PSMC: Provable and Scalable Algorithms for Motif Conductance Based Graph Clustering

Longlong Lin
College of Computer and Information Science, Southwest University
longlonglin@swu.edu.cn

Tao Jia
College of Computer and Information Science, Southwest University
tjia@swu.edu.cn

Zeli Wang
Chongqing University of Posts and Telecommunications
zlwang@cqupt.edu.cn

Jin Zhao
School of Computer Science and Technology, HuaZhong University of Science and Technology
zjin@hust.edu.cn

Rong-Hua Li
Shenzhen Institute of Technology
Beijing Institute of Technology
lironghuabit@126.com

ABSTRACT

Higher-order graph clustering aims to partition the graph using frequently occurring subgraphs (i.e., motifs), instead of the lower-order edges, as the atomic clustering unit, which has been recognized as the state-of-the-art solution in ground truth community detection and knowledge discovery. *Motif conductance* is one of the most promising higher-order graph clustering models due to its strong interpretability. However, existing motif conductance based graph clustering algorithms are mainly limited by a seminal two-stage reweighting computing framework, needing to enumerate all motif instances to obtain an edge-weighted graph for partitioning. However, such a framework has two-fold vital defects: (1) It can only provide a quadratic bound for the motif with three vertices, and whether there is provable clustering quality for other motifs is still an open question. (2) The enumeration procedure of motif instances incurs prohibitively high costs against large motifs or large dense graphs due to combinatorial explosions. Besides, expensive spectral clustering or local graph diffusion on the edge-weighted graph also makes existing methods unable to handle massive graphs with millions of nodes. To overcome these dilemmas, we propose a **P**rovable and **S**calable **M**otif **C**onductance algorithm *PSMC*, which has a *fixed* and *motif-independent* approximation ratio for any motif. Specifically, *PSMC* first defines a new vertex metric *Motif Resident* based on the given motif, which can be computed locally. Then, it iteratively deletes the vertex with the smallest motif resident value very efficiently using novel dynamic update technologies. Finally, it outputs the locally optimal result during the above iterative process. To further boost efficiency, we propose several effective bounds to estimate the motif resident value of each vertex, which can greatly reduce computational costs. Empirical results on real-life and synthetic demonstrate that our proposed algorithms achieve 3.2∼32 times speedup and improve the quality by at least 12 times than the state-of-the art baselines.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671666

CCS CONCEPTS

• **Mathematics of computing → Graph algorithms**.

KEYWORDS

Higher-order Graph Clustering; Motif Conductance

ACM Reference Format:
Longlong Lin, Tao Jia, Zeli Wang, Jin Zhao, and Rong-Hua Li. 2024. PSMC: Provable and Scalable Algorithms for Motif Conductance Based Graph Clustering. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3637528.3671666

1 INTRODUCTION

Graph clustering is a fundamental problem in machine learning and enjoys numerous applications, including image segmentation [50], anomaly detection [24], parallel computing [12], and graph representation learning [11, 32, 66, 68]. Therefore, many traditional graph clustering models have been proposed in the literature, such as null model based (e.g., modularity [46]), edge cut based (e.g., ratio cut or normalized cut [58]), and subgraph cohesiveness based (e.g., k-core or k-truss [9]). Informally, these models partition all vertices into several clusters, satisfying the vertices within the same cluster have more edges than the vertices in different clusters [54].

Nevertheless, these traditional graph clustering models ignore the significant motif connectivity patterns (i.e., small frequently occurring subgraphs), which are regarded as indispensable for modeling and understanding the higher-order organization of complex networks [45, 64]. Unlike dyadic edges, each motif (involves more than two nodes) indicates the unique interaction behavior among vertices and represents some particular functions. To name a few, the triangle is the stable relationship cornerstone of social networks [36, 52]. Cycles hint at some money laundering events in financial markets [35]. Feed-forward loops are basic transcription units in genetic networks [44]. As a consequence, adopting such mesoscopic level motif as the atomic clustering unit has been recognized as the state-of-the-art (SOTA) solution in ground truth community detection and knowledge discovery [13, 52, 62]. Such clustering methods are typically named higher-order graph clustering, which aims at capturing the higher-order community structures with dense motifs instead of edges [4]. This paper focuses on higher-order graph clustering on massive graphs with millions of nodes. Thus, perfect solutions must be highly scalable and utility.

Numerous higher-order graph clustering models have been proposed in the literature [2, 4, 55] (Section 5). Perhaps, the most representative and effective model is the *motif conductance* due to its strong interpretability and solid theoretical foundation [4] (Section 2.1). To be specific, motif conductance is the variant of conductance (conductance is an edge-based clustering model [21, 23, 70]), which indicates the ratio of the number of motif instances going out the cluster to the number of motif instances within the cluster. As a result, smaller motif conductance implies better higher-order clustering quality [4, 56, 65, 72, 73]. However, identifying the result with the smallest motif conductance raises significant challenges due to its NP-hardness [4]. Therefore, many approximate or heuristic algorithms have been proposed to either improve the clustering quality or reduce the computational costs. For example, the *Science* paper [4] proposed a seminal two-stage reweighting framework. In the first stage, the input graph G is transformed into an edge-weighted graph $\mathcal{G}^{\mathbb{M}}$, in which the weight of each edge e is the number of motif instances e participates in. In the second stage, the traditional spectral clustering is used to partition $\mathcal{G}^{\mathbb{M}}$. However, such a seminal framework can only obtain provable clustering qualities for the motif consisting of three vertices [4]. Thus, whether the motif with four or more vertices (such motifs are more realistic [47, 67]) has a provable clustering quality is still an open question. On top of that, the framework has to enumerate all motif instances in advance and computes the eigenvector of normalized Laplacian matrix of $\mathcal{G}^{\mathbb{M}}$, resulting in prohibitively high time and space costs (Section 2.2). To improve the efficiency, some local graph diffusion algorithms are proposed to replace the eigenvector calculation with various random walk distributions (e.g., Personalized PageRank and higher-order markov chain) (Section 2.2). However, these algorithms are heuristic, and their clustering qualities are heavily dependent on many hard-to-tune parameters and seeding strategies. So, their performance is unstable and in most cases very poor, as demonstrated in our experiments. Recently, Huang et al. [31] pointed out that almost all existing solutions are limited by the above two-stage reweighting framework, and then they proposed an adaptive sampling method to estimate the weights of the edges for reducing the 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 limitations, we propose a **P**rovable and **S**calable **M**otif **C**onductance algorithm *PSMC*. Since the purpose of optimizing motif conductance is to obtain target clusters rather than to obtain the intermediate edge-weighted graph $\mathcal{G}^{\mathbb{M}}$, it is not necessary to blindly spend much time on getting precise $\mathcal{G}^{\mathbb{M}}$. Instead, we deeply analyze the functional form of motif conductance (Lemma 3) and iteratively optimize motif conductance 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 during the above iterative process. As a consequence, *PSMC* is to integrate the computation and partition of edge-weighted graph $\mathcal{G}^{\mathbb{M}}$ in an iterative algorithm, thus eliminating the need for expensive spectral clustering or local graph diffusion. Besides, we also theoretically prove that *PSMC* has a *fixed* and *motif-independent* approximation ratio (Theorem 3). In other words, *PSMC* can output a fixed approximation ratio for any given motif, which solves the open question posed by the previous two-stage reweighting framework. Particularly, when the given motif has three vertices, *PSMC* improves the well-known quadratic bound (Table 1). On the other hand, the motif resident of vertex u implicitly depends on the number of motif instances u participates in, causing *PSMC* also indirectly calculates all motif instances. To further boost efficiency, we develop several effective bounds to estimate the motif resident of each vertex via the well-known Turan Theorem [57] and colorful h-star degree [22]. We highlight our main contributions as follows.

