MOCHI: Motif-based Community Search over Large Heterogeneous Information Networks

Yuhan Zhou[1], Qing Liu[1,3], Xin Huang[2], Jianliang Xu[2], Yunjun Gao[1,3]
[1]Zhejiang University, [2]Hong Kong Baptist University, [3]Zhejiang Key Laboratory of Big Data Intelligent Computing
{zhou_yh,qingliucs,gaoyj}@zju.edu.cn,{xinhuang,xujl}@comp.hkbu.edu.hk

ABSTRACT

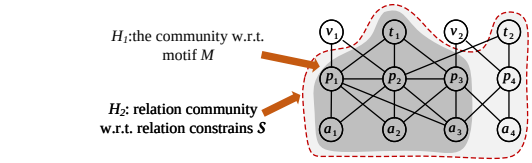
In this paper, we investigate the problem of motif-based community search over heterogeneous information networks (MOCHI). We introduce a novel motif density modularity (MDM) to measure the *motif cohesiveness* of communities. Based on MDM, we define the MOCHI problem as follows: given a heterogeneous information network (HIN) H, a motif M, and a query vertex set Q, the objective is to identify the largest subgraph of H connected by motif instances, containing Q, and maximizing MDM. Since motifs encapsulate rich semantics, the MOCHI problem enables the retrieval of semantically meaningful communities, facilitating applications like fraud detection and academic collaboration analysis. Due to the NP-hardness of MOCHI, we propose three algorithms. The basic algorithm iteratively removes vertices to maximize MDM. However, vertex selection and maintaining M-connectivity incur significant overhead. Hence, we devise an *MW-HIN-based algorithm* that employs a *vertex selection strategy* and a compact data structure *motif-based weighted HIN* to boost efficiency. Additionally, we propose a *motif-distance-based algorithm* to further improve performance by integrating *motif distance* and a lightweight goodness function named M-ratio to remove vertices. Extensive experiments on real-world HINs demonstrate the effectiveness and efficiency of our proposed methods.

PVLDB Reference Format:
Yuhan Zhou, Qing Liu, Xin Huang, Jianliang Xu, Yunjun Gao. MOCHI: Motif-based Community Search over Large Heterogeneous Information Networks. PVLDB, 14(1): XXX-XXX, 2026.
doi:XX.XX/XXX.XX

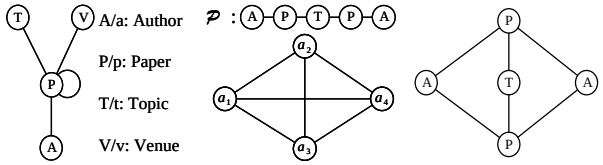
PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/ZJU-DAILY/MOCHI.

1 INTRODUCTION

Heterogeneous information networks (HINs), characterized by multiple types of vertices and edges, are prevalent in various real-world applications. Notable examples of HINs include bibliographic networks [1], movie networks [2], biological networks [3], and knowledge graphs [4, 5]. Figures 1(a) and 1(b) illustrate an HIN H of DBLP and its schema, respectively. Specifically, the vertices of H consist of *author* (A), *paper* (P), *topic* (T), and *venue* (V). The relationships among these different types of vertices include authorship ($A-P$), citation ($P-P$), publication ($P-V$), and mention ($P-T$).



(a) A heterogeneous information network H of DBLP



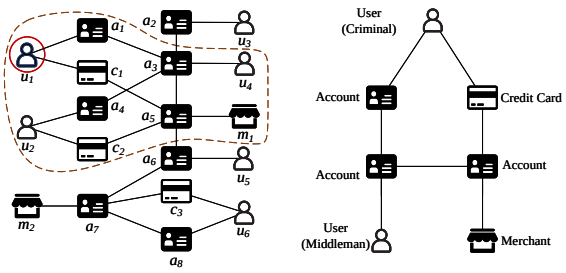
(b) Schema of H (c) $(3,\mathcal{P})$-core of H (d) A motif M in H

Figure 1: A motivating example.

Recently, the problem of community search over HINs has gained much attention [6–11]. The objective is to identify a cohesive subgraph containing the query vertex set from an HIN. In the literature, various models for community search over HINs have been proposed, which can be categorized into two types: homogeneous models [7–11] and heterogeneous models [6]. (1) The homogeneous models return communities composed of vertices of the same type. For example, Fang et al. [7] proposed $(k,\mathcal{P})$-core to model communities in HINs. Given an HIN H, a symmetric meta-path $\mathcal{P}$, and an integer k, the $(k,\mathcal{P})$-core consists of a set of vertices of the same type, where each vertex is adjacent to at least k other vertices through instances of the meta-path $\mathcal{P}$. Figure 1(c) shows an example of $(k,\mathcal{P})$-core for the HIN H in Figure 1(a). Let $k=3$ and $\mathcal{P}=(APTPA)$, the $(3,\mathcal{P})$-core consists of vertices a_1, a_2, a_3, and a_4. (2) The heterogeneous model retrieves communities that include different types of vertices. For example, Jian et al. [6] introduced a relation community, which must satisfy user-specified relation constraints. The relation constraints $\mathcal{S}=\{s_1,s_2,...,s_n\}$ consist of triplets $s_i=\langle l_i^1,l_i^2,k_i\rangle$, indicating that vertices of type l_i^1 should have at least k_i neighbors of type l_i^2. The subgraph H_2 in Figure 1(a) represents a relation community with $\mathcal{S}=\{\langle A,P,2\rangle,\langle P,A,2\rangle,\langle P,T,1\rangle,\langle T,P,2\rangle\}$.

While HINs and their communities inherently encapsulate rich semantics that can offer valuable insights, existing models fall short of fully capturing these semantics. For instance, in Figure 1(a), H_1 is a community where two authors have co-authored at least two papers on the same topic. The semantics of H_1 can be represented by the motif M in Figure 1(d). Although existing models, such as $(k,\mathcal{P})$-core and relation community, can convey some simple semantics, they struggle to represent complex semantics like the motif M. This limitation arises because motifs cannot be easily decomposed into a set of meta-paths or relational constraints. Moreover, these models

Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX



(a) A credit card transaction network (b) A motif M of credit card fraud

Figure 2: An application of the MOCHI problem.

primarily focus on intra-community cohesiveness, but overlook vertex relationships both inside and outside the community, thus failing to capture the global structure and semantics of HINs.

To address these limitations, in this paper, we study a novel problem of motif-based community search over heterogeneous information networks (MOCHI). The goal is to identify *a heterogeneous community with both structural and semantic cohesiveness*. We start by introducing a new community model called *motif density modularity* (MDM), which integrates both structural and semantic cohesiveness. Specifically, given an HIN H and a motif M, the MDM of H and M is defined as the difference between the actual motif density and the expected motif density in H. A larger MDM indicates a higher level of cohesiveness in H. Note that the computation of MDM involves the expected motif density, which relies on a random graph model. However, existing random graph models [12–15] fail to simultaneously capture the higher-order structure and heterogeneous information of HINs, leading to inaccurate estimations of the expected motif density. To overcome this challenge, we propose a novel *motif-based random graph model* that preserves both the higher-order structure and the heterogeneous information of HINs.

Based on MDM, we formulate the MOCHI problem as follows: given an HIN H, a query vertex set Q, and a motif M, the objective is to identify the largest subgraph of H that is connected by motif instances, contains Q, and maximizes MDM. The advantages of MOCHI include its parameter-free nature, support for multiple query vertices of different types, and the ability for users to specify a motif for personalized heterogeneous community search with complex semantics. The MOCHI problem has a broad range of applications, including fraud detection, neuron interaction analysis, movie recommendation, and academic collaboration analysis. For example, Figure 2(a) depicts a credit card transaction network consisting of four types of vertices: user, credit card, account, and merchant. The motif M in Figure 2(b) illustrates a type of credit card fraud [16]. Assume that the e-commerce platform has identified a suspect u_1 and aims to uncover the criminal network associated with this individual, including related credit cards, accounts, and merchants. In this scenario, the platform can use M and u_1 as the query motif and query vertex, respectively, and employ MOCHI to identify the community, as outlined by the brown dotted line in Figure 2(a). We can observe that the vertices within the community are densely connected w.r.t. M, indicating the criminal network of u_1.

