label propagation with structural + attribute weights	Fast; suitable for LP-friendly networks	Excels on small or LP-suited datasets; less stable on large graphs
CDSSA	Yes	Attribute + structural similarity with adaptive weighting	Good semantic preservation	Stable on medium-scale graphs; scalability limited
CPSK (proposed)	**Yes, linear weighting (α) + prepartition**	**k-core backbone + intra-community edge sampling (θ) + core–periphery reconnection**	**Preserves backbone & attributes; efficient at large scale**	**High modularity & purity across most datasets**

The comparison focuses on whether each method incorporates attribute information, the main sparsification or integration strategy, the ability to explicitly preserve core/periphery structures, and their typical performance characteristics. This overview shows that CPSK uniquely combines k-core–based sparsification with attribute-aware weighting and explicit core–periphery preservation, thereby achieving both scalability and semantic cohesion more effectively than prior hybrids.

3. Method

3.1. Dataset

For the evaluation, we used six datasets of networks with various attributes ranging from small to large scales, namely (i) Karate, a small network of friendship relationships between karate club members consisting of 34 nodes and 78 edges, (ii) Polblogs containing a network of political blogs consisting of 1,222 nodes and 16,714 edges, (iii) Citeseer, a network of citations consisting of 3,327 nodes and 4,732 edges, each node attribute in the form of a bag-of-word representation of the content of the publication, (iv)Cora, a citation network with 2,708 nodes and 5,429 edges, node attributes in the form of word vectors extracted from publication abstracts, (v) IMDB, the Movie Database network consists of 19,000 nodes and 96,000 edges, with textual and categorical attributes and (vi) Gplus, a social network with 107,614 nodes and 13,673,453 edges, with user profile attributes. The Karate and Polblogs data sets are small-scale networks; Citeseer and Cora are medium-scale, while IMDB and GPlus are large-scale networks.

3.2. Proposed Framework

We propose the Community Preserving Sparsification Based on K-core (CP-SK) framework, a preprocessing stage for community detection in attribute networks. The basic idea is to selectively reduce the graph by concentrating intra-communities and preserving core–periphery connectivity, so that community detection in the resulting graph is more efficient while maintaining or even improving its structural quality and semantic coherence. The CP-SK framework includes three stages, as shown in Fig.1. consists of: (1) Preprocessing, (2) Core–Periphery-Preserving k-Core Sparsification, and (3) Community Detection.

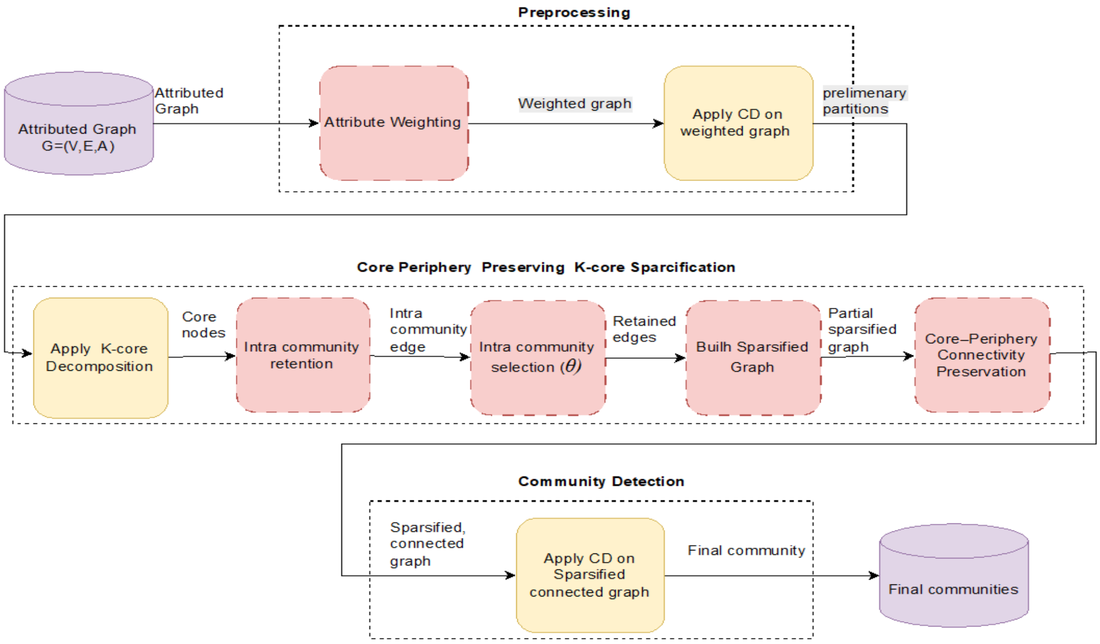


Fig. 1. Community Preserving Sparsification Based on K-core

Given an attributed network $G=(V,E,A)$, where is the set of nodes, E the set of edges, and $A=\{a_v|v\in V\}$the node attribute set, two hyper-parameters are used: the degree threshold k for k-core selection and the retention ratio $\theta \in(0,1]$ to control the sparsification size.

3.2.1. Preprocessing

3.2.1.1. Attribute Weighting

This initial step aims to combine the structural and semantic information of the graph, where the weights of each side are defined as a linear combination of *baseweights* that represent the existence of structural connections and similarity of attributes between nodes.

$$w(u,v) = \alpha.base\_weight + (1-\alpha).Attribute\_similarity \quad (1)$$

where $0 \le \alpha \le 1$, *base_weight* = 1 If the edge is connected and 0 is not, then the similarity attribute with the Jaccard similarity on the equation (2).

$$jaccard(u,v) = \frac{|A_u \cap A_v|}{|A_u \cup A_v|} \quad (2)$$

So that if the edge is connected, the weight of the edge becomes.

$$w(u,v) = \alpha + (1-\alpha).\frac{|A_u \cap A_v|}{|A_u \cup A_v|} \quad (3)$$

3.2.1.2. Apply CD to the weight graph

Next, we perform community detection using Louvain to obtain an initial partition, which guides the identification of potential intra-community edges. This partition serves only as a heuristic and does not constrain the final community structure, so the pruned network can be analyzed with any community detection method.

