Provable Higher-Order Graph Clustering: The Power of Peeling-Based Approaches

Longlong Lin, Zeli Wang, Rong-Hua Li, Qiyu Liu, Hongchao Qin, and Jin Zhao

***Abstract*—Higher-order graph clustering partitions graphs use frequently occurring subgraphs instead of edges, proving effective in community detection and knowledge discovery. Motif conductance, known for its strong interpretability, is a leading model. However, existing motif conductance algorithms are hindered by a two-stage reweighting framework that requires enumerating motif instances to generate an edge-weighted graph for partitioning. This framework has two major drawbacks: (1) It provides only a quadratic bound for three-vertex motifs, with no provable approximation guarantees for other motifs. (2) Enumerating motif instances is computationally prohibitive for large motifs or dense graphs due to combinatorial explosions. Besides, costly spectral clustering or local graph diffusion on the edge-weighted graph limits their scalability. In this paper, we propose a novel peeling-based clustering framework, PSMC, offering a motif-independent approximation ratio for any motif. Specifically, PSMC first defines a new locally computable vertex metric Motif Resident based on the given motif. Then, it iteratively deletes vertices with the smallest motif resident using efficient dynamic update techniques, outputting a locally optimal result with approximation guarantees. Besides, we introduce several powerful optimization techniques to further reduce computational costs. Empirical results on real-world and synthetic datasets showcase our proposed solutions' superiority over ten competitors.**

***Index Terms*—Graph clustering, motif, and cohesive subgraphs.**

I. Introduction

GRAPH clustering is a fundamental task in machine learning with numerous applications, including fraud detection [1], [2], [3] and graph representation learning [4], [5], [6]. Many traditional graph clustering models have been developed in the literature, including null model-based methods (e.g., modularity [7]), edge cut-based methods (e.g., ratio cut or normalized cut [8]), subgraph cohesiveness-based methods (e.g., k-core or k-truss [9]). These models aim to partition vertices into clusters such that vertices within the same cluster have more edges connecting them than vertices in different clusters [10], [11], [12].

Nevertheless, traditional graph clustering models overlook the significant motif connectivity patterns (i.e., small, frequently occurring subgraphs), which are essential for modeling and understanding the higher-order organization of complex networks [13]. Unlike dyadic edges, each motif (involving more than two nodes) indicates the unique interaction behaviors among vertices and represents some specific functions. For example, triangles are the cornerstone of stable relationships in social networks [14], [15], cycles can indicate money laundering events in financial markets [16], and feed-forward loops are basic transcription units in genetic networks [17]. Consequently, adopting mesoscopic level motifs as atomic clustering units has been recognized as the state-of-the-art (SOTA) solution in ground truth community detection and knowledge discovery [15], [18]. These clustering methods, referred to as higher-order graph clustering, aim to capture clusters with dense motifs rather than edges [19]. This paper focuses on higher-order graph clustering for massive million-node graphs, emphasizing the need for highly scalable and effective solutions.

Numerous higher-order graph clustering models have been proposed in the literature [19], [20], [21] (Section VII). Among these, the most representative and effective model is the *motif conductance* due to its strong interpretability and solid theoretical foundation [19] (Section II). Specifically, motif conductance is a variant of conductance (conductance is an edge-based clustering model [22], [23], [24], [25]), which measures the ratio of the number of motif instances exiting a cluster to the number within the cluster. Lower motif conductance implies better higher-order clustering quality [19], [26], [27], [28], [29]. However, identifying the result with the smallest motif conductance is challenging due to its NP-hardness [19]. Consequently, many approximate or heuristic algorithms have been proposed to either improve clustering quality or reduce computational costs. For instance, the seminal two-stage reweighting framework proposed in *Science* [19] transforms the input graph G into an edge-weighted graph $\mathcal{G}^{\mathbb{M}}$ in the first stage, where each edge's weight corresponds to the number of motif instances it participates in. In the second stage, traditional spectral clustering is applied to partition $\mathcal{G}^{\mathbb{M}}$. However, this framework only provides provable approximation guarantees for motifs consisting of three vertices [19]. Whether motifs with four or more vertices (which are more realistic [30], [31]) can achieve provable clustering quality remains an open question. Additionally, this

Received 12 August 2024; revised 28 May 2025; accepted 9 June 2025. Date of publication 13 June 2025; date of current version 24 July 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62402399, Grant U2241211, and Grant 62072034, in part by China Postdoctoral Science Foundation under Grant 2023T00325, and in part by Fundamental Research Funds for the Central Universities under Grant SWU-KQ22028. Recommended for acceptance by S. Whang. *(Corresponding author: Zeli Wang.)*

Longlong Lin and Qiyu Liu are with the College of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: longlonglin@swu.edu.cn; qyliu.cs@gmail.com).

Zeli Wang is with the Chongqing University of Posts and Telecommunications, Chongqing 400715, China (e-mail: zlwang@cqupt.edu.cn).

Rong-Hua Li and Hongchao Qin are with the Beijing Institute of Technology, Beijing 100081, China (e-mail: lironghuabit@126.com; qhc.neu@gmail.com).

Jin Zhao is with the Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: zjin@hust.edu.cn).

Digital Object Identifier 10.1109/TKDE.2025.3579811

framework requires enumerating all motif instances in advance and computing the eigenvector of the normalized Laplacian matrix of $\mathcal{G}^{\mathbb{M}}$, leading to prohibitively high time and space costs (Section III). To improve efficiency, some local graph diffusion algorithms have been proposed to replace eigenvector calculations with various random walk distributions (e.g., Personalized PageRank and higher-order Markov chains) (Section III). However, these algorithms are heuristic and heavily dependent on many hard-to-tune parameters and seeding strategies, resulting in unstable and often poor performance, as demonstrated in our experiments. Recently, Huang et al. [32] pointed out that almost all existing solutions are limited by the two-stage reweighting framework and proposed an adaptive sampling method to estimate edge weights, reducing computational time. However, this adaptive sampling method introduces randomness, leading to inaccurate results. Therefore, obtaining provable and scalable algorithms for motif conductance remains a challenging task.

To overcome the above challenges, we propose a novel and powerful peeling-based clustering framework, named *PSMC* (**P**rovable and **S**calable **M**otif **C**onductance). We observe that the goal of optimizing motif conductance is to obtain target clusters rather than the intermediate edge-weighted graph $\mathcal{G}^{\mathbb{M}}$, so it is unnecessary to spend excessive time on obtaining a precise $\mathcal{G}^{\mathbb{M}}$. Instead, we deeply analyze the functional form of motif conductance (Lemma 3) and iteratively optimize it starting from each vertex. Specifically, we first define a new vertex metric, *Motif Resident* (Definition 4), which can be computed locally. Then, *PSMC* iteratively deletes the vertex with the smallest motif resident value very efficiently using novel dynamic update technologies. Finally, *PSMC* returns the cluster with the smallest motif conductance found during this iterative process (also known as the peeling process). Consequently, *PSMC* integrates the computation and partition of the edge-weighted graph $\mathcal{G}^{\mathbb{M}}$ into an iterative algorithm, eliminating the need for expensive spectral clustering or local graph diffusion. Additionally, we theoretically prove that *PSMC* has a *fixed* and *motif-independent* approximation ratio (Theorem 4). This means *PSMC* can output a fixed approximation ratio for any given motif, solving the open question posed by the previous two-stage reweighting framework. However, the motif resident of a vertex u implicitly depends on the number of motif instances u participates in, causing *PSMC* to indirectly calculate all motif instances. Therefore, to further enhance efficiency, we develop several powerful optimization techniques. Specifically, we first devise the inclusion-exclusion-based Turan estimator and colorful wedge estimator (Section V-A) to estimate the motif resident of each vertex. Subsequently, a core-based optimization strategy is proposed to *directly* locate the target result by checking several small subgraphs (Section V-B). Our main contributions are highlighted as follows:

A Novel Higher-order Graph Clustering Framework with Approximation Guarantees: We introduce a novel and powerful peeling-based higher-order graph clustering framework *PSMC*, which can obtain a provable result based on our proposed vertex metric *Motif Resident*. *PSMC* has two striking features. First, it integrates the computation and partition of the edge-weighted graph into an iterative algorithm, reducing significantly computational costs. Second, it can output a fixed and motif-independent approximation ratio for any given motif, which existing SOTA frameworks cannot achieve.

Several Powerful Optimization Strategies: The inclusion-exclusion-based Turan estimator and colorful wedge estimator are proposed to estimate the motif resident of each vertex, which achieves a better trade-off between efficiency and accuracy. Besides, we propose a *practically* faster core-based optimization strategy to *directly* locate the target result by examining several small subgraphs.

Extensive Experiments: We conduct comprehensive experiments on nine datasets, comprising six real-world graphs and four synthetic graphs, and evaluate our proposed solutions against ten competing methods. The empirical results show that our algorithms are significantly more efficient, accurate, and scalable compared to the baselines.

II. Preliminaries

Given an unweighted and undirected graph $G(V, E)$, we use V and E to represent the vertex set and the edge set of G, respectively. We denote $|V| = n$ (resp.$|E| = m$) as the number of vertices (resp. edges) of G. Let $G_S(S, E_S)$ be the induced subgraph induced by S iff $S \subseteq V$ and $E_S = \{(u, v) \in E | u, v \in S\}$. We use $N_S(v) = \{u \in S | (u, v) \in E\}$ to denote the neighbors of v in S. We use $\mathbb{M}$ to denote the user-initiated query motif, which is a frequently occurring interaction pattern in complex networks. For convenience, we use $G_S \in \mathbb{M}$ to represent G_S as an instance of $\mathbb{M}$. Namely, $G_S \in \mathbb{M}$ iff G_S is isomorphic to $\mathbb{M}$. Let $k(\mathbb{M}) \geq 2$ be the order of $\mathbb{M}$, which is the number of vertices involved in $\mathbb{M}$. For example, an edge is a 2-order motif and a triangle is a 3-order motif. A high-level definition of higher-order graph clustering is as follows.

Definition 1 (Higher-Order Graph Clustering): For an unweighted and undirected graph $G(V, E)$ and a motif $\mathbb{M}$, the problem of the higher-order graph clustering aims to find a high-quality cluster $C \subseteq V$ has the following properties: (1) G_C contains many instances of $\mathbb{M}$; (2) there are few motif instances that cross G_C and $G_{V \setminus C}$.

According to the intuition of Definition 1, we use the most representative and effective *motif conductance* [19], [26], [27], [28], [29] to measure the clustering quality of an identified cluster C.

Definition 2 (Motif Conductance [19]): For an unweighted and undirected graph $G(V, E)$ and a motif $\mathbb{M}$, the motif conductance of C is defined as $\phi_{\mathbb{M}}(C) = \frac{cut_{\mathbb{M}}(C)}{\min\{vol_{\mathbb{M}}(C), vol_{\mathbb{M}}(V \setminus C)\}}$.

$$cut_{\mathbb{M}}(C) = |\{G_S \in \mathbb{M} | S \cap C \neq \emptyset, S \cap (V \setminus C) \neq \emptyset\}| \quad (1)$$

$$vol_{\mathbb{M}}(C) = \sum_{u \in C} |\{G_S \in \mathbb{M} | u \in S\}| \quad (2)$$

Where $cut_{\mathbb{M}}(C)$ is the number of motif instances with at least one vertex in C and at least one vertex in $V \setminus C$, and $vol_{\mathbb{M}}(C)$ (resp. $vol_{\mathbb{M}}(V \setminus C)$) is the number of the motif instance the vertices in C (resp. $V \setminus C$) participate in. When $\mathbb{M}$ is an edge, the motif conductance degenerates into classic conductance [22], [23], [33]. Thus, edges that do not participate in any motif instances do not contribute to the motif conductance. Namely, a cluster with many edges but few motif instances may also have poor motif conductance. Therefore, motif conductance has

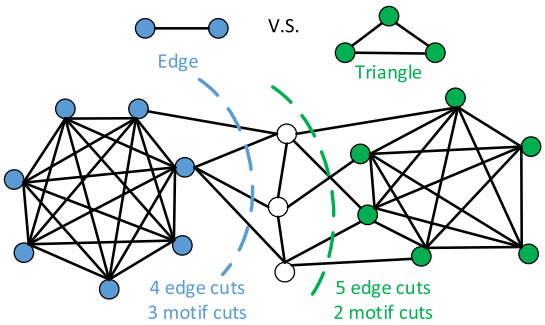


Fig. 1. Illustration of the traditional edge-based conductance and the motif conductance on a synthetic graph. There are 47 edges and 60 triangles. The blue dotted line indicates the optimal cut when the motif is an edge and the corresponding conductance is $\frac{4}{\min\{42,52\}}$. The green dotted line represents the optimal cut when the motif is a triangle and the corresponding triangle conductance is $\frac{2}{\min\{116,64\}}$. Motif conductance is more likely to preserve motif instances compared with edge-based conductance.

strong interpretability and can improve the quality of the resulting cluster by focusing on the particular motifs that are important higher-order structures of a given network [19]. Fig. 1 shows the difference between the traditional edge-based conductance and the motif conductance. Note that we have $\phi_{\mathbb{M}}(C) = \phi_{\mathbb{M}}(V \setminus C)$ by Definition 2.

Problem Statement: Given an unweighted and undirected graph $G(V, E)$ and a motif $\mathbb{M}$, the goal of motif conductance based graph clustering is to find a vertex subset $S^* \subseteq V$, satisfying $vol_{\mathbb{M}}(S^*) \leq vol_{\mathbb{M}}(V \setminus S^*)$ and $\phi_{\mathbb{M}}(S^*) \leq \phi_{\mathbb{M}}(S)$ for any $S \subseteq V$. $\phi^*_{\mathbb{M}}$ stands for $\phi_{\mathbb{M}}(S^*)$ for brevity.

III. Existing Solutions and Their Shortcomings

A. Seed-Free Global Clustering

Seed-free global clustering identifies the higher-order clusters by calculating the eigenvector of the normalized Laplacian matrix of the motif adjacency matrix. Let $\mathcal{G}^{\mathbb{M}} = (V, \mathcal{E}^{\mathbb{M}})$ be the motif-based weighted graph, in which V is the vertex set that is the same as the original graph G, and $\mathcal{E}^{\mathbb{M}} = \{(u, v, W_{uv}) | u, v \in V\}$ indicates the weighted edge set generated based on the given motif $\mathbb{M}$ and W_{uv} is the number of motif instances that contain u and v together. Therefore, we define $\mathcal{A}$ as the motif adjacency matrix with $\mathcal{A}_{uv} = W_{uv}$ for all $u, v \in V$. We use $\mathcal{D}$ to represent the diagonal degree matrix with $\mathcal{D}_{uu} = \sum_v \mathcal{A}_{uv}$ for all $u \in V$. The normalized Laplacian matrix of the motif adjacency matrix is defined as $\mathcal{L} = I - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$ where I is the identity matrix. Based on these symbols, Benson et al. [19] proposed the following two-stage higher-order spectral clustering (*HSC*), which is outlined in Algorithm 1. Specifically, Algorithm 1 first obtains the normalized Laplacian matrix $\mathcal{L}$ by enumerating motif instances (Lines 1-5). Then, Algorithm 1 computes the eigenvector x of the second smallest eigenvalue of $\mathcal{L}$ to execute the sweep procedure (Lines 6-8). Namely, it sorts all entries in x such that $x_1 \geq x_2 \geq \cdots \geq x_n$ and outputs $S = \arg\min \phi^{\mathcal{G}^{\mathbb{M}}}(S_i)$, in which $S_i = \{x_1, x_2, \ldots, x_i\}$ and $\phi^{\mathcal{G}^{\mathbb{M}}}(S_i)$ is the traditional edge-based conductance of S_i in terms of the weighted graph $\mathcal{G}^{\mathbb{M}}$ constructed by $\mathcal{A}$. Note that in Lines 9-12, we output the smaller of S and $\bar{S}$ due to $\phi^{\mathcal{G}^{\mathbb{M}}}(S) = \phi^{\mathcal{G}^{\mathbb{M}}}(\bar{S})$. The following theorems are important theoretical bases of *HSC*.

Algorithm 1: Higher-Order Spectral Clustering (*HSC*) [19].