We prove that the MOCHI problem is NP-hard. Due to its hardness, we propose a basic algorithm to address this problem and develop two efficient algorithms to enhance performance. The basic algorithm greedily deletes the vertices that maximize MDM. However, this approach incurs significant overhead because it requires evaluating every vertex in each iteration and maintaining M-connectivity via traversing the HIN. To mitigate these limitations, we propose a more efficient *MW-HIN-based algorithm* based on a *vertex selection strategy* and a compact data structure called *motif-based weighted HIN* (MW-HIN). The vertex selection strategy evaluates the MDM after removing each vertex efficiently and eliminates the candidate vertices whose removal does not disrupt the M-connectivity of the remaining vertices robustly. To expedite the process of maintaining M-connectivity after vertex removal, we design the MW-HIN to explicitly encode M-connectivity relationships between vertices through edge weights. Furthermore, we design two optimizations to boost efficiency by updating edge weights lazily and skipping vertices based on historical access information.

To efficiently handle HINs with numerous motif instances and vertices, we propose a third algorithm named *motif-distance-based algorithm*, which integrates the MDM with a novel metric *motif distance* to remove irrelevant vertices while maximizing MDM. It operates in two phases: coarse-grained deletion and fine-grained deletion. In the coarse-grained deletion, the search space is rapidly reduced by removing vertices that are farthest from the query vertices in batches, in terms of motif distance. In the fine-grained deletion, the community is further refined by removing vertices individually. Additionally, to reduce the overhead of updating vertex goodness, we introduce a lightweight goodness function named *M-ratio*, which efficiently and effectively determines the order in which vertices should be removed.

In summary, our contributions of this paper are as follows.

- We propose a novel motif density modularity using a motif-based random graph model to measure the structural and semantic cohesiveness of communities in HINs.
- We formulate the MOCHI problem and prove its hardness. To the best of our knowledge, we are the first to employ motif and modularity for community search over HINs.
- We propose a basic algorithm to address the MOCHI problem and develop two more efficient algorithms with a series of optimizations to enhance performance.
- We conduct extensive experiments on real-world HINs to demonstrate the effectiveness and efficiency of the proposed model and algorithms.

Roadmap. The rest of this paper is organized as follows. Section 2 reviews related work. We formulate and analyze the MOCHI problem in Section 3. Section 4 introduces algorithms to address the MOCHI problem. Section 5 reports the experimental results. We conclude this paper in Section 6.

2 RELATED WORK

Community Search. Community search (CS) aims to find query-dependent communities within graphs [17, 18]. It has been extensively studied in homogeneous networks since it was first proposed by Sozio and Gionis [19], and various models have been developed for community search, such as k-core [19, 20], k-truss [21, 22], k-edge connected component [23, 24], clique [25, 26], and density modularity [14]. Moreover, [27–31] explored learning-based CS,

which does not require users to specify concrete models. In addition to simple graphs, CS over complex graphs has also been investigated, including directed graphs [32, 33], attributed graphs [34, 35], bipartite graphs [36, 37], uncertain graphs [38, 39], and temporal graphs [40, 41].

CS over HINs has also received significant attention [6–11]. For example, Jian et al. [6] introduced the concept of a relation community, which allows users to specify relation constraints. To identify communities of the single vertex type over HINs, Fang et al. proposed a series of homogeneous community models based on meta-paths such as $(k, \mathcal{P})$-core [7], $(k, \mathcal{P})$-truss [8], Ψ-NMC [9], and HIC [10]. Li et al. extended the $(k, \mathcal{P})$-core model by incorporating meta-structure [11]. Additionally, [42, 43] investigated attributed community search over HINs. Despite these advancements in CS, existing research has yet to leverage motif, which can represent complex semantics, for heterogeneous community search over HINs. This paper aims to address this gap.

Graph Modularity. Graph modularity, introduced by Newman and Girvan, quantifies the quality of community structure in networks [12]. It has been extensively explored in subsequent studies [44–47]. For example, bipartite modularity [48, 49] and tripartite modularity [50, 51] have been proposed for k-partite network with $k > 1$. Arenas et al. [13] designed *motif modularity*, which is defined as the normalized fraction of motif instances within communities minus the normalized expected fraction in a random network. Note that this is different from our proposed motif density modularity. First, motif modularity is designed for homogeneous graphs. Second, it adopts the classic random graph model to derive the modularity function, which assumes that the degree distribution of a random network matches that of the original network. Although this maintains a similar one-hop structure for vertices between the random and origin networks, it neglects the motif structure, which carries complex semantics, thereby limiting its ability to identify communities with rich semantics. Recently, [52, 53] investigated motif-based modularity in multi-layer networks, where the vertices across different layers represent the same individuals. To date, no existing graph modularity framework accounts for motifs in HINs, which is the focus of this paper.

3 PROBLEM FORMULATION

We model an HIN as an undirected graph $H = (V_H, E_H, L_H, \ell_H)$, where V_H is the set of vertices, E_H is the set of edges, L_H is the set of vertex types, and $\ell_H : V_H \to L_H$ is a mapping function that assigns each vertex $v \in V_H$ a type $\ell_H(v) \in L_H$. The set $V_H(\ell_H(v))$ denotes all vertices in H with the same type as v. The *schema* of H, defined over L_H, describes all allowable edges between vertex types. For example, Figure 1(b) illustrates the schema of DBLP, which includes four relationships among vertex types. For a vertex set $S \subseteq V_H$, we use $H[S]$ to denote the subgraph of H induced by S. Additionally, $\deg_H(v)$ denotes the degree of vertex v in H.

3.1 Motif Density Modularity

To define motif density modularity, we first introduce the concepts of *motif* and *modularity*.

Motif. The motif is a fundamental building block of a graph [54, 55]. Formally, given an HIN H, a motif M of H can be modeled by $M = (V_M, E_M, L_M, \ell_M)$, where V_M is a set of vertices that represents vertex types, E_M is a set of edges between vertices in V_M, L_M is a set of vertex types, and ℓ_M is a vertex type mapping function for M such that $\forall v \in V_M$, $\ell_M(v) \in L_M$. It is noteworthy that M should adhere to the constraints imposed by H's schema.

Definition 1. *(Motif Instance). Given an HIN H, a motif M, and a subgraph $H' \subseteq H$, H' is a motif instance of M iff H' is isomorphic to M, i.e., there exists a bijective mapping $\phi : V_M \to V_{H'}$ such that (1) $\forall u \in V_M, \ell_M(u) = \ell_H(\phi(u))$ and (2) $\forall (u, v) \in E_M$, $(\phi(u), \phi(v)) \in E_{H'}$.*

For example, Figure 3(b) shows a motif M of an HIN H of DBLP. Correspondingly, in Figure 3(a), the subgraph induced by vertices $\{p_1, p_2, a_1\}$ is a motif instance of M. Given an HIN H and a motif M, we use $\mathcal{I}_H^M$ to denote all motif instances of M in H. The motif degree of a vertex $v \in V_H$, denoted by $\texttt{Mdeg}_H(v)$, is the number of motif instances of M containing v, i.e., $\texttt{Mdeg}_H(v) = |\{I \mid I \in \mathcal{I}_H^M \wedge v \in V_I\}|$. The motif volume of a vertex set $S \subseteq V_H$ is defined as the sum of the motif degree of all vertices in S, i.e., $\texttt{Mvol}_H(S) = \sum_{v \in S} \texttt{Mdeg}_H(v)$.

Modularity. The modularity is widely used to evaluate the community quality [56], which is defined as the differences in graph structures from an expected random graph. Formally, given a homogeneous graph $G = (V_G, E_G)$ and a set of vertices $S \subseteq V_G$, the modularity of S is

$$\begin{aligned} Modularity(G, S) &= \frac{1}{|E_G|}(|E_{G[S]}| - Exp[|E_{G[S]}|]) \\ &= \frac{1}{|E_G|}(|E_{G[S]}| - \frac{\texttt{vol}_G(S)^2}{4|E_G|}), \end{aligned} \tag{1}$$

where $Exp[|E_{G[S]}|]$ denotes the expected number of edges in $G[S]$ and $\texttt{vol}_G(S)$ denotes the sum of the degree of all vertices in S over G. Specifically, the modularity evaluates the community structures by comparing the edges within the community to a random distribution of edges. Higher modularity indicates well-partitioned communities with more internal connections and fewer external connections.