3.2.2. Core Periphery Preserving K-core Sparsification

3.2.2.1. Apply k-core decomposition

K-core decomposition is applied to extract the dense structural core (the maximal subgraph G_k), by computing the k-core, G_k=(V_k, E_k, A_k) where.

$$\mathrm{V}_k = v \in V | deg_{G_k}(v) \geq k, \mathrm{E}_k = (u, v) \in E | u, v \in \mathrm{V}_k \tag{4}$$

This core forms the structural anchor of the sparsified graph, while peripheral nodes with weaker connectivity are candidates for pruning.

3.2.2.2. Intra-community edge retention

Based on the preliminary partition Π_{pre}, we retain edges whose endpoints belong to the same community.

$$E_{intra} = (u, v) \in E: \Pi_{pre}(u) = \Pi_{pre}(v) \tag{5}$$

This restriction concentrates on edges that are structurally consistent with intra-community relations

3.2.2.3. Intra-community edge selection ($\boldsymbol{\theta}$)

To enforce sparsity, only a fraction θ of the intra-community edges is kept, forming.

$$E_{keep} \subseteq E_{intra}, and \left|E_{keep}\right| = \theta. \left|E_{intra}\right| \tag{6}$$

In the baseline implementation, edges are currently selected randomly according to θ, which may introduce some variability across runs. However, this approach is not binding, and alternative strategies such as weight or attribute-prioritized selection can also be applied within the CPSK framework. This ensures that the edge selection process remains flexible and the method robust, while maintaining both structural backbone and semantic consistency.

3.2.2.4. Build Sparsified graph

The reduced graph H is built from the core node set V_k and the retained edges. Because edges naturally bring their endpoints, some peripheral nodes are included if they are incident to selected edges, producing a compact subgraph concentrated around the core.

3.2.2.5. Core–periphery connectivity preservation

To avoid isolating the core, each core node $u \in V_k$ is reconnected with its original neighbors missing in H, along with their edges. This step preserves short paths between core and periphery, ensuring the sparsified graph remains both connected and representative. The output of Stage 2 is a sparsified graph H that emphasizes intra-community structure while retaining core–periphery accessibility.

3.2.3. Apply CD on Sparsified connected graph

Finally, we implemented this stage with Louvain method (although any off-the-shelf detector method can be used without changing the framework community detection) performed on the sparsified graph H, yielding the final partition. Running detection on H reduces runtime and often improves structural quality by eliminating noisy cross-community edges. Evaluation of structural quality (e.g., modularity) and semantic consistency (e.g., attribute purity and entropy) is reported in the experimental section.

3.3. Evaluation Metrics

The evaluation metrics we use for evaluation include:

- Modularity to assess the quality of community structure in the form of intra-community density relative to random connections (the higher the value, the better), the formula is:

$$Q_c = \sum_{c=1}^{n_c} \left[\frac{k_c^{in}}{2M} - \left(\frac{k_c}{2M} \right)^2 \right] \tag{7}$$

where k_c^{in} is the number of intra-community edges in the community, k_c is the total degree (intra-community and inter-community edges) of the community, n_c is the number of communities, and M is the total number of edges in the graph.

- Purity to measure the homogeneity of attributes in a community (the higher the value, the better), the formula is.

$$P_c = \prod_{a \in A} \frac{max\left(\sum_{v \in c}\{a(v)\right)\}}{|c|} \tag{8}$$

where A is the label set, $a \in A$ is a label, $a(v)$ is an indicator function that takes the value 1 if $a \in A(v)$.

- Entropy to assess attribute uniformity (the lower the better), the formula is.

$$H(C_i) = -\sum_{j=1}^{m} P_{ij} log\, P_{ij} \tag{9}$$

where m is the number of distinct attribute classes within the community, and represents the proportion of nodes in community that belong to class j. We pair purity and entropy to capture semantic cohesion in a complementary way.

- Normalized Mutual Information (NMI) to assess the suitability of the results with ground-truth (the higher the value, the better), the formula is:

$$NMI(A, B) = \frac{2 * I(A,B)}{H(A) + H(B)} \tag{10}$$

where $I(A, B)$ is the mutual information between the two partitions A and B, $H(A)$ and $H(B)$ are the entropies of partitions A and B, respectively.

- Execution time as an indicator of the efficiency of the entire pipeline (the lower the value, the better), measured in terms of wall-clock time.

3.4. Experimental Setup

The performance of the proposed CPSK framework is compared against six state-of-the-art methods: SAC [34], Eva [16] , MAGCN [17], KnnA [48], CDSSA [35], and SAS-LP [25].

For fairness, all methods are implemented in Python with standardized hyperparameter settings, including the same k-value for k-core decomposition, uniform clustering parameters, and identical maximum iteration limits for label propagation. Results are averaged across 10 independent runs to mitigate noise and variability.

In addition, we perform a sensitivity analysis of CPSK by varying three key parameters: the sparsification size (θ), the kernel threshold (k), and the attribute-structure trade-off (α). The purpose of this analysis is to examine the effects of these parameters on community quality. This approach allows us to demonstrate the robustness of CPSK across a range of settings and highlight how changes to each parameter affect the balance between structural preservation and semantic cohesion.

4. Results and Discussion

4.1. Modularity Performance

Fig. 2 shows that CPSK consistently achieves the highest modularity across all six datasets (Karate, Polblogs, Citeseer, Cora, IMDb, and Gplus).

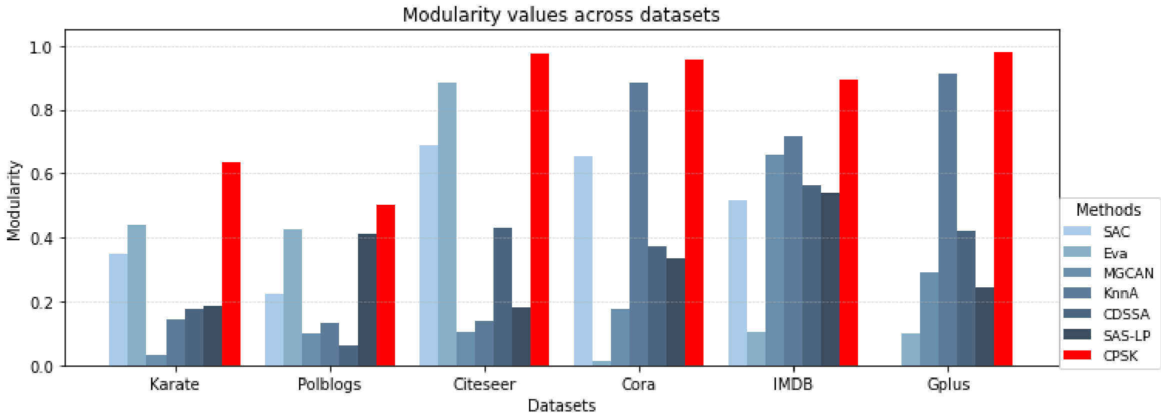


Fig. 2. Modularity values for comparison methods on real-world datasets

In medium- and large-scale networks such as Citeseer, Cora, IMDb, and Gplus, CPSK demonstrates a clear advantage over competing methods, with KnnA ranking second but at a noticeable distance. On smaller datasets such as Karate and Polblogs, CPSK still maintains the best performance, though the margin over EVA is narrower. These results indicate that the proposed framework is effective at preserving dense intra-community structures across network scales. The integration of core–periphery preservation with attribute-based weighting enables CPSK to retain structural backbone quality even after sparsification, resulting in consistently higher modularity than all baseline methods.