A Novel Computing Framework with Accuracy Guarantee. We introduce a provable and scalable motif conductance algorithm, called *PSMC*, based on the proposed vertex metric *Motif Resident*. *PSMC* has two striking features. One is that it is a novel high-order graph clustering framework by integrating the computation and partition of edge-weighted graph in an iterative algorithm, reducing the computational costs. The other is that it can output a fixed and motif-independent approximation ratio for any given motif, while the existing SOTA frameworks cannot.

Several Effective Optimization Strategies. To further boost efficiency, we develop several dynamic update technologies to incrementally maintain the motif resident of each vertex when its neighbor is deleted, without recomputing the motif resident from scratch. Besides, with the help of Turan Theorem and colorful h-star degree, several effective bound estimation strategies are proposed to obtain a better trade-off between efficiency and accuracy.

Extensive Experiments. We conduct extensive experiments on nine datasets (five real-world graphs and four synthetic graphs) and eight competitors to evaluate the scalability and effectiveness of our proposed solutions. These empirical results show that our algorithms achieve 3.2∼32 times speedup and improve the quality by at least 12 times than baselines. Besides, our algorithms realize up to an order of magnitude memory reduction when contrasted with baselines.

2 PRELIMINARIES

2.1 Problem Formulation

Given an unweighted and undirected graph $G(V, E)$[1], 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 use-initiated query motif, which is a frequently occurring interaction pattern (i.e., significant subgraph) in complex networks. For simplicity, $G_S \in \mathbb{M}$ means that G_S is an instance of $\mathbb{M}$. Namely, $G_S \in \mathbb{M}$ iff G_S is isomorphic to $\mathbb{M}$[2]. We let $k(\mathbb{M})$ be the order of $\mathbb{M}$, which is the number of vertices involved in $\mathbb{M}$. For example, an edge is a second-order motif and a triangle is a three-order motif. Following existing research work [20, 55, 65], unless otherwise

[1] For simplicity, we consider the unweighted and undirected graph in this work, while our proposal can be easily extended to the weighted or directed graphs.
[2] $G(V_1, E_1)$ and $G(V_2, E_2)$ are isomorphic if there exists a bijection $f: V_1 \to V_2$ such that $(u, v) \in E_2$ iff $(f(u), f(v)) \in E_2$.

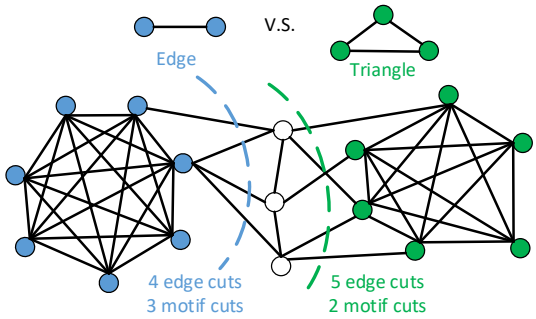


Figure 1: Illustration of the traditional edge-based conductance and the motif conductance on a synthetic graph. There are 47 edges and 60 triangles. The blue dotted line indicates the optimal cut when the motif is an edge and the corresponding conductance is $\frac{4}{\min\{42,52\}}$. The green dotted 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.

stated, we also take clique as the motif by default in this paper. 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 $G_{V\setminus C}$.*

Based on the intuition of Definition 1, we use the most representative and effective *motif conductance* [4, 56, 65, 72, 73] to measure the clustering quality of an identified cluster C.

Definition 2 (motif conductance). *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 instance 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 [21, 23, 40]. 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 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 [4]. Figure 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.

2.2 Existing Solutions and Their Shortcomings

In this subsection, we review several SOTA motif conductance algorithms, which can be roughly divided into two categories: seed-free global clustering and seed-dependent local clustering.

2.2.1 Seed-free global clustering. Seed-free global clustering identifies the higher-order clusters by calculating the eigenvector of the normalized Laplacian matrix of the edge-weighted graph $\mathcal{G}^{\mathbb{M}}$, in which the weight of each edge $e \in \mathcal{G}^{\mathbb{M}}$ is the number of motif instances e participates in. For example, Benson et al. [4] proposed the following two-stage higher-order spectral clustering (*HSC*). Specifically, *HSC* first obtains the normalized Laplacian matrix $\mathcal{L}$ by enumerating motif instances. Then, *HSC* computes the eigenvector x of the second smallest eigenvalue of $\mathcal{L}$ to execute the sweep procedure. Namely, it sorts all entries in x such that $x_1 \leq x_2 \leq \ldots \leq x_n$ and outputs $S = \arg\min \phi(S_i)$, in which $S_i = \{x_1, x_2, \ldots, x_i\}$. The following theorems are important theoretical basis of *HSC*.

Theorem 1 ([4]). *Given a graph $G(V,E)$ and a motif $\mathbb{M}$, for any $S \subseteq V$, we have*

$$\phi_{\mathbb{M}}(S) = \begin{cases} \phi^{\mathcal{G}^{\mathbb{M}}}(S), & if\ k(\mathbb{M}) = 3 \\ \phi^{\mathcal{G}^{\mathbb{M}}}(S) - \frac{\sum_{mi \in \mathbb{M}} I(|mi \cap S| = 2)}{\sum_{u \in S} \mathcal{D}_{uu}}, & if\ k(\mathbb{M}) = 4 \end{cases} \quad (3)$$

Where $\phi^{\mathcal{G}^{\mathbb{M}}}(S)$ is the edge-based conductance of S in terms of the weighted graph $\mathcal{G}^{\mathbb{M}}$ and $I(.)$ is the indicator function. Note that when $k(\mathbb{M}) > 4$, the relationship between $\phi_{\mathbb{M}}(S)$ and $\phi^{\mathcal{G}^{\mathbb{M}}}(S)$ is unclear.

Theorem 2 (Cheeger inequality [4]). *Given a graph $G(V,E)$ and a motif $\mathbb{M}$ with $k(\mathbb{M}) = 3$, let S be the vertex subset returned by HSC, 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.*

Discussions. Note that *HSC* can only derive the Cheeger inequality for motifs consisting of three vertices. However, it has not been proven whether there is quality guarantee for motifs with four or more vertices. On top of that, since *HSC* needs to enumerate motif instances in advance and calculate the eigenvector of $\mathcal{L}$, its time complexity is $O(k(\frac{\delta}{2})^{k-2}m) + O(n^3)$ and space complexity is $O(n^2)$ [4], resulting in poor scalability.

2.2.2 Seed-dependent local clustering. Seed-dependent local clustering executes the local graph diffusion from the given seed vertex q to identify higher-order clusters. Higher-order markov chain based random walk [61] and Personalized PageRank based random walk [59, 60] are two well-known local graph diffusion methods. The former uses state transition tensors to simulate the long-term dependence of states. The latter models a random walk with restart over the edge-weighted graph $\mathcal{G}^{\mathbb{M}}$. Based on these backgrounds, Zhou et al. [72, 73] and Yin et al. [65] proposed *HOSPLOC* and *MAPPR* to obtain higher-order clusters, respectively. To be specific, they first compute 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 $y = \pi\mathcal{D}^{-1}$. Then, they run the sweep procedure. Namely, it sorts all non-zero entries in y such that $y_1 \geq y_2 \geq \ldots \geq y_{sup(y)}$

Table 1: A comparison of motif conductance based graph clustering. $\phi^*_{\mathbb{M}} \in (0, 1]$ is the smallest motif conductance value. $k = k(\mathbb{M})$ is the order of $\mathbb{M}$. δ is the degeneracy and is often very small in real-life graphs [39]. 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. *PSMC+* is *PSMC* with estimate strategies proposed in Section 3.3. $\times$ represents the corresponding method has no accuracy guarantee.