Input: A graph $G(V, E)$ and a motif $\mathbb{M}$
Output: A higher-order cluster S
1: Initializing $\mathcal{A} \in R^{n*n}$ and $\mathcal{D} \in R^{n*n}$ are zero matrices
2: **for** each motif instance $mi \in \mathbb{M}$ of G **do**
3: **for** each node $u \in mi$ and node $v(\neq u) \in mi$ **do**
4: $\mathcal{A}_{uv} += 1$; $\mathcal{A}_{vu} += 1$; $\mathcal{D}_{uu} += 1$; $\mathcal{D}_{vv} += 1$
5: $\mathcal{L} \leftarrow I - \mathcal{D}^{-1/2}\mathcal{A}\mathcal{D}^{-1/2}$
6: $x \leftarrow$ eigenvector of the second smallest eigenvalue of $\mathcal{L}$
7: $x_i \leftarrow$ to be index of x with ith largest value
8: $S \leftarrow \arg\min \phi^{\mathcal{G}^{\mathbb{M}}}(S_i)$, where $S_i = \{x_1, x_2, \ldots, x_i\}$ and $\phi^{\mathcal{G}^{\mathbb{M}}}(S_i)$ is the traditional edge-based conductance of S_i in terms of the weighted graph $\mathcal{G}^{\mathbb{M}}$ constructed by $\mathcal{A}$
9: **if** $|S| > |\bar{S}|$ **then**
10: **return** $\bar{S}$
11: **else**
12: **return** S // return the smaller of S and $\bar{S}$ due to $\phi^{\mathcal{G}^{\mathbb{M}}}(S) = \phi^{\mathcal{G}^{\mathbb{M}}}(\bar{S})$

Theorem 1 ([19]): Given a graph $G(V, E)$ and a motif $\mathbb{M}$, for any $S \subseteq V$, we have

$$\phi_{\mathbb{M}}(S) = \begin{cases} \phi^{\mathcal{G}^{\mathbb{M}}}(S), & \text{if } k(\mathbb{M}) = 3 \\ \phi^{\mathcal{G}^{\mathbb{M}}}(S) - \frac{\sum_{mi \in \mathbb{M}} I(|mi \cap S| = 2)}{\sum_{u \in S} \mathcal{D}_{uu}}, & \text{if } k(\mathbb{M}) = 4 \\ \not\sim \phi^{\mathcal{G}^{\mathbb{M}}}(S) & \text{if } k(\mathbb{M}) > 4 \end{cases} \tag{3}$$

Where $\phi^{\mathcal{G}^{\mathbb{M}}}(S)$ is the traditional edge-based conductance of S in terms of the weighted graph $\mathcal{G}^{\mathbb{M}}$ and $I(.)$ is the indicator function. The symbol $\phi_{\mathbb{M}}(S) \not\sim \phi^{\mathcal{G}^{\mathbb{M}}}(S)$ indicates the relationship between $\phi_{\mathbb{M}}(S)$ and $\phi^{\mathcal{G}^{\mathbb{M}}}(S)$ is unclear. Since the proof of Theorem 1 directly comes from [19], we omit it for brevity. Next, we present the well-known cheeger inequality, followed by the theoretical bounds of Algorithm 1 when $k(\mathbb{M}) = 3$.

Theorem 2 (Cheeger Inequality [34]): Given a weighted graph $\mathcal{G}^{\mathbb{M}}$ and its normalized Laplacian matrix $\mathcal{L}$, we assume that λ_2 is the second smallest eigenvalue of $\mathcal{L}$. Then, we have $\lambda_2/2 \leq \phi^* \leq \sqrt{2\lambda_2}$, in which ϕ^* is the optimal edge-based conductance in terms of the weighted graph $\mathcal{G}^{\mathbb{M}}$.

Theorem 3: Given a graph $G(V, E)$ and a motif $\mathbb{M}$ with $k(\mathbb{M}) = 3$, let S be the vertex subset returned by Algorithm 1, we have $\phi^*_{\mathbb{M}} \leq \phi_{\mathbb{M}}(S) \leq 2\sqrt{\phi^*_{\mathbb{M}}}$, in which $\phi^*_{\mathbb{M}}$ is the optimal motif conductance.

Proof: Since S is equivalent to the vertex subset return by Fielder vector-based spectral clustering over the motif-based weighted graph $\mathcal{G}^{\mathbb{M}}$ (Algorithm 1), we can obtain $\phi^{\mathcal{G}^{\mathbb{M}}}(S) \leq \sqrt{2\lambda_2}$ [33], [35], where λ_2 is the second smallest eigenvalue of the normalized Laplacian matrix of $\mathcal{G}^{\mathbb{M}}$. Meanwhile, by Theorem 2, we have $\phi^{\mathcal{G}^{\mathbb{M}}}(S) \leq \sqrt{2\lambda_2} \leq \sqrt{2 * 2 * \phi^*} = 2\sqrt{\phi^*}$, in which ϕ^* is the optimal edge-based conductance in terms of the weighted graph $\mathcal{G}^{\mathbb{M}}$. Furthermore, by Theorem 1, we have $\phi_{\mathbb{M}}(H) = \phi^{\mathcal{G}^{\mathbb{M}}}(H)$ for any vertex subset $H \subseteq V$ due to $k(\mathbb{M}) = 3$. Thus, $\phi_{\mathbb{M}}(S) \leq 2\sqrt{\phi^*_{\mathbb{M}}}$. Clearly, $\phi^*_{\mathbb{M}} \leq \phi_{\mathbb{M}}(S)$ due to $\phi^*_{\mathbb{M}}$ is the smallest motif conductance. As a result, this theorem is proved. □

Algorithm 2: Seed-Dependent Local Clustering (e.g., *HOSPLOC* or *MAPPR* [26], [27], [29]).

Input: A graph $G(V, E)$, a motif $\mathbb{M}$, and a seed vertex q
Output: A higher-order cluster S
1: $\pi \leftarrow$ the probability distribution after the end of the corresponding local graph diffusion // π is π_{Markov} or π_{PPR}
2: $y \leftarrow \pi\mathcal{D}^{-1}$
3: $y_i \leftarrow$ to be index of y with ith largest non-zero value
4: $S \leftarrow \arg\min \phi_{\mathbb{M}}(S_i)$, where $S_i = \{y_1, y_2, \ldots, y_i\}$
5: **if** $|S| > |\bar{S}|$ **then**
6: **return** $\bar{S}$
7: **else**
8: **return** S // return the smaller of S and $\bar{S}$ due to $\phi_{\mathbb{M}}(S) = \phi_{\mathbb{M}}(\bar{S})$

B. Seed-Dependent Local Clustering

Seed-dependent local clustering executes the local graph diffusion from the given seed vertex q to identify higher-order clusters. Before proceeding further, we give some important symbols. Let $\mathcal{P} = \mathcal{D}^{-1}\mathcal{A}$ be the probability transition matrix of the motif-based weighted graph $\mathcal{G}^{\mathbb{M}}$. On top of that, we use $\mathcal{P}^k$ to represent the k-hop probability transition matrix of $\mathcal{G}^{\mathbb{M}}$. In other words, $\mathcal{P}^k_{uv}$ is the probability that a k-hop ($k \geq 1$) random walk from vertex u would end at vertex v. Based on these symbols, we elaborate two well-known local graph diffusion methods: Higher-order Markov chain [36] and Personalized PageRank [37], [38].

Higher-order Markov chain can use state transition tensors to simulate the long-term dependence of states [36]. Specifically, we let $\mathcal{T}$ be the k-order state transition tensor with $\mathcal{T}(u_1, u_2, \ldots u_k) = \frac{I((u_1,u_2,\ldots u_k)\ is\ motif\ instance)}{\sum_{v\in V} I((u_1,u_2,\ldots u_{k-1},v)\ is\ motif\ instance)}$ for all $u_1, u_2, \ldots u_k \in V$, in which k is the order of $\mathbb{M}$. Thus, $\mathcal{T}$ can be interpreted as a $(k-1)$-order markov chain. Namely, $Pr(S_t = s_t | S_1 = s_1, S_2 = s_2, \ldots, S_{t-1} = s_{t-1}) = Pr(S_t = s_t | S_{t-k+1} = s_{t-k+1}, S_{t-k+2} = s_{t-k+2}, \ldots, S_{t-1} = s_{t-1}) = \mathcal{T}(s_{t-k+1}, s_{t-k+2}, \ldots s_t)$. In other words, if each vertex represents an individual state, the future state (i.e., S_t) only depends on the past $k-1$ states. Since it is very expensive to store and calculate the stationary distribution of such a high-order Markov chain (e.g., requires $O(n^{k-1})$ space complexity), Li et al. [36] used "rank-one" to obtain an approximation solution, which can reduce the space complexity to $O(n)$. Specifically, let π^i be the distribution of i-th higher-order Markov random walk, we have

$$\pi^{(t)} = \bar{\mathcal{T}}\left(\pi^{(t-1)} \otimes \pi^{(t-2)} \ldots \otimes \pi^{(t-k+1)}\right) \tag{4}$$

Where $\otimes$ is the Kronecker product operator and $\bar{\mathcal{T}}$ is the $(k-2)$-mode unfolding matrix of the k-order tensor $\mathcal{T}$. Based on these theoretical knowledge, [26], [27] proposed *HOSPLOC* to obtain higher-order clusters by performing the truncated higher-order markov random walk. Specifically, for any vector π and error tolerance θ, we define a truncation operator π_θ on any vertex u as follows:

$$\pi_\theta[u] = \begin{cases} \pi[u], & \text{if } \pi[u] \geq |N_V(u)| \cdot \theta \\ 0, & \text{Otherwise} \end{cases} \tag{5}$$

Furthermore, we use $\pi^{(1)} = \chi_q$ to represent the one-hot vector with only a value-1 entry corresponding to the seed vertex q. The initial distribution vectors are computed as $\pi^{(i)} = (\mathcal{P}\pi^{(i-1)})_\theta$ for $i = 2, 3, \ldots k-1$. Subsequently, [26], [27] executes the "rank-one" procedure (i.e., (4)) to obtain the probability distribution of higher-order Markov random walk. Thus, we let $\pi_{Markov} = \pi^{(N)}$ be the final probability distribution after N iterations, in which N is an input parameter [26], [27].

Personalized PageRank is the procedure of the random walk with restart [37], [38]. Namely, given a teleportation parameter α, the α-discount random walk is denoted as follows: (1) It starts from the given seed vertex q. (2) At each step it stops in the current vertex with probability α, otherwise, it continues to walk according to the probability transition matrix $\mathcal{P}$. Thus, we let $\pi_{PPR}(u)$ be the probability that the α-discount random walk stops in u. Namely, $\pi_{PPR}(u) = \sum_{k=0}^{\infty} \alpha(1-\alpha)^k \mathcal{P}^k_{qu}$.

Based on these backgrounds, Zhou et al. [26], [27] and Yin et al. [29] proposed *HOSPLOC* and *MAPPR* to obtain higher-order clusters, respectively. Specifically, Algorithm 2 first computes the probability distribution π at the end of the corresponding graph diffusion (i.e., truncated higher-order Markov random walk or Personalized PageRank random walk), and let $y = \pi\mathcal{D}^{-1}$ (Lines 1-2). Subsequently, Algorithm 2 runs the sweep procedure (Lines 3-4). Namely, it sorts all non-zero entries in y such that $y_1 \geq y_2 \geq \ldots \geq y_{sup(y)}$ ($sup(y)$ is the number of the non-zero entries in y and $sup(y) << n$ [39]), and outputs $S = \arg\min \phi_{\mathbb{M}}(S_i)$, in which $S_i = \{y_1, y_2, \ldots, y_i\}$. Note that in Lines 5-8, we output the smaller of S and $\bar{S}$ due to $\phi_{\mathbb{M}}(S) = \phi_{\mathbb{M}}(\bar{S})$.

C. The Shortcomings of Existing Solutions

The seed-free global clustering *HSC* (i.e., Algorithm 1) can only derive the Cheeger inequality for motifs consisting of three vertices (Theorem 3). However, it has not been proven whether there is a quality guarantee for motifs with four or more vertices (which are more realistic [30], [31]). On top of that, since *HSC* needs to enumerate motif instances in advance and calculate the eigenvector of $\mathcal{L}$, its time complexity is $O(\mathcal{T}(\mathbb{M})) + O(n^3)$ ($O(\mathcal{T}(\mathbb{M}))$ is the time complexity of enumerating motif instances) and space complexity is $O(min\{\binom{k}{2} num(\mathbb{M}), n^2\})$ [19], resulting in poor scalability. On the other hand, since seed-dependent local clustering methods (i.e., Algorithm 2) aim to identify the higher-order clusters to which the given seed vertex q belongs, they only have locally-biased Cheeger-like quality for the motif consisting of three vertices [26], [27], [29]. Namely, seed-dependent local clustering methods do not give the theoretical gap to $\phi^*_{\mathbb{M}}$. Besides, their clustering qualities are heavily dependent on many hard-to-tune parameters and seeding strategies. Practically, their performance is unstable and even find degenerate solutions, as demonstrated in our experiments. For convenience, we give Table I to summarize the above SOTA motif conductance algorithms. By Table I, we know that the complexities of our solutions are lower than the baselines. This is because baselines need to enumerate all motif instances in advance, and then execute the expensive spectral clustering or local graph diffusion. However, we are to integrate the enumeration and partitioning in an iterative algorithm, which can greatly reduce computational costs. On top of that, our *PSMC*

TABLE I
A COMPARISON OF MOTIF CONDUCTANCE BASED GRAPH CLUSTERING

Methods	Accuracy Guarantee	Time Complexity	Space Complexity	Remark
HSC [19]	$O(\sqrt{\phi^*_{\mathbb{M}}})$ for $k=3$ $\times$ for $k>3$	$O(\mathcal{T}(\mathbb{M}))+O(n^3)$	$O(min\{\binom{k}{2}num(\mathbb{M}), n^2\})$	Eigenvector-based
HOSPLOC [26], [27]	$\times$	$O(t_{max}\frac{2^{bk}}{(\phi^*_{\mathbb{M}})^{2k}}\log^{3k} m)$	$O(n^k)$	Higher-order Markov Chain-based
MAPPR [29]	$\times$	$O(\mathcal{T}(\mathbb{M}))+O(\frac{\log\frac{1}{\epsilon}}{\epsilon})$	$O(min\{\binom{k}{2}num(\mathbb{M}), n^2\})$	Personalized PageRank-based
PSMC (This paper)	$O(1/2+1/2\phi^*_{\mathbb{M}})$ for any k	$O(\mathcal{T}(\mathbb{M}))$	$O(m+n)$	*Motif Resident*-based

$\phi^*_{\mathbb{M}} \in (0, 1]$ is the smallest motif conductance value. $k = k(\mathbb{M})$ is the order of $\mathbb{M}$. $O(\mathcal{T}(\mathbb{M}))$ is the time complexity of enumerating motif instances. num($\mathbb{M}$) is the number of motif instances. t_{max} and b are the maximum iteration number and motif volum parameter of HOSPLOC. ϵ is the error tolerance of MAPPR to execute forward push. $\times$ represents the corresponding method has no accuracy guarantee.

Algorithm 3: **P**rovable and **S**calable **M**otif **C**onductance (*PSMC*).

Input: A graph $G(V, E)$ and a motif $\mathbb{M}$ with $k = k(\mathbb{M})$
Output: A higher-order cluster $\hat{S}$ with motif-independent approximation ratio
1: Initializing the motif degree $\mathbb{M}(u) = 0$ for any $u \in V$
2: **for** each motif instance $mi \in \mathbb{M}$ of G **do**
3: **for** each node $u \in mi$ **do**
4: $\mathbb{M}(u) += 1$
5: $i \leftarrow 1$; $S_i \leftarrow V$; $\mathbb{M}_k^{S_i}(u) = \mathbb{M}(u)$ and $\mathbb{M}_1^{S_i}(u) = 0$ for $u \in S_i$
6: $Mr_{S_i}(u) \leftarrow \frac{\mathbb{M}(u)+\mathbb{M}_k^{S_i}(u)-\mathbb{M}_1^{S_i}(u)}{\mathbb{M}(u)}$ for any $u \in S_i$
7: **while** $S_i \neq \emptyset$ **do**
8: $u \leftarrow \arg\min\{Mr_{S_i}(u) | u \in S_i\}$
9: $i \leftarrow i + 1$
10: $S_i \leftarrow S_{i-1} \setminus \{u\}$
11: $\hat{S} \leftarrow \arg\min_{S \in \{S_1, S_2, \ldots, S_n\}}\{\phi_{\mathbb{M}}(S) | vol_{\mathbb{M}}(S) \leq vol_{\mathbb{M}}(V \setminus S)\}$
12: **return** $\hat{S}$

can output $O(1/2 + 1/2\phi^*_{\mathbb{M}})$ accuracy guarantee for any size of motif, while baselines cannot.