4.2. Clustering Accuracy (NMI)

Fig. 3 demonstrates that CPSK achieves the highest NMI in five out of six datasets (Karate, Citeseer, Cora, IMDb, and Gplus), confirming its effectiveness in aligning detected communities with ground truth. The only exception is Polblogs, where SAS-LP slightly outperforms CPSK, reflecting the suitability of label propagation for this network type. On smaller datasets like Karate, the advantage of CPSK over CDSSA and SAS-LP is marginal, whereas on Citeseer and Cora, CPSK slightly surpasses CDSSA. In IMDb, CPSK shows the clearest margin of superiority, and in Gplus, it remains the best despite all methods producing relatively low scores. These results indicate that the combination of core–periphery preservation and attribute weighting in CPSK enhances semantic consistency across various network scales, while highlighting that specific network characteristics may still favor alternative approaches such as SAS-LP.

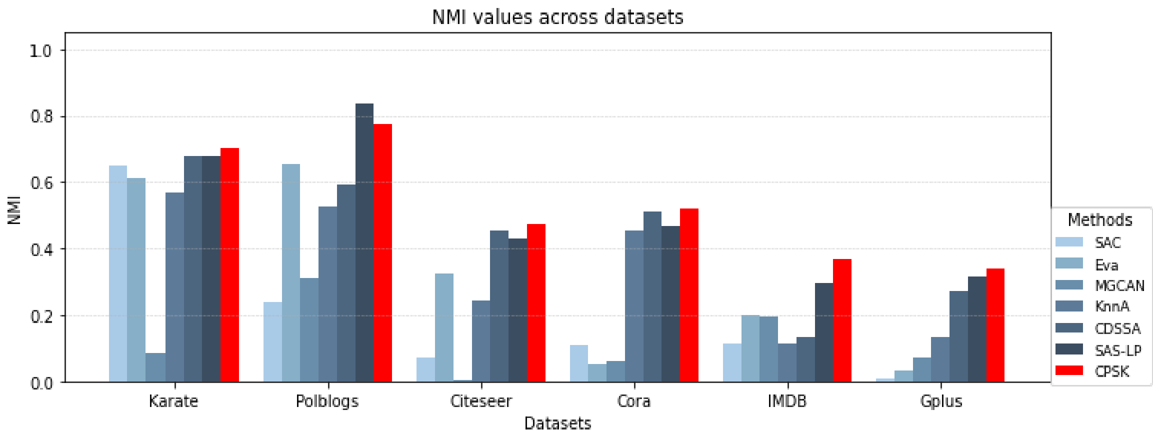


Fig. 3. NMI values for comparison methods on real-world datasets

4.3. Computational Efficiency

Fig. 4 shows that CPSK achieves notable time efficiency across all datasets. It records the fastest execution on Karate, Citeseer, Cora, and Gplus, while on Polblogs and IMDb it ranks slightly below MAGCN and CDSSA, which are the fastest on those datasets, respectively. In contrast, SAC consistently performs the worst and even fails to scale on Gplus. KnnA and MAGCN also show sharp increases in runtime on larger networks, confirming scalability limitations. These results suggest that the core-periphery-preserving mechanism in CPSK substantially reduces computational load without compromising quality, particularly in large-scale attributed networks.

Fig. 4. Runtime values for comparison methods on real-world datasets

4.4. Semantic Cohesion (Purity and Entropy)

The results of the semantic cohesion of the community are obtained through the purity and entropy of various data sets, as seen in Table 2 and Table 3.

Table 2. Purity scores across datasets for all methods

	karate	Polblogs	Citeseer	Cora	imdb	Gplus
SAC	0.9125	0.9595	0.4969	0.4513	0.7177	NA
Eva	0.9805	0.9224	0.8369	0.3021	0.5387	0.2049
MAGCN	0.607	0.5617	0.2399	0.3627	0.1462	0.2942
KnnA	0.26888	0.1766	0.2956	0.8369	0.7475	0.4141
CDSSA	0.9955	0.9734	0.9060	0.9455	0.9197	**0.8284**
SAS-LP	1.0000	0.9953	**0.9442**	0.9661	**0.9882**	0.8087
CPSK	**1.0000**	**0.9964**	0.9427	**0.9677**	0.9541	**0.8105**

Table 3. Entropy scores across datasets for all methods

	karate	Polblogs	Citeseer	Cora	imdb	Gplus
SAC	0.3201	0.2266	0.9398	0.74	0.7942	NA
Eva	0.0986	0.0468	0.4229	0.2904	0.1385	**0.3502**
MAGCN	0.284	0.1857	0.3544	0.71	0.8749	0.8049
KnnA	1.00	0.8749	0.7599	0.4229	0.7537	0.8682
CDSSA	0.0248	0.0704	0.2178	0.1372	0.2131	0.4128
SAS-LP	0.0051	0.0332	0.1461	**0.0988**	0.0596	0.5146
CPSK	**0.0014**	**0.0163**	**0.1417**	0.1166	**0.1236**	0.5639

CPSK consistently demonstrates very high purity across all datasets. The method achieves the highest scores on Karate (tied with SAS-LP), Polblogs, Cora, and Gplus; while on Citeseer and IMDb, CPSK ranks second with only a slight margin below SAS-LP. These results highlight CPSK's ability to preserve attribute homogeneity across networks of varying scales. In terms of entropy, CPSK produces the lowest values on Karate, Polblogs, and Citeseer. For Cora, CPSK ranks second, just below SAS-LP; similarly, for IMDb, CPSK also ranks second after SAS-LP. In the case of Gplus, a highly large and heterogeneous network, EVA achieves the lowest entropy, followed by CDSSA, SAS-LP, and CPSK.

