Towards Efficient Motif-based Graph Partitioning: An Adaptive Sampling Approach

Shixun Huang[†] Yuchen Li [‡] Zhifeng Bao[†] Zhao Li[§]
[†]*RMIT University* [‡]*Singapore Management University* [§]*Alibaba Group*
[†]{shixun.huang, zhifeng.bao}@rmit.edu.au, [‡]yuchenli@smu.edu.sg, [§]lizhao.lz@alibaba-inc.com

***Abstract*—In this paper, we study the problem of efficient motif-based graph partitioning (MGP). We observe that existing methods require to enumerate all motif instances to compute the exact edge weights for partitioning. However, the enumeration is prohibitively expensive against large graphs. We thus propose a sampling-based MGP (SMGP) framework that employs an unbiased sampling mechanism to efficiently estimate the edge weights while trying to preserve the partitioning quality. To further improve the effectiveness, we propose a novel adaptive sampling framework called SMGP+. SMGP+ iteratively partitions the input graph based on up-to-date estimated edge weights, and adaptively adjusts the sampling distribution so that edges that are more likely to affect the partitioning outcome will be prioritized for weight estimation. To our best knowledge, this is the first attempt to solve the MGP problem without employing exact edge weight computations, which gives hope for existing MGP methods to perform on complicated motifs in a scalable yet effective manner. Extensive experiments on seven real-world datasets have validated that our framework delivers competitive partitioning quality compared to existing workflows based on exact edge weights, while achieving orders of magnitude speedup.**

I. Introduction

Graph Partitioning (GP) is a fundamental yet core operator in processing large graphs and has been used in a wide range of domains including community detection [1], distributed computing [2] and image processing [3]. While traditional GP approaches focus on reducing the number of edges between partitions, recent studies have unveiled that the network motifs, defined by a particular pattern of interactions between vertices, may provide a deep insight into discovering high-order communities from complex networks [4], [5]. For example, triangular motifs are crucial to finding ground-truth communities in social networks [6] and identifying structural hubs in the brain [7], and multi-hop loops may indicate money laundering process in financial transactions [4]. It thus attracts many interests in developing motif-based GP (MGP) methods to preserve motif patterns in the clusters or communities after the partitioning process [6], [8], [9].

To preserve a specified motif pattern Q during the partitioning of a graph G, a general way is to minimize the number of subgraphs in G that are isomorphic to Q and consist of edges crossing different partitions. Existing MGP approaches employ a two-step workflow. In the first step, G is transformed into an edge-weighted graph where the weight of each edge e is the number of induced subgraphs in G that are isomorphic to Q and contain e. In the second step, an existing GP kernel is chosen to partition the weighted graph obtained from the first step. The GP kernels can be classified based on the objective function adopted. Two commonly adopted objectives are *modularity* [10] and *conductance* [11].

The first step is the major efficiency bottleneck since determining whether G contains a subgraph that is isomorphic to Q is NP-complete [12]. In practice, as evidenced by our experiments later, it takes about a week to compute the exact edge weights over a reasonably small-scale graph with a motif of four vertices (e.g., the motif Q_1 on the Catster dataset in our experiments).

To mitigate the inefficiency of exact edge weight computations, we first devise an edge-centric sampling method called SMGP which is an unbiased estimator of exact edge weights and tries to preserve the quality of partitioning result produced upon the estimated weights. SMGP first samples a target edge e as an induced subgraph of two vertices, and then gradually expands the subgraph by iteratively sampling neighborhood vertices for edge weight estimations. Upon sampling a subgraph I that is isomorphic to the query Q, we update the weights of all edges in I, and the estimated weighted graph is later fed to a GP kernel for partitioning the graph. This edge-centric sampling method is an *unbiased estimator* for the exact edge weight, which thereafter establishes the approximation guarantees of SMGP on the partitioning quality.

SMGP treats each edge equally and may have to generate a large number of samples to ensure an accurate weight estimation for every edge. In fact, edges have varying degrees of importance in affecting the outcomes of GP kernels. For example, edges in densely connected subgraphs are likely to have large exact edge weights, but they can be assigned with small estimated weights due to inaccurate estimations. Thus, it is very likely that the underlying GP kernel would place the endpoints of these edges into different partitions. In contrast, the accurate estimation for edges with small exact weights is relatively less crucial to producing high-quality partitioning results, because the decision on placing these edges may have minimal influence on the partitioning quality.

In order to further improve the partitioning quality with a limited budget (e.g., the sample size), we propose a novel *adaptive sampling framework* called SMGP+. In each iteration of SMGP+, it performs sampling for edge weight estimation followed by calling a GP kernel to partition the graph. The partitioning outcome from the previous iteration is used to adjust the sampling distribution for the next iteration, where edges that are likely to affect the partitioning outcome

TABLE I: Frequently used notations.

Notation	Description
$V(G)$	The set of vertices in graph G.
$E(G)$	The set of edges in graph G.
N_v	The set of neighbors of v (in G).
ϕ_k	A partitioning plan with k partitions.
$\phi_k(u)$	The ID of the partition to which vertex u belongs.
φ_i	The partition with an ID i.
w_e^i	The estimated weight of edge e in the i-th iteration.
$\pi(I)$	The probability of sampling the induced subgraph I.
$\widetilde{\pi}(e)$	The probability of sampling edge e as the starting edge.
$\mathcal{C}(u, \varphi_i)$	The set of edges connecting u to vertices in partition φ_i.

are prioritized for weight estimations. We also prove the theoretical guarantees of SMGP+ w.r.t. the partitioning quality based on *martingales* [13], a classical statistical tool.

To our best knowledge, this is the first attempt to solve the MGP problem without employing exact edge weight computations. Our sampling based methods give hope for existing motif-based graph partitioning (MGP) methods to perform on complex-structured motifs in a scalable yet effective manner. Our proposed methods can efficiently evaluate the potential of complex motifs with different underlying optimization goals (i.e., modularity and conductance). That can help existing MGP methods choose good motifs for large network analysis without costly trials via exact edge weight computations. To summarize, we have made the following contributions:

1) We devise a sampling-based approach called SMGP (in Section III) for the MGP problem and propose an adaptive sampling framework SMGP+ (in Section IV) that improves the effectiveness and efficiency of SMGP.
2) We prove the theoretical guarantees on the partitioning quality based on the weighted graph produced by SMGP (in Section III) and SMGP+ (in Section IV) respectively.
3) We conduct extensive experiments (in Section V) and verify: (1) SMGP can work with different GP kernels and produce competitive results compared with the one produced based on exact edge weights; (2) SMGP+ significantly outperforms SMGP with faster convergence and achieves two orders of magnitude speedup than the traditional MGP workflow while achieving nearly-identical partitioning quality.

II. Problem Formulation

Given an undirected, connected and unweighted graph G, the set of vertices and edges of G are represented as $V(G)$ and $E(G)$ respectively. For a vertex $v \in V(G)$, N_v denotes the set of v's neighbors, and $|N_v|$ denotes the degree of v. A graph G' is called a *subgraph* of G if $V(G') \subset V(G)$ and $E(G') \subset E(G)$. A subgraph G' is an *induced subgraph* of G if $\forall\, u \in V(G), \forall\, v \in V(G)$ and $(u, v) \in E(G)$, we have $(u, v) \in E(G')$. A query motif Q is defined to be an (un)directed, unweighted and connected graph. Table I summarizes some frequently used notations in problem definitions and solutions. Our proposed methods can support both *undirected* and *directed* cases. In this paper, we present undirected cases for ease of illustration.

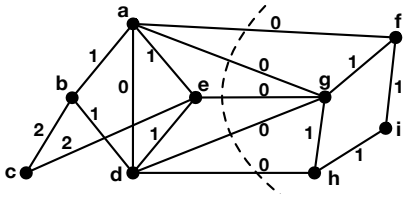


Fig. 1: A weighted graph based on the motif ⊐.

Definition 1 (Isomorphism)**.** *A graph G' is isomorphic to another graph G, denoted by $G' \simeq G$, if there is a bijection $f : V(G') \to V(G)$ such that $\forall u \in V(G)$, $\forall v \in V(G)$, $(u, v) \in E(G')$ if and only if $(f(u), f(v)) \in E(G)$.*

Definition 2 (Instance of a Motif)**.** *Given a graph G and a query motif graph Q, we call each induced subgraph of G that is isomorphic to Q as an instance of Q in G.*

Definition 3 (Motif-based Weighted Graph)**.** *Given a query motif Q and a graph G, we compute a motif-based weighted graph of G such that the weight w_e of each edge e represents the number of instances of Q in G that contain e.*

Definition 4 (Graph Partition)**.** *A partitioning plan ϕ_k of a graph G is a division of $V(G)$ into k disjoint vertex sets φ_i $(1 \le i \le k)$ such that $\cup_{i=1}^{k} \varphi_i = V(G)$. The partition ID assigned to a vertex v is denoted as $\phi_k(v)$. The size of a partition φ_i is denoted by $|\varphi_i|$ (i.e., the number of vertices assigned to φ_i).*

Definition 5 (Motif-based Edge-Cut Weight)**.** *Given a partitioning plan ϕ_k of a motif-based weighted graph G, the motif-based edge-cut weight of ϕ_k is the sum of weights of the edges crossing the partitions, i.e., $Cut(\phi_k) = \sum_{(u,v)\in E(G), \phi_k(u) \neq \phi_k(v)} w_{(u,v)}$.*

Definition 6 (Motif-based Graph Partition)**.** *Given a query motif Q and a graph G, motif-based graph partition (*MGP*) finds a partitioning plan ϕ_k that minimizes the motif-based edge-cut weight, i.e., $\min_{\phi_k} Cut(\phi_k)$.*

There are two popular metrics for MGP's quality evaluation: modularity [10] and conductance [11]. Modularity measures the density of edges inside partitions compared to crossing edges. Conductance measures the ratio of edge-cut weight over the sum of weights of edges in the smallest partition. It is not hard to see that edge-cut weight is a central factor in these objectives above. Thus, we design our approaches with the goal of minimizing the edge-cut weight such that our approaches can easily work with multiple GP kernels with different objective functions.