IV. PSMC: THE PROPOSED SOLUTION

In this section, we first devise a novel peeling-based higher-order graph clustering algorithm *PSMC* (**P**rovable and **S**calable **M**otif **C**onductance), which aims to output a high-quality cluster. It is important to highlight that PSMC can provide *fixed* and *motif-independent* approximation ratio for any motif. This significant feature addresses and resolves the open problem raised by [19]. Subsequently, we propose novel dynamic update technologies and effective bounds to further boost the efficiency of *PSMC*.

A. The PSMC Algorithm

Recall that our problem is to obtain a higher-order cluster rather than to obtain the intermediate edge-weighted graph $\mathcal{G}^{\mathbb{M}}$, thus it is not necessary to blindly spend much time on getting precise $\mathcal{G}^{\mathbb{M}}$. Based on in-depth observations, we reformulate motif conductance and propose a novel computing framework, which iteratively optimizes motif conductance starting from each vertex. Before describing our proposed algorithms, several useful definitions are stated as follows.

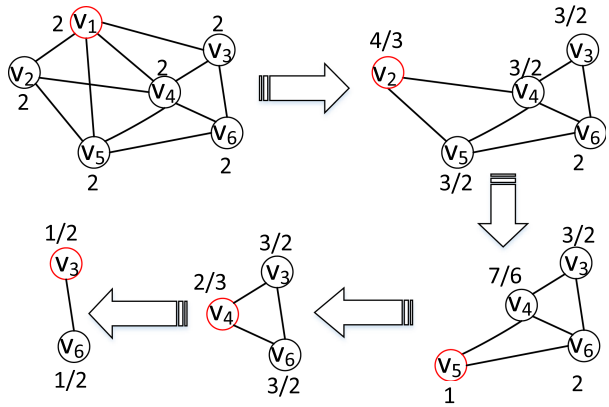


Fig. 2. The red circle refers to the currently deleted vertices. The number next to each vertex is the motif resident value (obtained by Definition 4).

Definition 3 (Motif degree): Given an unweighted and undirected graph $G(V, E)$ and a motif $\mathbb{M}$ with $k = k(\mathbb{M})$, the motif degree of u is defined as $\mathbb{M}(u) = |\{G_S \in \mathbb{M} | u \in S\}|$. For a positive integer $1 \leq j \leq k$, we let $\mathbb{M}_j^C(u) = |\{G_S \in \mathbb{M} | u \in S, |C \cap S| = j\}|$.

Definition 4 (Motif Resident): Given an unweighted and undirected graph $G(V, E)$, a motif $\mathbb{M}$ with $k = k(\mathbb{M})$, and a vertex subset S, the motif resident of $u \in S$ w.r.t. G_S is defined as $Mr_S(u) = \frac{\mathbb{M}(u)+\mathbb{M}_k^S(u)-\mathbb{M}_1^S(u)}{\mathbb{M}(u)}$.

Based on these definitions, we develop *PSMC* with a three-stage computing framework (Algorithm 3). In *Stage 1*, we compute the motif resident value for each vertex (Lines 1-6). In *Stage 2*, we iteratively remove the vertex with the smallest motif resident (Lines 7-10). Such an iterative deletion process is referred to as a *peeling* process. In *Stage 3*, we output the result with the smallest motif conductance during the peeling process (lines 11-12). The following synthetic example illustrates the process of Algorithm 3.

Example 1: Consider the input graph $G(V, E)$ in upper left corner of Fig. 2 and the query motif is the triangle (i.e., $k(\mathbb{M}) = 3$), the ordering $(v_1, v_2, v_5, v_4, v_3, v_6)$ is the order of vertices selected in Line 8 of Algorithm 3, which is illustrated in Fig. 2. This is because that v_1 has the minimum motif resident (i.e., 2) in $\{v_1, v_2, v_3, v_4, v_5, v_6\}$ (when there are multiple minimum values, use id to break them). Similarly, v_2 has the minimum motif resident (i.e., 4/3) in $\{v_2, v_3, v_4, v_5, v_6\}$. Following this ordering, we can derive that (v_3, v_4, v_6) is the resultant cluster returned by Algorithm 3. This is because they have the minimum motif conductance under the deleted ordering (Line 11 of Algorithm 3).

Lemma 1 (Monotonicity): Given an unweighted and undirected graph $G(V,E)$, a motif $\mathbb{M}$ with $k=k(\mathbb{M})$, and two vertex subset S and H, we have $Mr_S(u) \le Mr_H(u)$ if $u \in S \subseteq H$.

Proof: Let $I_j^C(u)=\{G_W \in \mathbb{M} | u \in W, |C \cap W| = j\}$, we have $\mathbb{M}_j^C(u)=|I_j^C(u)|$. For any motif instance $G_W \in I_k^S(u)$, we have $G_W \in I_k^H(u)$ due to $S \subseteq H$. Thus, $I_k^S(u) \subseteq I_k^H(u)$ and $\mathbb{M}_k^S(u)=|I_k^S(u)| \le |I_k^H(u)|=\mathbb{M}_k^H(u)$. Similarly, For any motif instance $G_W \in I_1^H(u)$, we have $G_W \in I_1^S(u)$ due to $u \in S$ and $S \subseteq H$. Thus, $I_1^H(u) \subseteq I_1^S(u)$ and $\mathbb{M}_1^H(u)=|I_1^H(u)| \le |I_1^S(u)|=\mathbb{M}_1^S(u)$. Thus, $Mr_S(u)=\frac{\mathbb{M}(u)+\mathbb{M}_k^S(u)-\mathbb{M}_1^S(u)}{\mathbb{M}(u)} \le \frac{\mathbb{M}(u)+\mathbb{M}_k^H(u)-\mathbb{M}_1^H(u)}{\mathbb{M}(u)}=Mr_H(u)$. □

Lemma 2: Given an unweighted and undirected graph $G(V,E)$, a motif $\mathbb{M}$ with $k=k(\mathbb{M})$, and a vertex subset S, we have $cut_{\mathbb{M}}(S \setminus \{u\})=cut_{\mathbb{M}}(S)-\mathbb{M}_1^S(u)+\mathbb{M}_k^S(u)$.

Proof: By Definition 2, we have $cut_{\mathbb{M}}(S)=\sum_{i=1}^{k-1} cut_{\mathbb{M}}^i(S)$ where $cut_{\mathbb{M}}^i(S)=|\{G_H \in \mathbb{M} | |H \cap S|=i\}|$ is the number of motif instances with exactly i vertices in S. Thus, when the vertex u is deleted from S, $cut_{\mathbb{M}}^i(S \setminus \{u\})=cut_{\mathbb{M}}^i(S)-\mathbb{M}_i^S(u)+\mathbb{M}_{i+1}^S(u)$. Moreover, $cut_{\mathbb{M}}(S \setminus \{u\})=\sum_{i=1}^{k-1} cut_{\mathbb{M}}^i(S \setminus \{u\})=cut_{\mathbb{M}}^1(S)-\mathbb{M}_1^S(u)+\mathbb{M}_2^S(u)+cut_{\mathbb{M}}^2(S)-\mathbb{M}_2^S(u)+\mathbb{M}_3^S(u)+\cdots cut_{\mathbb{M}}^{k-1}(S)-\mathbb{M}_{k-1}^S(u)+\mathbb{M}_k^S(u)=\sum_{i=1}^{k-1} cut_{\mathbb{M}}^i(S)-\mathbb{M}_1^S(u)+\mathbb{M}_k^S(u)=cut_{\mathbb{M}}(S)-\mathbb{M}_1^S(u)+\mathbb{M}_k^S(u)$. Therefore, Lemma 2 is true. □

Let $g_{\mathbb{M}}(S)=(\sum_{u\in S}\mathbb{M}(u)-cut_{\mathbb{M}}(S))/(\sum_{u\in S}\mathbb{M}(u))$ and assume that the larger of $g_{\mathbb{M}}(S)$, the better the quality of S. Let $\widetilde{S}$ be the optimal vertex set for $g_{\mathbb{M}}(.)$. That is, $g_{\mathbb{M}}(\widetilde{S}) \ge g_{\mathbb{M}}(S)$ for any vertex subset $S \subseteq V$. The following two lemmas are key theoretical bases of *PSMC*.

Lemma 3 (Reformulation of Motif Conductance): Given an unweighted and undirected graph $G(V,E)$, a motif $\mathbb{M}$ with $k=k(\mathbb{M})$, and a vertex subset S, we have $\phi_{\mathbb{M}}(S)=1-g_{\mathbb{M}}(S)$ if $vol_{\mathbb{M}}(S) \le vol_{\mathbb{M}}(V \setminus S)$.

Proof: By Definition 2, if $vol_{\mathbb{M}}(S) \le vol_{\mathbb{M}}(V \setminus S)$, we have $\phi_{\mathbb{M}}(S)=\frac{cut_{\mathbb{M}}(S)}{\min\{vol_{\mathbb{M}}(S),vol_{\mathbb{M}}(V\setminus S)\}}=\frac{cut_{\mathbb{M}}(S)}{vol_{\mathbb{M}}(S)}$. Furthermore, $\frac{cut_{\mathbb{M}}(S)}{vol_{\mathbb{M}}(S)}=\frac{cut_{\mathbb{M}}(S)}{\sum_{u\in S}\mathbb{M}(u)}=1-\frac{\sum_{u\in S}\mathbb{M}(u)-cut_{\mathbb{M}}(S)}{\sum_{u\in S}\mathbb{M}(u)}$. Thus, we have $\phi_{\mathbb{M}}(S)=1-g_{\mathbb{M}}(S)$ if $vol_{\mathbb{M}}(S) \le vol_{\mathbb{M}}(V \setminus S)$. □

Remark: Although Algorithm 3 may encounter $vol_{\mathbb{M}}(S) > vol_{\mathbb{M}}(V \setminus S)$, such case will not be the final result. The reasons can be analyzed as follows. Since $cut_{\mathbb{M}}(C)=cut_{\mathbb{M}}(V \setminus C)$, we have $\phi_{\mathbb{M}}(C)=\phi_{\mathbb{M}}(V \setminus C)$ for any $C \subseteq V$ (Definition 2 and (1) and (2)). Thus, assume that $\phi_{\mathbb{M}}(S)$ has the smallest value if $vol_{\mathbb{M}}(S) \ge vol_{\mathbb{M}}(V \setminus S)$, we can directly derive that $\phi_{\mathbb{M}}(V \setminus S)$ also has the smallest value. As a result, for convenience, we only need to output the small end in terms of $vol_{\mathbb{M}}$. Namely, *motif conductance based graph clustering* is to identify a vertex set S with $vol_{\mathbb{M}}(S) \le vol_{\mathbb{M}}(V \setminus S)$ and $\phi_{\mathbb{M}}(S)$ has the smallest value (Problem Statement of Section II). Based on the facts presented above, we can know that the vertex set S with $vol_{\mathbb{M}}(S) > vol_{\mathbb{M}}(V \setminus S)$ is meaningless and is not possible the resulting cluster. So, in Lemma 3, assume that $vol_{\mathbb{M}}(S) \le vol_{\mathbb{M}}(V \setminus S)$ is sufficient to solve our problem.

Lemma 4: Given an unweighted and undirected graph $G(V,E)$, a motif $\mathbb{M}$ with $k=k(\mathbb{M})$, we have $Mr_{\widetilde{S}}(u) \ge g_{\mathbb{M}}(\widetilde{S})$ for any $u \in \widetilde{S}$.

Proof: This lemma can be proved by contradiction. Let vertex $u \in \widetilde{S}$ such that $\frac{\mathbb{M}(u)+\mathbb{M}_k^{\widetilde{S}}(u)-\mathbb{M}_1^{\widetilde{S}}(u)}{\mathbb{M}(u)} < g_{\mathbb{M}}(\widetilde{S})$, we have $\mathbb{M}(u)+\mathbb{M}_k^{\widetilde{S}}(u)-\mathbb{M}_1^{\widetilde{S}}(u) < \mathbb{M}(u)\cdot g_{\mathbb{M}}(\widetilde{S})$. Therefore,

$$g_{\mathbb{M}}(\widetilde{S}\setminus\{u\})=\frac{\sum_{v\in\widetilde{S}}\mathbb{M}(v)-\mathbb{M}(u)-cut_{\mathbb{M}}(\widetilde{S}\setminus\{u\})}{\sum_{v\in\widetilde{S}}\mathbb{M}(v)-\mathbb{M}(u)}$$
$$=\frac{g_{\mathbb{M}}(\widetilde{S})\cdot\sum_{v\in\widetilde{S}}\mathbb{M}(v)+cut_{\mathbb{M}}(\widetilde{S})-\mathbb{M}(u)-cut_{\mathbb{M}}(\widetilde{S}\setminus\{u\})}{\sum_{v\in\widetilde{S}}\mathbb{M}(v)-\mathbb{M}(u)}$$
$$=\frac{g_{\mathbb{M}}(\widetilde{S})\cdot\sum_{v\in\widetilde{S}}\mathbb{M}(v)-\mathbb{M}(u)-\mathbb{M}_k^{\widetilde{S}}(u)+\mathbb{M}_1^{\widetilde{S}}(u)}{\sum_{v\in\widetilde{S}}\mathbb{M}(v)-\mathbb{M}(u)}$$
$$>\frac{g_{\mathbb{M}}(\widetilde{S})\cdot\sum_{v\in\widetilde{S}}\mathbb{M}(v)-\mathbb{M}(u)\cdot g_{\mathbb{M}}(\widetilde{S})}{\sum_{v\in\widetilde{S}}\mathbb{M}(v)-\mathbb{M}(u)}=g_{\mathbb{M}}(\widetilde{S})$$

Therefore, $g_{\mathbb{M}}(\widetilde{S}\setminus\{u\}) > g_{\mathbb{M}}(\widetilde{S})$, which contradicts that $\widetilde{S}$ is the optimal vertex set for $g_{\mathbb{M}}(.)$. As a consequence, $\frac{\mathbb{M}(u)+\mathbb{M}_k^S(u)-\mathbb{M}_1^S(u)}{\mathbb{M}(u)} \ge g_{\mathbb{M}}(\widetilde{S})$ for any $u \in \widetilde{S}$ holds. □

Implications of Lemma 3 and Lemma 4: Since $g_{\mathbb{M}}(\widetilde{S}) \ge g_{\mathbb{M}}(S)$ for any $S \subseteq V$, Lemma 3 indicates that $\phi_{\mathbb{M}}(\widetilde{S})=\phi_{\mathbb{M}}(S^*)$ where S^* is our optimal vertex set. This is because that S^* satisfies $vol_{\mathbb{M}}(S^*) \le vol_{\mathbb{M}}(V \setminus S^*)$, thus the condition of $vol_{\mathbb{M}}(S) \le vol_{\mathbb{M}}(V \setminus S)$ in Lemma 3 is always true for our problem. Please see Problem Statement in Section II for details. Meanwhile, Lemma 4 indicates that the motif resident of any vertex $u \in \widetilde{S}$ w.r.t $\widetilde{S}$ is at least $g_{\mathbb{M}}(\widetilde{S})$. Namely, the motif resident of any vertex in S^* w.r.t S^* is at least $1-\phi_{\mathbb{M}}^*$. Based on these implications, we can derive the following significant lemma and theorem to local a cluster with *fixed* and *motif-independent* approximation ratio for any motif.

Lemma 5: Given an unweighted and undirected graph $G(V,E)$, a motif $\mathbb{M}$ with $k=k(\mathbb{M})$, and a vertex subset S, we have $g_{\mathbb{M}}(S) \ge \frac{\sum_{u\in S}(\mathbb{M}(u)+\mathbb{M}_k^S(u)-\mathbb{M}_1^S(u))}{2\sum_{u\in S}\mathbb{M}(u)}$.