The combination of core–periphery preservation and attribute-based weighting enables CPSK to maintain semantic cohesion within communities, as reflected in its high purity and low entropy across most datasets. The exception observed in highly heterogeneous networks (e.g., Gplus) suggests that approaches emphasizing homophily or label propagation, such as EVA or SAS-LP, may be better suited to certain topological and attribute characteristics.

4.5. Discussion

The advantages of CPSK across multiple metrics can be explained by its design, which combines core–periphery preservation with attribute-based weighting. The core–periphery preservation mechanism ensures that critical edges within the community backbone are retained, thereby maintaining consistently high modularity even in large networks, as internal community density is not significantly distorted by sparsification. An additional point of consideration concerns peripheral nodes. While pruning may oversimplify networks in which low-degree nodes still carry semantic weight, CPSK mitigates this effect through its explicit core–periphery reconnection step, which ensures peripheral nodes remain attached to the structural backbone. The integration of attributes into the weighting process directly enhances semantic quality, as reflected in the high purity and low entropy values, since nodes with similar attributes are more likely to be grouped consistently. This also explains the improvement in NMI, as CPSK's balanced treatment of structure and attributes enables detection results to align more closely with the ground truth than purely structure-based methods or label-propagation methods.

Nevertheless, it should be noted that NMI values remain consistently low on certain datasets, such as IMDB. This phenomenon is largely due to weak alignment between the provided ground-truth labels and the actual structural communities, which often overlap and display high attribute heterogeneity. In such cases, even though CPSK produces communities with high purity and low entropy, the NMI metric underestimates their quality. This reflects a limitation of the ground-truth labels rather than the CPSK framework itself and suggests that adaptive attribute weighting or multi-label strategies may be helpful for highly heterogeneous networks.

From a computational efficiency perspective, selective pruning at the periphery reduces the processing burden while preserving the community backbone, allowing CPSK to remain lightweight and scalable on large graphs, unlike other methods that suffer sharp increases in runtime. The slightly lower performance observed against specific methods on particular datasets (e.g., SAS-LP on Polblogs or EVA on Gplus) reflects the inherent characteristics of those networks, which are better suited to specialized approaches, such as networks with exceptionally strong homophily (favoring EVA) or those where label diffusion is particularly effective (favoring SAS-LP). Overall, CPSK demonstrates adaptive superiority by balancing structural and attribute preservation while ensuring computational efficiency. This balanced performance contrasts with earlier hybrid sparsification–attribute approaches, which often improve semantic consistency at the expense of scalability (e.g., KNN-Atr, SAS-LP), or prioritize efficiency at the expense of semantic quality. CPSK's integration of k-core pruning and attribute weighting enables it to mitigate this trade-off more effectively. It is also important to note that although Louvain is used in our experiments as the main detector, the CPSK framework is algorithm-agnostic and can be integrated with other community detection approaches, reinforcing its general applicability.

The dominant costs in CPSK are: (i) k-core decomposition, which runs in $O(|V|+|E|)$; (ii) attribute-based edge weighting and preliminary partitioning, which are linear in the number of edges (approximately $O(|E|\cdot s)$ where s denotes the average cost of computing attribute similarity); and (iii) intra-community selection (edge sampling), again proportional to $|E|$. Thus, the framework's main steps scale near-linearly in typical sparse graphs. For comparison, different baselines have different scaling characteristics: KnnA constructions can be $O(|V|2)$ without acceleration, label-propagation variants are often near-linear in $|E|$, spectral methods may require costly eigen computations, and MAGCN methods involve iterative message passing with costs proportional to the number of edges per epoch (and multiplied by the number of layers and training epochs). Importantly, CPSK's practical speedup comes from performing the final, often expensive, community detection on the reduced graph H (with

$|EH| \ll |E|$), which lowers the overall wall-clock time even when individual preprocessing steps are linear.

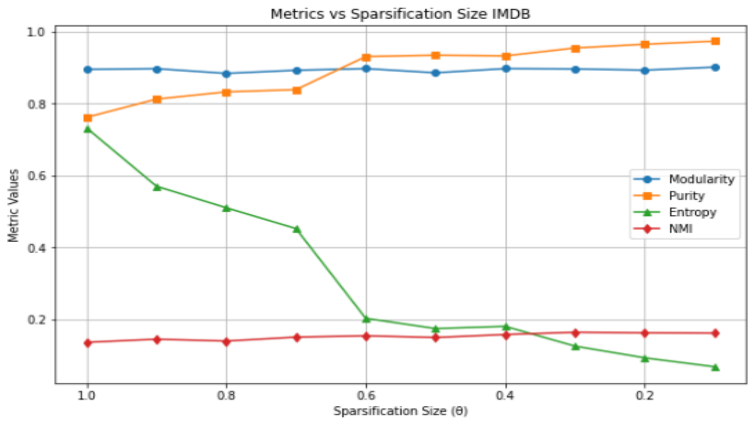
Beyond benchmark evaluations, CPSK achieves practical impact by pruning redundant edges while preserving the k-core backbone and attribute-consistent links, yielding a smaller yet semantically meaningful network. This enables social media platforms to identify cohesive user groups for content moderation, epidemiological networks to highlight key transmission pathways for targeted interventions, and recommendation systems to cluster users more effectively for personalization. These cases demonstrate how CPSK balances scalability with semantic cohesion in large, heterogeneous networks. While CPSK demonstrates superior performance across most datasets, we observed that in highly heterogeneous networks such as Gplus, methods like EVA or CDSSA may outperform CPSK on specific metrics such as entropy or purity. This is likely due to two factors: (i) the strong diversity of attributes and loose community structures, which favor label-propagation-based approaches; and (ii) the sparsification step emphasizing dense cores, which may reduce local heterogeneity patterns. To mitigate these limitations, adaptive parameter tuning (e.g., lowering θ for heterogeneous subgraphs) or non-uniform intra-edge selection strategies could help preserve more local diversity. These directions highlight opportunities for future extensions of CPSK to handle highly heterogeneous networks better.

4.6. Parameter Sensitivity Analysis

4.6.1. Sparsification Size ($\boldsymbol{\theta}$)

The sensitivity analysis of sparsification size reveals consistent patterns across both datasets. In Fig. 5(a) (IMDB), increasing the level of sparsification leads to purity approaching 1 and entropy decreasing significantly toward zero, while modularity remains relatively stable at a high level and NMI consistently low. This indicates that even as the graph is reduced, the community structure remains well preserved and semantic homogeneity among nodes improves.