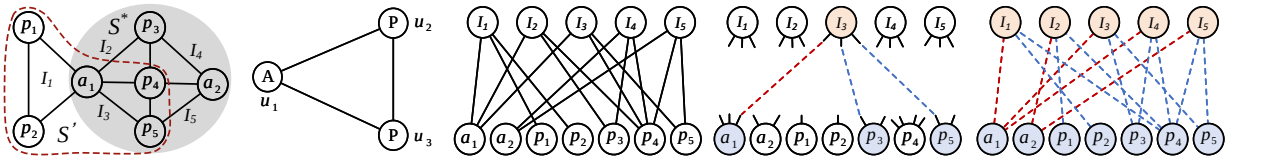
Generalized Motif Modularity. As analyzed in Section 2, directly applying existing modularity to HINs suffers from the loss of the higher-order structure as well as the heterogeneous information. Motivated by the classic modularity shown in Equation 1, we propose a new generalized motif modularity for HINs.

Definition 2. *(Generalized Motif Modularity (GMM)). Given an HIN H, a motif M, and a vertex set $S \subseteq V_H$, the generalized motif modularity of S is*

$$GMM(H, S, M) = \frac{1}{|\mathcal{I}_H^M|}(|\mathcal{I}_{H[S]}^M| - Exp[|\mathcal{I}_{H[S]}^M|]), \tag{2}$$

where $Exp[|\mathcal{I}_{H[S]}^M|]$ denotes the expected number of motif instances in $H[S]$.

In other words, GMM is defined as the difference in the number of motif instances between $H[S]$ and the random HIN induced by S. For GMM, a primary issue is how to compute $Exp[|\mathcal{I}_{H[S]}^M|]$. Recall that the classic modularity $Modularity(G, S)$ employs the random graph model that preserves the degree distribution of the original graph to compute $Exp[|E_{G[S]}|]$ [56]. However, it is specifically designed for homogeneous networks and does not consider motifs,



(a) An HIN H of DBLP (b) A motif M in H (c) A motif-based bipartite graph (d) The process of MGM (e) A random motif-based bipartite graph

Figure 3: An example of MGM.

making it unsuitable for HINs. To tackle this issue, we design a novel *motif-based random graph model*. Given an HIN H, the goal of the motif-based random graph model is to reconstruct H that preserves the vertices' motif degree as much as possible. To reconstruct H, we introduce the concept of *motif-based bipartite graph*.

Motif-based bipartite graph. Given an HIN H and a motif M, the motif-based bipartite graph of H and M is defined as $\mathcal{B}_M = (\mathcal{R}_{\mathcal{B}_M}, \mathcal{L}_{\mathcal{B}_M}, \mathcal{E}_{\mathcal{B}_M})$. Specifically, $\mathcal{R}_{\mathcal{B}_M}$ and $\mathcal{L}_{\mathcal{B}_M}$ are two disjoint vertex sets. $\mathcal{R}_{\mathcal{B}_M} = \{v : \exists I \in \mathcal{I}_H^M, v \text{ represents } I\}$ and $|\mathcal{R}_{\mathcal{B}_M}| = |\mathcal{I}_H^M|$. In other words, each vertex $v_I \in \mathcal{R}_{\mathcal{B}_M}$ represents a motif instance I of M in H. In addition, $\mathcal{L}_{\mathcal{B}_M}$ is the vertex set of H, i.e., $\mathcal{L}_{\mathcal{B}_M} = V_H$. $\mathcal{E}_{\mathcal{B}_M}$ is an undirected edge set that connects vertices of $\mathcal{R}_{\mathcal{B}_M}$ and $\mathcal{L}_{\mathcal{B}_M}$, i.e., $\mathcal{E}_{\mathcal{B}_M} \subseteq \mathcal{R}_{\mathcal{B}_M} \times \mathcal{L}_{\mathcal{B}_M}$. In particular, $\forall u \in \mathcal{L}_{\mathcal{B}_M}\ \forall v \in \mathcal{R}_{\mathcal{B}_M}$, if u is contained in the motif instance represented by v, the edge (u, v) is in $\mathcal{E}_{\mathcal{B}_M}$. For example, Figure 3(c) depicts the motif-based bipartite graph of H and M shown in Figures 3(a) and 3(b), respectively. $\mathcal{R}_{\mathcal{B}_M} = \{v_{I_1}, v_{I_2}, v_{I_3}, v_{I_4}, v_{I_5}\}$ and $\mathcal{L}_{\mathcal{B}_M} = \{a_1, a_2, p_1, p_2, p_3, p_4, p_5\}$. As shown in Figure 3(a), since p_1 is within the motif instance I_1, there is an edge between I_1 and p_1 in Figure 3(c). The HIN and its corresponding motif-based bipartite graph have the following relationships.[1]

Lemma 1. *Given an HIN H, a motif M, and a vertex set $S \subseteq V_H$, the motif-based bipartite graph $\mathcal{B}_M$ of M and H satisfies:*

(1) $\forall v \in \mathcal{L}_{\mathcal{B}_M}$, $\mathtt{deg}_{\mathcal{B}_M}(v) = \mathtt{Mdeg}_H(v)$;

(2) For a motif instance $I \in \mathcal{I}_{H[S]}^M$, the vertex $v_I \in \mathcal{R}_{\mathcal{B}_M}$ satisfies that $\mathcal{N}_{\mathcal{B}_M}(v_I) \subseteq S$;

(3) $\forall l \in L_M$, $\mathtt{vol}_{\mathcal{B}_M}(V_H(l)) = \mathtt{Mvol}_H(V_H(l))$.

Here, $\mathtt{deg}_{\mathcal{B}_M}(v)$ is the degree of v in $\mathcal{B}_M$, $\mathcal{N}_{\mathcal{B}_M}(v_I)$ is the neighbors of v_I in $\mathcal{B}_M$, and $\mathtt{vol}_{\mathcal{B}_M}(V_H(l)) = \sum_{v \in V_H(l)} \mathtt{deg}_{\mathcal{B}_M}(v)$.

On the basis of the motif-based bipartite graph, we formally introduce Motif-based Random Graph Model (MGM).

Motif-based random graph model. Given an HIN H, a motif M, and the corresponding motif-based bipartite graph $\mathcal{B}_M$ of M and H, MGM is to reconstruct a random motif-based bipartite graph $\mathcal{B}'_M$ by breaking and rewiring the edges in $\mathcal{B}_M$. Specifically, MGM first copies $\mathcal{B}'_M$ from $\mathcal{B}_M$ and breaks all the edges in $\mathcal{B}'_M$. Then, according to the motif M, $\forall v_I \in \mathcal{R}_{\mathcal{B}_M}$, MGM selects different types of vertices in $\mathcal{L}_{\mathcal{B}_M}$. Here, the probability of a vertex v to be selected is $p_v = \frac{\mathtt{deg}_{\mathcal{B}_M}(v)}{\mathtt{vol}_{\mathcal{B}_M}(V_H(\ell_H(v)))}$. If v is selected, MGM rewires an edge between v and v_I. After all vertices of $\mathcal{R}_{\mathcal{B}_M}$ are processed, $\mathcal{B}'_M$ is successfully reconstructed. For instance, in Figure 3(d), after breaking all edges of $\mathcal{B}_M$ in Figure 3(c), I_3 forms a new motif instance of M, which consists of a_1, p_3, and p_5. After all vertices are rewired, we get a new motif-based bipartite graph shown in Figure 3(e).

[1]Due to space limitations, some proofs of this paper are provided in the technical report [57].

The random motif-based bipartite graph generated by MGM has the following properties.

Lemma 2. *Given an HIN H, a motif M, the corresponding motif-based bipartite graph $\mathcal{B}_M$ of M and H, and the random motif-based bipartite graph $\mathcal{B}'_M$ generated by MGM, $\mathcal{B}_M$ and $\mathcal{B}'_M$ have the following relationships:*

(1) $\forall v \in \mathcal{L}_{\mathcal{B}_M}$, $Exp[\mathtt{deg}_{\mathcal{B}'_M}(v)] = \mathtt{deg}_{\mathcal{B}_M}(v) = \mathtt{Mdeg}_H(v)$.