Example 1. *Figure 1 shows an example of how the exact edge weights are computed, and an ideal partitioning result based on the query motif ⊐. The weight of an edge e indicates how many instances of ⊐ contain e. Here, the weight of (b, c) is 2 since it is contained in two instances of the query motif formed by vertices $\{a, b, c, e\}$ and $\{b, c, d, e\}$ respectively. The weight of some edges like (a, f) and (a, g) is 0 as they are not contained in any induced subgraphs isomorphic to ⊐. If all*

vertices are partitioned into two sets while preserving motif-based communities, the minimum sum of cross-partition edge weights is 0, as shown in Figure 1.

III. Sampling-based Estimation

In this section we first present an edge-centric sampling method called SMGP to estimate the edge weights, and then discuss the theoretical guarantee of SMGP.

A. Edge-centric Sampling

To estimate the edge weights in G, we need to compute how many instances of a given query motif Q contain any edge. A straightforward way is to sample many induced subgraphs of size $|V(Q)|$ uniformly, check whether they are instances of Q, and then update the edge weights if so. However, it is not efficient because the sampled subgraphs may be unconnected, and an extremely large number of samples are needed for accurate estimations. Inspired by a sampling approach for estimating subgraph concentrations [14], we devise an edge-centric sampling method SMGP. SMGP works by uniformly picking a starting edge (v_1, v_2) and then sampling the connected subgraphs expanded from the starting edge. In what follows, we will introduce how we conduct edge-centric sampling via a process called *subgraph expansion*, to compute the subgraph sampling probabilities and finally estimate the edge weights.

Subgraph Expansion. We perform the subgraph expansion process by adopting a randomized procedure [14], [15], [16]. At the initial stage, we uniformly select a starting edge as an induced subgraph of size two. Then we incrementally expand the size of the current induced subgraph S by adding a randomly selected vertex v in G as well as the edges that connect v and the vertices in S. We stop the expansion process once the size of S reaches $|V(Q)|$.

Subgraph Sampling Probability. Given a query motif Q and a graph G, for each sampled subgraph of size $|V(Q)|$ in G, we have a corresponding order of vertices $[v_1, v_2, ..., v_{|V(Q)|}]$ in the expansion procedure. Suppose we have a sampled subgraph S_i of size $i < |V(Q)|$ that is formed by the ordered vertices $[v_1, v_2, ..., v_i]$, then the probability of sampling the next vertex v_{i+1} is:

$$\mathbb{P}[v_{i+1}|S_i] = \frac{|E(S_{i+1})| - |E(S_i)|}{\sum_{j=1}^{i} |N_{v_j}| - 2|E(S_i)|} \tag{1}$$

As v_{i+1} is uniformly sampled from the neighborhood of S_i, the probability of sampling an induced subgraph $S_{|V(Q)|}$ is:

$$\pi(S_{|V(Q)|}) = \frac{1}{|E(G)|} \prod_{i=2}^{|V(Q)|-1} \mathbb{P}[v_{i+1}|S_i] \tag{2}$$

where $\frac{1}{|E(G)|}$ is the sampling probability of the edge (v_1, v_2).

Edge Weight Estimation. To this end, suppose we have sampled θ connected and induced subgraphs $I_1, I_2, ..., I_\theta$ through the above subgraph expansion process, then for any edge $e \in E(G)$, its weight can be estimated with the Horvitz-Thompson inverse probability weighting [17] as below:

$$w_e = \frac{1}{\theta} \sum_{j=1}^{\theta} \frac{\mathbb{1}(e \in E(I_j)) \cdot \mathbb{1}(I_j \simeq Q)}{|\mathcal{O}(I_j)| \cdot \pi(I_j)} \tag{3}$$

Here, $\mathbb{1}$ refers to the indicator function, $|\mathcal{O}(I_j)|$ refers to the number of possible orderings of vertices that can lead to I_j, and $\pi(\cdot)$ refers to the subgraph sampling distribution. As long as π is supported over all possible (connected) induced subgraphs of size $|V_Q|$ such that the expected value is the true edge weight, the estimator (i.e., Equation 3) is unbiased. Note that $|\mathcal{O}(I_j)|$ is only dependent on the query motif Q and can be precomputed by dynamic programming.

To extend the method to directed graphs, we only need to consider the directions of the final induced subgraphs for the subgraph expansion process.

B. Theoretical Analysis of SMGP

Intuitively, when the number of sampled subgraphs (aka. the sample size θ) is sufficiently large, we can get an approximation such that each estimated edge weight is close to its exact value. In such a case, we can feed the graph with the estimated weights to a GP kernel that produces a competitive partitioning result, compared with the one using exact edge weights. In this section, we will show how the estimated weights and the partitioning quality converge to their expected values as the sample size θ grows.

Lemma 1. *The Hoeffding's inequality [18]. Let $X_1, ..., X_\theta$ be θ independent and bounded random variables ($a_i \leq X_i \leq b_i$) and $\widetilde{X} = \frac{1}{\theta}\sum_{i=1}^{\theta} X_i$. For any $\epsilon \geq 0$, we have*

$$\mathbb{P}[|\widetilde{X} - \mathbb{E}[\widetilde{X}]| \geq \epsilon] \leq 2\exp(-\frac{2\theta^2\epsilon^2}{\sum_{i=1}^{\theta}(b_i - a_i)^2}).$$

Theorem 1. *Given the weights $X_1, ..., X_\theta$ estimated for an edge e where $X_i = \frac{\mathbb{1}(e \in E(I_i)) \cdot \mathbb{1}(I_i \simeq Q)}{|\mathcal{O}(I_i)| \cdot \pi(I_i)}$, the sample size θ, and the upper-bound η of the estimated weights, then the deviation of the average of the estimated weights of e from $\mathbb{E}[w_e]$ is bounded with a probability increased w.r.t. θ, such that for any $\epsilon > 0$,*

$$\mathbb{P}[|w_e - \mathbb{E}[w_e]| \geq \epsilon] \leq 2\exp(-\frac{2\theta\epsilon^2}{\eta^2})$$

where $w_e = \frac{1}{\theta}\sum_{i=1}^{\theta} X_i$.

Proof. Since each induced subgraph of size $|V(Q)|$ is sampled independently, we can apply Lemma 1 to give a concentration bound on estimating w_e (Equation 3) if we can bound the value of $X_i = \frac{\mathbb{1}(e \in E(I_i)) \cdot \mathbb{1}(I_i \simeq Q)}{|\mathcal{O}(I_i)| \cdot \pi(I_i)}$. The lower-bound of X_i is 0 if the sampled induced subgraph I_i is not isomorphic to the query motif. In order to decide the upper-bound, we calculate the lowest possible probability of sampling an isomorphic

subgraph, and it is decided by $\pi(\cdot)$ (as defined in Equation 2) that satisfies the following inequality:

$$\pi(S_{|V(Q)|}) = \frac{1}{|E(G)|}\prod_{i=2}^{|V(Q)|-1}\frac{|E(S_{i+1})|-|E(S_i)|}{\sum_{j=1}^{i}|N_{v_j}|-2|E(S_i)|} > \frac{1}{|E(G)|}\cdot\frac{1}{\prod_{i=2}^{|V(Q)|-1}\sum_{j=1}^{i}|N_{v_j}|}$$

Clearly, $\pi(S_{|V(Q)|})$ is lower bounded in a connected graph, which derives that $\frac{1}{\pi(\cdot)}$ is upper bounded. We use η to denote an upper bound on $\frac{1}{\pi(\cdot)}$. Furthermore, each weight estimation is independent and bounded. Based on the Hoeffding's inequality, Theorem 1 is deduced. □

Next, we study the impact of the sample size θ on the quality of partitions. Note that the sum of weights of all edges is fixed as a constant. Thus, the problem of minimizing the edge-cut weight can be transformed to the problem of maximizing the sum of weights of non-crossing edges. Given a partitioning plan ϕ_k, let $z(\phi_k)$ and $\overline{z}(\phi_k)$ denote the sum of weights of non-crossing edges based on our estimated edge weights and the exact edge weights, respectively. We have the following Lemma to establish the convergence.

Lemma 2. *$z(\phi_k) \to \overline{z}(\phi_k)$ as $\theta \to \infty$.*

Proof. By Theorem 1, for any edge $(u,v) \in E(G)$, when $\theta \to \infty$, we have $w_{(u,v)} \to \overline{w}_{(u,v)}$ where $\overline{w}_{(u,v)}$ is the exact weight value. Then the following holds:

$$z(\phi_k) = \sum_{\substack{\phi_k(u)=\phi_k(v)\\(u,v)\in E(G)}} w_{(u,v)} \quad \to \sum_{\substack{\phi_k(u)=\phi_k(v)\\(u,v)\in E(G)}} \overline{w}_{(u,v)} = \overline{z}(\phi_k) \tag{4}$$

The Lemma is thus deduced. □

In the next step, we analyze the error between $z(\phi_k)$ and $\overline{z}(\phi_k)$ w.r.t. the sample size θ.

Lemma 3. *Let OPT be the optimal objective value for $\overline{z}(\cdot)$ among all partitioning plans. For a constant $\epsilon \in [0,1]$, we have the following inequality $|z(\phi_k)-\overline{z}(\phi_k)| \le \epsilon \cdot OPT$ hold for any partitioning plan ϕ_k with at least a probability of $1-\delta$, when the sample size θ satisfies:*

$$\theta \ge \frac{\eta^2 \log(\frac{2}{\delta})}{2\epsilon^2 OPT^2}$$

Proof. According to Theorem 1 and Equation 4, we have

$$\mathbb{P}[|z(\phi_k)-\overline{z}(\phi_k)| \ge \epsilon \cdot OPT] \le 2\exp(-\frac{2\theta\epsilon^2 OPT^2}{\eta^2})$$

The Lemma can be deduced by substituting $\theta = \frac{\eta^2\log(\frac{2}{\delta})}{2\epsilon^2 OPT^2}$. □

Theorem 2. *Given any partitioning algorithm that can achieve an approximation ratio of α $(0 < \alpha < 1)$ for maximizing the sum of weights of non-crossing edges based on the objective function $\overline{z}(\cdot)$, applying this algorithm to* SMGP *can result in an $(\alpha-2\epsilon)$-approximate solution with at least a probability of $1-\delta$ when $\theta \ge \frac{\eta^2\log(\frac{1}{\delta})}{2\epsilon^2 OPT^2}$.*

Proof. Let ϕ_k^* denote the optimal partitioning plan on the graph with exact edge weights and $\overline{z}(\phi_k^*) = OPT$. Assuming SMGP returns a partitioning plan ϕ_k with θ samples and let ϕ_k^s be the optimal partitioning plan on the graph with estimated edge weights. According to Lemma 3, we know that $|z(\phi_k)-\overline{z}(\phi_k)| \le \epsilon \cdot OPT$ holds with at least a probability of $1-\delta$ for all possible partitioning plans. Subsequently, we have:

$$\begin{aligned} z(\phi_k) &\ge \alpha z(\phi_k^s) \ge \alpha z(\phi_k^*),\\ \text{and } \overline{z}(\phi_k) &\ge z(\phi_k) - \epsilon\cdot OPT\\ &\ge \alpha z(\phi_k^s) - \epsilon\cdot OPT\\ &\ge \alpha z(\phi_k^*) - \epsilon\cdot OPT. \qquad (5)\\ \text{Since } z(\phi_k^*) &\ge z(\phi_k^*) - \epsilon\cdot OPT,\\ \text{then } \overline{z}(\phi_k) &\ge \alpha(1-\epsilon)OPT - \epsilon\cdot OPT\\ &> (\alpha-2\epsilon)OPT. \end{aligned}$$

Thus, the theorem is established. □

Algorithm 1: SMGP+

Input : Graph G, motif Q, the number of iteration t and partitions k, a GP kernel M, the sample size θ.
Output: A partitioning plan ϕ_k.