(a)



(b)

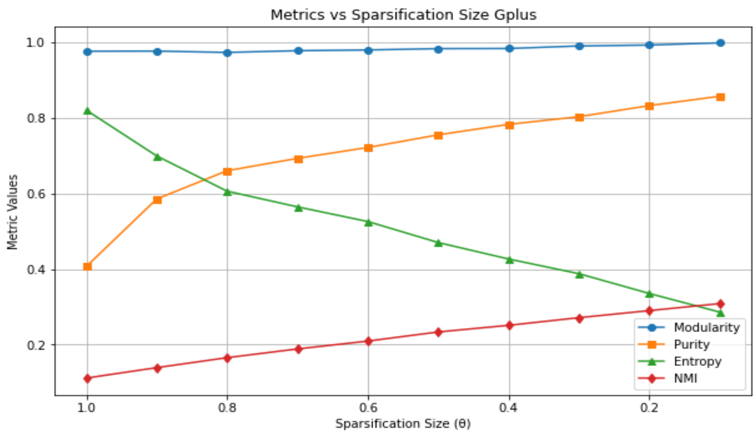
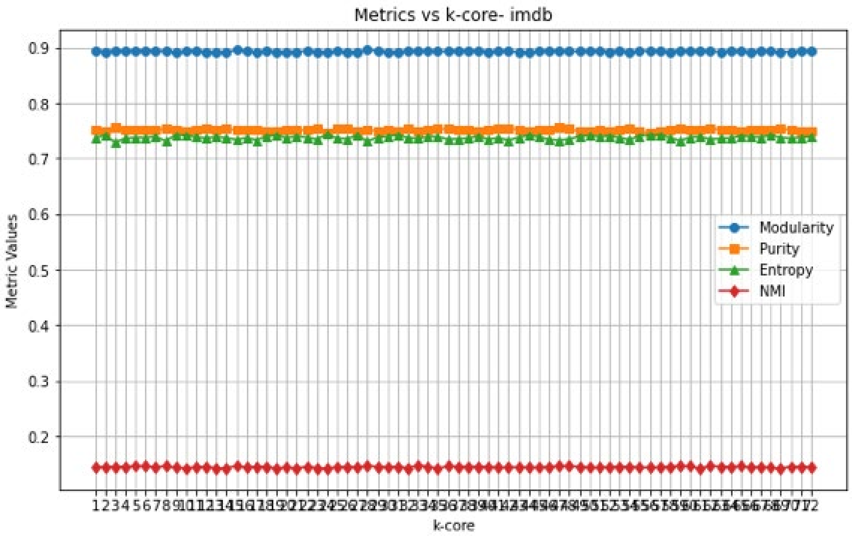


Fig. 5. Impact of Sparsification Size on Community Quality for (a) IMDB and (b) Gplus (k=2, α=0.5)

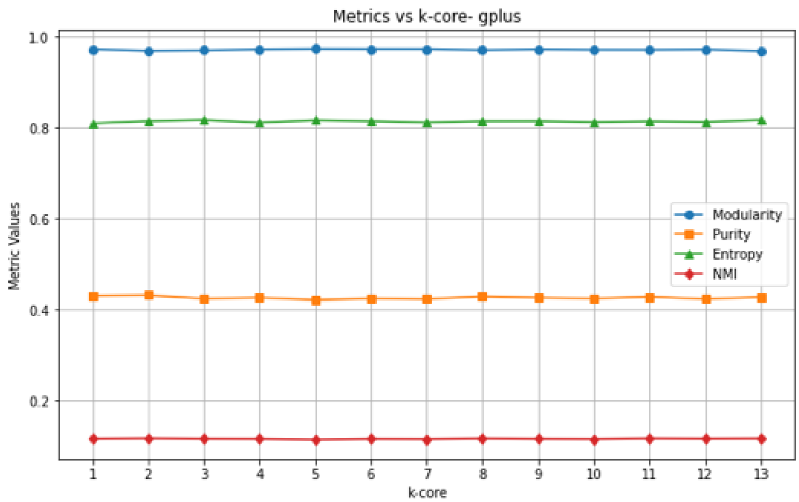
In Fig. 5(b) (Gplus), a similar trend is observed, with increasing purity and decreasing entropy, alongside a gradual rise in NMI, suggesting that community detection results become more consistent with the ground truth as sparsification increases. Overall, these findings demonstrate that sparsification can strengthen semantic cohesion and preserve modularity without compromising community detection quality.

4.6.2. Core Threshold (*k*)

Fig. 6 shows that variations in the k-core value do not significantly affect community quality in either dataset.



(a)



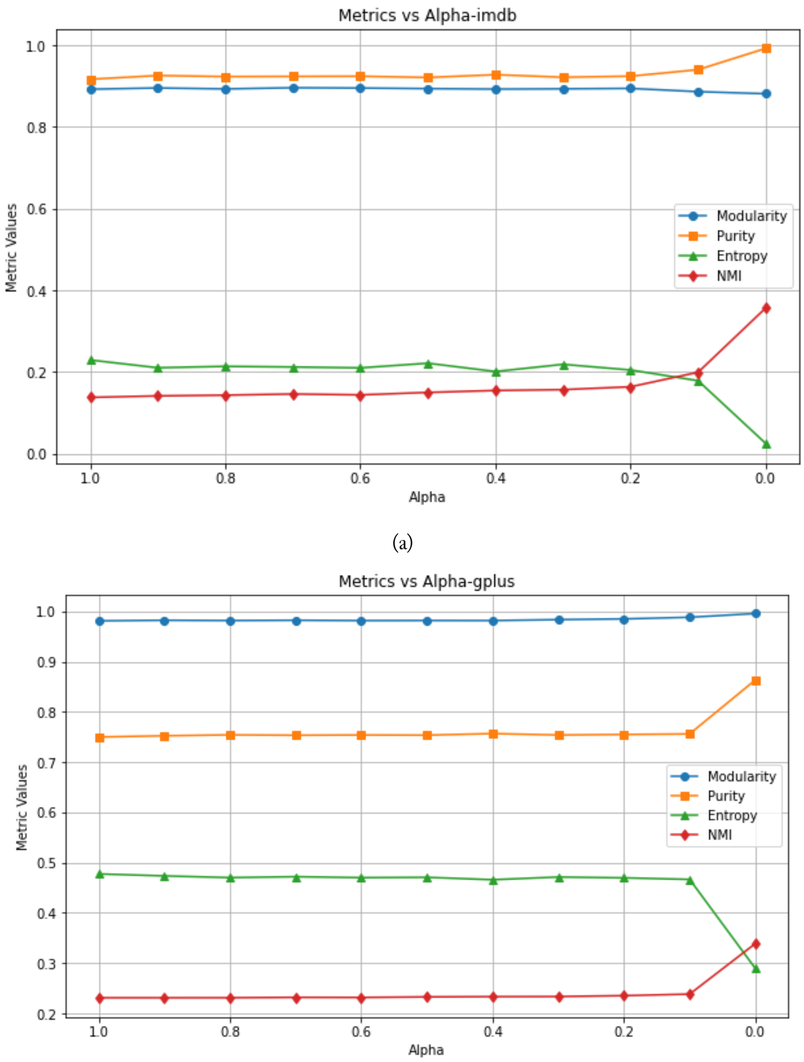
(b)