(2) $\forall l \in L_M$, $Exp[\mathtt{vol}_{\mathcal{B}'_M}(V_H(l))] = \mathtt{Mvol}_H(V_H(l))$.

Note that $Exp[\mathtt{deg}_{\mathcal{B}'_M}(v)]$ denotes the expected degree of v in $\mathcal{B}'_M$ and $Exp[\mathtt{vol}_{\mathcal{B}'_M}(V_H(l))]$ denotes the expected sum of degree of vertices with type l in $\mathcal{B}'_M$.

Lemma 2 reveals that $\mathcal{B}'_M$ generated by MGM preserves the expected degree distribution of $\mathcal{B}_M$. Moreover, it is equal to the expected motif degree distribution of H in terms of M, thereby facilitating a more accurate estimation of the expected number of motif instances $Exp[|\mathcal{I}_{H[S]}^M|]$ in Equation 2.

Theorem 1. *Given an HIN H, a motif M, and a vertex set $S \subseteq V_H$, the expected number of motif instances in $H[S]$ under MGM is*

$$Exp[|\mathcal{I}_{H[S]}^M|] = |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\mathtt{Mvol}_H(S(\ell_M(u)))}{\mathtt{Mvol}_H(V_H(\ell_M(u)))}, \tag{3}$$

where $S(\ell_M(u))$ is a set of vertices in S with type $\ell_M(u)$.

Proof. We estimate $Exp[|\mathcal{I}_{H[S]}^M|]$ by the expected number of motif instances in $\mathcal{B}'_M$, which all consist of vertices in S, i.e., $Exp[|\mathcal{I}_{H[S]}^M|] = Exp[|\{v_I \in \mathcal{R}_{\mathcal{B}'_M} : \forall (v_I, v) \in \mathcal{E}_{\mathcal{B}'_M}, v \in S\}|]$. Given a motif instance I in $\mathcal{B}'_M$, the MGM selects different types of vertices according to M. The probability of selecting a vertex with the type l from S is $\frac{\mathtt{Mvol}_H(S(l))}{\mathtt{Mvol}_H(V_H(l))}$. Hence, the probability that all selected vertices of a motif instance are from S is $\prod_{u \in V_M} \frac{\mathtt{Mvol}_H(S(\ell_M(u)))}{\mathtt{Mvol}_H(V_H(\ell_M(u)))}$. Since MGM needs to select vertices for all the motif instances, the process will be repeated for $|\mathcal{I}_H^M|$ times. Therefore, $Exp[|\mathcal{I}_{H[S]}^M|] = |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\mathtt{Mvol}_H(S(\ell_M(u)))}{\mathtt{Mvol}_H(V_H(\ell_M(u)))}$. □

According to Theorem 1, the generalized motif modularity is

$$GMM(H, S, M) = \frac{1}{|\mathcal{I}_H^M|}(|\mathcal{I}_{H[S]}^M| - |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\mathtt{Mvol}_H(S(\ell_M(u)))}{\mathtt{Mvol}_H(V_H(\ell_M(u)))}).$$

Nevertheless, like the classic modularity, GMM also suffers from the resolution limit [58] and the free-rider effect [59]. Specifically, the resolution limit means that it falls short of discovering small communities, even if they are densely connected. The free-rider effect indicates that the community may contain some vertices that

are irrelevant to the query vertices. To alleviate these problems, we propose an improved version of GMM, called *motif density modularity*, which exploits motif instance density to enhance community cohesiveness.

Definition 3. *(Motif Density Modularity (MDM)). Given an HIN H, a motif M, and a set of vertices $S \subseteq V_H$, the motif density modularity of S is:*

$$MDM(H,S,M) = \frac{1}{|S|}(|\mathcal{I}^M_{H[S]}| - Exp[|\mathcal{I}^M_{H[S]}|]) = \frac{1}{|S|}(|\mathcal{I}^M_{H[S]}| - |\mathcal{I}^M_H| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}).$$

The difference between MDM and GMM is that we use $|S|$ instead of $|\mathcal{I}^M_H|$ as the normalization term. By normalizing with $|S|$, the MDM of a community can be interpreted as the density of motif instances minus the expected density of motif instances within the community. MDM can alleviate both the free-rider effect and the resolution limit.

Lemma 3. *Whenever MDM suffers from the free-rider effect and resolution limit, GMM also suffers from these limitations.*

3.2 Problem Definition

Based on the MDM, we define the MOCHI problem. First, we introduce the concepts of M-adjacency and M-connectivity.

Definition 4. *(M-adjacency, M-connectivity). Given an HIN H, a motif M, and two motif instances I_s and I_t of M in H,*

(i) I_s and I_t are M-adjacent if $V_{I_s} \cap V_{I_t} \neq \emptyset$.

(ii) I_s and I_t are M-connected, denoted by $I_s \leftrightarrow I_t$, if there exists a sequence of motif instances $I_1, I_2, ..., I_n (n \geq 2)$ in H, such that $I_s = I_1$, $I_t = I_n$, and for $1 \leq i < n$, $V_{I_i} \cap V_{I_{i+1}} \neq \emptyset$.

For example, in Figure 3(a), the motif instances I_1 and I_2 are M-adjacency since $V_{I_1} \cap V_{I_2} = a_1$. All motif instances in Figure 3(a) are M-connected. We can also extend M-connectivity to vertices and subgraphs. Specifically, for any two vertices $u, v \in V_H$, u and v are M-connected, denoted by $u \leftrightarrow v$, iff (1) u and v belong to the same motif instance, or (2) $\exists I_s, I_t \in \mathcal{I}^M_H$, such that $u \in V_{I_s}$, $v \in V_{I_t}$, and $I_s \leftrightarrow I_t$. Similarly, the subgraph $H[S] \subseteq H$ is M-connected if $\forall u, v \in S$, $u \leftrightarrow v$. Note that, if an HIN is M-connected, it must be connected, but not vice versa. Next, we formally define the MOCHI problem.

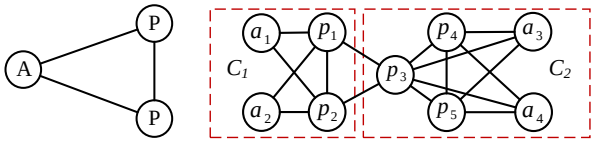
Problem 1. *(MOCHI Problem). Given an HIN H, a motif M, and a query vertex set $Q \subseteq V_H$, the MOCHI problem is to find the maximum subgraph $H[S] \subseteq H$ satisfying:*

(1) $Q \subseteq S$,

(2) $H[S]$ is M-connected,

(3) $MDM(H,S,M)$ is maximized.

In a word, the MOCHI problem returns the community that contains the query vertex set, is M-connected, and maximizes motif density modularity. Take the HIN H in Figure 3(a) and the motif M in Figure 3(b) as an example. Let $Q = \{a_1, p_5\}$, $S' = \{a_1, p_1, p_2, p_4, p_5\}$, and $S^* = \{a_1, a_2, p_3, p_4, p_5\}$. $MDM(H,S',M) = \frac{1}{5}(2 - 5 \times \frac{3}{5} \times \frac{8}{10} \times \frac{8}{10}) = 0.016$ and $MDM(H,S^*,M) = \frac{1}{5}(4 - 5 \times \frac{5}{5} \times \frac{8}{10} \times \frac{8}{10}) = 0.16$. Since $H[S^*]$ has the maximum MDM among all subgraphs, $H[S^*]$ is returned as the community.



(a) The motif M (b) Connected but not M-connected communities

Figure 4: An examples of M-connectivity

For the MOCHI problem, we would like to highlight two issues. (1) Why should the community be M-connected? Although a community is connected, it may not be M-connected, which is undesirable for communities. For example, the communities C_1 and C_2 in Figure 4(b) are connected but not M-connected in terms of the motif M in Figure 4(a). It is obvious that the authors a_1 and a_2 in the community C_1 do not collaborate with authors a_3 and a_4 in C_2. Therefore, we apply M-connectivity to ensure that the community is densely connected via motif instances. (2) The MOCHI problem takes the motif M as one of the inputs. Below, we provide some guidelines for *motif selection*. (i) *Predefined semantic motifs*: In some scenarios, users can employ known motifs with specific semantics. For instance, the motif M in Figure 2(b) represents credit card fraud [16] and we can use it for credit card fraud detection. (ii) *Custom motifs*: If users know the schema of HINs, they can design personalized motifs based on the schema. (iii) *Discovered motifs*: If users are unaware of the schema of HINs, they can leverage motif discovery techniques [60–62] to identify motifs in a given graph and use them as query motifs.