Proof: By the definitions of $cut_{\mathbb{M}}(S)$, $\mathbb{M}(u)$, and $\mathbb{M}_j^S(u)$, we have $\mathbb{M}(u)=\sum_{j=1}^{k}\mathbb{M}_j^S(u)$ and $cut_{\mathbb{M}}(S)=\sum_{u\in S}\sum_{j=1}^{k-1}\frac{1}{j}\mathbb{M}_j^S(u)$. By the definition of $g_{\mathbb{M}}(S)$, we have

$$g_{\mathbb{M}}(S)=\frac{\sum_{u\in S}\mathbb{M}(u)-cut_{\mathbb{M}}(S)}{\sum_{u\in S}\mathbb{M}(u)}$$
$$=\frac{\sum_{u\in S}(\mathbb{M}(u)-\sum_{j=1}^{k-1}\frac{1}{j}\mathbb{M}_j^S(u))}{\sum_{u\in S}\mathbb{M}(u)}$$
$$=\frac{\sum_{u\in S}(\sum_{j=2}^{k-1}(1-\frac{1}{j})\mathbb{M}_j^S(u)+\mathbb{M}_k^S(u))}{\sum_{u\in S}\mathbb{M}(u)}$$
$$\ge\frac{\sum_{u\in S}(\sum_{j=2}^{k-1}\frac{1}{2}\mathbb{M}_j^S(u)+\mathbb{M}_k^S(u))}{\sum_{u\in S}\mathbb{M}(u)}$$
$$=\frac{\sum_{u\in S}(\sum_{j=2}^{k-1}\mathbb{M}_j^S(u)+2\cdot\mathbb{M}_k^S(u))}{2\sum_{u\in S}\mathbb{M}(u)}$$
$$=\frac{\sum_{u\in S}(\mathbb{M}(u)+\mathbb{M}_k^S(u)-\mathbb{M}_1^S(u))}{2\sum_{u\in S}\mathbb{M}(u)}$$

Thus, this lemma is proved. □

Theorem 4: Algorithm 3 can identify a higher-order cluster with motif conductance $1/2+1/2\phi_{\mathbb{M}}^*$.

Proof: Let $\widetilde{S}$ is the optimal vertex set for $g_{\mathbb{M}}(.)$. In Lines 7-10, Algorithm 3 executes the peeling process. That is, in each round, it greedily deletes the vertex with the smallest motif

resident. Consider the round t when the first vertex v of $\widetilde{S}$ is deleted. Let V_t be the vertex set from the beginning of round t. $\widetilde{S}$ is the subset of V_t (i.e., $\widetilde{S} \subseteq V_t$) because v is the first deleted vertex of $\widetilde{S}$. This implies that $\min_{u \in V_t} Mr_{V_t}(u) = Mr_{V_t}(v) \geq Mr_{\widetilde{S}}(v) \geq g_{\mathbb{M}}(\widetilde{S})$ according to Lemmas 1 and 4. Therefore, for any $u \in V_t$, we have $\frac{\mathbb{M}(u)+\mathbb{M}_k^{V_t}(u)-\mathbb{M}_1^{V_t}(u)}{\mathbb{M}(u)} \geq g_{\mathbb{M}}(\widetilde{S})$. Furthermore, by Lemma 5, $g_{\mathbb{M}}(V_t) \geq \frac{\sum_{u \in V_t}(\mathbb{M}(u)+\mathbb{M}_k^{V_t}(u)-\mathbb{M}_1^{V_t}(u))}{2\sum_{u \in V_t}\mathbb{M}(u)} \geq \frac{1}{2}\frac{\sum_{u \in V_t} g_{\mathbb{M}}(\widetilde{S})\cdot\mathbb{M}(u)}{\sum_{u \in V_t}\mathbb{M}(u)} = \frac{1}{2}g_{\mathbb{M}}(\widetilde{S})$. Since Algorithm 3 maintains the optimal solution during the peeling process in Lines 11-12, $\phi_{\mathbb{M}}(\hat{S}) = 1 - g_{\mathbb{M}}(\hat{S}) \leq 1 - g_{\mathbb{M}}(V_t) \leq 1 - \frac{g_{\mathbb{M}}(\widetilde{S})}{2}$ due to Lemma 3. On the other hand, According to the definition of $\widetilde{S}$, we know that $g_{\mathbb{M}}(\widetilde{S}) \geq g_{\mathbb{M}}(S^*)$, in which S^* is the vertex set with optimal motif conductance. Thus, $\phi_{\mathbb{M}}(\hat{S}) \leq 1 - \frac{g_{\mathbb{M}}(\widetilde{S})}{2} \leq 1 - \frac{g_{\mathbb{M}}(S^*)}{2} = 1 - \frac{1-\phi_{\mathbb{M}}(S^*)}{2}$. Namely, $\phi_{\mathbb{M}}(\hat{S}) \leq 1/2 + 1/2\phi_{\mathbb{M}}(S^*)$. So, Algorithm 3 can identify a higher-order cluster with motif conductance $1/2 + 1/2\phi^*_{\mathbb{M}}$. □

B. *Efficient Dynamic Update of* Motif Resident

The computational challenge of Algorithm 3 is how to incrementally maintain Mr_{S_i} in Line 8 and $\phi_{\mathbb{M}}(S_i)$ in Line 11 when a vertex u is removed. Note that since $\phi_{\mathbb{M}}(S_i) = 1 - g_{\mathbb{M}}(S_i)$ by Lemma 3, we can maintain $\phi_{\mathbb{M}}(S_i)$ by maintaining $g_{\mathbb{M}}(S_i)$. Therefore, we propose the following efficient dynamic update technologies to solve the challenge.

Lemma 6: Given the current search space S_i, if a vertex $u \in S_i$ is removed, let $S_{i+1} = S_i \setminus \{u\}$ and for any $v \in S_{i+1}$, we have the following equation:

$$g_{\mathbb{M}}(S_{i+1}) = \frac{vol_{\mathbb{M}}(S_i)g_{\mathbb{M}}(S_i) - Mr_{S_i}(u)\mathbb{M}(u)}{vol_{\mathbb{M}}(S_i) - \mathbb{M}(u)} \tag{6}$$

$$Mr_{S_{i+1}}(v) = Mr_{S_i}(v) - \frac{\mathbb{M}_k^{S_i}(u,v) + \mathbb{M}_2^{S_i}(u,v)}{\mathbb{M}(v)} \tag{7}$$

Where $\mathbb{M}_k^{S_i}(u,v) = |\{G_S \in \mathbb{M} | (\{u,v\} \subseteq S, |S_i \cap S| = k\}|$ and $\mathbb{M}_2^{S_i}(u,v) = |\{G_S \in \mathbb{M} | \{u,v\} \subseteq S, |S_i \cap S| = 2\}|$. That is $\mathbb{M}_k^{S_i}(u,v)$ (resp., $\mathbb{M}_2^{S_i}(u,v)$) is the number of motif instances containing the node pair $\{u,v\}$ with exactly k (resp., 2) vertices in S_i.

Proof: By the proof process of Lemma 4, we have $g_{\mathbb{M}}(S_{i+1}) = \frac{g_{\mathbb{M}}(S_i)\cdot\sum_{w \in S_i}\mathbb{M}(v) - \mathbb{M}(u) - \mathbb{M}_k^{S_i}(u) + \mathbb{M}_1^{S_i}(u)}{\sum_{w \in S_i}\mathbb{M}(v) - \mathbb{M}(u)}$. Besides, we have $\mathbb{M}(u) + \mathbb{M}_k^{S_i}(u) - \mathbb{M}_1^{S_i}(u) = Mr_{S_i}(u)\mathbb{M}(u)$ by Definition 4. Thus, $g_{\mathbb{M}}(S_{i+1}) = \frac{vol_{\mathbb{M}}(S_i)g_{\mathbb{M}}(S_i) - Mr_{S_i}(u)\mathbb{M}(u)}{vol_{\mathbb{M}}(S_i) - \mathbb{M}(u)}$ due to $\sum_{w \in S_i}\mathbb{M}(v) = vol_{\mathbb{M}}(S_i)$. On the other hand, by definitions of $\mathbb{M}_k^S(u)$, $\mathbb{M}_1^S(u)$, $\mathbb{M}_k^S(u,v)$, and $\mathbb{M}_2^S(u,v)$, we have $\mathbb{M}_k^{S_{i+1}}(v) = \mathbb{M}_k^{S_i}(v) - \mathbb{M}_k^{S_i}(u,v)$ and $\mathbb{M}_1^{S_{i+1}}(v) = \mathbb{M}_1^{S_i}(v) + \mathbb{M}_2^{S_i}(u,v)$. According to Definition 4, we can further obtain that $Mr_{S_{i+1}}(v) = \frac{\mathbb{M}(v)+\mathbb{M}_k^{S_{i+1}}(v)-\mathbb{M}_1^{S_{i+1}}(v)}{\mathbb{M}(v)} = \frac{\mathbb{M}(v)+\mathbb{M}_k^{S_i}(v)-\mathbb{M}_k^{S_i}(u,v)-\mathbb{M}_1^{S_i}(v)-\mathbb{M}_2^{S_i}(u,v)}{\mathbb{M}(v)} = Mr_{S_i}(v) - \frac{\mathbb{M}_k^{S_i}(u,v)+\mathbb{M}_2^{S_i}(u,v)}{\mathbb{M}(v)}$. In short, the lemma is hold. □

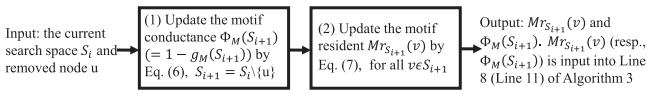


Fig. 3. The flowchart of motif resident dynamic update process.

Let $N_{\mathbb{M}}(u) = \{v | u, v \in mi, mi \in \mathbb{M}\}$ be the motif neighbor set of u. Namely, for any node $v \in N_{\mathbb{M}}(u)$, we have the two nodes u and v contained in at least one motif instance. By Lemma 6, if $v \notin S_{i+1} \cap N_{\mathbb{M}}(u)$, we have $Mr_{S_{i+1}}(v) = Mr_{S_i}(v)$. Thus, Algorithm 3 can incrementally update the motif resident for each vertex when its motif neighbor is removed. Fig. 3 shows how motif resident dynamic update process works.

Theorem 5: The worse-case time complexity and space complexity of Algorithm 3 are $O(\mathcal{T}(\mathbb{M}))$ and $O(m+n)$, respectively. Where $O(\mathcal{T}(\mathbb{M}))$ is the time complexity of enumerating motif instances.

Proof: Algorithm 3 first takes $O(\mathcal{T}(\mathbb{M}))$ time to calculate the motif degree $\mathbb{M}(u)$ for any $u \in V$ by enumerating motif instances (Lines 1-4). In Lines 7-11, Algorithm 3 can incrementally calculate Mr_{S_i} and $\phi_{\mathbb{M}}(S_i)$ (note that $\phi_{\mathbb{M}}(S_i) = 1 - g_{\mathbb{M}}(S_i)$ by Lemma 3) by Lemma 6. Specifically, we assume that u_i is the deleted vertex in i-th round, it takes $cost(u_i)$ time to incrementally update Mr_{S_i} and $O(1)$ time to update $g_{\mathbb{M}}(S_i)$, in which $cost(u_i)$ is the time cost of computing $\mathbb{M}(u_i)$. This is because calculating $\mathbb{M}_k^{S_i}(u_i, v)$ and $\mathbb{M}_2^{S_i}(u_i, v)$ for all $v \in S_i \cap N_{\mathbb{M}}(u)$ take at most $cost(u_i)$. For $v \notin S_i \cap N_{\mathbb{M}}(u)$, we have $Mr_{S_{i+1}}(v) = Mr_{S_i}(v)$ by (7). Thus, it takes $O(\sum_{i=1}^{n} cost(u_i)) = O(\mathcal{T}(\mathbb{M}))$ to execute Lines 7-11. Algorithm 3 takes $O(\mathcal{T}(\mathbb{M}))$ time in total.

For the space complexity, Algorithm 3 just needs an extra $O(n)$ to store $\mathbb{M}(u)$, $\mathbb{M}_k^{S_i}(u)$, $\mathbb{M}_1^{S_i}(u)$, $Mr_{S_i}(u)$ and $O(m+n)$ to store the inputed graph G. As a consequence, Algorithm 3 needs $O(n+m)$ space in total. □

C. *Discussions*

Comparison with [33]. *PCon* is an excellent conductance-based graph clustering framework [33], which aims to sort the vertices and then perform the sweep-cut operation on the sorted vertex ordering. This framework covers the mainstream ideas of existing SOTA conductance-based graph clustering algorithms [33]. Different algorithms distinguish themselves in how they sort the vertices and whether the sweep-cut operation can provide theoretical guarantees. Existing conductance-based graph clustering approaches typically sort the vertices by performing expensive eigenvector computations or heuristic local graph diffusion, and they provide poor clustering qualities. Thus, [33] sorts the vertices using an easy-to-compute degree-like (they call degeneracy or degree ratio) metric and obtains high-quality clusters. Inspired by this, our paper is the first to non-trivially apply this peeling architecture to devise scalable and provable algorithms for higher-order graph clustering, which are our main highlight contributions. Several pivotal differences between [33] and we are described below: (1) when the input motif is an edge, the motif conductance degenerates into classic conductance (Definition 2). Thus, our proposed algorithm is a generalization of [33]. As a result, the proof of our algorithm's approximation ratio is more challenging and requires some careful design tricks. For example, Lemmas

1, 2, and 5 are unique properties of our problem. (2) When the input motif is not an edge, our proposed motif resident (Definition 4) is more complex than the degree ratio of [33]. For example, [33] can calculate the degree ratio of all vertices (the degree ratio of node $u \in S$ w.r.t. G and G_S is defined as $Dr_S(u) = \frac{|N_S(u)|}{|N_V(u)|}$) in linear time by simply visiting all edges. However, existing methods take $O(\mathcal{T}(\mathbb{M}))$ time to calculate the proposed motif resident (Definition 4), in which $O(\mathcal{T}(\mathbb{M}))$ is the time complexity of enumerating motif instances. For example, when the motif is a k-clique, $O(\mathcal{T}(\mathbb{M})) = O(k(\frac{\delta}{2})^{k-2}m)$ by running the SOTA clique enumeration algorithms [40] where δ is the degeneracy (Table II). (3) Updating the motif resident is more difficult than the degree ratio of [33]. In particular, assume that S is the current search space, updating the degree ratio of node u requires only $O(|N_S(u)|)$, but updating motif resident of u requires $O(|N_S(u)|^k)$. By exploring the intrinsic properties of motif resident, we devise the powerful dynamic update technologies to incrementally maintain the motif resident of each vertex, thereby avoiding recomputing the motif resident from scratch (Lemma 6). (4) Estimating the motif resident needs more theoretical knowledge. For example, the degree ratio needs only one edge information, while motif resident has a more complex motif relationship. We propose effective inclusion-exclusion-based Turan estimator and colorful wedges estimator, which are our novel optimizers (Section V-A).

Summing up, the highlight novelty of our paper is to peel the vertices by our proposed motif resident metric. There are two merits of using the motif resident metric. First, the motif resident is faster to compute, update (Section IV-B), and estimate (Section V-A) than existing spectral clustering or local graph diffusion based on motif instances. Second, obtaining theoretically guaranteed higher-order clustering quality for arbitrary motifs has been a long-standing open problem. To address this issue, we theoretically prove our *PSMC* has a motif-independent approximation ratio by elegantly exploiting the relationship between the motif resident and motif conductance.

V. Optimizations

In this section, we propose two optimization strategies: one involves estimating the bounds of motif resident, which is the most time-consuming part of *PSMC*; the other is iteratively identifying small subgraphs that potentially contain the target results, and then conducting the search only within these small subgraphs, thereby significantly reducing the time overhead.

A. PSMC With Bound Estimation Strategies (PSMC+)

Although our proposed *PSMC* (i.e., Algorithm 3) can improve the worse-case time&space complexities and clustering qualities of SOTA methods (Table I), it still requires implicitly enumerating motif instances (Theorem 5), resulting in poor scalability. This is because the motif resident of a vertex u implicitly depends on the number of motif instances u participates in, causing *PSMC* to indirectly calculate all motif instances. For example, on the DBLP dataset with millions of edges, our *PSMC* takes around 2500 seconds to process a query when the motif is a 6-clique (Section VI), which is not acceptable for the online user experience. Inspired by this, we devise the following lower and upper bounds of motif resident and further propose heuristic solutions for faster queries.