1 $\widetilde{\pi} \leftarrow$ the uniform sampling distribution of starting edges;
2 **for** $j = 1$ *to* t **do**
3 $\quad w^j_{(\cdot,\cdot)} \leftarrow AdaptiveSampling(G, Q, \widetilde{\pi}, \theta)$;
4 $\quad$ **for** *each edge* $(u,v) \in E(G)$ **do**
5 $\quad\quad w^{avg}_{(u,v)} \leftarrow \frac{1}{j}\sum_{i=1}^{j} w^i_{(u,v)}$;
6 $\quad \phi_k \leftarrow M(G, w^{avg}_{(\cdot,\cdot)}, k)$;
7 $\quad \widetilde{\pi} \leftarrow UpdateDistribution(G, w^{avg}_{(\cdot,\cdot)}, \phi_k, \widetilde{\pi})$;
8 **return** ϕ_k;

IV. An Adaptive Sampling Framework

While SMGP can deliver accurate partition results, it needs to sample an excessive number of subgraphs to ensure that the weight for each edge is estimated with a small error. For the MGP problem, we observe that not every single edge plays a critical role in affecting the partition results, and the samples dedicated to these non-significant edges are largely wasted. Motivated by this, we propose an adaptive SMGP framework, namely SMGP+, which can dynamically adjust the sampling distribution of the starting edge to improve the partitioning quality while maintaining the same sample size.

A. An Overview of SMGP+

Algorithm 1 presents an overview of the SMGP+ framework that works by iteratively partitioning the original graph. In each iteration, we sample θ subgraphs according to a sampling distribution $\widetilde{\pi}$ for the starting edge (line 3). The estimated weights for iteration j, i.e., $w^j_{(\cdot,\cdot)}$, will be aggregated with the weights estimated from previous iterations to produce the up-to-date estimations $w^{avg}_{(\cdot,\cdot)}$ (line 5). The weight estimator averaged across all t iterations will be close to its expected

value with a small error as the number of iteration increases. We will show the convergence rate of $w^{avg}_{(u,v)}$ in Section IV-C. Subsequently, a GP kernel is invoked to partition G w.r.t. the weights $w^{avg}_{(\cdot,\cdot)}$ and generate a partitioning plan ϕ_k (line 6). Based on ϕ_k and $w^{avg}_{(\cdot,\cdot)}$, we will update the sampling distribution $\widetilde{\pi}$ to prioritize accurate weight estimations for those edges that could significantly affect the partitioning result. In this way, given a fixed sample size, we can put more focus on sampling "important" edges to enable a better partitioning result. We will present our approach of updating the sampling distribution $\widetilde{\pi}$ in Section IV-B. Since these subgraphs are not sampled independently, we show a theoretical guarantee of SMGP+ by exploiting the concept of Martingale in Section IV-C.

B. Updating the Sampling Distribution

We update the existing sampling distribution $\widetilde{\pi}$ with a normalization as below:

$$\widetilde{\pi}((u,v)) \leftarrow \widetilde{\pi}((u,v)) \cdot \mathsf{PF}[(u,v)] \cdot \mathsf{NF}[(u,v)] \qquad (6)$$

In Equation 6, we consider two factors in updating the sampling probability for (u,v): (1) the *Partition Factor* ($\mathsf{PF}[(u,v)]$) and (2) the *Neighborhood Factor* ($\mathsf{NF}[(u,v)]$).

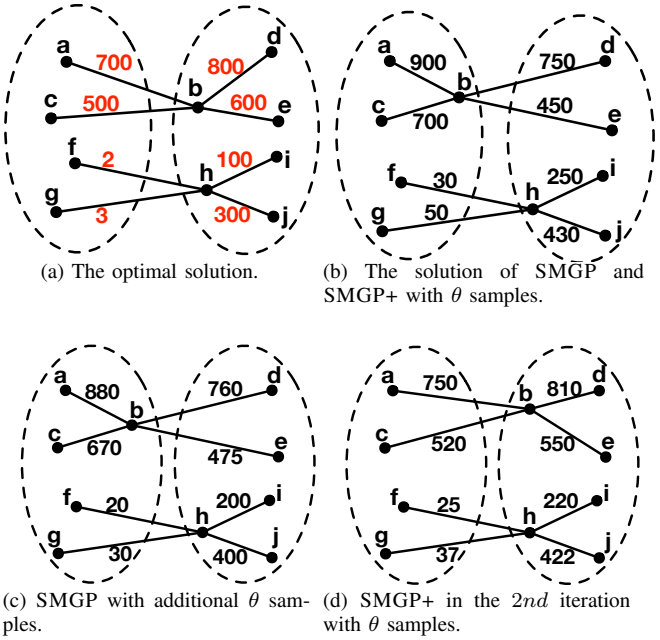
Partition Factor. We set a higher partition factor to the edge (u,v), if the inaccurate weight estimation of this edge can easily change the current partitioning plan on placing this edge. Let $\mathcal{C}(u,\varphi_i)$ denote the set of edges connecting the vertex u to vertices in the partition φ_i. If $i = \phi_k(u)$, then $\mathcal{C}(u,\varphi_i)$ contains all non-crossing edges incident on u. Otherwise, it contains all crossing edges connecting u to vertices in a different partition φ_i. Let $|\mathcal{C}(u,\varphi_i)|$ denote the sum of the weight of the edges in $\mathcal{C}(u,\varphi_i)$. The following equation measures how well u is connected to vertices in the same partition, as compared to those in a different partition:

$$\xi(u,\varphi_i) = |\mathcal{C}(u,\varphi_j)| - |\mathcal{C}(u,\varphi_i)|$$

where $j = \phi_k(u)$ and $i \neq j$.

We assume that the underlying GP kernel is effective such that $\xi(u,\varphi_i)$ is positive. If $\xi(u,\varphi_i)$ is negative, we can always move u to φ_i and reduce the sum of weights of the crossing edges. When $\xi(u,\varphi_i)$ is relatively small or close to zero, it is very likely that the inaccurate weight estimation of edges in $\mathcal{C}(u,\varphi_i)$ or $\mathcal{C}(u,\varphi_j)$ leads to a negative value of $\xi(u,\varphi_i)$. Consequently, the vertex u needs to be moved from φ_j to φ_i to reduce the edge-cut weight (i.e., to optimize for a positive $\xi(u,\varphi_i)$ and improve the partitioning objective value). The consequence caused by inaccurate weight estimations of such edges can potentially trigger a domino effect that changes the entire partitioning plan. On the contrary, if $\xi(u,\varphi_i)$ is estimated as a large positive value, then the exact value of $\xi(u,\varphi_i)$ is likely positive even when the weight estimations of involved edges are not accurate.

Example 2. *Figure 2 shows an example on how the* partition factor *is used to improve the partitioning quality, where we only show a small part of the whole graph for easy illustration.*



(a) The optimal solution.

(b) The solution of SMGP and SMGP+ with θ samples.

(c) SMGP with additional θ samples.

(d) SMGP+ in the $2nd$ iteration with θ samples.

Fig. 2: An example of using the partition factor.

Figure 2(a) refers to the optimal partition result based on the exact edge weights. When we set the sample size as θ, both SMGP *and* SMGP+ *uniformly sample θ starting edges for weight estimation and may produce the partition result as shown in Figure 2(b). In this result, the sum of weights of edges (a,b) and (b,c) is close to the sum of weights of edges (b,d) and (b,e). The inaccurate weight estimation of these edges with high partition factors can easily influence the following decision: which partition should the vertex b be placed into, such that the edge-cut weight can be reduced?*

On the other hand, the sum of weights of (f,h) and (g,h) is much smaller than that of (h,i) and (h,j). More accurate weight estimations of these edges with low partition factors are unlikely to change the relationship between the sum of weights of these two pairs of edges. Thus, in the next iteration, SMGP+ *will assign (a,b), (b,c), (b,d) and (b,e) with higher sampling probabilities to compute more accurate weight estimations for them, and thereby produce a better partition result, as shown in Figure 2(d). On the contrary,* SMGP *still adopts the uniform sampling strategy and may fail to improve the partitioning quality even with a larger sample size as shown in Figure 2(c).*

Motivated by the above intuition, for edges whose inaccurate weight estimations have great impacts on the partitioning result, we need to sample them as the starting edges with high probabilities, such that the partitioning result can be stabilized with more accurate edge weights. When we try to define the partition factor, several aspects need to be considered:

1) For a crossing edge (u,v) where $\phi_k(u) = j, \phi_k(v) = i$, the values of $\xi(u,\varphi_i)$ and $\xi(v,\varphi_j)$ are different. It indicates that the inaccurate weight estimation for (u,v) has different levels of impact on movements of vertices u and v. Thus, the partition factor of (u,v) is different

when analyzing vertex u and vertex v individually. Since we need to make sure the weight estimation of (u, v) is accurate enough such that neither of these two vertices needs to be moved, a larger partition factor between u and v should be used as the final partition factor of (u, v) (Equation 7).