Theorem 2. *The MOCHI problem is NP-hard.*

Proof. We reduce the DMCS problem [14], which has been proved to be NP-hard, to the MOCHI problem. In particular, given a homogeneous network $G = (V_G, E_G)$, a query vertex set Q, the DMCS problem is to find a connected subgraph $G[S] \subseteq G$ that contains Q and has the maximum density modularity. Here, the density modularity is

$$DM(G,S) = \frac{1}{|S|}(|E_{G[S]}| - \frac{\texttt{vol}_G(S)^2}{4|E_G|}). \quad (4)$$

We show that the DMCS problem is a special case of the MOCHI problem. Specifically, for the MOCHI problem, if we set that (1) the vertices of HIN are of the same type and (2) the query motif M is an edge, the MOCHI problem is equivalent to the DMCS problem. Since the DMCS problem is NP-hard, the MOCHI problem is also NP-hard. A complete proof is available in our technical report [57]. □

4 ALGORITHMS

Due to the hardness of the MOCHI problem, finding an exact solution is computationally difficult. In this section, we first present a basic algorithm to solve the problem. Subsequently, we propose two more efficient algorithms.

4.1 Basic Algorithm

A naive solution to solving the MOCHI problem is to iteratively remove a vertex, which maximizes MDM after its removal and M-connectivity maintenance. Algorithm 1 outlines the pseudo-code of the basic algorithm. Specifically, Algorithm 1 first finds all motif

Algorithm 1 Basic Algorithm

Input: an HIN H, a motif M, and a query vertex set Q
Output: the M-connected subgraph containing Q with the maximum MDM

1: $\mathcal{I}_H^M \leftarrow$ find all motif instances of M in H;
2: $i \leftarrow 0$;
3: $S_i \leftarrow$ the vertices being M-connected to Q in H;
4: **while** $S_i \neq \emptyset$ **do**
5: $\quad MDM_{max} \leftarrow -\infty$; $S_{i+1} \leftarrow \emptyset$;
6: $\quad$ **for** *each vertex* $v \in S_i \setminus Q$ **do**
7: $\quad\quad S' \leftarrow$ the vertices being M-connected to Q after removing v from S_i;
8: $\quad\quad$ **if** $S' \neq \emptyset$ and $MDM(H, S', M) > MDM_{max}$ **then**
9: $\quad\quad\quad MDM_{max} \leftarrow MDM(H, S', M)$; $S_{i+1} \leftarrow S'$;
10: $\quad i \leftarrow i + 1$;
11: $S^* \leftarrow \text{argmax}_{S \in \{S_0, S_1, ..., S_{i-1}\}} MDM(H, S, M)$;
12: **return** $H[S^*]$;

instances of M in H and the M-connected subgraph that contains the query vertex set Q (lines 1–3). Then, in each round (lines 4-10), Algorithm 1 deletes each vertex to get a subgraph $H[S_i]$, which is M-connected and has maximal MDM. Since the function MDM is non-monotonic, $H[S^*]$ with the maximum MDM is returned (lines 11-12).

For Algorithm 1, we would like to highlight the issue of motif instances computation in line 1. Currently, there are many algorithms designed for subgraph match [63, 64], which can be used to compute the motif instances. Since the motif size is usually bounded from 3 to 7 in practice [54, 65, 66], in our implementation, we employ *RapidMatch* [67], a well-known join-based subgraph matching algorithm that is efficient for handling small query graphs. Next, we analyze the time complexity of Algorithm 1.

Complexity analysis. Let $\texttt{Mdeg}_H^{max}$ be the maximum motif degree of H. In each round of vertex removal, Algorithm 1 evaluates every vertex by temporarily removing it and maintaining the M-connectivity of the remaining HIN, which takes $O(\texttt{Mdeg}_H^{max} \times |V_M| \times |V_H|)$ time per vertex. Since there are $|V_H|$ vertices to consider in each round and up to $|V_H|$ rounds in total, the overall time complexity of Algorithm 1 is $O(|V_H|^3 \times \texttt{Mdeg}_H^{max} \times |V_M|)$. A detailed analysis is provided in our technical report [57].

4.2 MW-HIN-based Algorithm

The basic algorithm incurs significant overhead since it requires evaluating all vertices in each iteration. In addition, traversing the HIN to maintain M-connectivity is computationally expensive. To address these inefficiencies, we propose an efficient *vertex selection strategy* and a compact data structure called *motif-based weighted HIN* (MW-HIN) for M-connectivity checks. Based on these, we design our second algorithm.

4.2.1 Vertex Selection.

The basic algorithm examines the goodness of a vertex v by deleting v and the corresponding vertices that violate the M-connectivity and computing the MDM of the remaining HIN. This process is inefficient. We find that if only one vertex is removed, the MDM of the remaining HIN can be easily calculated.

Lemma 4. *Given an HIN H, a motif M, and a subgraph $H[S] \subseteq H$. For a vertex $v \in S$, if we delete v from $H[S]$, the MDM of $H[S \setminus \{v\}]$ is*

$$MDM(H, S \setminus \{v\}, M) = \frac{|\mathcal{I}_{H[S]}^M| - \texttt{Mdeg}_{H[S]}(v)}{|S| - 1} - \frac{|\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u))) - \texttt{Mdeg}_H(v)\eta(u,v)}{\texttt{Mvol}_H(V_H(\ell_M(u)))}}{|S| - 1}. \quad (5)$$

If $\ell_M(u) = \ell_H(v)$, $\eta(u, v) = 1$; otherwise, $\eta(u, v) = 0$.

Lemma 4 shows that the MDM of $H[S \setminus \{v\}]$ can be computed based on the MDM of $H[S]$, thereby improving the MDM computation efficiency. Motivated by Lemma 4, in line 8 of Algorithm 1, we can use $MDM(H, S \setminus \{v\}, M)$ to approximate $MDM(H, S', M)$. However, it overlooks the impact of additional vertices that may be removed during the M-connectivity maintenance phase. Therefore, a more natural approach is to select candidate vertices whose deletion does not disrupt the M-connectivity of the remaining vertices. To this end, we introduce two cases.

Case I: Single candidate vertex

Definition 5. *(Removable Vertex). Given an HIN H, a motif M, and a query vertex set $Q \subseteq V_H$. Let $H[S]$ be a subgraph of H satisfying that $H[S]$ is M-connected and contains Q. For a vertex $v \in S$, v is a removable vertex iff (1) $v \notin Q$; and (2) $H[S \setminus \{v\}]$ is M-connected.*

If we delete a removable vertex from the M-connected HIN, the remaining HIN is still M-connected. Thus, we can directly use Equation 5 to compute MDM. Take Figure 5(a) as an example, the removable vertices of HIN are $\{a_1, a_3\}$.

Case II: Batch candidate vertices

In Figure 5(a), after removing a_1 and a_3 from the HIN, the remaining vertices are not removable. But, if we remove some vertices together, e.g., $\{p_1, p_2\}$ or $\{p_3, p_4\}$, the remaining vertices are still M-connected. Based on the above observation, we introduce the second case.

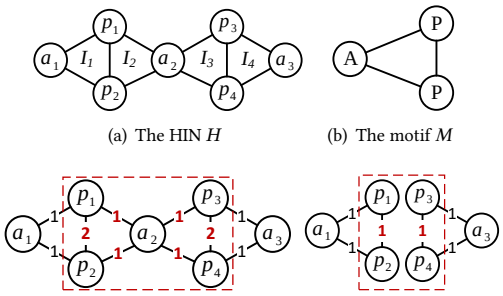
Definition 6. *(Equivalent Vertices). Given an HIN H, a motif M, and a vertex $v \in V_H$, the equivalent vertices of v in H are defined as $Eq_H^M(v) = \{u \mid u \in V_H, \mathcal{I}_H^M(u) = \mathcal{I}_H^M(v)\}$. Here, $\mathcal{I}_H^M(u)$ denotes all motif instances of M in H containing u.*

In other words, the equivalent vertices of v share the same motif instances as v. Note that (1) the equivalent vertices of v contain itself and (2) if v is a removable vertex, $Eq_H^M(v) = \{v\}$. If we delete v's equivalent vertices from HIN and the remaining HIN is still M-connected, the equivalent vertices of v are also candidate vertices. It is because their removal does not disrupt the M-connectivity of the remaining vertices.