Fig. 6. Impact of Core Threshold (k) on Community Quality for (a) IMDB and (b) Gplus (spar_size=0.5, α=0.5)

In IMDB, modularity remains consistently high at around 0.9 across the entire range of k, while purity and entropy exhibit only minimal fluctuations around stable averages, and NMI stays low without notable variation. This indicates that the community structure in IMDB is relatively insensitive to changes in the k-core parameter. Similarly, in Gplus, the same pattern is observed: modularity remains high and stable, entropy and purity are relatively consistent, and NMI remains low with negligible change. These findings suggest that variations in k have only a minimal impact on community quality, indicating that CPSK is robust to changes in the k-core parameter and does not depend on the selection of a specific k value.

4.6.3. Attribute–Structure Trade-off (α)

Fig. 7. illustrates that the parameter α exhibits relatively stable patterns across both datasets, with significant changes occurring only at the extreme value of α = 0.0. In IMDB, modularity remains consistently high at around 0.9 across the entire range of α, while purity is generally stable with a slight increase at α = 0.0. Entropy remains low overall and decreases further as α approaches 0.0, whereas NMI, which is otherwise consistently low, rises sharply only at α = 0.0. A similar trend is observed in Gplus, where modularity remains high and stable, purity and entropy show negligible changes except at α = 0.0, where purity increases, entropy decreases, and NMI rises substantially. Overall, these results indicate that the method is robust to variations in α, with major shifts occurring only under the extreme condition in which structural information is completely ignored (α = 0.0), leading attributes to dominate the community detection process.



(b)

Fig. 7. Impact of Attribute–Structure Trade-off (α) on Community Quality for (a) IMDB and (b) Gplus (k=2, spar_size=0.5)

The sensitivity analysis across the three parameters shows that CPSK is robust in preserving community quality under different configurations. Varying the sparsification size demonstrates that even as the graph is reduced, modularity remains high while purity increases and entropy decreases, indicating stronger semantic cohesion and improved attribute consistency. The results for k-core show minimal impact on evaluation metrics, confirming that CPSK is independent of a specific backbone and can be applied flexibly across networks with diverse degree distributions. Similarly, the α analysis shows stable performance across most values, with major changes only occurring at the extreme case of α = 0, where structural information is entirely ignored. Overall, these findings confirm that CPSK consistently maintains high-quality community detection across parameter variations, highlighting its suitability for large-scale, heterogeneous networks.

5. Conclusion

A new community detection framework, CPSK (Community Preserving Sparsification based on K-core), is proposed in this paper. In this framework, a community-preserving mechanism is introduced to retain the network's structural backbone, while attribute–aware weighting emphasizes semantic similarity between nodes. By integrating these two characteristics, the original graph is sparsified to preserve dense intra-community links while discarding redundant edges, thereby reducing computational load without compromising community integrity. Experimental results on six benchmark datasets demonstrate that CPSK consistently achieves higher modularity, improved NMI, and better semantic cohesion in terms of purity and entropy, while also providing significant time efficiency compared to existing methods. These findings confirm that CPSK effectively balances structural preservation and attribute homogeneity, making community detection more scalable and reliable on large heterogeneous networks. For future work, we plan to explore adaptive strategies for the attribute–structure trade-off parameter α at the community level, extend CPSK to dynamic attributed networks, and incorporate more robust weighting schemes to handle highly sparse or noisy attributes.

Acknowledgment

The research is supported by Ministry of Higher Education Malaysia under the Fundamental Research Grant Scheme FRGS/1/2020/ICT02/UKM/01/2.

Declarations

Author contribution. Tedy Setiadi: conceptualization, methodology, investigation, data curation, writing, and original draft; Mohd Ridzwan Yaakub: supervision, resources, validation, writing, and reviewing and editing; Azuraliza Abu Bakar: conceptualization, methodology, supervision, project administration, funding acquisition, writing, reviewing and editing. All authors had approved the final version.

Funding statement. The research is supported by Ministry of Higher Education Malaysia under the Fundamental Research Grant Scheme FRGS/1/2020/ICT02/UKM/01/2.

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

References

[1] J. Li *et al.*, “A comprehensive review of community detection in graphs,” *Neurocomputing*, vol. 600, p. 128169, Oct. 2024, doi: 10.1016/j.neucom.2024.128169.

[2] G. Sperlì, “Community detection in Multimedia Social Networks using an attributed graph model,” *Online Soc. Networks Media*, vol. 46, p. 100312, May 2025, doi: 10.1016/j.osnem.2025.100312.

[3] M. Rostami, M. Oussalah, K. Berahmand, and V. Farrahi, “Community Detection Algorithms in Healthcare Applications: A Systematic Review,” *IEEE Access*, vol. 11, no. March, pp. 30247–30272, 2023, doi: 10.1109/ACCESS.2023.3260652.

[4] J. Guo, Q. Liang, and J. Zhao, “F-Deepwalk: A Community Detection Model for Transport Networks,” *Entropy*, vol. 26, no. 8, p. 715, Aug. 2024, doi: 10.3390/e26080715.

[5] T. Setiadi, M. R. Yaakub, and A. A. Bakar, “Community Detection Methods in Library ’ s Books and Borrowers Social Network Segmentation,” *J. Adv. Inf. Technol.*, vol. 14, no. 6, pp. 1177–1185, 2023, doi: 10.12720/jait.14.6.1177-1185.