Methods	Accuracy Guarantee	Time Complexity	Space Complexity	Remark
HSC [4]	$O(\sqrt{\phi^*_{\mathbb{M}}})$ for $k = 3$ $\times$ for $k > 3$	$O(k(\frac{\delta}{2})^{k-2}m) + O(n^3)$	$O(n^2)$	Eigenvector-based
HOSPLOC [72, 73]	$\times$	$O(t_{max}\frac{2^{bk}}{(\phi^*_{\mathbb{M}})^{2k}}\log^{3k} m)$	$O(n^k)$	Higher-order Markov Chain-based
MAPPR [65]	$\times$	$O(k(\frac{\delta}{2})^{k-2}m) + O(\frac{\log\frac{1}{\epsilon}}{\epsilon})$	$O(n^2)$	Personalized PageRank-based
PSMC (This paper)	$O(1/2 + 1/2\phi^*_{\mathbb{M}})$ for any k	$O(k(\frac{\delta}{2})^{k-2}m)$	$O(m + n)$	*Motif Resident*-based
PSMC+ (This paper)	$\times$	$O(km)$	$O(m + n)$	*Motif Resident*-based

($sup(y)$ is the number of the non-zero entries in y), and outputs $S = \arg\min \phi(S_i)$, in which $S_i = \{y_1, y_2, ..., y_i\}$.

Discussions. Since seed-dependent local clustering methods 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 [65, 72, 73]. 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.

2.2.3 The Shortcomings of Existing Solutions. Table 1 summarizes the above SOTA motif conductance algorithms for comparison. By Table 1, 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* can output $O(1/2 + 1/2\phi^*_{\mathbb{M}})$ accuracy guarantee for any size of motif, while baselines cannot. Note that although the proposed *PSMC+* has no accuracy guarantee (the practical performance of *PSMC+* is comparable to the baselines in our empirical results), it has the excellent property of near-linear time complexity, which is vital for dealing with massive graphs with millions of nodes.

Comparison with [40]. Although our algorithm architecture has many similarities with [40], there are still also pivotal differences. In particular, we are the first to apply this architecture to higher-order graph clustering in a non-trivial way. Specifically, (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 [40]. As a result, the proof of our algorithm's approximation ratio is more challenging and requires some tricks. For example, Lemma 1 and Lemma 2 are unique properties of our problem. Besides, the inequality bounding in the proof process of Theorem 3 is more challenging. (2) When the input motif is not an edge, our proposed motif resident (Definition 4) is more complex than the degree ratio of [40]. For example, [40] can calculate the degree ratio in linear time by simply visiting all edges. However, existing SOTA methods take $O(k(\frac{\delta}{2})^{k-2}m)$ time to calculate the proposed motif resident (Table 1). Thus, we propose several powerful pruning techniques and effective bounds to estimate the motif resident of each vertex, thereby reducing computational costs (Section 3.3). (3) Updating the motif resident is more difficult than the degree ratio of [40]. 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 efficient dynamic update technologies to incrementally maintain the motif resident of each vertex, thereby avoiding recomputing the motif resident from scratch (Lemma 5).

3 PSMC: THE PROPOSED SOLUTION

In this section, we devise a **P**rovable and **S**calable **M**otif **C**onductance algorithm *PSMC*, which aims to output a high-quality cluster. It is important to highlight that PSMC has the capability to provide *fixed* and *motif-independent* approximation ratio for any motif. This significant feature addresses and resolves the open problem raised by [4]. Then, we propose novel dynamic update technologies and effective bounds to further boost efficiency of *PSMC*.

3.1 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 optimized motif conductance starting from each vertex. Before describing our proposed algorithms, several useful definitions are stated as follows.

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 $1 \le j \le k$, we let $\mathbb{M}^C_j(u) = |\{G_S \in \mathbb{M} | u \in S, |C \cap S| = j\}|$.*

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 $Mr_S(u) = \frac{\mathbb{M}(u)+\mathbb{M}^S_k(u)-\mathbb{M}^S_1(u)}{\mathbb{M}(u)}$.*

Based on these definitions, we develop *PSMC* with three-stage computing framework (Algorithm 1). In *Stage 1*, we compute the motif resident value for each vertex (Lines 1-7). In *Stage 2*, we iteratively remove the vertex with the smallest motif resident (Lines 8-11). 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 12-13). Next, we prove that this simple *PSMC* algorithm can produce the high-quality cluster with *fixed* and *motif-independent* approximation ratio.

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 $\mathbb{M}_k^H(u) \geq \mathbb{M}_k^S(u)$ and $\mathbb{M}_1^H(u) \leq \mathbb{M}_1^S(u)$ if $u \in S \subseteq H$.*

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 $cut_{\mathbb{M}}(S \setminus \{u\}) = cut_{\mathbb{M}}(S) - \mathbb{M}_1^S(u) + \mathbb{M}_k^S(u)$.*

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, $g_{\mathbb{M}}(\widetilde{S}) \geq g_{\mathbb{M}}(S)$ for any vertex subset $S \subseteq V$. The following two lemmas are key theoretical basis 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 $vol_{\mathbb{M}}(S) \leq vol_{\mathbb{M}}(V \setminus S)$.*

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

Implications of Lemma 3 and Lemma 4. Since $g_{\mathbb{M}}(\widetilde{S}) \geq g_{\mathbb{M}}(S)$ 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 $vol_{\mathbb{M}}(S^*) \leq vol_{\mathbb{M}}(V \setminus S^*)$, thus the condition of $vol_{\mathbb{M}}(S) \leq vol_{\mathbb{M}}(V \setminus S)$ in Lemma 3 is always true for our problem. Please see Problem Statement in Section 2.1 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 theorem to local a cluster with *fixed* and *motif-independent* approximation ratio.

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

Proof. According to the definitions of $cut_{\mathbb{M}}(C)$, $\mathbb{M}(u)$, and $\mathbb{M}_j^C(u)$, we know that $\mathbb{M}(u) = \sum_{j=1}^{k} \mathbb{M}_j^C(u)$ and $cut_{\mathbb{M}}(C) = \sum_{u\in C} \sum_{j=1}^{k-1} \frac{1}{j} \mathbb{M}_j^C(u)$. Let $\widetilde{S}$ is the optimal vertex set for $g_{\mathbb{M}}(.)$. In Lines 8-11, Algorithm 1 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 because v is the first deleted vertex of $\widetilde{S}$. This implies that $\min_{u\in V_t} Mr_{V_t}(u) = Mr_{V_t}(v) = \frac{\mathbb{M}(v)+\mathbb{M}_k^{V_t}(v)-\mathbb{M}_1^{V_t}(v)}{\mathbb{M}(v)} \geq \frac{\mathbb{M}(v)+\mathbb{M}_k^{\widetilde{S}}(v)-\mathbb{M}_1^{\widetilde{S}}(v)}{\mathbb{M}(v)} = Mr_{\widetilde{S}}(v) \geq g_{\mathbb{M}}(\widetilde{S})$ according to Lemma 4 and Lemma 1. 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,