By (7) and (6), we know that the bottleneck of Algorithm 3 is how to quickly obtain the motif degree $\mathbb{M}(u)$, $\mathbb{M}_k^{S_i}(u,v)$ and $\mathbb{M}_2^{S_i}(u,v)$. Inspired by this, we propose effective lower and upper bounds to estimate them, which can be computed locally. Specifically, let NS_u, NS_{uv}^S, and $NS_{uv}^{V\setminus S}$ be the *neighbor subgraph* induced by $N(u)$, $N_S(u) \cap N_S(v)$, and $N_{V\setminus S}(u) \cap N_{V\setminus S}(v)$, respectively. Following existing research work [21], [29], [41], we take acquiescently the given motif is a clique (a clique is a complete graph such that there is an edge between every pair of vertices). This is because clique-based higher-order graph clustering is widely used in numerous applications [21], [29], [41], we mainly focus on bounds of motif resident for clique in this paper (other motifs are more challenging and is our future work). Thus, we have $\mathbb{M}(u)$ and $\mathbb{M}_k^{S_i}(u,v)$(resp., $\mathbb{M}_2^{S_i}(u,v)$) are the number of $(k-1)$-cliques in NS_u and the number of $(k-2)$-cliques in $NS_{uv}^{S_i}$ (resp., $NS_{uv}^{V\setminus S_i}$), respectively. As a consequence, the estimate of $\mathbb{M}(u)$, $\mathbb{M}_k^{S_i}(u,v)$ and $\mathbb{M}_2^{S_i}(u,v)$ becomes to estimate the number of cliques in the corresponding subgraph.

Lower Bounds: For convenience, we use NS to specify which neighbor subgraph is adopted, i.e., NS_u, $NS_{uv}^{S_i}$, and $NS_{uv}^{V\setminus S_i}$. The following Theorem is one of the most important results in extremal graph theory, which can be used to estimate the number of cliques.

Theorem 6 (Turan Theorem [42]): For any subgraph NS, if $\frac{2E(NS)}{|V(NS)|(|V(NS)|-1)} > 1 - \frac{1}{r-1}$, then NS contains a r-clique.

According to Theorem 6, we have the following facts.

Fact 1: Let $D = \frac{2E(NS)}{|V(NS)|(|V(NS)|-1)}$ and $r = \lfloor \frac{1}{1-D} \rfloor + 1$, we have: (1) $\mathbb{M}(u) \geq \binom{r}{k-1}$ if $NS = NS_u$; (2) $\mathbb{M}_k^{S_i}(u,v) \geq \binom{r}{k-2}$ if $NS = NS_{uv}^{S_i}$; (3) $\mathbb{M}_2^{S_i}(u,v) \geq \binom{r}{k-2}$ if $NS = NS_{uv}^{V\setminus S_i}$.

The well-known graph theory expert *Paul Erdos* proposed the following tighter theorem to expand the Turan Theorem.

Theorem 7: [43] For any subgraph NS, if $\frac{2E(NS)}{|V(NS)|(|V(NS)|-1)} > 1 - \frac{1}{r-1}$, then NS contains at least $(\frac{|V(NS)|}{r-1})^{r-2}$ r-cliques.

Let h is an integer and $A_i = \{C_i^1, C_i^2, \ldots, C_i^{\binom{r}{h}}\}$ be the h-clique set obtained from the i-th r-clique of Theorem 7, in which $i \in \{1, 2, \ldots, (\frac{|V(NS)|}{r-1})^{r-2}\}$. Since two r-cliques have at most $r-1$ common vertices, $|A_i \cap A_j| \leq \binom{r-1}{h}$. According to the *inclusion-exclusion principle* [44] and let $t = (\frac{|V(NS)|}{r-1})^{r-2}$, we have $|\bigcup_{i=1}^t A_i| \geq \sum_{i=1}^t |A_i| - \sum_{1\leq i<j\leq t} |A_i \cap A_i| \geq t\binom{r}{h} - \binom{t}{2}\binom{r-1}{h}$. So, we have the following facts.

Fact 2: Let $D = \frac{2E(NS)}{|V(NS)|(|V(NS)|-1)}$, $r = \lfloor \frac{1}{1-D} \rfloor + 1$, and $t = (\frac{|V(NS)|}{r-1})^{r-2}$, we have: (1) $\mathbb{M}(u) \geq t\binom{r}{k-1} - \binom{t}{2}\binom{r-1}{k-1}$ if $NS = NS_u$; (2) $\mathbb{M}_k^{S_i}(u,v) \geq t\binom{r}{k-2} - \binom{t}{2}\binom{r-1}{k-2}$ if $NS = NS_{uv}^{S_i}$; (3) $\mathbb{M}_2^{S_i}(u,v) \geq t\binom{r}{k-2} - \binom{t}{2}\binom{r-1}{k-2}$ if $NS = NS_{uv}^{V\setminus S_i}$.

In a nutshell, we can obtain the lower bounds of $\mathbb{M}(u)$, $\mathbb{M}_k^{S_i}(u,v)$ and $\mathbb{M}_2^{S_i}(u,v)$ according to Fact 1 and Fact 2.

Upper Bounds: Gao et al. proposed the concept of colorful h-star degree $csd_h(u)$ [45], which can be computed in $O(h|N(u)|)$ time. Specifically, $csd_h(u)$ is the number of colorful h-stars centered on u, in which a colorful h-star is a star with h vertices having different colors (Fig. 4(a)). Since each vertex in h-clique must have a different color, we can obtain $csd_h(u) \geq \mathbb{M}(u)$

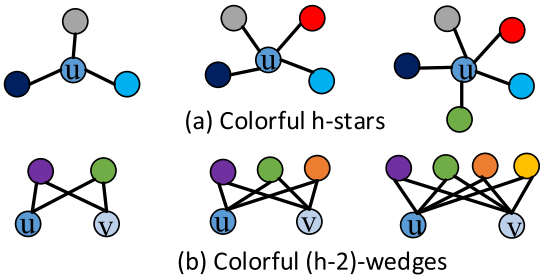


Fig. 4. Colorful h-stars and colorful $(h-2)$-wedges.

if $h = k(\mathbb{M})$. For estimating $\mathbb{M}_k^{S_i}(u,v)$ and $\mathbb{M}_2^{S_i}(u,v)$, we propose a novel concept of colorful $(h-2)$-wedge degree $cwd_h^S(u,v)$ w.r.t. the vertex set S. $cwd_h^S(u,v)$ is the number of colorful $(h-2)$-wedges of S with u and v as endpoints (there may be no edge between u and v), in which a colorful $(h-2)$-wedge of S is a set of $h-2$ wedges (a wedge is a path with three nodes) such that all vertices are in S and having different colors (Fig. 4(b)). Assume that (u,v) is an edge and $hc(u,v)$ is any h-clique containing the edge (u,v), each vertex in $hc(u,v)$ must have a different color. Therefore, $cwd_h^{S_i}(u,v) \geq \mathbb{M}_k^{S_i}(u,v)$ and $cwd_h^{V\setminus S_i}(u,v) \geq \mathbb{M}_2^{S_i}(u,v)$ if $h = k(\mathbb{M})$.

So far we have obtained the upper and lower bounds of $\mathbb{M}(u)$, $\mathbb{M}_k^{S_i}(u,v)$ and $\mathbb{M}_2^{S_i}(u,v)$, which can be computed and updated locally. Furthermore, we can also directly take the average of their upper and lower bounds as the corresponding estimated value. Based on these bounds, we propose a heuristic algorithm *PSMC+* (Algorithm 4) which aims to optimize efficiency. Specifically, Line 1 initializes some variables, in which $\widetilde{g_{\mathbb{M}}(V)} \leftarrow 1$ and $\widetilde{Mr_V(u)} \leftarrow 2$ for any $u \in V$ are calculated from their definitions. Then, Lines 2-13 utilize the lower&upper bounds proposed in Section V-A to estimate the motif degree of nodes and edges, which will be used to estimate the motif resident for each node. Subsequently, Lines 14-22 iteratively remove the vertex with the smallest estimated motif resident (Lines 15, 19, and 20) and update the variables (Lines 17, 18, 21, 22) needed for the next iteration by the proposed dynamic update strategies (Section IV-B) and estimation strategies (Section V-A). Finally, Lines 23-24 output the result with the smallest estimated motif conductance during the iterative process.

Algorithm 4: *PSMC* With Bound Estimation Strategies (*PSMC+*).

Input: A graph $G(V,E)$ and a motif $\mathbb{M}$ with $k = k(\mathbb{M})$
Output: A higher-order cluster $\widetilde{S}$

1: $i \leftarrow 1$; $S_i \leftarrow V$; $\widetilde{g_{\mathbb{M}}(S_i)} \leftarrow 1$; $\widetilde{vol_{\mathbb{M}}(S_i)} \leftarrow 0$; $\widetilde{Mr_{S_i}(u)} \leftarrow 2$ for any $u \in S_i$
2: **for** $u \in S_i$ **do**
3: $L\mathbb{M}(u) \leftarrow$ the lower bound of $\mathbb{M}(u)$ by Section V-A
4: $U\mathbb{M}(u) \leftarrow$ the upper bound of $\mathbb{M}(u)$ by Section V-A
5: $\widetilde{\mathbb{M}(u)} \leftarrow \frac{L\mathbb{M}(u)+U\mathbb{M}(u)}{2}$
6: $\widetilde{vol_{\mathbb{M}}(S_i)} \leftarrow \widetilde{vol_{\mathbb{M}}(S_i)} + \widetilde{\mathbb{M}(u)}$
7: **for** $v \in N_{S_i}(u)$ **do**
8: $L\mathbb{M}_k^{S_i}(u,v) \leftarrow$ the lower bound of $\mathbb{M}_k^{S_i}(u,v)$ by Section V-A
9: $L\mathbb{M}_2^{S_i}(u,v) \leftarrow$ the lower bound of $\mathbb{M}_2^{S_i}(u,v)$ by Section V-A
10: $U\mathbb{M}_k^{S_i}(u,v) \leftarrow$ the upper bound of $\mathbb{M}_k^{S_i}(u,v)$ by Section V-A
11: $U\mathbb{M}_2^{S_i}(u,v) \leftarrow$ the upper bound of $\mathbb{M}_2^{S_i}(u,v)$ by Section V-A
12: $\widetilde{\mathbb{M}_k^{S_i}(u,v)} \leftarrow \frac{L\mathbb{M}_k^{S_i}(u,v)+U\mathbb{M}_k^{S_i}(u,v)}{2}$
13: $\widetilde{\mathbb{M}_2^{S_i}(u,v)} \leftarrow \frac{L\mathbb{M}_2^{S_i}(u,v)+U\mathbb{M}_2^{S_i}(u,v)}{2}$
14: **while** $S_i \neq \emptyset$ **do**
15: $u \leftarrow \arg\min\{\widetilde{Mr_{S_i}(u)} | u \in S_i\}$
16: **for** $v \in N_{S_i}(u)$ **do**
17: $\widetilde{Mr_{S_{i+1}}(v)} \leftarrow \widetilde{Mr_{S_i}(v)} - \frac{\widetilde{\mathbb{M}_k^{S_i}(u,v)}+\widetilde{\mathbb{M}_2^{S_i}(u,v)}}{\widetilde{\mathbb{M}(v)}}$
18: $\widetilde{g_{\mathbb{M}}(S_{i+1})} \leftarrow \frac{\widetilde{vol_{\mathbb{M}}(S_i)}\widetilde{g_{\mathbb{M}}(S_i)} - \widetilde{Mr_{S_i}(u)}\widetilde{\mathbb{M}(u)}}{\widetilde{vol_{\mathbb{M}}(S_i)} - \widetilde{\mathbb{M}(v)}}$
19: $i \leftarrow i+1$
20: $S_i \leftarrow S_{i-1} \setminus \{u\}$
21: $\widetilde{vol_{\mathbb{M}}(S_i)} \leftarrow \widetilde{vol_{\mathbb{M}}(S_{i-1})} - \widetilde{\mathbb{M}(u)}$
22: Updating $\widetilde{\mathbb{M}_2^{S_i}(u,v)}$ and $\widetilde{\mathbb{M}_k^{S_i}(u,v)}$by Section V-A
23: $\widetilde{S} \leftarrow \arg\min_{S\in\{S_1,S_2,\ldots,S_n\}}\{1 - \widetilde{g_{\mathbb{M}}(S)} | \widetilde{vol_{\mathbb{M}}(S)} \leq \widetilde{vol_{\mathbb{M}}(V \setminus S)}\}$
24: **return** $\widetilde{S}$

B. Core-Based PSMC (PSMC-C)

Although *PSMC+* can achieve lower complexity through motif resident estimation strategies, it is heuristic and lacks an accuracy guarantee, resulting in instability and very poor quality in some cases (Section VI). Thus, an important question is whether we can improve efficiency while maintaining quality. We answer this question affirmatively. Specifically, existing studies [19], [26], [27], [28], [29] have shown that higher-order clusters tend to be concentrated in small areas of the graph, which is also consistent with our experimental results (e.g., approximately 1/10000 of the original graph, as indicated by the "Size Column" in Table III). Inspired by this, we propose a core-based approach to quickly identify smaller subgraphs that are more likely to include the target results.

Definition 5 ($(\beta,\mathbb{M})$-Resident Core): Given an unweighted and undirected graph $G(V,E)$, a motif $\mathbb{M}$ with $k = k(\mathbb{M})$, and a positive β, the $(\beta,\mathbb{M})$-resident core, denoted $\mathcal{C}_\beta$, is the largest subgraph of G such that $\forall u \in \mathcal{C}_\beta$, $Mr_{\mathcal{C}_\beta}(u) \geq \beta$.

Note that the $(\beta,\mathbb{M})$-resident core is similar to the k-core [46], which is how our method got its name. Where a k-core is the largest subgraph such that each node in the subgraph has at least k neighbors. Similar to the core number [46], we define the *motif resident core number* of a vertex u, $\mathcal{C}(u)$, is the maximum β of a $\mathcal{C}_\beta$ containing u.

Lemma 7: Given an unweighted and undirected graph $G(V,E)$, a motif $\mathbb{M}$ with $k = k(\mathbb{M})$, and any $(\beta,\mathbb{M})$-resident core $\mathcal{C}_\beta \subseteq V$, we have $\frac{\beta}{2} \leq g_{\mathbb{M}}(\mathcal{C}_\beta) \leq \beta_{max}$. Where $\beta_{max} = \max\arg\{\beta | \mathcal{C}_\beta \neq \emptyset\}$.

Proof: (1) Let $\widetilde{S}$ be the optimal vertex set for $g_{\mathbb{M}}(.)$. That is, $g_{\mathbb{M}}(\widetilde{S}) \geq g_{\mathbb{M}}(S)$ for any vertex subset $S \subseteq V$. Thus, $g_{\mathbb{M}}(\widetilde{S}) \geq g_{\mathbb{M}}(\mathcal{C}_\beta)$ due to $\mathcal{C}_\beta \subseteq V$. Assume that $g_{\mathbb{M}}(\widetilde{S}) > \beta_{max}$, by Lemma 4, we have $Mr_{\widetilde{S}}(u) \geq g_{\mathbb{M}}(\widetilde{S}) > \beta_{max}$ for any $u \in \widetilde{S}$, which contradicts the definition of β_{max}. As a consequence, we have $g_{\mathbb{M}}(\widetilde{S}) \leq \beta_{max}$, further $g_{\mathbb{M}}(\mathcal{C}_\beta) \leq \beta_{max}$.

(2) According to Definition 4, for any node $u \in \mathcal{C}_\beta$, we have $Mr_{\mathcal{C}_\beta}(u)M(u) = \mathbb{M}(u) + \mathbb{M}_k^{\mathcal{C}_\beta}(u) - \mathbb{M}_1^{\mathcal{C}_\beta}(u)$. Moreover, by Lemma 5 and Definition 5, we can obtain $g_{\mathbb{M}}(\mathcal{C}_\beta) \geq \frac{\sum_{u\in\mathcal{C}_\beta}(\mathbb{M}(u)+\mathbb{M}_k^{\mathcal{C}_\beta}(u)-\mathbb{M}_1^{\mathcal{C}_\beta}(u))}{2\sum_{u\in\mathcal{C}_\beta}\mathbb{M}(u)} = \frac{\sum_{u\in\mathcal{C}_\beta} Mr_{\mathcal{C}_\beta}(u)M(u)}{2\sum_{u\in\mathcal{C}_\beta}\mathbb{M}(u)} \geq \beta/2$. □

Lemma 8: Given an unweighted and undirected graph $G(V, E)$, the $(\beta_{max}, \mathbb{M})$-resident core $\mathcal{C}_{\beta_{max}}$ is a higher-order cluster with motif conductance $1/2 + 1/2\phi^*_{\mathbb{M}}$.