[6] M. E. J. Newman and M. Girvan, “Finding and evaluating community structure in networks,” *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 69, no. 2 2, pp. 1–15, 2004, doi: 10.1103/PhysRevE.69.026113.

[7] T. Li *et al.*, "Hierarchical Community Detection by Recursive Partitioning," *J. Am. Stat. Assoc.*, vol. 117, no. 538, pp. 951–968, Apr. 2022, doi: 10.1080/01621459.2020.1833888.

[8] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech. Theory Exp.*, vol. 2008, no. 10, p. P10008, Oct. 2008, doi: 10.1088/1742-5468/2008/10/P10008.

[9] S. Aref and M. Mostajabdaveh, "Analyzing modularity maximization in approximation, heuristic, and graph neural network algorithms for community detection," *J. Comput. Sci.*, vol. 78, p. 102283, Jun. 2024, doi: 10.1016/j.jocs.2024.102283.

[10] C. Huang and Y. Zhong, "An Algorithm Based on Non-Negative Matrix Factorization for Detecting Communities in Networks," *Mathematics*, vol. 12, no. 4, p. 619, Feb. 2024, doi: 10.3390/math12040619.

[11] I. Inuwa-Dutse, M. Liptrott, and I. Korkontzelos, "A multilevel clustering technique for community detection," *Neurocomputing*, vol. 441, pp. 64–78, Jun. 2021, doi: 10.1016/j.neucom.2021.01.059.

[12] V. A. Traag and L. Šubelj, "Large network community detection by fast label propagation," *Sci. Rep.*, vol. 13, no. 1, pp. 1–11, 2023, doi: 10.1038/s41598-023-29610-z.

[13] M. Rostami and M. Oussalah, "A novel attributed community detection by integration of feature weighting and node centrality," *Online Soc. Networks Media*, vol. 30, p. 100219, 2022, doi: 10.1016/j.osnem.2022.100219.

[14] E. Alinezhad, B. Teimourpour, M. M. Sepehri, and M. Kargari, "Community detection in attributed networks considering both structural and attribute similarities: two mathematical programming approaches," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3203–3220, 2020, doi: 10.1007/s00521-019-04064-5.

[15] X. Zhou, L. Su, X. Li, Z. Zhao, and C. Li, "Community detection based on unsupervised attributed network embedding," *Expert Syst. Appl.*, vol. 213, p. 118937, 2023, doi: 10.1016/j.eswa.2022.118937.

[16] S. Citraro and G. Rossetti, "Eva: Attribute-Aware Network Segmentation," in Studies in Computational Intelligence, vol. 881 SCI, Springer, Cham, 2020, pp. 141–151, doi: 10.1007/978-3-030-36687-2_12.

[17] K. Yao, J. Liang, J. Liang, M. Li, and F. Cao, "Multi-view graph convolutional networks with attention mechanism," *Artif. Intell.*, vol. 307, p. 103708, Jun. 2022, doi: 10.1016/j.artint.2022.103708.

[18] J. Kim, S. Jeong, and S. Lim, "Link Pruning for Community Detection in Social Networks," *Appl. Sci.*, vol. 12, no. 13, 2022, doi: 10.3390/app12136811.

[19] Z. Liu, H. Wang, G. Wang, Z. Guo, and Y. Zhou, "Link community detection combined with network pruning and local community expansion," *Mod. Phys. Lett. B*, vol. 35, no. 5, pp. 4–7, 2021, doi: 10.1142/S0217984921500986.

[20] A. Socievole and C. Pizzuti, "Community Detection in Complex Networks Exploiting Spectral Graph Sparsification for Efficient Disaster Response," 2025, pp. 189–203, doi: 10.1007/978-3-031-85386-9_14.

[21] A. Socievole and C. Pizzuti, "Efficient community detection in disaster networks using spectral sparsification," *Pervasive Mob. Comput.*, vol. 113, p. 102106, Oct. 2025, doi: 10.1016/j.pmcj.2025.102106.

[22] A. Socievole and C. Pizzuti, "Community Detection in Dense Networks using Effective Resistance and Link Pruning," in *2023 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*, Sep. 2023, pp. 1–7, doi: 10.1109/ICT-DM58371.2023.10286913.

[23] M. Hashemi, S. Gong, J. Ni, W. Fan, B. A. Prakash, and W. Jin, "A Comprehensive Survey on Graph Reduction: Sparsification, Coarsening, and Condensation," in *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence*, Aug. 2024, pp. 8058–8066, doi: 10.24963/ijcai.2024/891.

[24] G. Wan and H. Schweitzer, "Edge Sparsification for Graphs via Meta-Learning," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, Apr. 2021, pp. 2733–2738, doi: 10.1109/ICDE51399.2021.00316.

[25] K. Berahmand, S. Haghani, M. Rostami, and Y. Li, "A new attributed graph clustering by using label propagation in complex networks," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 5, pp. 1869–1883, May 2022, doi: 10.1016/j.jksuci.2020.08.013.

[26] X.-L. Xu, Y.-Y. Xiao, X.-H. Yang, L. Wang, and Y.-B. Zhou, "Attributed network community detection based on network embedding and parameter-free clustering," *Appl. Intell.*, vol. 52, no. 7, pp. 8073–8086, May 2022, doi: 10.1007/s10489-021-02779-4.

[27] D. Malhotra and A. Chug, "A modified label propagation algorithm for community detection in attributed networks," *Int. J. Inf. Manag. Data Insights*, vol. 1, no. 2, p. 100030, Nov. 2021, doi: 10.1016/j.jjimei.2021.100030.

[28] H. Qing, "Community detection by spectral methods in multi-layer networks," *Appl. Soft Comput.*, vol. 171, p. 112769, Mar. 2025, doi: 10.1016/j.asoc.2025.112769.

[29] M. Huang and A. Ma, "Community detection in stochastic block models via penalized variational estimation," *J. Stat. Comput. Simul.*, vol. 95, no. 4, pp. 810–824, Mar. 2025, doi: 10.1080/00949655.2024.2439481.

[30] V. A. Traag, L. Waltman, and N. J. van Eck, "From Louvain to Leiden: guaranteeing well-connected communities," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, 2019, doi: 10.1038/s41598-019-41695-z.