Algorithm 1 Provable and Scalable Motif Conductance (*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 fixed and motif-independent approximation ratio

1: $MI \leftarrow$ all the motif instances of $\mathbb{M}$
2: Initializing the motif degree $\mathbb{M}(u) = 0$ for any $u \in V$
3: **for** each motif instance $mi \in MI$ **do**
4: **for** each edge $(u,v) \in mi$ **do**
5: $\mathbb{M}(u)+ = 1$; $\mathbb{M}(v)+ = 1$
6: $i \leftarrow 1$; $S_i \leftarrow V$; $\mathbb{M}_k^{S_i}(u) = \mathbb{M}(u)$ and $\mathbb{M}_1^{S_i}(u) = 0$ for $u \in S_i$
7: $Mr_{S_i}(u) \leftarrow \frac{\mathbb{M}(u)+\mathbb{M}_k^{S_i}(u)-\mathbb{M}_1^{S_i}(u)}{\mathbb{M}(u)}$ for any $u \in S_i$
8: **while** $S_i \neq \emptyset$ **do**
9: $u \leftarrow \arg\min\{Mr_{S_i}(u) | u \in S_i\}$
10: $i \leftarrow i + 1$
11: $S_i \leftarrow S_{i-1} \setminus \{u\}$
12: $\hat{S} \leftarrow \arg\min_{S\in\{S_1,S_2,\ldots,S_n\}} \{\phi_{\mathbb{M}}(S) | vol_{\mathbb{M}}(S) \leq vol_{\mathbb{M}}(V \setminus S)\}$
13: **return** $\hat{S}$

$$g_{\mathbb{M}}(V_t) = \frac{\sum_{u\in V_t} \mathbb{M}(u) - cut_{\mathbb{M}}(V_t)}{\sum_{u\in V_t} \mathbb{M}(u)} = \frac{\sum_{u\in V_t} (\mathbb{M}(u) - \sum_{j=1}^{k-1} \frac{1}{j} \mathbb{M}_j^{V_t}(u))}{\sum_{u\in V_t} \mathbb{M}(u)} \tag{4}$$

$$= \frac{\sum_{u\in V_t} (\sum_{j=2}^{k-1} (1-\frac{1}{j}) \mathbb{M}_j^{V_t}(u) + \mathbb{M}_k^{V_t}(u))}{\sum_{u\in V_t} \mathbb{M}(u)} \geq \frac{\sum_{u\in V_t} (\sum_{j=2}^{k-1} \frac{1}{2} \mathbb{M}_j^{V_t}(u) + \mathbb{M}_k^{V_t}(u))}{\sum_{u\in V_t} \mathbb{M}(u)} \tag{5}$$

$$= \frac{\sum_{u\in V_t} (\sum_{j=2}^{k-1} \mathbb{M}_j^{V_t}(u) + 2 \cdot \mathbb{M}_k^{V_t}(u))}{2 \sum_{u\in V_t} \mathbb{M}(u)} = \frac{\sum_{u\in V_t} (\mathbb{M}(u) + \mathbb{M}_k^{V_t}(u) - \mathbb{M}_1^{V_t}(u))}{2 \sum_{u\in V_t} \mathbb{M}(u)} \tag{6}$$

$$\geq \frac{1}{2} \frac{\sum_{u\in V_t} g_{\mathbb{M}}(\widetilde{S}) \cdot \mathbb{M}(u)}{\sum_{u\in V_t} \mathbb{M}(u)} = \frac{1}{2} g_{\mathbb{M}}(\widetilde{S}). \tag{7}$$

Since Algorithm 1 maintains the optimal solution during the peeling process in Lines 12-13, $\phi_{\mathbb{M}}(\hat{S}) = 1 - g_{\mathbb{M}}(\hat{S}) \leq 1 - g_{\mathbb{M}}(V_t) \leq 1 - \frac{g_{\mathbb{M}}(\widetilde{S})}{2}$ 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, $\phi_{\mathbb{M}}(\hat{S}) \leq 1/2 + 1/2\phi_{\mathbb{M}}(S^*)$. As a result, Algorithm 1 can identify a higher-order cluster with motif conductance $1/2 + 1/2\phi^*_{\mathbb{M}}$. □

3.2 Dynamic Update of *Motif Resident*

The computational challenge of Algorithm 1 is how to incrementally maintain Mr_{S_i} in Line 9 and $\phi_{\mathbb{M}}(S_i)$ in Line 12 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 $g_{\mathbb{M}}(S_i)$. We propose efficient dynamic update technologies to solve the challenge as follows.

Lemma 5. *For the current search space S_i, if a vertex $u \in S_i$ is removed, for any $v \in N_{S_i}(u)$ and $S_{i+1} = S_i \setminus \{u\}$, we have the following equation:*

$$Mr_{S_{i+1}}(v) = Mr_{S_i}(v) - \frac{\mathbb{M}_k^{S_i}(u,v) + \mathbb{M}_2^{S_i}(u,v)}{\mathbb{M}(u)} \tag{8}$$

$$g_{\mathbb{M}}(S_{i+1}) = \frac{vol_{\mathbb{M}}(S_i)g_{\mathbb{M}}(S_i) - Mr_{S_i}(u)\mathbb{M}(u)}{vol_{\mathbb{M}}(S_i) - \mathbb{M}(u)} \quad (9)$$

Where $\mathbb{M}_k^{S_i}(u,v) = |\{G_S \in \mathbb{M} | (u,v) \in G_S, |S_i \cap S| = k\}|$ and $\mathbb{M}_2^{S_i}(u,v) = |\{G_S \in \mathbb{M} | (u,v) \in G_S, |S_i \cap S| = 2\}|$. 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 edge (u,v) with exactly k (resp., 2) vertices in S_i.

Based on Lemma 5, Algorithm 1 can incrementally update the motif resident for each vertex when its neighbor is removed. The time complexity of Algorithm 1 is analyzed as follows.

Theorem 4. *The worse-case time complexity of Algorithm 1 is $O(k(\delta/2)^{k-2}m)$. k is the order of $\mathbb{M}$. δ is the degeneracy and is often very small in real-world graphs (Table 2).*

3.3 Lower and Upper Bounds of *Motif Resident*

According to the definitions of $Mr_S(.)$ and $g_{\mathbb{M}}(.)$, we can know that the initial $Mr_{S_1}(u) = 2$ for any vertex $u \in S_1$ and $g_{\mathbb{M}}(S_1) = 1$. Thus, by Equation 8 and Equation 9, we further know that the bottleneck of Algorithm 1 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. 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. Therefore, 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.

3.3.1 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 5 (Turan Theorem [57]). *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 5, 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 further expand the Turan Theorem.

Theorem 6. *[14] 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 $A_i = \{C_i^1, C_i^2, ..., C_i^{\binom{r}{h}}\}$ be the h-clique set obtained from the i-th r-clique of Theorem 6, in which $i \in \{1, 2, ..., (\frac{|V(NS)|}{r-1})^{r-2}\}$. 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 [5] 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.