2) Similarly, for a non-crossing edge (u, v) where $\phi_k(u) = \phi_k(v) = j$, its estimated weight is involved in every pair of $\xi(u, \varphi_i)$ $(1 \leq i \leq k, i \neq j)$. That means the inaccurate estimation for (u, v) has a different influence on which partition φ_i we should move u into. Thus, we also consider the smallest $\xi(u, \cdot)$ among all pairs to compute the partition factor of a non-crossing edge (u, v) (Equation 8).

By accounting for the above two aspects, we devise the partition factor $\mathsf{PF}[(u, v)]$ of an edge (u, v) as follows:

$$\mathsf{PF}[(u,v)] = max(\mathsf{PF}_u[(u,v)], \mathsf{PF}_v[(u,v)]) \tag{7}$$

where $\mathsf{PF}_u[(u, v)]$ and $\mathsf{PF}_v[(u, v)]$ are defined symmetrically:

$$\mathsf{PF}_u[(u,v)] = \begin{cases} \exp(-\frac{\xi(u,\varphi_i)}{d_u} + 1), & \text{if } \phi_k(u) \neq \phi_k(v) = i \\ \exp(-\frac{\xi_{min}(u)}{d_u} + 1), & \text{otherwise} \end{cases}$$

$$\text{and } \xi_{min}(u) = \min_{\phi_k(u)=j, 1\leq i\leq k, i\neq j} |\mathcal{C}(u,\varphi_j)| - |\mathcal{C}(u,\varphi_i)| \tag{8}$$

where d_u is the weighted degree of u.

Neighborhood Factor. It is worth mentioning that there could be a notably higher number of motif instances discovered in the dense region of the graph than those discovered in the sparse region. The different number of motif instances results in weight difference, and higher weights have a significant impact on the partition result. An edge may appear in either a sparse or a dense region of the graph. Hence we penalize an edge e if it appears in a sparse region and lowers the sampling probability. To achieve this goal, we look at the neighborhood of an edge $e = (u, v)$ and use the number of common neighbors of u and v to indicate the 'density' of the neighborhood (i.e., $|N_u \cup N_v|$). Additionally, it is possible that the number of common neighbors of high-degree nodes is significantly larger than that of the remaining nodes in power-law graphs. To ensure that each edge is assigned with a reasonable sampling probability and outliers are penalized, we apply the log function in $\mathsf{NF[(u,v)]}$ as: $\mathsf{NF[(u,v)]} = \log |N_u \cup N_v|$.

C. Theoretical Analysis

In contrast to the theoretical analysis of SMGP (in Section III-B), the samples in SMGP+ are not independently generated due to the adaptive sampling strategy devised. Thus, the Hoeffdding inequality is inapplicable to SMGP+. Instead, we propose to employ the martingale [13], a classical statistic tool, to restore the theoretical guarantee of SMGP+.

Lemma 4. ***Martingale*** *[13]. A sequence of random variables $Z_0, Z_1, ..., Z_t$ (possibly correlated) is a martingale if the following conditions hold for all $0 \leq i \leq t$: (1) $\mathbb{E}[|Z_i|] < \infty$; (2) $\mathbb{E}[Z_{i+1}|Z_1, Z_2, ..., Z_i] = Z_i$.*

Lemma 5. *A sequence of random variables $R_1, R_2, ..., R_t$ is a martingale where $R_i = \sum_{j=1}^{i}(w^j_{(u,v)} - w_{(u,v)})$ and $1 \leq i \leq t$.*

Proof. In SMGP+, we iteratively update the weight estimation $w^i_{(u,v)}$. To connect $w^i_{(u,v)}$ to martingales, we have:

$$w_{(u,v)} = \frac{1}{t}\mathbb{E}[\sum_{i=1}^{t} w^i_{(u,v)}] \tag{9}$$

Furthermore, only the starting edge's sampling distributions $\widetilde{\pi}_1, \widetilde{\pi}_2, ..., \widetilde{\pi}_t$ in each iteration are correlated. Given the query motif Q, $\widetilde{\pi}_i (1 \leq i \leq t)$ is supported over all edges and thus unbiased, and the subgraphs of size $|V(G)|$ expanded from the starting edges are sampled uniformly. Thereby, for any $i \in [1, t]$, we have

$$\mathbb{E}[w^i_{(u,v)}|w^1_{(u,v)}, w^2_{(u,v)}, ..., w^{i-1}_{(u,v)}] = \mathbb{E}[w^i_{(u,v)}] = w_{(u,v)}.$$

Let $R_i = \sum_{j=1}^{i}(w^j_{(u,v)} - w_{(u,v)})$, we have $\mathbb{E}[R_i] = 0$ and

$$\mathbb{E}[R_i|R_1, R_2, ..., R_{i-1}] = R_{i-1}.$$

Therefore, $R_1, ..., R_t$ is a martingale based on Lemma 4. □

Lemma 6. *The martingale $R_1, ..., R_t$ satisfies the three properties: (1) $|R_1| < \eta$, (2) $|R_i - R_{i-1}| < \eta$ $(2 \leq i \leq t)$ and (3) $Var[R_1] + \sum_{i=2}^{t} Var[R_i|R_1, R_2, ..., R_{i-1}] = \sum_{i=1}^{t} Var[R_i] < t\eta^2$.*

Proof. Recall the definition of R_i, we can show that $R_1, R_2, ..., R_h$ have the properties above by showing: $|w^i_{(u,v)}|$ can be bounded by the same constant and the (conditional) variance of each $w^i_{(u,v)}$ can be bounded as well.

Based on our analysis in Section III, the lower-bound of each $w^i_{(u,v)}$ is 0 and we can find the upper-bound for each $w^i_{(u,v)}$ as η. Thus, we have $|R_1| < \eta$ and $|R_i - R_{i-1}| < \eta$ $(2 \leq i \leq t)$ by the definition of R_i. This in turn derives the following: $Var[R_1] + \sum_{i=2}^{t} Var[R_i|R_1, ..., R_{i-1}] = \sum_{i=1}^{t} Var[R_i] < t\eta^2$ □

Lemma 7. ***Concentration bound*** *[13]. Let $Z_1, Z_2, ..., Z_t$ be a martingale which satisfies the properties that (1) $|Z_1| \leq a$, (2) $|Z_i - Z_{i-1}| \leq a$ $(2 \leq i \leq t)$, and (3) $Var[Z_1] + \sum_{i=2}^{t} Var[Z_i|Z_1, Z_2, ..., Z_{i-1}] \leq b$, where $Var[\cdot]$ refers to the variance of a random variable. Then for any β,*

$$\mathbb{P}(Z_i - \mathbb{E}[Z_i] \geq \beta) \leq \exp(-\frac{\beta^2}{\frac{2}{3}a\beta + 2b}).$$

Theorem 3. *Given the edge weights $w^1_{(u,v)}, w^2_{(u,v)}, ..., w^t_{(u,v)}$ computed in t iterations based on the query motif Q, and the maximum possible estimated edge weight η, the deviation of the averaged edge weight of (u, v) from $\mathbb{E}[w_{(u,v)}]$ is bounded with a probability increased with t, such that, for any $\epsilon > 0$,*

$$\mathbb{P}\left[\left|\frac{1}{t}\sum_{i=1}^{t} w^i_{(u,v)} - \mathbb{E}[w_{(u,v)}]\right| \geq \epsilon\right] \leq \exp(-\frac{\epsilon^2}{\frac{2}{3}\epsilon\eta + 2\eta^2}t).$$

Proof. Given the concentration bound of martingale (Lemma 7), $R_i = \sum_{j=1}^{i}(w^j_{(u,v)} - w_{(u,v)})$ and $\mathbb{E}[R_i] = 0$, for any $\epsilon > 0$, we have

$$\mathbb{P}\left[\sum_{i=1}^{t} w^i_{(u,v)} - t\mathbb{E}[w_{(u,v)}\right] \geq \epsilon t] \leq \exp(-\frac{\epsilon^2 t^2}{\frac{2}{3}\epsilon t\eta + 2t\eta^2}).$$

Similarly, by applying the concentration bound on the martingale $-R_1$,$-R_2$,...,$-R_t$, we have

$$\mathbb{P}\left[\sum_{i=1}^{t} w^i_{(u,v)} - t\mathbb{E}[w_{(u,v)}\right] \leq -\epsilon t] \leq \exp(-\frac{\epsilon^2 t^2}{\frac{2}{3}\epsilon t\eta + 2t\eta^2}).$$

Thus, this theorem is deduced. □

By following the arguments made in Lemmas 2-3 and Theorem 2 for SMGP, Theorem 3 naturally leads to Theorem 4.

Given a partitioning plan ϕ_k, let $z(\phi_k)$ and $\overline{z}(\phi_k)$ denote the weighted sum of non-crossing edges based on the edge weights estimate by SMGP+ and the ground-truth edge weights respectively. We have the following Lemma to establish the convergence.