Based on the above discussions, we briefly summarize our vertex selection strategy. Specifically, we first calculate $MDM(H, S \setminus \{v\}, M)$ for each vertex and sort the vertices in descending order of $MDM(H, S \setminus \{v\}, M)$. Then, we find the a vertex v such that (1) after deleting $Eq_H^M(v)$, the remaining HIN is still M-connected, and (2) $MDM(H, S \setminus \{v\}, M)$ is maximal.

4.2.2 Motif-based Weighted HIN.

After deleting $Eq_H^M(v)$, a straightforward way to check the M-connectivity of the remaining HIN is to traverse the remaining HIN via motif instances, which is computationally inefficient. To address this, we design a compact data structure *motif-based weighted HIN*,



(a) The HIN H (b) The motif M

(c) The MW-HIN H_M of H and M (d) H_M after removing a_2

Figure 5: An example of the MW-HIN.

which explicitly encodes M-connectivity relationships between vertices via edge weights for faster M-connectivity checks. Formally, the motif-based weighted HIN is defined as follows.

Definition 7. *(Motif-based Weighted HIN (MW-HIN)). Given an HIN H, a motif M, and a motif instance set $\mathcal{I}_H^M$, the MW-HIN of M and H is $H_M = (V_{H_M}, E_{H_M}, L_{H_M}, \ell_{H_M}, \omega_{H_M})$. Specifically, $V_{H_M} = \{v \mid v \in V_H \wedge \mathcal{I}_H^M(v) \neq \emptyset\}$, $E_{H_M} = \{(u,v) \mid u,v \in V_{H_M}, \mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(v) \neq \emptyset\}$, $L_{H_M} = L_H$, and $\ell_{H_M} = \ell_H$. ω_{H_M} is a weight mapping function to assign weight for each edge in E_{H_M}, i.e., $\forall (u,v) \in E_{H_M}$, $\omega_{H_M}(u,v) = |\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(v)|$.*

The MW-HIN consists of vertices that have motif instances in the HIN. The edge between two vertices indicates they share some common motif instances, which is quantified by the edge weight. For instance, Figure 5(c) shows the MW-HIN of HIN and motif in Figures 5(a) and 5(b), respectively. Since p_1 and p_2 share two common motif instances I_1 and I_2, in MW-HIN, the edge weight of (p_1, p_2) is 2. Based on MW-HIN, we can easily check the M-connectivity of the corresponding HIN. First, we introduce the following lemma to ensure correctness.

Lemma 5. *Given an HIN H, a motif M, and an MW-HIN H_M of M and H, H is M-connected iff (1) $\forall v \in V_H$, $\mathtt{Mdeg}_H(v) \geqslant 1$, and (2) H_M is connected.*

In other words, if the motif degrees of all vertices are no less than 1 and the MW-HIN is connected, the HIN is M-connected. For example, in Figure 5(c), since the MW-HIN satisfies the above two constraints, the corresponding HIN is M-connected. If we delete the vertex a_2 from the HIN, the corresponding MW-HIN is shown in Figure 5(d), which is not connected. Thus, we can infer that the deletion of a_2 will result in the HIN being M-disconnected.

Construction of MW-HIN. The MW-HIN can be constructed by visiting all the motif instances. For any two vertices within each motif instance, if they do not have an edge in the MW-HIN, we create an edge between them and mark the edge weight as one. Otherwise, we increase the edge weight by one.

4.2.3 MW-HIN-based Algorithm.

Based on the vertex selection and motif-based weighted HIN, we propose the second algorithm, called MW-HIN-based algorithm. The basic idea of the MW-HIN-based algorithm is as follows. It first finds all motif instances of M and constructs the MW-HIN H_M. Then, it finds M-connected component containing Q with H_M. Next, in each round, it selects a vertex v to delete $Eq_H^M(v)$ such that (1) after deleting $Eq_H^M(v)$, the remaining HIN is still M-connected, and (2) $MDM(H, S \setminus \{v\}, M)$ is maximal. The algorithm continues until no vertex can be deleted. Note that we employ MW-HIN H_M to check whether the deletion of $Eq_H^M(v)$ will lead the remaining HIN to be M-disconnected. Specifically, we maintain the MW-HIN by recomputing the weight of influenced edges after deleting $Eq_H^M(v)$. Once the edge weight is zero, we remove this edge from the MW-HIN. If the updated MW-HIN is not connected, the deletion of $Eq_H^M(v)$ will lead the remaining HIN to be M-disconnected. However, this method may suffer from two efficiency limitations: (1) edge weights of the MW-HIN are updated exactly every time for M-connectivity maintenance; (2) the vertex with high goodness but not a candidate vertex will be identified repeatedly across iterations. To address these two limitations, we propose two optimizations, namely, *lazy update* and *vertex skipping*.

Optimization 1: Lazy Update. The intuition behind the lazy update is that we only focus on whether the edge weight is greater than zero instead of the specific number. Therefore, the lazy update strategy aims to avoid the real-time exact updates of the edge weight, which is based on two crucial properties of MW-HIN, i.e., the *limited influence scope* and the *bounded influence strength*.

Lemma 6. *Given an HIN H, a motif M, an MW-HIN H_M of M and H, a vertex $v \in V_{H_M}$, and an edge $(u,w) \in E_{H_M}$ with $u, w \neq v$. Let H'_M be the MW-HIN obtained by deleting v from H_M, and $\mathcal{N}_{H_M}(v)$ be the neighbor set of v in H_M.*

(1) Limited Influence Scope: If $u \notin \mathcal{N}_{H_M}(v)$ or $w \notin \mathcal{N}_{H_M}(v)$, $\omega_{H_M}(u,w) = \omega_{H'_M}(u,w)$.

(2) Bounded Influence Strength: $0 \leq \omega_{H_M}(u,w) - \omega_{H'_M}(u,w) \leq |\mathcal{I}_H^M(v)|$.

Lemma 6 indicates that removing a vertex will only affect the weights of edges within its one-hop-induced subgraph in the MW-HIN. Furthermore, the weight change does not exceed the number of motif instances of the removed vertex. For example, in Figure 5(d), after removing a_2, the weights of edges within the red rectangle may change. Moreover, since the number of motif instances of a_2 is two, the change of weights for these edges will not exceed two.

Based on Lemma 6, if we delete the vertex v from H_M to get H'_M, we can update H_M as follows. First, the edges within v's one-hop neighbors induced subgraph in the MW-HIN are candidate edges whose weight may change. Next, we can use $\omega_{H_M}(u,w) - |\mathcal{I}_H^M(v)|$ to estimate the updated weight for each candidate edge (u,w). If $\omega_{H_M}(u,w) - |\mathcal{I}_H^M(v)| \leq 0$, we should compute the exact edge weight of (u,w) in H'_M and delete the edges whose exact edge weight is zero. Otherwise, we do not need to calculate the exact edge weight. Finally, the equivalent vertices of v and their incident edges should be deleted.

Optimization 2: Vertex Skipping. In each round, we should select a vertex v to delete $Eq_H^M(v)$ such that (1) after deleting $Eq_H^M(v)$, the remaining HIN is still M-connected, and (2) $MDM(H, S \setminus \{v\}, M)$ is maximal. Assume that in a certain round, we have examined vertex v and found that the removal of $Eq_H^M(v)$ will make some vertices M-disconnected. Then, we can use a reuse set $R(v)$ to record these M-disconnected vertices. In the following rounds, we can directly use $R(v)$ to judge whether $Eq_H^M(v)$ can be removed. Specifically, for

an HIN H, if $R(v)$ is not empty, $Eq_H^M(v)$ cannot be deleted from H. Note that in each round, the reuse set also should be updated by considering the following two cases. (1) If the deletion of $Eq_H^M(v)$ from H does not influence the M-connectivity of other vertices, the reuse set of all vertices should be updated as $R() \setminus Eq_H^M(v)$. (2) If the deletion of $Eq_H^M(v)$ from H makes a set of vertices S being M-disconnected from the query vertex set Q, only the reuse set of v should be updated as $R(v) \cup S$.