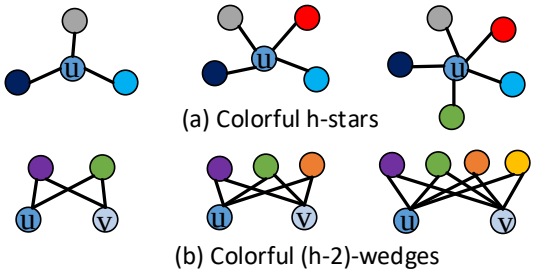


Figure 2: Colorful h-stars and colorful $(h-2)$-wedges.

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.

3.3.2 Upper Bounds. Gao et al. proposed the concept of colorful h-star degree $csd_h(u)$, 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 (Figure 2 (a)). Since each vertex in h-clique must have a different color, we can obtain $csd_h(u) \geq \mathbb{M}(u)$ 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 (Figure 2 (b)). Assume that (u,v) is an edge and $hc(u,v)$ is any h-clique containing the edge (u,v), we know that each vertex in $hc(u,v)$ must have a different color. Thus, $cwd_h^{S_i}(u,v) \geq \mathbb{M}_k^{S_i}(u,v)$ 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 in near-linear time by the dynamic programming. Therefore, we directly take the average of their upper and lower bounds as the corresponding estimated value.

4 EXPERIMENTAL EVALUATION

4.1 Experimental Setup

Datasets. Our solutions are evaluated on five real-world graphs with ground-truth clusters[3], which are widely used benchmarks for higher-order graph clustering [4, 16, 29]. Besides, we also use four types of synthetic graphs to test the scalability and the effectiveness of our solutions: *LFR* [37], *PLC* [28], *ER* [15], and *BA* [3], which can be generated by the well-known NetworkX Python package [25]. Note that they can be used to simulate the degree distributions, communities, and small-world properties in the real world.

[3]All datasets can be downloaded from http://snap.stanford.edu/

Table 2: Dataset statistics. δ is the degeneracy.

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
LFR [37]	$10^3 \sim 10^7$	$10^3 \sim 10^7$	$3 \sim 5$	Synthetic network
PLC [28]	$10^3 \sim 10^7$	$10^3 \sim 10^7$	$3 \sim 5$	Synthetic network
ER [15]	$10^3 \sim 10^7$	$10^3 \sim 10^7$	$5 \sim 11$	Synthetic network
BA [3]	$10^3 \sim 10^7$	$10^3 \sim 10^7$	$3 \sim 5$	Synthetic network

Competitors. The following SOTA baselines are implemented for comparison. (1) Traditional graph clustering: *SC* [1], *Louvain* [7] and *KCore* [49]; (2) Cohesive subgraph based higher-order graph clustering: *HD* [16, 27, 53]; (3) Modularity based higher-order graph clustering: *HM* [2, 29]; (4) Motif conductance based higher-order graph clustering: *HSC* [4], *MAPPR* [65], and *HOSPLOC* [72, 73]. *PSMC* is our proposed Algorithm 1 and *PSMC+* is *PSMC* with bound estimation strategies in Section 3.3.

Parameters and Implementations. Unless specified otherwise, we take the default parameters of these baselines 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 [20, 55, 65], we also limit ourselves to the representative k-clique motif to illustrate the main patterns observed. All experiments are conducted on a Linux server with an Intel(R) Xeon(R) E5-2683 v3@2.00GHZ CPU and 256GB RAM running CentOS 6.10.

4.2 Efficiency Testing

Since the objective functions of traditional graph clustering (e.g., *SC* [1], *Louvain* [7] and *KCore* [49]), cohesive subgraph based higher-order graph clustering (e.g., *HD* [16, 27, 53]), and modularity based higher-order graph clustering (e.g., *HM* [2, 29]) 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* and *PSMC+* with varying $k(\mathbb{M})$ on five real-world networks is detailed in Figure 3. Note that we do not report the empirical results for LiveJ and Orkut on $k(\mathbb{M}) = 6$. This is because we cannot obtain the results of baselines (i.e., *MAPPR*, *HOSPLOC*, and *HSC*) within 7 days. By Figure 3, we have: (1) *PSMC+* is consistently faster than other methods. This is because *PSMC+* non-trivially adopts the technologies of Turan Theorem, colorful h-star degree, and colorful $(h-2)$-wedge degree to estimate the motif resident with near-linear time (Table 1). (2) *PSMC* is the runner-up on four of the five 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 1). 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 [4]. (4) The sparser the graph (i.e., the smaller the δ), the faster our algorithms (i.e., *PSMC* and *PSMC+*). For example, our algorithms are faster on Youtube compared to DBLP, despite Youtube having more vertices and more edges (Table 2). This is because Youtube has a smaller δ (Table 2). These results give preliminary evidence that the proposed solutions are indeed high efficiency in practice.

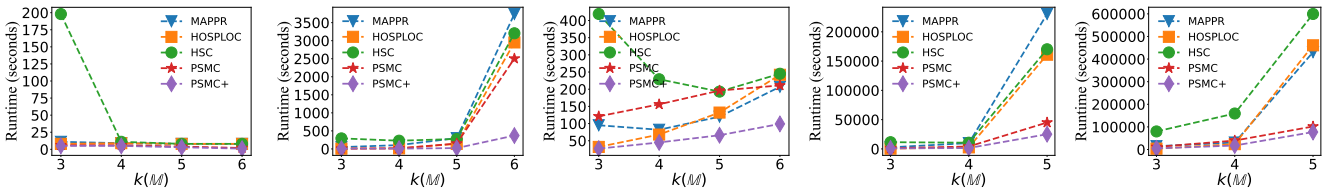
Exp-2: Scalability testing on synthetic graphs. Extensive synthetic graphs are generated to further test the scalability of our solutions. Figure 4 only presents the results when the given motif is a triangle, with comparable trends across other motifs. By Figure 4, we know that *PSMC* and *PSMC+* scale 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 1). These results indicate that our algorithms have excellent scalability over massive graphs while the baselines do not.

Exp-3: Memory overhead comparison. Figure 5 displays the memory overhead of the evaluated algorithms when the motif is a triangle, with comparable trends across other motifs. As excepted, our porposed *PSMC* and *PSMC+* 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 need to maintain some simple data structures, such as motif resident of each vertex (Algorithm 1). Note that *PSMC+* is slightly worse than *PSMC*. This is because *PSMC+* needs to maintain more intermediate variables to estimate motif resident (Section 3.3). 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 ordering 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 2.2). These results affirm the memory efficiency of our algorithms.

4.3 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 [4, 16, 29]. Besides, we also use motif conductance (*MC* for short) calculated by Definition 2 to evaluate the quality of the identified community. The smaller the value of the *MC*(C), the better the partition of community C. We also report the size of the identified community for completeness. Note that we do not report traditional edge-based metric (e.g., density, conductance) because they are mainly used to measure the quality of communities with edges as atomic clustering units (Section 1).

Exp-4: Effectiveness of various graph clustering methods. Table 3 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

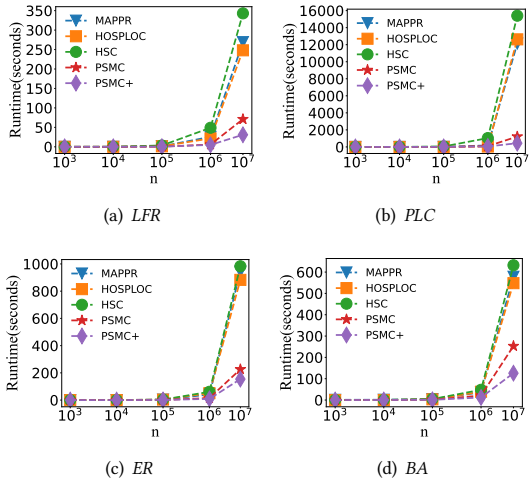


(a) Amazon (b) DBLP (c) Youtube (d) LiveJ (e) Orkut

Figure 3: Runtime (seconds) of different motif conductance algorithms with varying $k(\mathbb{M})$.