Lemma 8. *$z(\phi_k) \to \overline{z}(\phi_k)$ as $k \to \infty$.*

Proof. We can deduce this Lemma based on Theorem 3 and similar argument made in Lemma 2. □

Lemma 9. *Let OPT be the optimal objective value for $\overline{z}(\cdot)$ among all partitioning plans. For a constant $0 \leq \epsilon \leq 1$, we have the inequality $|z(\phi_k) - \overline{z}(\phi_k)| \leq \epsilon \cdot OPT$ which holds for any partitioning plan ϕ_k with at least $1-\delta$ probability when the number of iteration $t \geq \frac{(\frac{2}{3}\epsilon\eta + 2\eta^2)\log(\frac{1}{\delta})}{\epsilon^2 OPT^2}$.*

Proof. According to Theorem 3 and Lemma 8, we have

$$\mathbb{P}[|z(\phi_k) - \overline{z}(\phi_k)| \geq \epsilon \cdot OPT] \leq \exp(-\frac{\epsilon^2 t OPT^2}{\frac{2}{3}\epsilon\eta + 2\eta^2}).$$

We can deduce the Lemma with $\delta = \exp(-\frac{\epsilon^2 t OPT^2}{\frac{2}{3}\epsilon\eta + 2\eta^2})$. □

Theorem 4. *Given any α-approximate $(0 < \alpha < 1)$ partitioning algorithm for maximizing the weighted sum of non-crossing edges defined based on the objective function $z(\cdot)$, applying this partitioning algorithm to* SMGP+ *can result in an $(\alpha - 2\epsilon)$-approximate solution with at least $1-\delta$ probability when $t \geq \frac{(\frac{2}{3}\epsilon\eta + 2\eta^2)\log(\frac{1}{\delta})}{\epsilon^2 OPT^2}$.*

Proof. Let ϕ^*_k denote the optimal partitioning plan and $\overline{z}(\phi^*_k) = OPT$. Assuming Algorithm 1 returns a partitioning plan ϕ_k with t iterations and let ϕ^s_k be the optimal partitioning plan on the graph with sampled edge weights. According to Lemma 9, we know that $|z(\phi_k) - \overline{z}(\phi_k)| \leq \epsilon \cdot OPT$ holds with at least $1-\delta$ probability for all possible partitioning plans. Subsequently, we have:

$$\begin{aligned} z(\phi_k) &\geq \alpha z(\phi^s_k) \geq \alpha z(\phi^*_k), \\ \text{and } \overline{z}(\phi_k) &\geq \alpha z(\phi^*_k) - \epsilon \cdot OPT. \\ \text{Since } z(\phi^*_k) &\geq \overline{z}(\phi^*_k) - \epsilon \cdot OPT, \\ \text{then } \overline{z}(\phi_k) &> (\alpha - 2\epsilon)OPT. \end{aligned} \tag{10}$$

Thus, Theorem 4 is established. □

TABLE II: The statistics of datasets

Datasets	Nodes	Edges	Average Degree
Epinion	75,879	405,740	10.69
Gowalla	196,591	950,327	9.66
Flixster	2,523,386	7,918,801	6.28
Digg	279,630	1,548,126	11.07
Dogster	426,820	8,543,549	40.03
Catster	149,700	5,448,197	72.79
Orkut	3,072,441	117,184,899	76.28

TABLE III: Statistics of EMGP

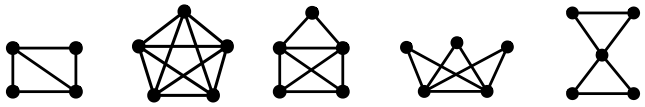
Dataset	Statistics	Query Motif				
		Q_1	Q_2	Q_3	Q_4	Q_5
Epinion	**Time (s)**	4.5E+02	3.3E+02	9.9E+03	2.6E+04	4.1E+04
	Count	7.8E+07	1.7E+07	1.1E+09	1.7E+09	5.7E+09
Gowalla	**Time (s)**	5.0E+02	3.0E+02	1.6E+04	3.2E+04	7.2E+04
	Count	8.6E+07	1.5E+07	1.1E+09	5.4E+09	7.3E+09
Flixster	**Time (s)**	1.6E+03	1.8E+03	5.5E+04	1.1E+05	1.8E+05
	Count	2.3E+08	9.6E+07	2.5E+09	5.2E+09	8.6E+09
Digg	**Time (s)**	1.2E+04	2.3E+04	\	\	\
	Count	2.3E+09	1.7E+09	\	\	\
Dogster	**Time (s)**	1.1E+05	2.9E+05	\	\	\
	Count	3.5E+10	1.2E+10	\	\	\
Catster	**Time (s)**	5.4E+05	\	\	\	\
	Count	2.0E+11	\	\	\	\
Orkut	**Time (s)**	2.8E+05	\	\	\	\
	Count	4.8E+10	\	\	\	\

TABLE IV: Average partitioning time across motifs (seconds).

Graph Kernel	Epinion	Gowalla	Flixster	Digg	Dogster	Catster	Orkut
Fennel	2	7	50	9	30	16	470
MAPPR	2	5	40	7	29	20	497

V. Experiment

In this section, we conduct experiments over seven real-world datasets to compare our sampling-based approaches with the exact MGP method. Please refer to appendix for more experiments.



(a) Q_1 (b) Q_2 (c) Q_3 (d) Q_4 (e) Q_5

Fig. 3: Query motifs used in the experiments.

Query motifs. We use five commonly used motifs in the experiment discussion (Figure 3). As reported in [19], [20], [21], [22], these motifs are discovered in many real-wold networks across different categories. Our proposed methods can support more complex motifs but we present the above queries which the exact method can handle.

Datasets. Seven real-world datasets, i.e., Gowalla, YouTube, Flixster, Digg, Dogster, Catster and Orkut [23], are used. They range from small-scale to large-scale and from sparse to dense graphs. Table II summarizes the statistics of each dataset.

Methods for comparison. Since this is the first work to solve the MGP problem without employing exact edge weight computations, we compare our sampling methods SMGP and SMGP+ with EMGP [24], the state-of-the-art exact motif counting method. To extend EMGP, we generate a subgraph

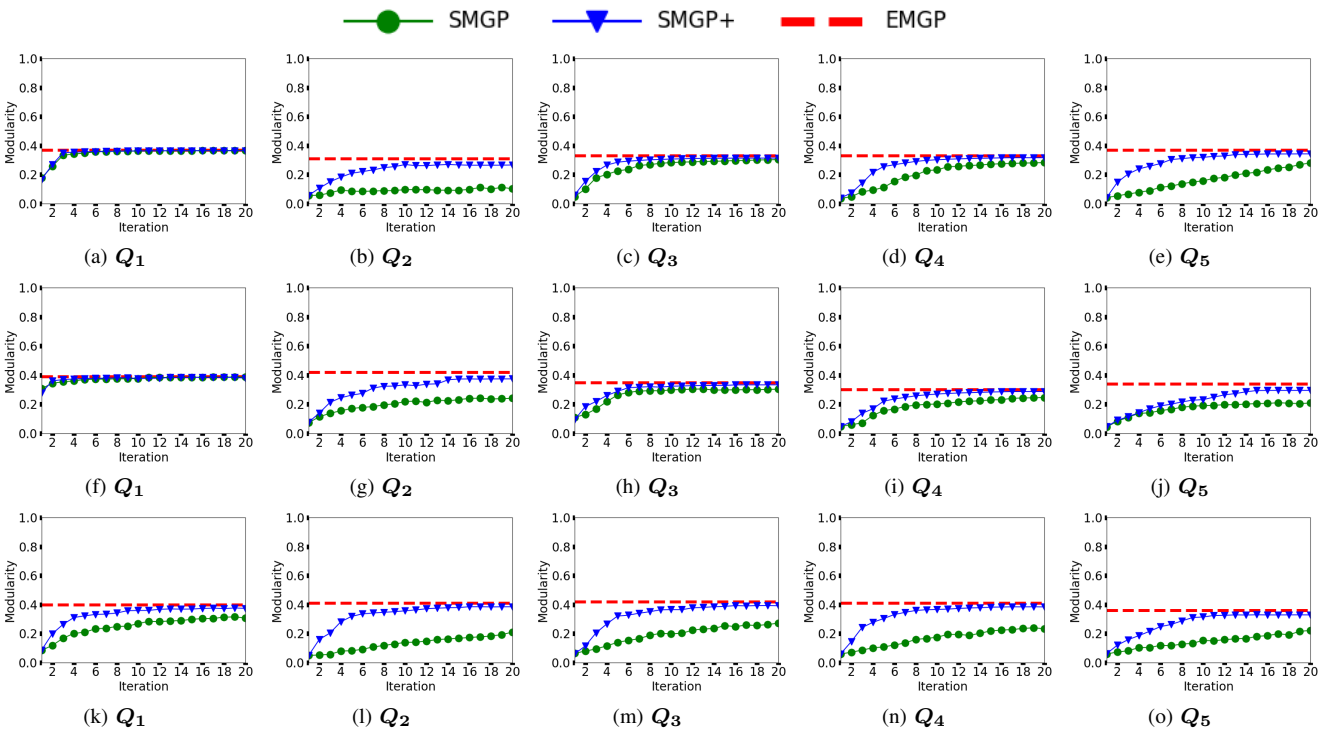


Fig. 4: Effectiveness comparison on Epinion (1st row), Gowalla (2nd row) and Flixster (3rd row) with `Fennel` as the kernel.

instance by only considering an induced subgraph during the enumeration process and feed the instance to the pruning conditions of EMGP. The exact motif count and runtime of EMGP are reported in Table III. The counting statistics of some motifs on large dense graphs are not reported as they cannot be computed by EMGP *within a week*.

Underlying GP kernels and evaluation metrics. To show that our methods are robust to work with different kernels and objective functions for MGP, we use two widely-adopted GP kernels with different evaluation metrics for experiments:

- `Fennel` [25]: it is a streaming GP kernel aiming for optimal k-graph partitioning, which admits an equivalent formulation of maximizing the *modularity* scores.
- `MAPPR` [6]: it aims to compute the partitioning results with the minimum *conductance*, by sweeping over a motif-based personalized PageRank vector of vertices.

Environments. We conduct all experiments on a Linux server with Intel Xeon E5 (2.60 GHz) CPUs and 512 GB RAM. All codes are implemented in Python and we use a state-of-the-art method [24] for the exact edge weight computation. For a fair comparison, all algorithms are executed with a single thread.

Parameter settings for GP kernels We set the number of partitions $k = 10$ for `Fennel` and set the load threshold parameter of `Fennel` as 5. Since `MAPPR` is a local clustering method, we follow the common standard [6] to set $k = 2$ for `MAPPR`. For all other parameters of these partitioning methods, we adopt the default setting as reported in the original papers.

Parameter settings for our methods. There are two important parameters in our methods: the number of iterations and the sampling size for each iteration. We run 20 iterations to study the convergence of SMGP and SMGP+. SMGP+ needs to invoke the GP kernel for each iteration to adjust the sampling distribution whereas SMGP does not. Note that the theoretical sample size could be large. We hence employ a heuristic method to determine the stopping sample criterion for SMGP+. We use multiple batches of samples for every iteration and the size of each batch will be illustrated in the next paragraph. We stop the sampling of the current iteration when the estimated difference ratio (i.e., $|b_i^c - b_{i-1}^c|/b_i^c$) is below a threshold 0.1 for two consecutive batches, where b_i^c denotes the estimated motif count of the first i batches. For a fair comparison, we assign SMGP with the same sample size as SMGP+ in each iteration.

Sampling batch size. For each test case, we sweep over different choices of batch sizes $\{2.5 \times 10^4, 7.5 \times 10^4, 1.25 \times 10^5, 1.75 \times 10^5\}$ and choose the smallest one that leads to high-quality results when SMGP+ converges. The convergence of SMGP+ will be further introduced in Section V-B.