Incorporating the above two optimizations, Algorithm 2 shows the pseudo-code of the MW-HIN-based algorithm. Firstly, Algorithm 2 initializes the reuse set $R()$, finds all motif instances of M in H, constructs the MW-HIN H_M, and finds all vertices connected to Q in H_M (lines 1-4). Then, Algorithm 2 iteratively selects a vertex to delete according to the vertex selection strategy (lines 5-18). Specifically, Algorithm 2 first computes $MDM(H, S_i \setminus \{v\}, M)$ for each vertex in $v \in S_i \setminus Q$ and sorts them in descending order of $MDM(H, S_i \setminus \{v\}, M)$ (lines 6-7). Then, Algorithm 2 visits vertices in $S_i \setminus Q$ from the vertex with the maximum $MDM(H, S_i \setminus \{v\}, M)$ (line 8). For the visited vertex v, if the reuse set of v is not empty, v can be skipped since its removal will make some vertices M-disconnected from Q (line 9). Otherwise, Algorithm 2 computes equivalent vertices $Eq_{H[S_i]}^M(v)$ of v, removes them from $H_M[S_i]$, and lazily updates edges' weights of $H_M[S_i \setminus Eq_{H[S_i]}^M(v)]$ (lines 10-12). If the updated $H_M[S_i \setminus Eq_{H[S_i]}^M(v)]$ is connected, it means that the removal of v does not influence the other vertices' M-connectivity to Q. Hence, v is selected for deletion in this round. Algorithm 2 sets S_{i+1} by $S_i \setminus Eq_{H[S_i]}^M(v)$ and removes $Eq_{H[S_i]}^M(v)$ from the reuse set for the next round (lines 13-15). Otherwise, removing $Eq_{H[S_i]}^M(v)$ will break the M-connectivity of some vertices. Therefore, v cannot be removed in this round and Algorithm 2 adds the vertices disconnected from Q in $H_M[S_i \setminus Eq_{H[S_i]}^M(v)]$ into $R(v)$ for reuse in the following rounds (line 16-17). The vertex deletion continues until no more vertices can be deleted. Finally, $H[S_i]$ with the maximum MDM is returned (lines 19-20).

Complexity analysis. Let σ_{check} denote the number of vertex checks performed per round to determine candidate vertices, and β_{update} denote the number of exact edge weight updates after vertex removal. In each round, Algorithm 2 checks whether each vertex is a candidate vertex by temporarily removing it, updating the edge weight of MW-HIN, and checking the connectivity of the updated MW-HIN. The time complexity of edge weight update is $O(\beta_{\mathsf{update}} \times \mathtt{Mdeg}_H^{max} \times \log \mathtt{Mdeg}_H^{max})$ and the time complexity of checking connectivity based on MW-HIN is $O(|V_{H_M}| + |E_{H_M}|)$. Since there are up to $|V_{H_M}|$ rounds in total and each round examines σ_{check} vertices, the time complexity of Algorithm 2 is $O(|V_{H_M}| \times \sigma_{\mathsf{check}} \times (\beta_{\mathsf{update}} \times \mathtt{Mdeg}_H^{max} \times \log \mathtt{Mdeg}_H^{max} + |V_{H_M}| + |E_{H_M}|))$. A detailed analysis is provided in our technical report [57].

4.3 Motif-distance-based Algorithm

The MW-HIN-based algorithm improves upon the basic algorithm by incorporating vertex selection and MW-HIN techniques. However, its computational efficiency may degrade for large HINs with many motif instances and vertices. To overcome this, we propose a third algorithm designed to improve efficiency. Unlike the basic algorithm and MW-HIN-based algorithm, which rely solely on MDM for vertex deletion, this approach introduces a novel metric, *motif distance*, to guide vertex selection.

Algorithm 2 MW-HIN-based Algorithm

Input: an HIN H, a motif M, and a query vertex set Q
Output: the M-connected subgraph containing Q with the maximum MDM

1: $i \leftarrow 0$; $R() \leftarrow \emptyset$; //initialize the reuse set
2: $\mathcal{I}_H^M \leftarrow$ find all motif instances of M in H;
3: $H_M \leftarrow$ construct the MW-HIN with $\mathcal{I}_H^M$;
4: $S_i \leftarrow$ the vertices connected to Q in H_M;
5: **while** $S_i \neq \emptyset$ **do**
6: **for** *each vertex* $v \in S_i \setminus Q$ **do** compute $MDM(H, S_i \setminus \{v\}, M)$;
7: sort all vertices of $S_i \setminus Q$ in descending order of $MDM(H, S_i \setminus \{v\}, M)$;
8: **for** *each vertex* $v \in S_i \setminus Q$ **do**
9: **if** $R(v) \neq \emptyset$ **then continue**;
10: compute $Eq_{H[S_i]}^M(v)$;
11: $H_M[S_i \setminus Eq_{H[S_i]}^M(v)] \leftarrow$ remove $Eq_{H[S_i]}^M(v)$ from $H_M[S_i]$;
12: lazily update edges' weights of $H_M[S_i \setminus Eq_{H[S_i]}^M(v)]$;
13: **if** $H_M[S_i \setminus Eq_{H[S_i]}^M(v)]$ is connected **then**
14: $S_{i+1} \leftarrow S_i \setminus Eq_{H[S_i]}^M(v)$; $R() \leftarrow R() \setminus Eq_{H[S_i]}^M(v)$;
15: **break**;
16: **else**
17: $R(v) \leftarrow R(v) \cup$ (the vertices disconnected from Q in $H_M[S_i \setminus Eq_{H[S_i]}^M(v)]$);
18: $i \leftarrow i + 1$;
19: $S^* \leftarrow \mathrm{argmax}_{S \in \{S_0, S_1, \ldots, S_{i-1}\}} MDM(H, S, M)$;
20: **return** $H[S^*]$;

Definition 8. *(Motif Distance). Given an HIN H, a motif M, and two vertices $u, v \in V_H$, the motif distance of u, v in H, denoted by $\mathtt{Mdist}_H(u, v)$, is*

(1) if $u \leftrightarrow v$, $\mathtt{Mdist}_H(u, v) = \min\{n \mid \exists I_1, I_2, \ldots, I_n \in \mathcal{I}_H^M, u \in V_{I_1}, v \in V_{I_n}$, such that $V_{I_1} = V_{I_n}$ or $V_{I_i} \cap V_{I_{i+1}} \neq \emptyset (1 \leq i < n)\}$;
(2) if $u \nleftrightarrow v$, $\mathtt{Mdist}_H(u, v) = +\infty$.

In other words, if u and v are M-connected, the motif distance of u and v in H is the minimum number of motif instances that connect u and v. Otherwise, the motif distance of u and v is infinity. Based on Definition 8, the motif distance between the vertex $u \in V_H$ and the vertex set $S \subseteq V_H$ in H can be defined as $\mathtt{Mdist}_H(S, u) = \min_{v \in S}\{\mathtt{Mdist}_H(u, v)\}$. Take the HIN H in Figure 5(a) and the motif M in Figure 5(b) as an example. Let $S = \{a_2, a_3\}$. Since a_1 and a_2 are M-connected via motif instances I_1 and I_2, $\mathtt{Mdist}_H(a_1, a_2) = 2$. $\mathtt{Mdist}_H(S, a_1) = \min\{\mathtt{Mdist}_H(a_1, a_2), \mathtt{Mdist}_H(a_1, a_3)\} = 2$. Next, we show how to use the MW-HIN H_M of H to compute the motif distance. For any two vertices u, v in the MW-HIN H_M, we use $\mathtt{dist}_{H_M}(u, v)$ to denote the unweighted shortest path distance between u and v in H_M. We present the following lemma.

Lemma 7. *Given an HIN H, a motif M, and an MW-HIN H_M of M and H. $\forall u, v \in V_H$, if $u, v \in V_{H_M}$, $\mathtt{Mdist}_H(u, v) = \mathtt{dist}_{H_M}(u, v)$. Otherwise, $\mathtt{Mdist}_H(u, v) = +\infty$.*

Lemma 7 indicates that we can compute the motif distance by computing the distance between vertices in the MW-HIN. For example, in Figure 5(c), $\mathtt{dist}_{H_M}(a_1, a_2) = \mathtt{Mdist}_H(a_1, a_2) = 2$.