Table 3: Effectiveness of various graph clustering methods. 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 community size is better. We are simply presenting the community size objectively to provide an intuitive experience.

Model	Amazon			DBLP			Youtube			LiveJ			Orkut		
	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
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
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
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
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
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
MAPPR	0.015	0.457	175	0.115	0.339	28796	0.132	0.116	15810	0.102	0.257	307740	0.104	0.233	1343943
HOSPLOC	0.062	0.467	90	0.260	0.283	414	0.103	0.128	434	0.266	0.342	3586	0.381	0.237	44651
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
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



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

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

Figure 4: Scalability testing on synthetic graphs.

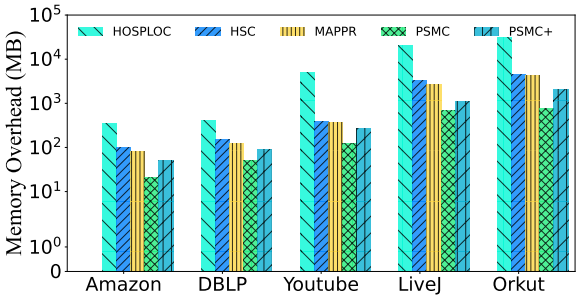
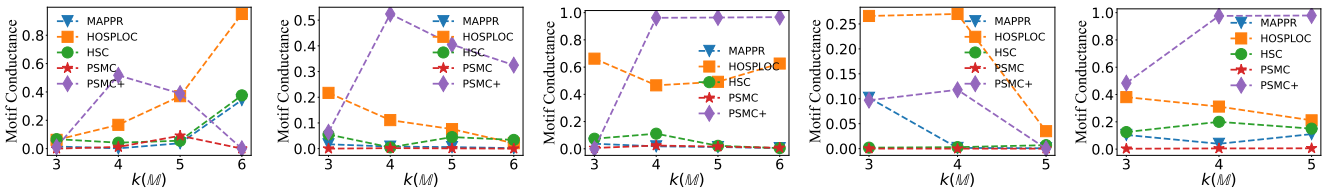


Figure 5: Memory overhead on real-world graphs (excluding the size of the graph itself).

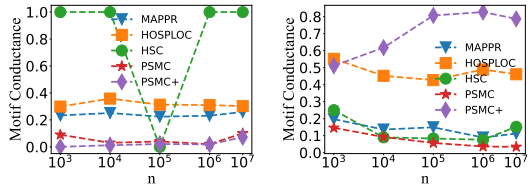
other methods on four of the five datasets (on Youtube, *PSMC+* is the champion and *PSMC* is the runner-up). In particular, *PSMC* is 167, 12, 20, and 41 times better than *HSC* on Amazon, Youtube, LiveJ, and Orkut, respectively. This is because *PSMC* can find clusters with near-liner approximation ratio, while *HSC* has quadratic bound (Table 1). (2) *PSMC+* outperforms *MAPPR* and *HOSPLOC* on four of the five datasets. This is because *MAPPR* and *HOSPLOC* are heuristic and have no guarantee of clustering quality (Table 1). However, *PSMC+* is built on top of *PSMC*, so even if *PSMC+* has no theoretical guarantee, it can still get good quality in practice. For F1-Score metric, we have: (1) *PSMC* consistently outperforms other methods (including *HD* and *HM*). (2) *SC*, *Louvain*, and *KCore* have poor F1-Scores on most cases. This is because they are traditional clustering methods which cannot capture higher-order structural information for graph clustering. For Size metric, on average, the community sizes found by different algorithms from largest to smallest are *SC*, *PSMC+*, *MAPPR*, *HD*, *PSMC*, *HM*, *Louvain*, *HOSPLOC*, *HSC*, and *KCore*. Our algorithm *PSMC* returns the community size that is ranked in the middle, so it tends to find communities of moderate size. However, other baselines either find the community that is too large or too small, leading to poor interpretability. Thus, these results give clear evidence that our solutions can indeed find higher quality clusters when contrasted with baselines.

Exp-5: Quality of various motif conductance algorithms. Figure 6 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+* and *HOSPLOC* fluctuates greatly as $k(\mathbb{M})$ increases. However, *PSMC+* performs well in synthetic graphs (see Figure 7 for details). One possible explanation is that as $k(\mathbb{M})$ increases, the actual effect of estimation bounds in *PSMC+* depends on the type

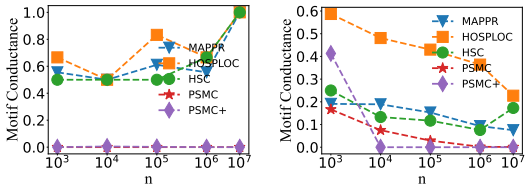


(a) Amazon (b) DBLP (c) Youtube (d) LiveJ (e) Orkut

Figure 6: Quality of various motif conductance algorithms with varying $k(\mathbb{M})$ on real-world graphs.



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



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

Figure 7: Quality of various motif conductance algorithms on synthetic graphs.

of network (e.g. real-world graphs have poor pruning effects, while synthetic graphs have good pruning effects). Moreover, we also report these qualities on extensive synthetic graphs. As shown in Figure 7, *PSMC+* outperforms other methods in most cases. Besides, *PSMC* is runner up and slightly worse than *PSMC+*. 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. These results indicate that our algorithms can identify higher quality clusters than the baselines on real-world&synthetic graphs.

5 RELATED WORK

Traditional Graph Clustering. Graph clustering has received much cattention over past decades [17, 18]. Modularity [34, 46, 51] and conductance [21, 23, 26, 40, 58] 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 [21, 46]. 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 [6, 7]. *Fiedler* vector-based spectral clustering algorithm can output a cluster with a quadratic factor of optimal conductance [40]. 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. Notable examples include average-degree densest subgraph, k-core, and k-truss [9]. However, 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 [63, 71], heterogeneous information networks [10, 33], and temporal networks [41–43, 69, 74]. Obviously, these methods are orthogonal to our work.

Higher-order Graph Clustering. In addition to the motif conductance studied in this paper [4], 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 [2, 29]. Higher-order densest subgraph model was proposed where the density is defined as the number of motif instances divided by the size of vertices [16, 27, 53]. Li et al. proposed an edge enhancement approach to overcome the hypergraph fragmentation issue appearing in the seminal reweighting framework [38]. 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 [8], labeled networks [20, 48], multi-layer networks [30], dynamic networks [19]. Clearly, these methods on complicated networks are orthogonal to our work.

6 CONCLUSION

We first propose a *simple* but *provable* algorithm *PSMC* for motif conductance based graph clustering. 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 technologies and effective bounds to further boost efficiency of *PSMC*. Finally, empirical results on real-life and synthetic datasets demonstrate the superiority of the proposed algorithms on both clustering accuracy and running time.

7 ACKNOWLEDGMENTS

The work was supported by (i) the National Natural Science Foundation of China under Grant Nos. 62072205 and 61932004, (ii) NSFC Grants U2241211, 62072034, (iii) China Postdoctoral Science Foundation under Grant 2023T00325, (iv) Fundamental Research Funds for the Central Universities under Grant SWU-KQ22028. Zeli Wang and Rong-Hua Li are the corresponding authors of this paper.