A. *Effectiveness Evaluation*

When evaluating the quality of the partitioning results respectively produced by EMGP, SMGP and SMGP+, we adopt the evaluation metric corresponding to the objective function of the underlying GP kernel.

Modularity evaluation with Fennel. Figure 4 shows the modularity achieved by SMGP and SMGP+ on Epinion, Gowalla and Flixster, with `Fennel` as the GP kernel. A

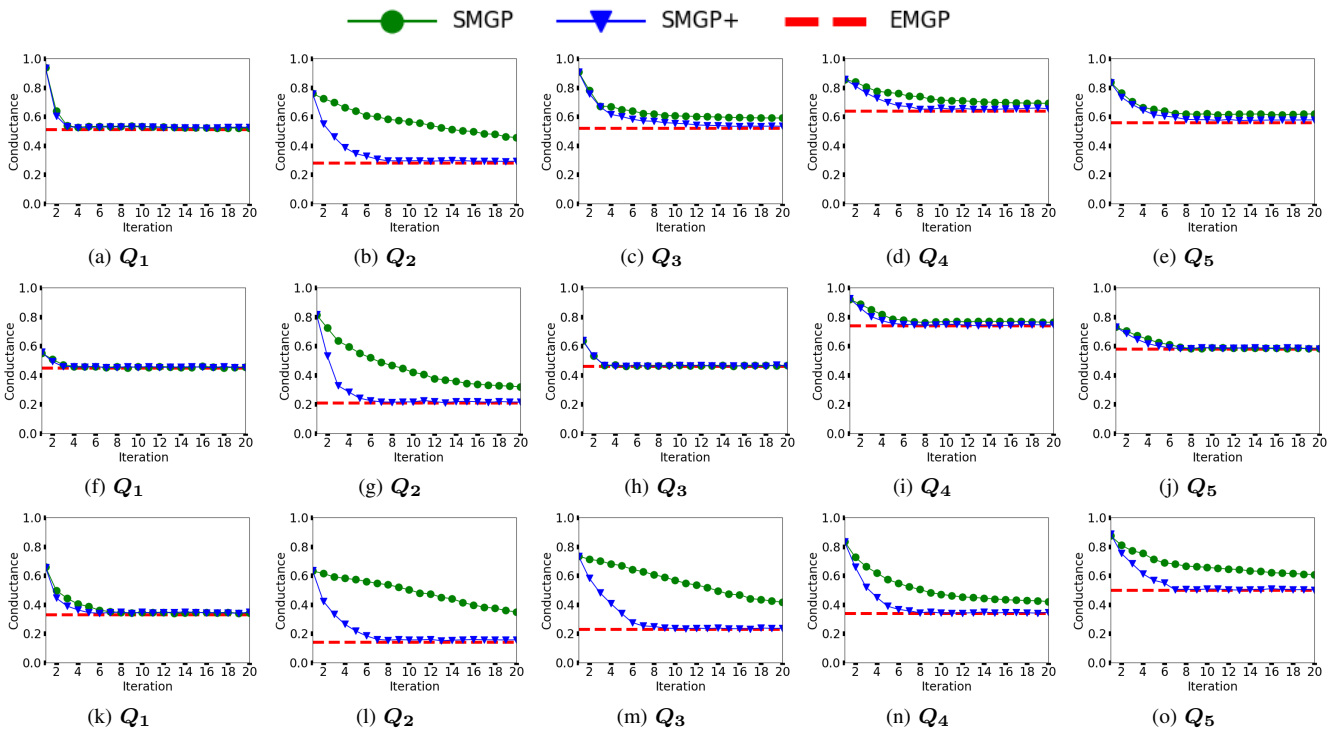


Fig. 5: Effectiveness comparison on Epinion (1st row), Gowalla (2nd row) and Flixster (3rd row) with MAPPR as the kernel.

higher modularity score indicates a better partitioning result. As the iteration increases, the modularity score achieved by SMGP+ grows and can be very competitive with EMGP when SMGP+ converges. In contrast, the modularity score achieved by SMGP generally grows much slower and in many cases, SMGP produces inferior results when it finishes all iterations (e.g., Figure 4(b),(l),(m) and (n)).

Conductance evaluation with MAPPR. Figure 5 compares the conductance achieved by different weight estimation methods, using MAPPR as the GP kernel. A lower conductance score indicates a better partitioning result. SMGP and SMGP+ show better performance and faster convergence here than using Fennel as the GP kernel in some cases (e.g., on Gowalla with motif Q_5). We suspect that MAPPR is less sensitive to the inaccurate edge weight estimation and thus smaller sample sizes are needed to achieve promising results. It is interesting to see that the conductance achieved by EMGP on different motifs varies significantly, which indicates the importance of choosing appropriate motifs for high-order social network analysis. Our methods are good alternatives to efficiently evaluate the potential of motifs w.r.t. the conductance without employing exact edge weight computations.

The performance difference of the query motifs in the same graph can be explained by but not limited to the following: (1) The number of motif instances are different. Suppose the motif 4-clique has more instances than other motifs of size 4 in the graph, it will be easier to sample 4-clique than any other motifs of size 4. (2) The distributions of motifs could be different. For instance, it may be easier to find a triangle subgraph in the neighborhood of a node with low degrees than finding a 5-clique. Thus, the accuracy of estimation is data dependent.

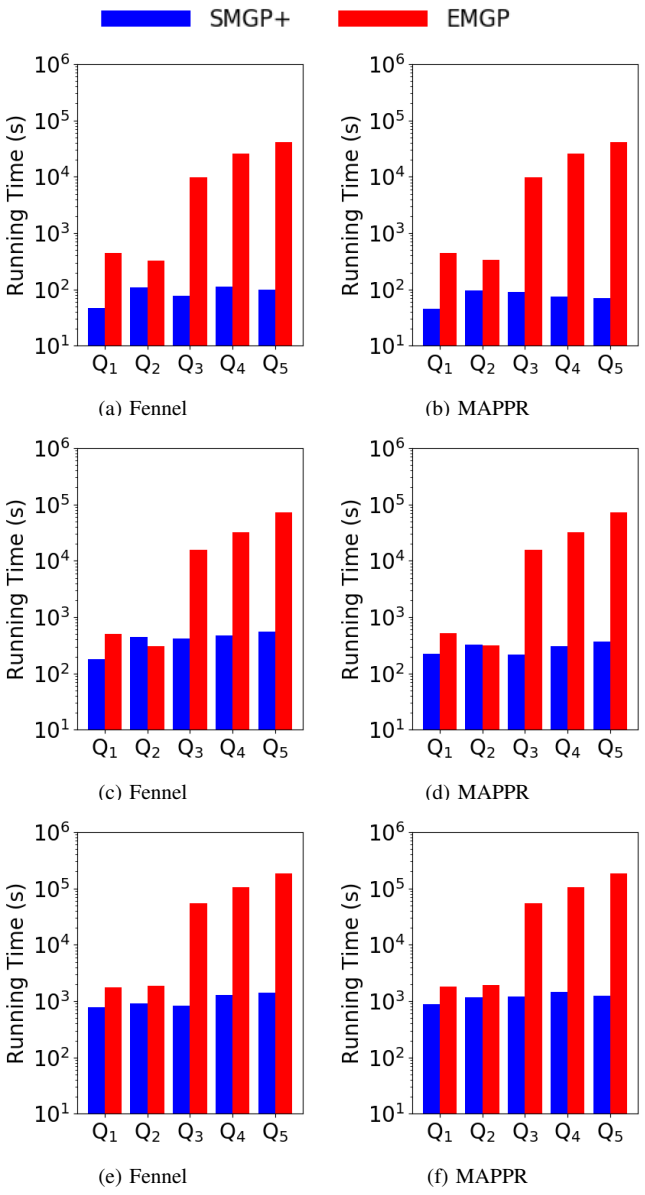
B. *Efficiency Evaluation*

Figure 6 compares the runtime of SMGP+ and EMGP with different GP kernels. We do not include SMGP in this comparison since SMGP may not even produce a good partitioning result after the maximum iteration, as shown in Section V-A. On the other hand, SMGP+ can quickly converge to EMGP in a few iterations. For a fair comparison, we measure the runtime of SMGP+ when it converges.

The convergence of SMGP+. It is determined based on a stopping criterion of how many iterations SMGP+ actually needs to return promising and stable solutions. Suppose O_i is the objective score returned by the respective GP kernel at iteration i. We stop SMGP+ once the error difference (i.e., $|O_i - O_{i-1}|$) is below 0.01 for two consecutive iterations.

Figure 6 compares the runtime of SMGP+ and EMGP. In most cases, SMGP+ notably outperforms EMGP and can achieve up to two orders of magnitude speedup compared to EMGP (e.g., Q_4 and Q_5 in Figure 6(b)). The computational cost of EMGP is very sensitive to the pattern of the input motif. In contrast, the performance of SMGP+ is much more stable as it can avoid expensive weight estimations for edges that are not critical to produce high-quality partitioning results. The results confirm that SMGP+ can help existing MGP methods choose good motifs for high-order network analysis without costly trials via exact edge weight computations.

Let $\frac{|O_{\text{SMGP+}} - O_{\text{EMGP}}|}{O_{\text{EMGP}}}$ be the relative error ratio where $O_{\text{SMGP+}}$ is the conductance score when SMGP+ converges and O_{EMGP}



(a) Fennel (b) MAPPR

(c) Fennel (d) MAPPR

(e) Fennel (f) MAPPR

Fig. 6: Runtime comparison on Epinion (1st row), Gowalla (2nd row) and Flixster (3rd row).

is the exact conductance score achieved by EMGP with ground-truth edge weights. On Epinion, the average relative error ratio is 5% and the speedup ranges from 3x to 590x. On Gowalla, the average error ratio is 5% and the speedup ranges from 0.95x to 196x. On Flixster, the average error ratio is 6% and the speedup ranges from 2x to 147x. The results show that SMGP+ can achieve a good trade-off between efficiency and accuracy.