Proof: By Lemma 7, we have $\frac{\beta_{max}}{2} \leq g_{\mathbb{M}}(\mathcal{C}_{\beta_{max}})$ and $g_{\mathbb{M}}(S^*) \leq \beta_{max}$ due to $S^* \subseteq V$, in which S^* is the vertex set with optimal motif conductance. Thus, by Lemma 3, we have $\phi_{\mathbb{M}}(\mathcal{C}_{\beta_{max}}) = 1 - g_{\mathbb{M}}(\mathcal{C}_{\beta_{max}}) \leq 1 - \frac{\beta_{max}}{2} \leq 1 - \frac{g_{\mathbb{M}}(S^*)}{2} = 1 - \frac{1-\phi_{\mathbb{M}}(S^*)}{2}$. Namely, $\phi_{\mathbb{M}}(\mathcal{C}_{\beta_{max}}) \leq 1/2 + 1/2\phi_{\mathbb{M}}(S^*)$. So, $\mathcal{C}_{\beta_{max}}$ is a cluster with motif conductance $1/2 + 1/2\phi^*_{\mathbb{M}}$. □

By Theorem 4 and Lemma 8, we know that $\mathcal{C}_{\beta_{max}}$ and Algorithm 3 provide the same theoretical guarantee; therefore, $\mathcal{C}_{\beta_{max}}$ is the result we are striving for. Since the motif resident is monotonicity (Lemma 1), by Definition 5, a straightforward approach is to use the core decomposition-style [46] to compute all the $\mathcal{C}_\beta$, and then output the $\mathcal{C}_{\beta_{max}}$. Clearly, such a solution has the same time complexity as *PSMC* (i.e., Algorithm 3). To further improve *practical* efficiency, we devise another advanced solution *PSMC*-C (Algorithm 5), which aims to identify the $\mathcal{C}_{\beta_{max}}$ directly without obtaining all the intermediate $\mathcal{C}_\beta$. Specifically, Algorithm 5 first computes the upper bound $\widetilde{\mathbb{C}(u)}$ for each vertex $u \in V$ (Lines 1-6) and sorts vertices based on their $\widetilde{\mathbb{C}(u)}$ (Line7). Then, Line 8 initializes three variables R as the candidate result, $\hat{\beta}$ as the currently maximum β, and $\hat{S}$ as the output result. Subsequently, Lines 9-19 identify iteratively the $\mathcal{C}_{\beta_{max}}$. Specifically, Line 10 first computes the motif resident $Mr_R(u)$ for any $u \in R$, then Lines 12-16 iteratively remove the vertice with the smallest motif resident and maintain the currently maximum $\hat{\beta}$ and the corresponding subgraph $\hat{S}$. After that, Line 19 doubles the size of R for the next iteration. Once Line 17 is satisfied, Algorithm 5 stops the iteration and outputs $\hat{S}$ in Line 18.

Theorem 8: Algorithm 5 can identify a higher-order cluster with motif conductance $1/2 + 1/2\phi^*_{\mathbb{M}}$. The worse-case time complexity and space complexity of Algorithm 5 are $O(\mathcal{T}(\mathbb{M}))$ and $O(m + n)$, respectively. Where $O(\mathcal{T}(\mathbb{M}))$ is the time complexity of enumerating motif instances.

Proof: Once the stopping criterion (i.e., Line 17) is satisfied, the maximum motif resident core number of all the remaining vertices (i.e., $V \setminus R_1$) is less than $\hat{\beta}$, $V \setminus R_1$ must not be in $(\hat{\beta}, \mathbb{M})$-resident core. Meanwhile, Algorithm 5 maintains the optimal β during the iteration process (Lines 14-15), thus it can correctly obtain $\mathcal{C}_{\beta_{max}}$ (i.e., $\hat{S} = \mathcal{C}_{\beta_{max}}$). As a result, by Lemma 8, we have $\hat{S}$ is a higher-order cluster with motif conductance $1/2 + 1/2\phi^*_{\mathbb{M}}$.

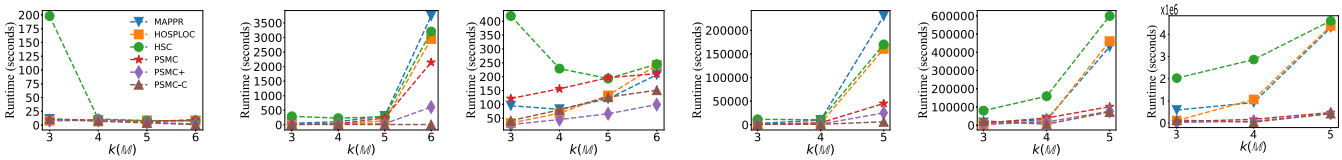
Algorithm 5: Core-Based *PSMC* (*PSMC-C*).

Input: A graph $G(V, E)$ and a motif $\mathbb{M}$ with $k = k(\mathbb{M})$
Output: A higher-order cluster $\hat{S}$ with motif-independent approximation ratio

1: **for** each node $u \in V$ **do**
2: $L\mathbb{M}(u) \leftarrow$ the lower bound of $\mathbb{M}(u)$ by Section V-A
3: $U\mathbb{M}(u) \leftarrow$ the upper bound of $\mathbb{M}(u)$ by Section V-A
4: $\widetilde{\mathbb{M}(u)} \leftarrow \frac{L\mathbb{M}(u)+U\mathbb{M}(u)}{2}$
5: **for** each node $u \in V$ **do**
6: $\widetilde{\mathbb{C}(u)} \leftarrow$ the core number of u by execute the core decomposition on $\widetilde{\mathbb{M}()}$ [46]
7: Sorting all vertices $u \in V$ such that $\widetilde{\mathbb{C}(u_1)} \geq \widetilde{\mathbb{C}(u_2)} \ldots \geq \widetilde{\mathbb{C}(u_n)}$
8: Initializing $R = \{u_1\}$, $\hat{\beta} \leftarrow 0$, and $\hat{S} \leftarrow \emptyset$
9: **while** True **do**
10: $Mr_R(u) \leftarrow \frac{\mathbb{M}(u)+\mathbb{M}_k^R(u)-\mathbb{M}_1^R(u)}{\mathbb{M}(u)}$ for any $u \in R$
11: $rlen \leftarrow |R|$, $R_1 \leftarrow R$
12: **while** $|R| > 0$ **do**
13: $u \leftarrow \arg\min\{Mr_R(u) | u \in R\}$ and $\beta \leftarrow Mr_R(u)$
14: **if** $\beta > \hat{\beta}$ **then**
15: $\hat{\beta} \leftarrow \beta$, $\hat{S} \leftarrow R$
16: $R \leftarrow R \setminus \{u\}$ and updating motif resident by Lemma 6
17: **if** $\max_{u\in V\setminus R_1} \widetilde{\mathbb{C}(u)} \leq \hat{\beta}$ **then**
18: **return** $\hat{S}$
19: $R \leftarrow \{u_1, u_2, \ldots u_{\min\{2*rlen,n\}}\}$

For complexities, $\widetilde{\mathbb{C}(u)}$ can be obtained with linear time and space complexities by using bounds proposed in Section V-A to execute the core decomposition-style [46]. Moreover, let the number of iterations be t. Since we use the exponential growth strategy, the number of vertices involved in these iterations are at most $(1/2)^{t-1} * n, (1/2)^{t-2} * n, \ldots, n$, which form a geometric sequence. In the i-th iteration, Algorithm 5 takes $O(\sum_{j=1}^{(1/2)^{t-i}*n} cost(u_j))$ time and $O(m)$ space by Theorem 5. By summing the time cost of all iterations, Algorithm 5 takes $O(2 * \sum_{i=1}^{n} cost(u_i)) = O(\mathcal{T}(\mathbb{M}))$ time. The space complexity is $O(m)$ due to the iteration being sequentially executed. □

Discussion of PSMC+ *and* PSMC-*C:* (1) The sparser the graph (i.e., the smaller the δ), the faster *PSMC+* and *PSMC-C* perform. This is due to the fact that they requires less time to estimate the motif resident and localize the target result when δ is small. (2) For small and sparse datasets (e.g., those with less than millions of edges and $\delta < 51$, as stated in our empirical results), *PSMC+* outperforms *PSMC-C* in terms of running time. Conversely, in large and dense datasets, *PSMC-C* is faster than *PSMC+*. This is because *PSMC-C*, a core-based algorithm, can efficiently decompose such graphs into smaller, sparse subgraphs and then find the results within these subgraphs. Besides, *PSMC+* requires a considerable time to execute the inclusion-exclusion-based Turan estimator and the colorful wedge estimator for estimating the motif conductance.



(a) Amazon (b) DBLP (c) Youtube (d) LiveJ (e) Orkut (f) Friendster

Fig. 5. Runtime (seconds) of different motif conductance algorithms with varying $k(\mathbb{M})$.

TABLE II
DATASET STATISTICS

Dataset	$\|V\|$	$\|E\|$	δ	Description
Amazon	334,863	925,872	6	Co-purchase
DBLP	317,080	1,049,866	113	Collaboration
Youtube	1,134,890	2,987,624	51	Social network
LiveJ	3,997,962	34,681,189	360	Social network
Orkut	3,072,441	117,185,083	253	Social network
Friendster	65,608,336	1,806,067,135	304	Social network
LFR [47]	$10^3 \sim 10^7$	$10^3 \sim 10^7$	$3 \sim 5$	Synthetic network
PLC [48]	$10^3 \sim 10^7$	$10^3 \sim 10^7$	$3 \sim 5$	Synthetic network
ER [49]	$10^3 \sim 10^7$	$10^3 \sim 10^7$	$5 \sim 11$	Synthetic network
BA [50]	$10^3 \sim 10^7$	$10^3 \sim 10^7$	$3 \sim 5$	Synthetic network

δ is the degeneracy.

VI. Experimental Evaluation

A. Experimental Setup

Datasets: Our solutions are evaluated on six real-world graphs with ground-truth clusters (all datasets can be downloaded from http://snap.stanford.edu/), which are widely used benchmarks for higher-order graph clustering [19], [51], [52]. Besides, we also use four types of synthetic graphs to test the scalability and the effectiveness of our solutions: *LFR* [47], *PLC* [48], *ER* [49], and *BA* [50], which can be generated by the well-known NetworkX Python package [53]. Note that they can be used to simulate the degree distributions, clusters, and small-world properties in the real world.

Competitors: PSMC, *PSMC+*, and *PSMC-C* are our proposed Algorithms 3, 4, and 5, respectively. Besides, we also describe the following ten competitors evaluated in our experiments. (1) Traditional graph clustering: *SC* [34], *Louvain* [54], *KCore* [46], *PCon_core* [33], and *PCon_de* [33]; (2) Cohesive subgraph based higher-order graph clustering: *HD* [51], [55], [56]; (3) Modularity based higher-order graph clustering: *HM* [20], [52]; (4) Motif conductance based higher-order graph clustering: *HSC* [19], *MAPPR* [29], and *HOSPLOC* [26], [27].

Parameters and Implementations: Unless specified otherwise, we take the default parameters of these competitors in our experiments. Since both *HOSPLOC* and *MAPPR* take a seed vertex as input, to be more reliable, we randomly select 50 vertices as seed vertices and report the average runtime and quality. Following previous work [21], [29], [41], we also limit ourselves to the representative k-clique motif to illustrate the main patterns observed. All experiments are conducted on a Linux machine with an Intel Xeon(R) Silver 4210 @2.20 GHz CPU and 1 TB RAM.

B. Efficiency Testing

Since the objective functions of traditional graph clustering (e.g., *SC* [34], *Louvain* [54], *KCore* [46], *PCon_core* [33], and *PCon_de* [33]), cohesive subgraph based higher-order graph clustering (e.g., *HD* [51], [55], [56]), and modularity based higher-order graph clustering (e.g., *HM* [20], [52]) are different from the motif conductance studied in this work, it is meaningless and unnecessary to compare their efficiency.

Exp-1. Runtime of different motif conductance algorithms with varying $k(\mathbb{M})$: The runtime of *HSC*, *MAPPR*, *HOSPLOC*, *PSMC*, *PSMC+*, and *PSMC-C* with varying $k(\mathbb{M})$ on six real-world networks is detailed in Fig. 5. Note that we do not report the empirical results for LiveJ, Orkut, and Friendster on $k(\mathbb{M}) = 6$. This is because we cannot obtain the results of baselines (i.e., *MAPPR*, *HOSPLOC*, and *HSC*) within 7 days on LiveJ and Orkut, or within 60 days on Friendster. By Fig. 5, we have: (1) *PSMC+* and *PSMC-C* are consistently faster than other methods, and *PSMC-C* is better than *PSMC+* in most cases. This is because *PSMC+* proposes the inclusion-exclusion-based Turan estimator and colorful wedge estimator to estimate the motif resident with near-linear time (Table I). *PSMC-C* is a core-based optimization strategy to directly local the target result by only visiting several small subgraphs (Section V-B). (2) *PSMC* is better than baselines (i.e., *MAPPR*, *HOSPLOC*, and *HSC*) on five of the six networks. The efficiency of *PSMC* can be attributed to its novel computing framework (i.e., integrating the enumeration and partitioning in an iterative algorithm). However, *MAPPR*, *HOSPLOC*, and *HSC* depend on the weight graph $\mathcal{G}^{\mathbb{M}}$ obtained by enumerating the motif instances, which increases exponentially with the size of the motif (Table I). In particular, *PSMC* achieves the speedups of 3.2∼32 times over *HSC*. For example, on DBLP and $k(\mathbb{M}) = 3$, *PSMC* takes 9 seconds to obtain the result, while *HSC* takes 292 seconds. (3) The runtime of all methods increases with increasing $k(\mathbb{M})$ except for *HSC* on Amazon and Youtube. This is because when $k(\mathbb{M})$ increases, we need more time to count/estimate motif instances. However, for *HSC*, a possible explanation is that the weighted graph $\mathcal{G}^{\mathbb{M}}$ gets smaller as $k(\mathbb{M})$ increases, resulting in very little time spent in the spectral clustering stage of the two-stage reweighting method [19]. (4) The sparser the graph (i.e., the smaller the δ), the faster our algorithms (i.e., *PSMC*, *PSMC+*, *PSMC-C*). For example, our algorithms are faster on Youtube compared to DBLP, despite Youtube having more vertices and more edges (Table II). This is because Youtube has a smaller δ (Table II). These results give preliminary evidence that the proposed solutions are indeed high efficiency.

Exp-2. Scalability testing on synthetic graphs: Extensive synthetic graphs are generated to further test the scalability of our solutions. Fig. 6 only presents the results when the given motif

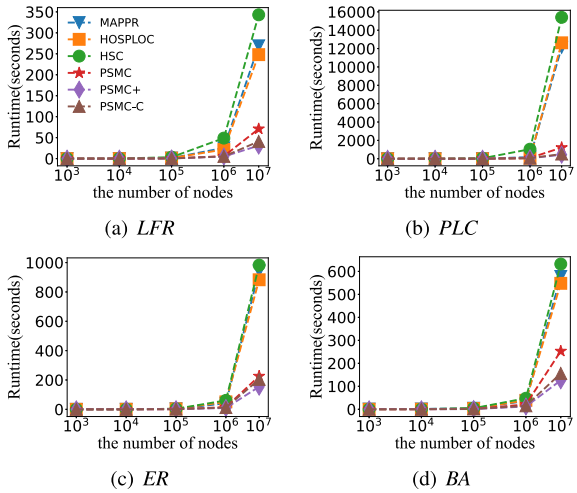


Fig. 6. Scalability testing on synthetic graphs.

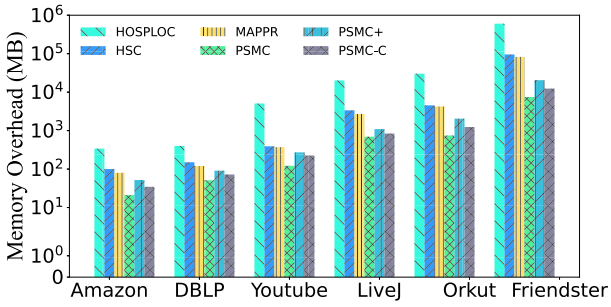


Fig. 7. Memory overhead comparison (excluding the size of the graph itself).

is a triangle, with comparable trends across other motifs. By Fig. 6, we know that our proposed algorithms (i.e., *PSMC*, *PSMC+*, and *PSMC-C*) scale near-linear with respect to the graph size. However, the runtime of other baselines fluctuates greatly as the graph size increases. This is because their time complexity is nonlinear depending on n, or even is n^3 for *HSC* (Table I). These results indicate that our algorithms have excellent scalability over massive graphs while the baselines do not.