REFERENCES

[1] Noga Alon and Vitali D Milman. 1985. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B* 38, 1 (1985), 73–88.

[2] Alex Arenas, Alberto Fernandez, Santo Fortunato, and Sergio Gomez. 2008. Motif-based communities in complex networks. *Journal of Physics A: Mathematical and Theoretical* 41, 22 (2008), 224001.

[3] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[4] Austin R Benson, David F Gleich, and Jure Leskovec. 2016. Higher-order organization of complex networks. *Science* 353, 6295 (2016), 163–166.

[5] Andreas Björklund, Thore Husfeldt, and Mikko Koivisto. 2009. Set Partitioning via Inclusion-Exclusion. *SIAM J. Comput.* 39, 2 (2009), 546–563.

[6] Vincent D. Blondel, Jean-Loup Guillaume, and Renaud Lambiotte. 2023. Fast unfolding of communities in large networks: 15 years later. *CoRR* abs/2311.06047 (2023).

[7] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[8] Aldo G. Carranza, Ryan A. Rossi, Anup Rao, and Eunyee Koh. 2020. Higher-order Clustering in Complex Heterogeneous Networks. In *KDD*. 25–35.

[9] Lijun Chang and Lu Qin. 2019. Cohesive Subgraph Computation Over Large Sparse Graphs. In *Proceedings of ICDE*. 2068–2071.

[10] Lu Chen, Yunjun Gao, Yuanliang Zhang, Christian S. Jensen, and Bolong Zheng. 2019. Efficient and Incremental Clustering Algorithms on Star-Schema Heterogeneous Graphs. In *ICDE*. 256–267.

[11] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. 2019. Cluster-GCN: An Efficient Algorithm for Training Deep and Large Graph Convolutional Networks. In *KDD*. 257–266.

[12] Karen D. Devine, Erik G. Boman, and George Karypis. 2006. Partitioning and Load Balancing for Emerging Parallel Applications and Architectures. In *Parallel Processing for Scientific Computing*. 99–126.

[13] Alexandre Duval and Fragkiskos D. Malliaros. 2022. Higher-order Clustering and Pooling for Graph Neural Networks. In *CIKM*. 426–435.

[14] P Erdos. 1969. On the number of complete subgraphs and circuits contained in graphs. *Casopis Pest. Mat.* 94 (1969), 290–296.

[15] Paul Erdos, Alfréd Rényi, et al. 1960. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* 5, 1 (1960), 17–60.

[16] Yixiang Fang, Kaiqiang Yu, Reynold Cheng, Laks V. S. Lakshmanan, and Xuemin Lin. 2019. Efficient Algorithms for Densest Subgraph Discovery. *Proc. VLDB Endow.* 12, 11 (2019), 1719–1732.

[17] Santo Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3-5 (2010), 75–174.

[18] Santo Fortunato and Darko Hric. 2016. Community detection in networks: A user guide. *CoRR* abs/1608.00163 (2016).

[19] Dongqi Fu, Dawei Zhou, and Jingrui He. 2020. Local Motif Clustering on Time-Evolving Graphs. In *KDD*. 390–400.

[20] Dongqi Fu, Dawei Zhou, Ross Maciejewski, Arie Croitoru, Marcus Boyd, and Jingrui He. 2023. Fairness-Aware Clique-Preserving Spectral Clustering of Temporal Graphs. In *WWW*. 3755–3765.

[21] Sainyam Galhotra, Amitabha Bagchi, Srikanta Bedathur, Maya Ramanath, and Vidit Jain. 2015. Tracking the Conductance of Rapidly Evolving Topic-Subgraphs. *Proc. VLDB Endow.* 8, 13 (2015), 2170–2181.

[22] Sen Gao, Rong-Hua Li, Hongchao Qin, Hongzhi Chen, Ye Yuan, and Guoren Wang. 2022. Colorful h-star Core Decomposition. In *ICDE*.

[23] David F. Gleich and C. Seshadhri. 2012. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *KDD5*.

[24] Manish Gupta, Jing Gao, Yizhou Sun, and Jiawei Han. 2012. Integrating community matching and outlier detection for mining evolutionary community outliers. In *KDD*. 859–867.

[25] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. 2008. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proceedings of the 7th Python in Science Conference*. 11 – 15.

[26] Yue He, Longlong Lin, Pingpeng Yuan, Ronghua Li, Tao Jia, and Zeli Wang. 2024. CCSS: Towards conductance-based community search with size constraints. *Expert Systems with Applications* 250 (2024), 123915.

[27] Yizhang He, Kai Wang, Wenjie Zhang, Xuemin Lin, and Ying Zhang. 2023. Scaling Up k-Clique Densest Subgraph Detection. *Proc. ACM Manag. Data* 1, 1 (2023), 69:1–69:26.

[28] Petter Holme and Beom Jun Kim. 2002. Growing scale-free networks with tunable clustering. *Physical review E* 65, 2 (2002), 026107.

[29] Ling Huang, Hong-Yang Chao, and Guangqiang Xie. 2020. MuMod: A Micro-Unit Connection Approach for Hybrid-Order Community Detection. In *AAAI*. 107–114.

[30] Ling Huang, Chang-Dong Wang, and Hongyang Chao. 2021. HM-Modularity: A Harmonic Motif Modularity Approach for Multi-Layer Network Community Detection. *IEEE Trans. Knowl. Data Eng.* 33, 6 (2021), 2520–2533.

[31] Shixun Huang, Yuchen Li, Zhifeng Bao, and Zhao Li. 2021. Towards Efficient Motif-based Graph Partitioning: An Adaptive Sampling Approach. In *ICDE*. 528–539.

[32] Zengfeng Huang, Shengzhong Zhang, Chong Xi, Tang Liu, and Min Zhou. 2021. Scaling Up Graph Neural Networks Via Graph Coarsening. In *KDD*. 675–684.

[33] Xun Jian, Yue Wang, and Lei Chen. 2020. Effective and Efficient Relational Community Detection and Search in Large Dynamic Heterogeneous Information Networks. *Proc. VLDB Endow.* 13, 10 (2020), 1723–1736.

[34] Junghoon Kim, Siqiang Luo, Gao Cong, and Wenyuan Yu. 2022. DMCS : Density Modularity based Community Search. In *SIGMOD*. 889–903.

[35] Brian James Kloostra, Chaitanya Dalvi, and Brookton Noah Behm. 2009. System and method for analyzing and dispositioning money laundering suspicious activity alerts. US Patent App. 12/258,784.

[36] Christine Klymko, David F. Gleich, and Tamara G. Kolda. 2014. Using Triangles to Improve Community Detection in Directed Networks. *CoRR* abs/1404.5874 (2014).

[37] Andrea Lancichinetti, Santo Fortunato, and János Kertész. 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New journal of physics* 11, 3 (2009), 033015.

[38] Pei-Zhen Li, Ling Huang, Chang-Dong Wang, and Jian-Huang Lai. 2019. EdMot: An Edge Enhancement Approach for Motif-aware Community Detection. In *KDD*. 479–487.

[39] Ronghua Li, Sen Gao, Lu Qin, Guoren Wang, Weihua Yang, and Jeffrey Xu Yu. 2020. Ordering Heuristics for k-clique Listing. *Proc. VLDB Endow.* 13, 11 (2020), 2536–2548.