[31] K. Sismanis, P. Potikas, D. Souliou, and A. Pagourtzis, "Overlapping community detection using graph attention networks," *Futur. Gener. Comput. Syst.*, vol. 163, p. 107529, Feb. 2025, doi: 10.1016/j.future.2024.107529.

[32] P. Harsha, S. Obaid, D. Mazumder, Y. Nagendar, and Z. A. Almoussawi, "Multi Resolution based Graph Convolutional Networks for Detecting Communities in Social Networks," in *2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)*, Aug. 2024, pp. 1–4, doi: 10.1109/IACIS61494.2024.10721782.

[33] G. Bokov, A. Konovalov, A. Uporova, S. Moiseev, I. Safonov, and A. Radionov, "A Parallel Hierarchical Approach for Community Detection on Large-scale Dynamic Networks," *arXiv*, pp. 1–46, 2025, [Online]. Available at: https://arxiv.org/abs/2502.18497.

[34] F. Tang and W. Ding, "Community detection with structural and attribute similarities," *J. Stat. Comput. Simul.*, vol. 89, no. 4, pp. 668–685, Mar. 2019, doi: 10.1080/00949655.2019.1568435.

[35] J. Cai, J. Hao, H. Yang, Y. Yang, X. Zhao, and Y. Xun, "A new community detection method for simplified networks by combining structure and attribute information," *Expert Syst. Appl.*, vol. 246, no. June 2023, p. 123103, 2024, doi: 10.1016/j.eswa.2023.123103.

[36] S. Kojaku, F. Radicchi, Y. Y. Ahn, and S. Fortunato, "Network community detection via neural embeddings," *Nat. Commun.* , vol. 15, no. 1, pp. 1–10, 2024, doi: 10.1038/s41467-024-52355-w.

[37] M. Malekzadeh and J. A. Long, "A network community structure similarity index for weighted networks," *PLoS One*, vol. 18, no. 11, p. e0292018, Nov. 2023, doi: 10.1371/journal.pone.0292018.

[38] A. Ahmad, L. Yuan, D. Yan, G. Guo, J. Chen, and C. Zhang, "Accelerating k-Core Decomposition by a GPU," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, Apr. 2023, pp. 1818–1831, doi: 10.1109/ICDE55515.2023.00142.

[39] J. Tu, G. Mei, and F. Piccialli, "An improved Nyström spectral graph clustering using k-core decomposition as a sampling strategy for large networks," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 6, pp. 3673–3684, Jun. 2022, doi: 10.1016/j.jksuci.2022.04.009.

[40] G. Mei, J. Tu, L. Xiao, and F. Piccialli, "An efficient graph clustering algorithm by exploiting k-core decomposition and motifs," *Comput. Electr. Eng.*, vol. 96, p. 107564, Dec. 2021, doi: 10.1016/j.compeleceng.2021.107564.

[41] K. Sotiropoulos and C. E. Tsourakakis, "Triangle-aware Spectral Sparsifiers and Community Detection," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Aug. 2021, pp. 1501–1509, doi: 10.1145/3447548.3467260.

[42] Q. Ni, J. Wang, and Z. Tang, "Degree and betweenness-based label propagation for community detection," *J. Comb. Optim.*, vol. 49, no. 2, p. 21, Mar. 2025, doi: 10.1007/s10878-024-01254-3.

[43] Y. Liu, X. Dong, Y. Gu, and Y. Sun, "Parallel k -Core Decomposition: Theory and Practice," *Proc. ACM Manag. Data*, vol. 3, no. 3, pp. 1–27, Jun. 2025, doi: 10.1145/3725332.

[44] X. Zhang, D. Song, and D. Tao, “Ricci Curvature-Based Graph Sparsification for Continual Graph Representation Learning,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 35, no. 12, pp. 17398–17410, Dec. 2024, doi: 10.1109/TNNLS.2023.3303454.

[45] X. Wu, C.-D. Wang, J.-Q. Lin, W.-D. Xi, and P. S. Yu, “Motif-Based Contrastive Learning for Community Detection,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 35, no. 9, pp. 11706–11719, Sep. 2024, doi: 10.1109/TNNLS.2024.3367873.

[46] M. Li *et al.*, “Topology-Driven Attribute Recovery for Attribute Missing Graph Learning in Social Internet of Things,” *IEEE Internet Things J.*, vol. 12, no. 11, pp. 16298–16313, 2025, doi: 10.1109/JIOT.2025.3531985.

[47] A. Polanco and M. E. J. Newman, “Hierarchical core-periphery structure in networks,” *Phys. Rev. E*, vol. 108, no. 2, p. 024311, Aug. 2023, doi: 10.1103/PhysRevE.108.024311.

[48] C. Jia, Y. Li, M. B. Carson, X. Wang, and J. Yu, “Node Attribute-enhanced Community Detection in Complex Networks,” *Sci. Rep.*, vol. 7, no. 1, pp. 1–15, 2017, doi: 10.1038/s41598-017-02751-8.

[49] S. Ali, M. Ahmad, M. A. Beg, I. U. Khan, S. Faizullah, and M. A. Khan, “<scp>SsAG</scp> : Summarization and Sparsification of Attributed Graphs,” *ACM Trans. Knowl. Discov. Data*, vol. 18, no. 6, pp. 1–22, Jul. 2024, doi: 10.1145/3651619.

[50] S. S. Das, N. A. Arafat, M. Rahman, S. M. Ferdous, A. Pothen, and M. M. Halappanavar, “SGS-GNN: A Supervised Graph Sparsification method for Graph Neural Networks,” *arXiv*, pp. 1–26, 2025, [Online]. Available at: https://arxiv.org/abs/2502.10208.

[51] J. Wang, K. Wang, X. Lin, W. Zhang, and Y. Zhang, “Neural Attributed Community Search at Billion Scale,” *Proc. ACM Manag. Data*, vol. 1, no. 4, pp. 1–25, Dec. 2023, doi: 10.1145/3626738.

[52] F. Tang, J. Li, X. Liu, C. Chang, and L. Teng, “GATFELPA integrates graph attention networks and enhanced label propagation for robust community detection,” *Sci. Rep.*, vol. 15, no. 1, pp. 1–17, 2025, doi: 10.1038/s41598-024-84962-4.

[53] J. Zhao *et al.*, “FS-GNN: Improving Fairness in Graph Neural Networks via Joint Sparsification,” *Neurocomputing*, vol. 648, p. 130641, Oct. 2025, doi: 10.1016/j.neucom.2025.130641.