Exp-3. Memory overhead comparison: Fig. 7 displays the memory overhead of the evaluated algorithms when the motif is a triangle, with comparable trends across other motifs. As excepted, our proposed *PSMC*, *PSMC+*, and *PSMC-C* are consistently less than other baselines on all datasets and realize up to an order of magnitude memory reduction on most cases. This advantage can be attributed to their novel computing framework, which only needs to maintain some simple data structures, such as motif resident of each vertex (Algorithm 3). Note that *PSMC* is the champion and *PSMC-C* is the runner-up (*PSMC+* is slightly worse than *PSMC-C*). This is because *PSMC+* needs to maintain more intermediate variables to estimate motif resident (Section V-A). On the other hand, *HSC* and *MAPPR* exhibit comparable memory overheads, while *HOSPLOC* has the worst performance. This is because *HOSPLOC* requires storing the expensive state transition tensor (the worst space is $O(n^3)$) to calculate the vertex order required by the sweep procedure. Similarly, *HSC* and *MAPPR* need to store the edge-weighted graph $\mathcal{G}^{\mathbb{M}}$ to calculate this vertex ordering (Section III). These results affirm the memory efficiency of our algorithms.

C. Effectiveness Testing

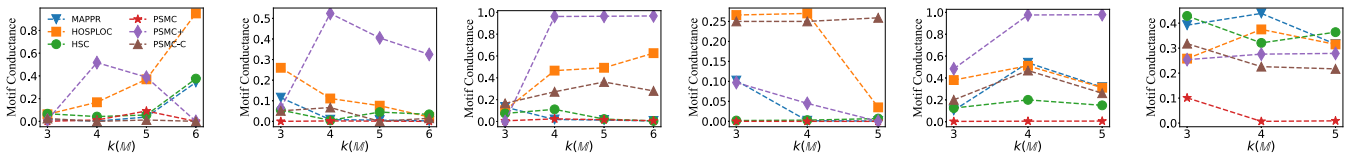
Effectiveness Metric: We use the F1-Score metric to measure how "close" each detected cluster C is to the ground-truth one. Note that since F1-Score is the harmonic mean of precision and recall, the larger the F1-Score, the better the quality of C [19], [51], [52]. Besides, we also use motif conductance (*MC* for short) calculated by Definition 2 to evaluate the quality of the identified cluster. The smaller the value of the *MC*(C), the better the partition of cluster C. We also report the size of the identified cluster for completeness. Note that we do not report traditional edge-based metrics (e.g., density, conductance) because they are used to measure the quality of clusters with edges as atomic clustering units (Section I).

Exp-4. Effectiveness of various graph clustering methods: Table III only reports these results when the given motif is a triangle, with analogous trends observed across other motifs. For motif conductance (*MC* for short) metric, we have: (1) *PSMC* outperforms other methods on five of the six datasets (on Youtube, *PSMC+* is the champion and *PSMC* is the runner-up). In particular, *PSMC* is 167, 12, 20, and 41 times better than *HSC* on Amazon, Youtube, LiveJ, and Orkut, respectively. This is because *PSMC* can find clusters with near-liner approximation ratio, while *HSC* has quadratic bound (Table I). Besides, *PSMC* is 255 (resp., 212), 4×10^{10} (resp., 1×10^{10}), 87 (resp., 52), 4×10^3 (resp., 2×10^3), and 267 (resp., 191) times better than *PCore_core* (resp., *PCore_de*) on Amazon, DBLP, Youtube, LiveJ, and Orkut, respectively. This is because *PCore_core* and *PCore_de* are lower-order graph clustering methods that ignore the significant higher-order motif structures. (2) *PSMC+* and *PSMC-C* outperform *MAPPR* and *HOSPLOC* on most cases. This is because *MAPPR* and *HOSPLOC* are heuristic and have no guarantee of clustering quality (Table I). However, *PSMC+* is built on top of *PSMC*, so even if *PSMC+* has no theoretical guarantee, it can still get good quality in practice. *PSMC-C* has strict theoretical guarantees as stated in Section V-B. Note that although *PSMC+* outperforms *PSMC-C* in four of the six datasets, *PSMC-C* has better performance when $k(\mathbb{M})$ increases (Fig. 8). For F1-Score metric, we have: (1) *PSMC* consistently outperforms other methods (including *HD* and *HM*). (2) *SC*, *Louvain*, *KCore*, *PCore_de*, and *PCore_core* have poor F1-Scores on most cases, and *PCore_de* consistently better than *PCore_core*. This is because they are traditional clustering methods that cannot capture higher-order structural information for graph clustering. For Size metric, on average, the cluster sizes found by different algorithms from largest to smallest are *SC*, *PSMC+*, *MAPPR*, *HD*, *PSMC*, *PCon_core*, *PCon_de*, *HM*, *Louvain*, *HOSPLOC*, *HSC*, *KCore*, and *PSMC-C*. Our algorithm *PSMC* returns the cluster size that is ranked in the middle, so it tends to find clusters of moderate size. However, other baselines either find the cluster that be too large or too small, leading to poor interpretability. So, these results give clear evidence that our solutions can indeed find higher-quality clusters when contrasted with baselines.

TABLE III
EFFECTIVENESS OF VARIOUS GRAPH CLUSTERING METHODS

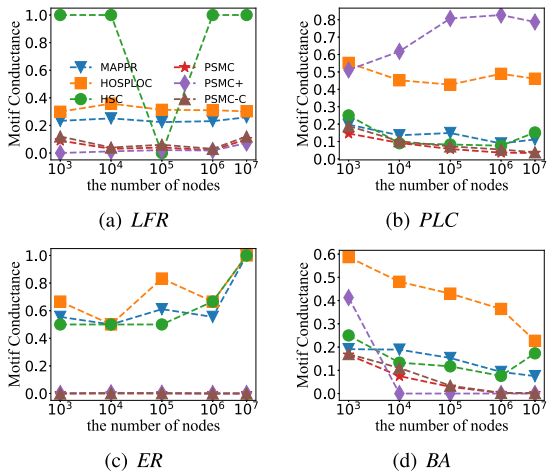
Model	Amazon			DBLP			Youtube			LiveJ			Orkut			Friendster		
	MC	F1-Score	Size	MC	F1-Score	Size	MC	F1-Score	Size	MC	F1-Score	Size	MC	F1-Score	Size	MC	F1-Score	Size
SC	0.704	0.226	80793	0.467	0.103	47891	0.773	0.061	38849	0.315	0.306	408010	0.499	0.020	476537	0.581	0.109	468085
Louvain	0.007	0.431	239	0.071	0.230	232	0.467	0.013	7480	0.071	0.277	2412	0.074	0.225	33	0.681	0.110	83470
KCore	0.109	0.138	497	0.018	0.273	114	0.470	0.095	845	0.035	0.313	377	0.141	0.165	15706	0.520	0.119	5043059
PCon_core	0.102	0.127	23014	0.339	0.248	309	0.523	0.101	40704	0.435	0.206	11513	0.799	0.061	514268	0.423	0.149	340754
PCon_de	0.085	0.371	17594	0.103	0.283	359	0.314	0.125	223	0.215	0.277	372	0.574	0.175	1402991	0.247	0.123	2340834
HD	0.269	0.182	30852	0.013	0.242	309	0.419	0.074	1239	0.011	0.125	7700	0.128	0.076	114187	0.398	0.159	2384002
HM	0.007	0.494	528	0.048	0.301	237	0.146	0.022	1999	0.016	0.120	674	0.082	0.248	96	0.136	0.168	3480
HSC	0.067	0.488	10	0.055	0.239	427	0.074	0.102	18	0.002	0.358	93	0.125	0.215	6	0.429	0.148	3457
MAPPR	0.015±0.01	0.457±0.11	175±19	0.115±0.08	0.339±0.18	28796±629	0.132±0.09	0.116±0.06	15810±4801	0.102±0.07	0.257±0.12	307740±49106	0.104±0.08	0.233±0.13	1343943±398103	0.392±0.18	0.165±0.11	32573797±3058731
HOSPLOC	0.062±0.03	0.467±0.21	90±12	0.260±0.17	0.283±0.14	414±98	0.103±0.02	0.128±0.09	434±137	0.266±0.11	0.342±0.16	3586±47	0.381±0.19	0.237±0.14	44651±346	0.258±0.16	0.187±0.09	3240830±4581
PSMC-C	0.025	0.417	7	0.051	0.342	20	0.166	0.113	5	0.25	0.302	3	0.2	0.197	3	0.318	0.198	32480
PSMC+	0.012	0.317	138098	0.064	0.353	87846	**0.000**	0.138	229147	0.097	0.326	88385	0.483	0.232	231334	0.254	0.201	3050234
PSMC	$\mathbf{4*10^{-4}}$	**0.511**	74991	$\mathbf{7*10^{-12}}$	**0.382**	141	0.006	**0.202**	21342	$\mathbf{1*10^{-4}}$	**0.413**	12458	**0.003**	**0.312**	1368793	**0.101**	**0.237**	3925021

The best and second-best results in each metric are marked in bold and underlined, respectively. Note that there is no clear evidence to suggest whether a larger or smaller cluster size is better. We report the cluster size to provide an intuitive motivation for our optimization algorithm psmc-c in section v-b. For HOSPLOC and MAPPER, each result is shown in average ± variance.



(a) Amazon (b) DBLP (c) Youtube (d) LiveJ (e) Orkut (f) Friendster

Fig. 8. Quality of various motif conductance algorithms with varying $k(\mathbb{M})$ on real-world graphs.



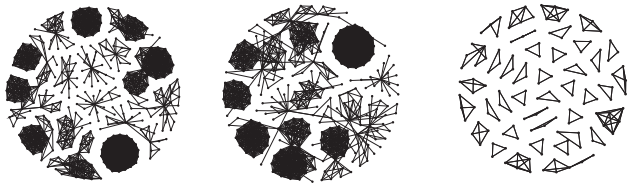
(a) *LFR* (b) *PLC*

(c) *ER* (d) *BA*

Fig. 9. Quality of various motif conductance algorithms on synthetic graphs.

Exp-5. Quality of various motif conductance algorithms: Fig. 8 depicts the quality of different motif conductance algorithms with varying $k(\mathbb{M})$ on real-world graphs. We have the following observations: (1) *PSMC* always outperforms other methods under different $k(\mathbb{M})$. Besides, *PSMC* is almost stable with increasing $k(\mathbb{M})$, while other methods have no obvious change trend with increasing $k(\mathbb{M})$. (2) *PSMC-C* is better than *PSMC+* on most cases. This is because *PSMC-C* is a core-based optimization algorithm with strict theoretical guarantees, while *PSMC+* is heuristic. (3) *PSMC+* and *HOSPLOC* fluctuates greatly as $k(\mathbb{M})$ increases. However, *PSMC+* performs well in synthetic graphs (see Fig. 9 for details). One possible explanation is that as $k(\mathbb{M})$ increases, the actual effect of estimation bounds in *PSMC+* depends on the type of network (e.g., real-world graphs have poor pruning effects, while synthetic graphs have good pruning effects). In particular, *PSMC+* and *HOSPLOC* have different trends of motif conductance in Fig. 8(a) and (e). This reason can be explained as follows: when k ranges from 3 to 6, the number of cliques for Amazon are 6×10^5, 2×10^5, 6×10^4, and 5×10^3, and the number of cliques for Orkut are 6×10^8, 3×10^9, 1×10^{10}, and 7×10^{10} [40], [57]. So, the number of cliques for Amazon is decreasing, while Orkut is increasing. Since *HOSPLOC* is based on random walks, so as the number of cliques increases, the probability of it escaping from within the cluster may decrease, thus motif conductance becomes smaller [26], [27]. The quality of *PSMC+* increases first and then decreases, indicating that the estimated bounds can achieve the best effect in the middle of k. Moreover, we also report these qualities on extensive synthetic graphs. As shown in Fig. 9, *PSMC* and *PSMC-C* have comparable performances and better than other baselines. Besides, *PSMC+* outperforms other baselines in most cases (an unusual situation occurred in *PLC* in Fig. 9(b)). However, the performance of *MAPPR*, *HOSPLOC*, and *HSC* vary significantly depending on the dataset. For example, *MAPPR* $<$ *HOSPLOC* $<$ *HSC* on *LFR* synthetic graphs, but *HSC* $<$ *MAPPR* $<$ *HOSPLOC* on *BA* synthetic graphs, where $A > B$ means A has larger motif conductance. Note that *HSC* has an obvious drop trend in Fig. 9(a). This reason can be explained as follows: Since the synthetic network *LFR* can simulate the cluster structures, when the number of nodes is small (e.g., $n < 10^5$), the cluster structure is not very obvious, resulting in a poor motif conductance value. However, as the number of nodes increases (e.g., $n = 10^5$), the cluster structure becomes obvious, resulting in a better motif conductance. When the number of nodes exceeds a certain value (e.g., $n > 10^5$), the overlap between clusters becomes high, making it different to detect high-quality clusters. These results indicate that our algorithms can identify higher-quality clusters than the baselines on real-world&synthetic graphs.

Exp-6. Case Studies on DBLP: Although our proposed clustering framework has many similarities with [33], there are still also pivotal differences that have been discussed in Section IV-C. To further illustrate, we perform some case studies to compare our method with [33]. Note that the effectiveness of other graph clustering methods (including lower-order/higher-order graph clustering) have been well validated by extensive existing case



(a) *PCon_de* (b) *PSMC* ($k(\mathbb{M}) = 2$) (c) *PSMC* ($k(\mathbb{M}) = 3$)

Fig. 10. Case studies on DBLP.

studies [19], [26], [27], [28], [29], we ignore them for brevity. Specifically, we compare the performance of *PCon_de* [33] and our *PSMC* on the well-known and widely-used DBLP dataset, where vertices represent authors and edges represent two authors who jointly published at least one paper. In this instance, *PCon_de* identifies a cluster comprising 359 authors closely connected each other (Fig. 10(a)). When $k(\mathbb{M}) = 2$, in Fig. 10(b), our *PSMC* also obtains a comparable size with 309 authors and elaborates similar structure with *PCon_de*. Through in-depth analysis of Fig. 10(a) and (b), we find that they present more star-shaped substructures, indicating that the cooperative relationship between them is mainly dominated by a certain person. However, our method *PSMC* for $k(\mathbb{M}) = 3$ (Fig. 10(c)) can obtain more triangular substructures, which are stable scientific partnerships. On top of that, Fig. 10(a) and (b) acquire clusters that are too large, leading to poor interpretability. This result exemplifies the potential of *PSMC* for effective higher-order graph clustering and its applications.

VII. Related Work

Traditional Graph Clustering: Graph clustering has received much attention over past decades [10], [58], [59]. Modularity [7], [60], [61] and conductance [8], [22], [23], [33] are two representative models to evaluate the clustering quality of the identified cluster. Informally, they aim to optimize the difference or ratio of edges between the internal and external of the cluster. However, finding the cluster with optimal modularity or conductance is NP-hard [7], [23]. Thus, many heuristic or approximate algorithms have been proposed in the literature. For example, the heuristic algorithm *Louvain* was proposed to iteratively optimize modularity in a greedy manner [54], [62]. *Fiedler* vector-based spectral clustering algorithm can output a cluster with a quadratic factor of optimal conductance [33]. Recently, some polynomial solvable cohesive subgraph models also have been proposed to partition the graph, which are to only optimize the internal denseness of the identified cluster [11], [12], [63], [64]. Notable examples include average-degree densest subgraph, k-core, and k-truss [9]. But these traditional methods mainly focus on the internal or external *lower-order edges* of the cluster, resulting in that cannot capture higher-order structural information for graph clustering. Besides simple graphs, more complex graphs has also been explored. For example, the graph clustering on attribute graphs [65], [66], heterogeneous information networks [67], [68], and temporal networks [69], [70], [71], [72], [73]. These methods are orthogonal to our work.