[40] Longlong Lin, Ronghua Li, and Tao Jia. 2023. Scalable and Effective Conductance-Based Graph Clustering. In *AAAI*. 4471–4478.

[41] Longlong Lin, Pingpeng Yuan, Rong-Hua Li, and Hai Jin. 2022. Mining Diversified Top-rr Lasting Cohesive Subgraphs on Temporal Networks. *IEEE Trans. Big Data* 8, 6 (2022), 1537–1549.

[42] Longlong Lin, Pingpeng Yuan, Rong-Hua Li, Jifei Wang, Ling Liu, and Hai Jin. 2022. Mining Stable Quasi-Cliques on Temporal Networks. *IEEE Trans. Syst. Man Cybern. Syst.* 52, 6 (2022), 3731–3745.

[43] Longlong Lin, Pingpeng Yuan, Rong-Hua Li, Chun-Xue Zhu, Hongchao Qin, Hai Jin, and Tao Jia. 2024. QTCS: Efficient Query-Centered Temporal Community Search. *Proc. VLDB Endow.* 17, 6 (2024), 1187–1199.

[44] Shmoolik Mangan and Uri Alon. 2003. Structure and function of the feed-forward loop network motif. *Proceedings of the National Academy of Sciences* 100, 21 (2003), 11980–11985.

[45] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. 2002. Network motifs: simple building blocks of complex networks. *Science* (2002).

[46] Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E* 69, 2 (2004), 026113.

[47] Pedro Ribeiro, Pedro Paredes, Miguel E. P. Silva, David Aparício, and Fernando M. A. Silva. 2022. A Survey on Subgraph Counting: Concepts, Algorithms, and Applications to Network Motifs and Graphlets. *ACM Comput. Surv.* 54, 2 (2022), 28:1–28:36.

[48] Ahmet Erdem Sariyüce. 2021. Motif-driven Dense Subgraph Discovery in Directed and Labeled Networks. In *WWW*. 379–390.

[49] S. B. Seidman. 1983. Network structure and minimum degree. *Social Networks* 5, 3 (1983), 269–287.

[50] Jianbo Shi and Jitendra Malik. 1997. Normalized Cuts and Image Segmentation. In *CVPR*. 731–737.

[51] Hiroaki Shiokawa, Yasuhiro Fujiwara, and Makoto Onizuka. 2013. Fast Algorithm for Modularity-Based Graph Clustering. In *AAAI*. AAAI Press, 1170–1176.

[52] Konstantinos Sotiropoulos and Charalampos E. Tsourakakis. 2021. Triangle-aware Spectral Sparsifiers and Community Detection. In *KDD*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). 1501–1509.

[53] Bintao Sun, Maximilien Danisch, T.-H. Hubert Chan, and Mauro Sozio. 2020. KClist++: A Simple Algorithm for Finding k-Clique Densest Subgraphs in Large Graphs. *Proc. VLDB Endow.* 13, 10 (2020), 1628–1640.

[54] Luca Trevisan. 2017. Lecture notes on graph partitioning, expanders and spectral methods. *University of California, Berkeley, https://people. eecs. berkeley. edu/luca/books/expanders-2016. pdf* (2017).

[55] Charalampos E. Tsourakakis. 2015. The K-clique Densest Subgraph Problem. In *WWW*. 1122–1132.

[56] Charalampos E. Tsourakakis, Jakub Pachocki, and Michael Mitzenmacher. 2017. Scalable Motif-aware Graph Clustering. In *WWW*. 1451–1460.

[57] P Turan. 1941. On an extremal problem in graph theory. *Mat. Fiz. Lapok* 48, 137 (1941), 436–452.

[58] Ulrike Von Luxburg. 2007. A tutorial on spectral clustering. *Statistics and computing* 17, 4 (2007), 395–416.

[59] Sibo Wang, Renchi Yang, Xiaokui Xiao, Zhewei Wei, and Yin Yang. 2017. FORA: Simple and Effective Approximate Single-Source Personalized PageRank. In *KDD*. 505–514.

[60] Zhewei Wei, Xiaodong He, Xiaokui Xiao, Sibo Wang, Shuo Shang, and Ji-Rong Wen. 2018. TopPPR: Top-k Personalized PageRank Queries with Precision Guarantees on Large Graphs. In *SIGMOD*. 441–456.

[61] Li Wen and Michael K. Ng. 2014. "On the limiting probability distribution of a transition probability tensor. *Linear and Multilinear Algebra* 62, 3 (2014), 362–385.

[62] Feng Xia, Shuo Yu, Chengfei Liu, Jianxin Li, and Ivan Lee. 2022. CHIEF: Clustering With Higher-Order Motifs in Big Networks. *IEEE Trans. Netw. Sci. Eng.* 9, 3 (2022),

990–1005.
[63] Renchi Yang, Jieming Shi, Yin Yang, Keke Huang, Shiqi Zhang, and Xiaokui Xiao. 2021. Effective and Scalable Clustering on Massive Attributed Graphs. In *WWW*. 3675–3687.
[64] Ömer Nebil Yaveroğlu, Noël Malod-Dognin, Darren Davis, Zoran Levnajic, Vuk Janjic, Rasa Karapandza, Aleksandar Stojmirovic, and Nataša Pržulj. 2014. Revealing the hidden language of complex networks. *Scientific reports* 4, 1 (2014), 4547.
[65] Hao Yin, Austin R. Benson, Jure Leskovec, and David F. Gleich. 2017. Local Higher-Order Graph Clustering. In *KDD*. 555–564.
[66] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. 2018. Hierarchical Graph Representation Learning with Differentiable Pooling. In *NeurIPS*. 4805–4815.
[67] Shuo Yu, Yufan Feng, Da Zhang, Hayat Dino Bedru, Bo Xu, and Feng Xia. 2020. Motif discovery in networks: A survey. *Comput. Sci. Rev.* (2020).
[68] Shengzhong Zhang, Zengfeng Huang, Haicang Zhou, and Ziang Zhou. 2020. SCE: Scalable Network Embedding from Sparsest Cut. In *kdd*. 257–265.
[69] Yifei Zhang, Longlong Lin, Pingpeng Yuan, and Hai Jin. 2022. Significant Engagement Community Search on Temporal Networks. In *DASFAA*. 250–258.
[70] Yilin Zhang and Karl Rohe. 2018. Understanding Regularized Spectral Clustering via Graph Conductance. In *NeurIPS*. 10654–10663.
[71] Chen Zhe, Aixin Sun, and Xiaokui Xiao. 2019. Community Detection on Large Complex Attribute Network. In *KDD*. 2041–2049.
[72] Dawei Zhou, Si Zhang, Mehmet Yigit Yildirim, Scott Alcorn, Hanghang Tong, Hasan Davulcu, and Jingrui He. 2017. A Local Algorithm for Structure-Preserving Graph Cut. In *KDD*. 655–664.
[73] Dawei Zhou, Si Zhang, Mehmet Yigit Yildirim, Scott Alcorn, Hanghang Tong, Hasan Davulcu, and Jingrui He. 2021. High-Order Structure Exploration on Massive Graphs: A Local Graph Clustering Perspective. *ACM Trans. Knowl. Discov. Data* 15, 2 (2021), 18:1–18:26.
[74] Chunxue Zhu, Longlong Lin, Pingpeng Yuan, and Hai Jin. 2022. Discovering Cohesive Temporal Subgraphs with Temporal Density Aware Exploration. *Journal of Computer Science and Technology* (2022). https://doi.org/10.1007/s11390-022-2431-z