C. Scalability Evaluation on Dense Datasets

As shown in Table III, EMGP is only able to compute the exact edge weights for all denser datasets (i.e., Digg, Dogster, Catster and Orkut) with motif Q_1. Thus, we present the experiments on these datasets with motif Q_1 to demonstrate the scalability of SMGP+. Figure 7 and Figure 8 compare the effectiveness of SMGP and SMGP+ with motif Q_1. On the smaller datasets Digg, Dogster and Catster, SMGP is very competitive with SMGP+ but is notably outperformed by SMGP+ on the large dataset Orkut, which is consistent with our observation made in Section V-A. As shown in this Section and Section V-A, SMGP+ performs notably better than SMGP which may not produce satisfying results after the maximum iteration in many cases. Thus, we will focus on analyzing the scalability of SMGP+ on these datasets.

TABLE V: The speedup of SMGP+ with different k.

Dataset	Motif	k=10	k=20	k=30	k=40	k=50
Epinion	Q3	140x	136x	132x	125x	117x
	Q5	517x	507x	492x	475x	449x
Gowalla	Q3	40x	38x	35x	30x	25x
	Q5	148x	140x	133x	125x	114x
Flixster	Q3	65x	52x	42x	37x	30x
	Q5	176x	167x	153x	142x	133x

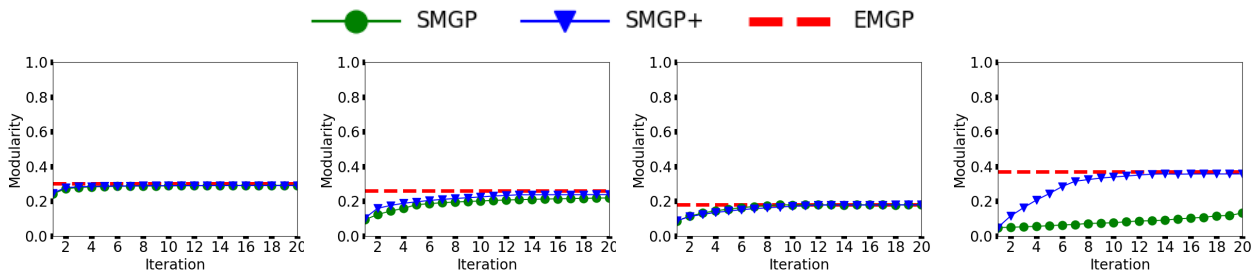
TABLE VI: The modularity scores with different k.

Dataset	Motif	k=10		k=50	
		EMGP	SMGP+	EMGP	SMGP+
Epinion	Q3	0.326	0.325	0.323	0.321
	Q5	0.369	0.365	0.358	0.352
Gowalla	Q3	0.346	0.341	0.330	0.328
	Q5	0.343	0.332	0.298	0.289
Flixster	Q3	0.442	0.437	0.436	0.428
	Q5	0.355	0.343	0.360	0.352

Figure 9 shows the *speedup per* 1% *relative error* achieved by SMGP+ in each iteration. The speedup is the running time of EMGP divided by the cumulative running time of SMGP+. Unless specified otherwise, we use the speedup per 1% error ratio and the speedup per error ratio interchangeably. We have the following observations: (1) Compared to EMGP, SMGP+ can achieve up to three orders of magnitude speedup per error ratio while maintaining high-quality solutions (e.g., Figure 9(b)). That demonstrates the effectiveness and scalability of SMGP+ on dense datasets. (2) Greater exact objective scores lead to a larger speedup per error ratio. A small error difference $|O_{\text{SMGP+}} - O_{\text{EMGP}}|$ with a greater exact objective score will lead to a smaller relative error ratio than the one with a lower exact objective score. Thus, in Figure 9, the speedup per error ratio achieved by `MAPPR` is generally greater than `Fennel` on Digg, Dogster and Catster. (3) Smaller exact objective scores lead to a more stable trend of speedup per error ratio. Thus, EMGP demonstrates more stable performance with `Fennel` on Digg, Dogster and Catster, and shows a more stable trend with `MAPPR` as the kernel on Orkut.

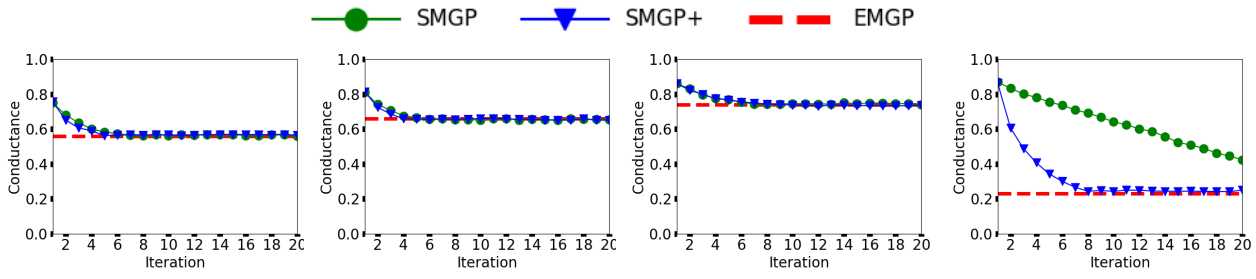
D. Ablation study on k

We study the impact of k with Fennel as the graph kernel, since MAPPR is a *local* partitioning method with k fixed as 2. We choose two motifs Q_2 and Q_5 for experiments as other query motifs show similar trends. Table V and Table VI shows the speedup of SMGP+ against EMGP and their modularity scores achieved respectively. The speedup decreases as k increases because of the growing cost of partition factor computation of crossing edges, the number of which increases as k increases. The modularity scores achieved by SMGP+ is very



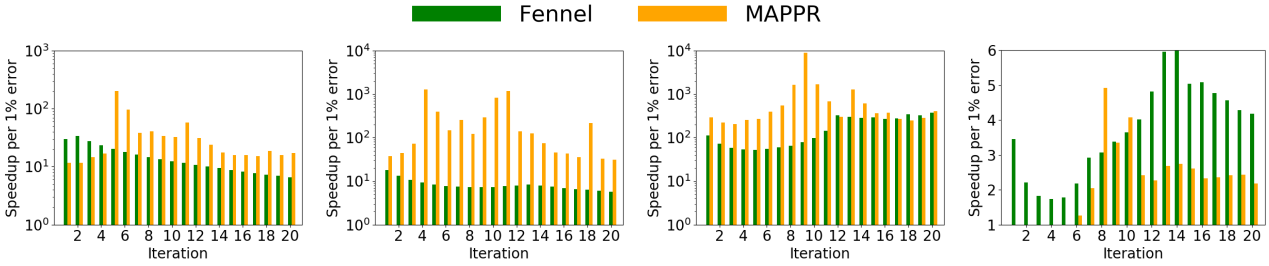
(a) Digg (b) Dogster (c) Catster (d) Orkut

Fig. 7: Effectiveness comparison with `Fennel` as the kernel and Q_1 as the motif.



(a) Digg (b) Dogster (c) Catster (d) Orkut

Fig. 8: Effectiveness comparison with `MAPPR` as the kernel and Q_1 as the motif.



(a) Digg (b) Dogster (c) Catster (d) Orkut

Fig. 9: Speedup per 1% error achieved by SMGP+ in each iteration.

competitive with EMGP as k increases, which demonstrates the effectiveness of our solution.

VI. Related Work

We summarize the literature related to this work from two fields, namely graphlet computation and graph partitioning.

Graphlet Computation is a closely related field to motif analysis. Given a data graph G and a connected motif graph Q, graphlet count is the number of induced sub-graphs on G that are isomorphic to Q. Computing the exact graphlet count is computationally intensive [26], [27], [28] since determining whether G contains a subgraph that is isomorphic to Q is NP-complete [12]. Thus, many sampling-based techniques are proposed. Monte Carlo based methods are shown to be very successful in accurately estimating the graphlet count with guaranteed high probabilities [29], [30], [14]. In [30], a random walk method is proposed to estimate the 3-vertex graphlet count so as to analyze the clustering coefficient of social networks. To improve upon prior work, the authors in [31], [32] generate statistics of graphlet count on high order graphs. Paramonov et al. [16] propose a new Monte Carlo algorithm, called lifting, for graphlet count computation. The aforementioned solutions lay the theoretical foundation of the sampling strategy in this work.

Graph Partitioning is a fundamental problem in graph theory that has wide applications in community detection [33], distributed computing and image processing [34]. *Conductance* [11] and *modularity* [10] are two popular metrics to evaluate the partition quality for community detection. Given a connected graph G, finding a bisection with the minimum conductance [35] or the maximum modularity [36] is NP-hard. Moreover, many practical requirements for graph partition make the problem even more intricate. For example, the problem of balanced k-way graph partition with minimum ratio-cut is proven to be NP-hard as well [34]. As such, there is a long line of efforts aiming to develop efficient k-way partition algorithms that can achieve good empirical results and easy-to-adjust optimization objectives [37], [38], [25].

The aforementioned existing studies focus on analyzing only simple edges across the partitions and designing opti-

mization algorithms accordingly. Recently, it has been shown that high-order connected subgraphs, known as network motifs, can provide more insights to understand fundamental graph structures and behaviors of many complex systems [4], [6], [7]. This finding drives a surge in developing methods for motif-based graph partitioning [6], [8], [9]. Existing motif-based partition methods mainly have two drawbacks. First, they only focus on the conductance metric while it is not straightforward to extend these solutions to handle more requirements raised by varying applications in reality, e.g., the modularity-based graph partitioning offered by many graph partitioning libraries [39], [10]. Second, they have to compute the *entire motif adjacency matrix* by invoking existing algorithms on graphlet computation, which is computationally expensive as mentioned in Section I. To mitigate these issues, we propose an adaptive sampling framework to estimate edge weights, and our framework is robust to the choice of partitioning objectives (i.e., conductance and modularity) to cater for different application needs, as shown in the experiments.

VII. Conclusion and Future Work

In this paper, we studied how to efficiently estimate motif-based edge weights for the motif-based partitioning problem (MGP) for the first time. Our proposed methods are able to effectively estimate the potential of motifs w.r.t. the conductance and modularity for the motif-based partitioning, and can help users quickly make decisions on the choice of motifs. We also estimate the potential of motifs individually, leaving much room for improvement when there are many query motifs. It will be interesting if we can use the estimation of one motif to accurately indicate how other similar motifs will behave when used as the query motifs. On the other hand, existing studies including this work all require a motif as input. However, it is difficult for users without specialized knowledge on the input graphs and graph patterns to select the query motif. Thus, another future direction is how to automatically compute motifs with which the underlying GP kenmel is able to produce high-quality partitioning results.