Higher-order Graph Clustering: In addition to the motif conductance studied in this paper [19], other higher-order graph clustering models also have been proposed in the literature. For example, motif modularity was proposed to extend the traditional modularity by optimizing the difference between the fraction of motif instances within the cluster and the fraction in a random network preserving the same degree of vertices [20], [52]. Higher-order densest subgraph model was proposed where the density is defined as the number of motif instances divided by the size of vertices [51], [55]. Li et al. proposed an edge enhancement approach to overcome the hypergraph fragmentation issue appearing in the seminal reweighting framework [74]. Unfortunately, they are still essentially optimizing the objective function for traditional lower-order clustering. Besides simple graphs, higher-order graph clustering on more complicated networks also have been studied, such as heterogeneous information networks [75], labeled networks [41], [76], multi-layer networks [77], dynamic networks [78]. Clearly, these methods on complicated networks are orthogonal to our work.

VIII. Conclusion

We first devise a *simple* but *provable* peeling-based higher-order graph clustering framework *PSMC* for motif conductance. Most notably, *PSMC* can output the result with *fixed* and *motif-independent* approximation ratio, which solves the open question posed by the seminal two-stage reweighting framework. We then devise novel dynamic update techniques and optimization strategies to further boost the efficiency of *PSMC*. Finally, empirical results on real-life and synthetic datasets demonstrate our solutions' superiority over ten competitors on both clustering accuracy and running time.

References

[1] M. Gupta, J. Gao, Y. Sun, and J. Han, "Integrating community matching and outlier detection for mining evolutionary community outliers," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 859–867.

[2] S. Liu, B. Hooi, and C. Faloutsos, "A contrast metric for fraud detection in rich graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2235–2248, Dec. 2019.

[3] B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos, "FRAUDAR: Bounding graph fraud in the face of camouflage," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 895–904.

[4] Y. Meng, R. Li, L. Lin, X. Li, and G. Wang, "Topology-preserving graph coarsening: An elementary collapse-based approach," *Proc. VLDB Endowment*, vol. 17, no. 13, pp. 4760–4772, 2024.

[5] W. Chiang, X. Liu, S. Si, Y. Li, S. Bengio, and C. Hsieh, "Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 257–266.

[6] Y. Yu, L. Lin, Q. Liu, Z. Wang, X. Ou, and T. Jia, "GSD-GNN: Generalizable and scalable algorithms for decoupled graph neural networks," in *Proc. Int. Conf. Multimedia Retrieval*, 2024, pp. 64–72.

[7] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E*, vol. 69, no. 2, 2004, Art. no. 026113.

[8] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[9] L. Chang and L. Qin, "Cohesive subgraph computation over large sparse graphs," in *Proc. IEEE Int. Conf. Data Eng.*, 2019, pp. 2068–2071.

[10] W. Feng, L. Wang, B. Hooi, S. Ng, and S. Liu, "Interrelated dense pattern detection in multilayer networks," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 11, pp. 6462–6476, Nov. 2024.

[11] W. Feng, S. Liu, D. Koutra, H. Shen, and X. Cheng, "SpecGreedy: Unified dense subgraph detection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases*, 2020, pp. 181–197.

[12] W. Feng, S. Liu, D. Koutra, and X. Cheng, "Unified dense subgraph detection: Fast spectral theory based algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 3, pp. 1356–1370, Mar. 2024.

[13] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, pp. 824–827, 2002.

[14] C. Klymko, D. F. Gleich, and T. G. Kolda, "Using triangles to improve community detection in directed networks," 2014, *arXiv:1404.5874*.

[15] K. Sotiropoulos and C. E. Tsourakakis, "Triangle-aware spectral sparsifiers and community detection," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 1501–1509.

[16] B. J. Kloostra, C. Dalvi, and B. N. Behm, "System and method for analyzing and dispositioning money laundering suspicious activity alerts," U.S. Patent App. 12/258,784, 2009.

[17] S. Mangan and U. Alon, "Structure and function of the feed-forward loop network motif," *Proc. Nat. Acad. Sci. USA*, vol. 100, no. 21, pp. 11980–11985, 2003.

[18] A. Duval and F. D. Malliaros, "Higher-order clustering and pooling for graph neural networks," in *Proc. ACM Int. Conf. Inf. Knowl. Manage.*, 2022, pp. 426–435.

[19] A. R. Benson, D. F. Gleich, and J. Leskovec, "Higher-order organization of complex networks," *Science*, vol. 353, no. 6295, pp. 163–166, 2016.

[20] A. Arenas, A. Fernandez, S. Fortunato, and S. Gomez, "Motif-based communities in complex networks," *J. Phys. A: Math. Theor.*, vol. 41, no. 22, 2008, Art. no. 224001.

[21] C. E. Tsourakakis, "The K-clique densest subgraph problem," in *Proc. Int. Conf. World Wide Web*, 2015, pp. 1122–1132.

[22] D. F. Gleich and C. Seshadhri, "Vertex neighborhoods, low conductance cuts, and good seeds for local community methods," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 597–605.

[23] S. Galhotra, A. Bagchi, S. Bedathur, M. Ramanath, and V. Jain, "Tracking the conductance of rapidly evolving topic-subgraphs," *Proc. VLDB Endowment*, vol. 8, no. 13, pp. 2170–2181, 2015.

[24] Y. Zhang and K. Rohe, "Understanding regularized spectral clustering via graph conductance," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 10654–10663.

[25] Y. He, L. Lin, P. Yuan, R. Li, T. Jia, and Z. Wang, "CCSS: Towards conductance-based community search with size constraints," *Expert Syst. Appl.*, vol. 250, 2024, Art. no. 123915.

[26] D. Zhou et al., "A local algorithm for structure-preserving graph cut," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 655–664.

[27] D. Zhou et al., "High-order structure exploration on massive graphs: A local graph clustering perspective," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 2, pp. 18:1–18:26, 2021.

[28] C. E. Tsourakakis, J. Pachocki, and M. Mitzenmacher, "Scalable motif-aware graph clustering," in *Proc. Int. Conf. World Wide Web*, 2017, pp. 1451–1460.

[29] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 555–564.

[30] S. Yu, Y. Feng, D. Zhang, H. D. Bedru, B. Xu, and F. Xia, "Motif discovery in networks: A survey," *Comput. Sci. Rev.*, vol. 37, 2020, Art. no. 100267.

[31] P. Ribeiro, P. Paredes, M. E. P. Silva, D. Aparício, and F. M. A. Silva, "A survey on subgraph counting: Concepts, algorithms, and applications to network motifs and graphlets," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 28:1–28:36, 2022.

[32] S. Huang, Y. Li, Z. Bao, and Z. Li, "Towards efficient motif-based graph partitioning: An adaptive sampling approach," in *Proc. IEEE Int. Conf. Data Eng.*, 2021, pp. 528–539.

[33] L. Lin, R. Li, and T. Jia, "Scalable and effective conductance-based graph clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 4471–4478.

[34] N. Alon and V. D. Milman, ", isoperimetric inequalities for graphs, and superconcentrators," *J. Combinatorial Theory, Ser. B*, vol. 38, no. 1, pp. 73–88, 1985.

[35] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Math. J.*, vol. 23, no. 2, pp. 298–305, 1973.

[36] L. Wen and M. K. Ng, "On the limiting probability distribution of a transition probability tensor," *Linear Multilinear Algebra*, vol. 62, no. 3, pp. 362–385, 2014.

[37] S. Wang, R. Yang, X. Xiao, Z. Wei, and Y. Yang, "FORA: Simple and effective approximate single-source personalized PageRank," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 505–514.

[38] Z. Wei, X. He, X. Xiao, S. Wang, S. Shang, and J. Wen, "TopPPR: Top-k personalized PageRank queries with precision guarantees on large graphs," in *Proc. 2018 Int. Conf. Manage. Data*, 2018, pp. 441–456.

[39] R. Andersen, F. R. K. Chung, and K. J. Lang, "Local graph partitioning using PageRank vectors," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci.*, 2006, pp. 475–486.

[40] R. Li, S. Gao, L. Qin, G. Wang, W. Yang, and J. X. Yu, "Ordering heuristics for k-clique listing," *Proc. VLDB Endowment*, vol. 13, no. 11, pp. 2536–2548, 2020.

[41] D. Fu, D. Zhou, R. Maciejewski, A. Croitoru, M. Boyd, and J. He, "Fairness-aware clique-preserving spectral clustering of temporal graphs," in *Proc. Int. Conf. World Wide Web*, 2023, pp. 3755–3765.

[42] P. Turan, "On an extremal problem in graph theory," *Mat. Fiz. Lapok*, vol. 48, no. 137, pp. 436–452, 1941.

[43] P. Erdos, "On the number of complete subgraphs and circuits contained in graphs," *Časopis Pěstování Matematiky*, vol. 94, pp. 290–296, 1969.

[44] A. Björklund, T. Husfeldt, and M. Koivisto, "Set partitioning via inclusion-exclusion," *SIAM J. Comput.*, vol. 39, no. 2, pp. 546–563, 2009.

[45] S. Gao, R. Li, H. Qin, H. Chen, Y. Yuan, and G. Wang, "Colorful h-star core decomposition," in *Proc. IEEE Int. Conf. Data Eng.*, 2022, pp. 2588–2601.

[46] S. B. Seidman, "Network structure and minimum degree," *Social Netw.*, vol. 5, no. 3, pp. 269–287, 1983.

[47] A. Lancichinetti, S. Fortunato, and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks," *New J. Phys.*, vol. 11, no. 3, 2009, Art. no. 033015.

[48] P. Holme and B. J. Kim, "Growing scale-free networks with tunable clustering," *Phys. Rev. E*, vol. 65, no. 2, 2002, Art. no. 026107.

[49] P. Erdos et al., "On the evolution of random graphs," *Pub. Math. Inst. Hung. Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.

[50] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[51] Y. Fang, K. Yu, R. Cheng, L. V. S. Lakshmanan, and X. Lin, "Efficient algorithms for densest subgraph discovery," *Proc. VLDB Endowment*, vol. 12, no. 11, pp. 1719–1732, 2019.

[52] L. Huang, H. Chao, and G. Xie, "MuMod: A micro-unit connection approach for hybrid-order community detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 107–114.

[53] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkX," in *Proc. 7th Python Sci. Conf.*, 2008, pp. 11–15.

[54] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mechanics: Theory Experiment*, vol. 2008, no. 10, 2008, Art. no. P10008.

[55] B. Sun, M. Danisch, T. H. Chan, and M. Sozio, "KClist++: A simple algorithm for finding k-clique densest subgraphs in large graphs," *Proc. VLDB Endowment*, vol. 13, no. 10, pp. 1628–1640, 2020.

[56] Y. He, K. Wang, W. Zhang, X. Lin, and Y. Zhang, "Scaling up k-clique densest subgraph detection," *Proc. ACM Manage. Data*, vol. 1, no. 1, pp. 69:1–69:26, 2023.

[57] M. Danisch, O. Balalau, and M. Sozio, "Listing k-cliques in sparse real-world graphs," in *Proc. Int. Conf. World Wide Web*, 2018, pp. 589–598.

[58] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, no. 3–5, pp. 75–174, 2010.

[59] S. Fortunato and D. Hric, "Community detection in networks: A user guide," 2016, *arXiv:1608.00163*.

[60] J. Kim, S. Luo, G. Cong, and W. Yu, "DMCS : Density modularity based community search," in *Proc. 2022 Int. Conf. Manage. Data*, 2022, pp. 889–903.

[61] H. Shiokawa, Y. Fujiwara, and M. Onizuka, "Fast algorithm for modularity-based graph clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2013, pp. 1170–1176.

[62] V. D. Blondel, J. Guillaume, and R. Lambiotte, "Fast unfolding of communities in large networks: 15 years later," 2023, *arXiv:2311.06047*.

[63] M. Charikar, "Greedy approximation algorithms for finding dense components in a graph," in *Proc. Int. Workshop Approximation Algorithms Combinatorial Optim.*, Springer, 2000, pp. 84–95.

[64] C. Tsourakakis, F. Bonchi, A. Gionis, F. Gullo, and M. Tsiarli, "Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 104–112.

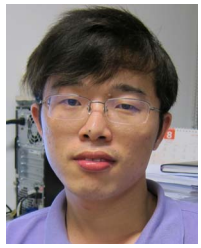
[65] R. Yang, J. Shi, Y. Yang, K. Huang, S. Zhang, and X. Xiao, "Effective and scalable clustering on massive attributed graphs," in *Proc. Int. Conf. World Wide Web*, 2021, pp. 3675–3687.

[66] C. Zhe, A. Sun, and X. Xiao, "Community detection on large complex attribute network," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 2041–2049.

[67] L. Chen, Y. Gao, Y. Zhang, C. S. Jensen, and B. Zheng, "Efficient and incremental clustering algorithms on star-schema heterogeneous graphs," in *Proc. IEEE Int. Conf. Data Eng.*, 2019, pp. 256–267.
[68] X. Jian, Y. Wang, and L. Chen, "Effective and efficient relational community detection and search in large dynamic heterogeneous information networks," *Proc. VLDB Endowment*, vol. 13, no. 10, pp. 1723–1736, 2020.
[69] C. Zhu, L. Lin, P. Yuan, and H. Jin, "Discovering cohesive temporal subgraphs with temporal density aware exploration," *J. Comput. Sci. Technol.*, vol. 37, pp. 1068–1085, 2022.
[70] Y. Zhang, L. Lin, P. Yuan, and H. Jin, "Significant engagement community search on temporal networks," in *Proc. Int. Conf. Database Syst. Adv. Appl.*, 2022, pp. 250–258.
[71] L. Lin et al., "QTCS: Efficient query-centered temporal community search," *Proc. VLDB Endowment*, vol. 17, no. 6, pp. 1187–1199, 2024.
[72] L. Lin, P. Yuan, R. Li, and H. Jin, "Mining diversified top-r lasting cohesive subgraphs on temporal networks," *IEEE Trans. Big Data*, vol. 8, no. 6, pp. 1537–1549, Dec. 2022.
[73] L. Lin, P. Yuan, R. Li, J. Wang, L. Liu, and H. Jin, "Mining stable quasi-cliques on temporal networks," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 6, pp. 3731–3745, Jun. 2022.
[74] P. Li, L. Huang, C. Wang, and J. Lai, "EdMot: An edge enhancement approach for motif-aware community detection," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 479–487.
[75] A. G. Carranza, R. A. Rossi, A. Rao, and E. Koh, "Higher-order clustering in complex heterogeneous networks," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 25–35.
[76] A. E. Sariyüce, "Motif-driven dense subgraph discovery in directed and labeled networks," in *Proc. Int. Conf. World Wide Web*, 2021, pp. 379–390.
[77] L. Huang, C. Wang, and H. Chao, "HM-modularity: A harmonic motif modularity approach for multi-layer network community detection," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2520–2533, Jun. 2021.
[78] D. Fu, D. Zhou, and J. He, "Local motif clustering on time-evolving graphs," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 390–400.

Longlong Lin received the PhD degree from the Huazhong University of Science and Technology (HUST) in 2022. He is currently an associate professor with the College of Computer and Information Science, Southwest University, Chongqing, China. His current research interests include (temporal) graph clustering and graph-based machine learning.

Zeli Wang received the PhD degree from the Huazhong University of Science and Technology (HUST) in 2022. She is now a lecturer with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China. Her current research interests include blockchain smart contract security and natural language processing security.



Rong-Hua Li received the PhD degree from the Chinese University of Hong Kong in 2013. He is currently a professor with the Beijing Institute of Technology (BIT), Beijing, China. His research interests include graph data management and mining, social network analysis, graph computation systems, and graph-based machine learning.

Qiyu Liu received the PhD degree in computer science from the Hong Kong University of Science and Technology in 2022. He was a postdoctoral fellow from 2022 to 2023 with HKUST and a research scientist from 2023 to 2024 with the Shenzhen Institute of Computing Science. Since 2024, he has been a full professor with the Southwest University.

Hongchao Qin received the BS degree in mathematics and the ME and PhD degrees in computer science from Northeastern University, China, in 2013, 2015, and 2020, respectively. He is currently an assistant professor with the Beijing Institute of Technology, China. His current research interests include social network analysis and data-driven graph mining.

Jin Zhao received the PhD degree from the Huazhong University of Science and Technology (HUST) in 2022. He is now working toward the postdoctoral fellow with Zhejiang Lab, in China. His current research interests include graph processing, system software, and architecture.