References

[1] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Physical Review E*, vol. 74, no. 1, p. 016110, 2006.

[2] B. Hendrickson and T. G. Kolda, "Graph partitioning models for parallel computing," *Parallel computing*, vol. 26, no. 12, pp. 1519–1534, 2000.

[3] L. Grady and E. L. Schwartz, "Isoperimetric graph partitioning for image segmentation," *TPAMI*, vol. 28, no. 3, pp. 469–475, 2006.

[4] K.-K. R. Choo, "Money laundering risks of prepaid stored value cards." *Trends & Issues in Crime & Criminal Justice*, no. 363, 2008.

[5] F. Chierichetti, R. Kumar, P. Raghavan, and T. Sarlos, "Are web users really markovian?" in *WWW*, 2012, pp. 609–618.

[6] H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich, "Local higher-order graph clustering," in *KDD*, 2017, pp. 555–564.

[7] O. Sporns and R. Kötter, "Motifs in brain networks," *PLoS biology*, vol. 2, no. 11, p. e369, 2004.

[8] K. Rohe and T. Qin, "The blessing of transitivity in sparse and stochastic networks," *arXiv preprint arXiv:1307.2302*, 2013.

[9] C. E. Tsourakakis, J. Pachocki, and M. Mitzenmacher, "Scalable motif-aware graph clustering," in *WWW*, 2017, pp. 1451–1460.

[10] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Physical review E*, vol. 70, no. 6, p. 066111, 2004.

[11] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *FOCS*, 2006, pp. 475–486.

[12] S. A. Cook, "The complexity of theorem-proving procedures," in *STOC*, 1971, pp. 151–158.

[13] F. Chung and L. Lu, "Concentration inequalities and martingale inequalities: a survey," *Internet Mathematics*, vol. 3, no. 1, pp. 79–127, 2006.

[14] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs," *Bioinformatics*, vol. 20, no. 11, pp. 1746–1758, 2004.

[15] X. Lu and S. Bressan, "Sampling connected induced subgraphs uniformly at random," in *SSDBM*, 2012, pp. 195–212.

[16] K. Paramonov, D. Shemetov, and J. Sharpnack, "Estimating graphlet statistics via lifting," in *KDD*, 2019, pp. 587–595.

[17] D. G. Horvitz and D. J. Thompson, "A generalization of sampling without replacement from a finite universe," *JASA*, vol. 47, no. 260, pp. 663–685, 1952.

[18] W. Hoeffding, "Probability inequalities for sums of bounded random variables," in *The Collected Works of Wassily Hoeffding*, 1994, pp. 409–426.

[19] N. T. L. Tran, L. DeLuccia, A. F. McDonald, and C.-H. Huang, "Cross-disciplinary detection and analysis of network motifs," *Bioinformatics and Biology insights*, vol. 9, pp. BBI–S23 619, 2015.

[20] P. Cunningham, M. Harrigan, G. Wu, and D. O'CALLAGHAN, "Characterizing ego-networks using motifs," *Network Science*, vol. 1, no. 2, pp. 170–190, 2013.

[21] S. Wuchty, Z. N. Oltvai, and A.-L. Barabási, "Evolutionary conservation of motif constituents in the yeast protein interaction network," *Nature genetics*, vol. 35, no. 2, pp. 176–179, 2003.

[22] Y. Al Rozz and R. Menezes, "Author attribution using network motifs," in *International Workshop on Complex Networks*, 2018, pp. 199–207.

[23] T. K. N. Collection, http://konect.uni-koblenz.de, 2017.

[24] S. Sun, Y. Che, L. Wang, and Q. Luo, "Efficient parallel subgraph enumeration on a single machine," in *ICDE*, 2019, pp. 232–243.

[25] C. Tsourakakis, C. Gkantsidis, B. Radunovic, and M. Vojnovic, "Fennel: Streaming graph partitioning for massive scale graphs," in *WSDM*, 2014, pp. 333–342.

[26] N. Chiba and T. Nishizeki, "Arboricity and subgraph listing algorithms," *SIAM Journal on computing*, vol. 14, no. 1, pp. 210–223, 1985.

[27] J. A. Grochow and M. Kellis, "Network motif discovery using subgraph enumeration and symmetry-breaking," in *RECOMB*, 2007, pp. 92–106.

[28] H. Kim, J. Lee, S. S. Bhowmick, W.-S. Han, J. Lee, S. Ko, and M. H. Jarrah, "Dualsim: Parallel subgraph enumeration in a massive graph on a single machine," in *SIGMOD*, 2016, pp. 1231–1245.

[29] M. A. Bhuiyan, M. Rahman, M. Rahman, and M. Al Hasan, "Guise: Uniform sampling of graphlets for large graph analysis," in *ICDM*, 2012, pp. 91–100.

[30] L. Katzir and S. J. Hardiman, "Estimating clustering coefficients and size of social networks via random walk," *TWEB*, vol. 9, no. 4, p. 19, 2015.

[31] P. Wang, J. Lui, B. Ribeiro, D. Towsley, J. Zhao, and X. Guan, "Efficiently estimating motif statistics of large networks," *TKDD*, vol. 9, no. 2, pp. 8:1–8:27, 2014.

[32] X. Chen, Y. Li, P. Wang, and J. Lui, "A general framework for estimating graphlet statistics via random walk," *VLDB*, vol. 10, no. 3, pp. 253–264, 2016.

[33] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[34] K. Andreev and H. Racke, "Balanced graph partitioning," *Theory of Computing Systems*, vol. 39, no. 6, pp. 929–939, 2006.

[35] D. Wagner and F. Wagner, "Between min cut and graph bisection," in *MFCS*, 1993, pp. 744–750.

[36] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On finding graph clusterings with maximum modularity," in *WG*, 2007, pp. 121–132.

[37] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

[38] I. Stanton and G. Kliot, "Streaming graph partitioning for large distributed graphs," in *KDD*, 2012, pp. 1222–1230.

[39] H. Shiokawa, Y. Fujiwara, and M. Onizuka, "Fast algorithm for modularity-based graph clustering," in *AAAI*, 2013.

[40] J. Zhang, B. Liu, J. Tang, T. Chen, and J. Li, "Social influence locality for modeling retweeting behaviors." in *IJCAI*, vol. 13, 2013, pp. 2761–2767.

TABLE VII: Performance w.r.t. effectiveness on triangle motif.

Dataset	Modularity with Fennel		Conductance with MAPPR	
	EMGP	SMGP+	EMGP	SMGP+
Epinion	0.418	0.415	0.429	0.432
Gowalla	0.455	0.453	0.293	0.304
Flixster	0.393	0.369	0.424	0.439
Digg	0.354	0.352	0.479	0.481
Dogster	0.383	0.376	0.508	0.523
Caster	0.291	0.286	0.552	0.557
Orkut	0.454	0.428	0.126	0.134
Sina	0.445	0.433	0.477	0.509

[41] R. Pagh and C. E. Tsourakakis, "Colorful triangle counting and a mapreduce implementation," *Information Processing Letters*, vol. 112, no. 7, pp. 277–281, 2012.

[42] M. N. Kolountzakis, G. L. Miller, R. Peng, and C. E. Tsourakakis, "Efficient triangle counting in large graphs via degree-based vertex partitioning," *Internet Mathematics*, vol. 8, no. 1-2, pp. 161–185, 2012.

[43] T. Schank and D. Wagner, "Finding, counting and listing all triangles in large graphs, an experimental study," in *International workshop on experimental and efficient algorithms*. Springer, 2005, pp. 606–609.

[44] A. Azad, A. Buluç, and J. Gilbert, "Parallel triangle counting and enumeration using matrix algebra," in *2015 IEEE International Parallel and Distributed Processing Symposium Workshop*. IEEE, 2015, pp. 804–811.

[45] H. Yin, A. R. Benson, and J. Leskovec, "Higher-order clustering in networks," *Physical Review E*, vol. 97, no. 5, p. 052306, 2018.

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

[47] 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, no. 5594, pp. 824–827, 2002.

APPENDIX

In this section, we will show more experimental results on more motifs.

Experimental results on the triangle motifs. We include another large dataset Sina [40] for experiments. Sina has 1,787,443 nodes, 355,095,815 edges and is three times greater w.r.t. edge size and five times denser than the current largest dataset Orkut. Table VII and Table VIII compares the effectivenss and efficiency of EMGP and SMGP+ respectively. The partitioning quality and efficiency of SMGP+ is very competitive with EMGP in many cases (e.g., on Digg, Catster and Sina) and SMGP+ can even outperform EMGP in terms of the efficiency on Catster. Computing the exact triangle count can be efficiently implemented with specialized algorithms [41], [42], [43], [44] and thus the need of sampling is not that critical. However, there still lacks of efficient exact algorithms for general motifs and many of these motifs have been shown to be useful and effective for motif-based clustering [45], [46], [47], which is the main focus of our work.

Experimental results on motifs of size 6. We add the motif $\bowtie$, which is reported to be a motif with very high statistical significance across datasets of different disciplinary [19]. Here, the statistical significance refers to z-sore [47] and a motif with a large z-score is highly overrepresented in the original network as compared to randomized ones. We only report the results on Epinion and Gowalla as EMGP fails to compute the edge weights on other datasets within a month. Table IX compares the performance of EMGP and SMGP+ when it converges. SMGP+ is able to achieve three orders of magnitude speedup while producing high-quality results compared with EMGP.

TABLE VIII: Performance w.r.t. efficiency on triangle motif.

Dataset	Running time (s) with Fennel		Running time (s) with MAPPR	
	EMGP	SMGP+	EMGP	SMGP+
Epinion	10	24	10	27
Gowalla	19	73	17	64
Flixster	110	364	100	344
Digg	62	89	60	78
Dogster	296	321	295	529
Catster	531	181	535	297
Orkut	3123	5061	3150	5146
Sina	22290	24878	22893	26272

TABLE IX: Experimental results on motif $\bowtie$.

Dataset	Kernel	Speedup of SMGP+	Score	
			EMGP	SMGP+
Epinion	Fennel	5640x	0.411	0.403
	MAPPR	4980x	0.484	0.506
Gowalla	Fennel	2924x	0.418	0.407
	MAPPR	2651x	0.463	0.502