Based on the motif distance, we propose our third algorithm, named *motif-distance-based algorithm*. The design of the motif-distance-based algorithm is motivated by the small world theory

[68, 69], which shows that the vertices within the same cohesive community exhibit strong access locality [70]. In other words, the vertex close to the query vertices tends to be within the query vertices' corresponding community. Motivated by it, we try to remove the vertices farthest from query vertices w.r.t., motif distance. Specifically, the motif-distance-based algorithm consists of two phases: *coarse-grained deletion* and *fine-grained deletion*. Next, we introduce these two phases in detail.

Coarse-grained deletion. The coarse-grained deletion aims to narrow down the search space. Specifically, it iteratively deletes the vertices with the maximum motif distance from query vertices in batches until no vertices can be deleted. In this process, each time the farthest vertices are deleted, a subgraph of HIN can be generated. Then, the motif-distance-based algorithm selects the one with the maximal MDM for the fine-grained deletion. Since the returned community should be M-connected, the coarse-grained deletion should also ensure the M-connectivity of the remaining subgraph, which can be guaranteed by the following lemma.

Lemma 8. *Given a motif M, an M-connected HIN H. Let $S \subseteq V_H$, and $S' = \{v \in V_H \mid \texttt{Mdist}_H(S, v) = \max_{u \in V_H} \texttt{Mdist}_H(S, u)\}$. If $H[S]$ is M-connected and $S \subseteq V_H \setminus S'$, $H[V_H \setminus S']$ is M-connected .*

Lemma 8 provides guarantees of M-connectivity when removing vertices in batches, which avoids the M-connectivity checks. To find an M-connected subgraph $H[S_0]$ containing Q as small as possible, we adopt the well-known approximate Steiner tree algorithm [71]. The construction of $H[S_0]$ contains two steps. (1) Computing an approximate Steiner tree of Q in H_M to connect the query vertices. (2) Greedily adding the motif instances that share the most vertices with the approximate Steiner tree until finding a subgraph $H[S_0]$ that is M-connected.

Overall, the procedure of coarse-grained deletion is as follows. If $H[Q]$ is not M-connected, we constructed a small M-connected subgraph $H[S_0]$ containing Q. Then, we compute the motif distance between S_0 and other vertices in H_M. Next, we iteratively remove the vertices farthest from S_0 in batches and return the subgraph with the maximal MDM.

Fine-grained deletion. The coarse-grained deletion removes the vertices farthest from the query vertices in batches, which may miss some subgraphs with large MDM. To address this issue, we design the fine-grained deletion to fine-tune the subgraph returned by the coarse-grained deletion. Specifically, given the subgraph obtained from the coarse-grained deletion, we further refine it by iteratively removing the vertices farthest from the query vertices one by one. After each vertex's removal, we also eliminate the influenced vertices whose motif instances have all been removed. We repeat this procedure until the resulting HIN reduces to $H[S_0]$, and return the subgraph with the maximum MDM encountered during the process.

For the fine-grained deletion, a key question is how to select a vertex to delete if there are many vertices with the same motif distance to S_0. A straightforward option is the MDM after each vertex removal. However, it varies for the change of current HIN, leading to a huge update overhead. Hence, we design a lightweight goodness function called *M-ratio* to efficiently guide the vertex selection in the fine-grained deletion process. Note that the M-ratio is motivated by the density ratio in homogeneous networks [14],

Algorithm 3 Motif-distance-based Algorithm

Input: an HIN H, a motif M, and a query vertex set Q
Output: the M-connected subgraph containing Q with the maximum MDM

1: $\mathcal{I}_H^M \leftarrow$ collect the motif instances of M in H;
2: $H_M \leftarrow$ construct the MW-HIN with $\mathcal{I}_H^M$;
3: $S_0 \leftarrow$ expand Q to find a small M-connected subgraph;
4: **for** each vertex $v \in S \setminus S_0$ **do**
5: compute $\texttt{Mdist}_H(S_0, v)$;
6: $S \leftarrow$ the vertices connected to S_0 in H_M;
7: $S^* \leftarrow S$;
8: **while** $S \neq S_0$ **do**
9: $S' \leftarrow \text{argmax}_{v \in S} \texttt{Mdist}_H(S_0, v)$;
10: $S \leftarrow S \setminus S'$;
11: **if** $MDM(H, S, M) > MDM(H, S^*, M)$ **then** $S^* \leftarrow S$;
12: $S \leftarrow S^*$;
13: **for** each vertex $v \in S$ **do**
14: compute Θ_v^S;
15: **while** $S \neq S_0$ **do**
16: $S' \leftarrow \text{argmax}_{v \in S} \texttt{Mdist}_H(S_0, v)$;
17: **while** $S' \neq \emptyset$ **do**
18: $u \leftarrow \text{argmax}_{v \in S'} \Theta_v^S$;
19: **for** each motif instance $I \in \mathcal{I}_{H[S]}^M(u)$ **do**
20: **for** each vertex $v \in V_I$ **do**
21: $\mathcal{I}_{H[S]}^M(v) \leftarrow \mathcal{I}_{H[S]}^M(v) \setminus \{I\}$;
22: Update Θ_v^S;
23: **if** $\mathcal{I}_{H[S]}^M(v) = \emptyset$ **then**
24: $S \leftarrow S \setminus \{v\}$; $S' \leftarrow S' \setminus \{v\}$;
25: **if** $MDM(H, S, M) > MDM(H, S^*, M)$ **then** $S^* \leftarrow S$;
26: **return** $H[S^*]$;

which has been used to identify the vertex removal order effectively and efficiently to find a community with maximum density modularity in [14].

Definition 9. *(M-ratio). Given an HIN H, a motif M, a set of vertices $S \subseteq V_H$, and a vertex $v \in S$, the M-ratio of v in $H[S]$ is defined as*

$$\Theta_v^S = \frac{\texttt{Mdeg}_H(v)}{\texttt{Mdeg}_{H[S]}(v)}. \tag{6}$$

The M-ratio quantifies the tendency of vertices to form motif instances within $H[S]$. Intuitively, a smaller $\texttt{Mdeg}_{H[S]}(v)$ suggests that removing the vertex has less impact on the number of motif instances within the community. In contrast, a larger $\texttt{Mdeg}_H(v)$ indicates a higher probability of the vertex forming motif instances in a random network, thereby exerting a greater influence on the expected number of motif instances. Therefore, removing the vertex with a larger M-ratio in the community is more likely to maximize MDM. Next, we show how to update the M-ratio.

Lemma 9. *Given an HIN H, a motif M, an M-connected subgraph $H[S] \subseteq H$, and a vertex $u \in S$. Let $S' = S \setminus \{u\}$. For a vertex $v \in S'$, if $\mathcal{I}_{H[S]}^M(v) \cap \mathcal{I}_{H[S]}^M(u) = \emptyset$, $\Theta_v^S = \Theta_v^{S'}$.*

Lemma 9 reveals that after removing a vertex u, we only need to recompute M-ratios of the vertices that share the same motif instances with u.

Based on the above introductions, the coarse-grained deletion and fine-grained deletion constitute the motif-distance-based algorithm, whose pseudo-code is outlined in Algorithm 3. It first computes the motif instances and constructs the MW-HIN of H (lines

1-2). Then, Algorithm 3 expands Q to find a small M-connected subgraph $H[S_0]$ using approximate Steiner tree algorithm, computes the motif distance of other vertices to S_0, finds the vertices connected to S_0 in H_M, and initializes the optimal community vertex set S^* (lines 3-7). Next, Algorithm 3 performs coarse-grained deletion by removing vertices farthest from S_0 in batches to identify a promising subgraph with maximal MDM (lines 8-12). After that, Algorithm 3 performs fine-grained deletion (lines 13-25). In particular, it firstly computes the M-ratio for each vertex in $H[S]$ (lines 13-14). Then, Algorithm 3 identifies the vertices farthest from S_0 and removes them in descending order of M-ratios (lines 18-24). If $MDM(H, S, M) > MDM(H, S^*, M)$, S^* should be updated (line 25). Algorithm 3 iteratively removes vertices until $S = S_0$. Finally, $H[S^*]$ is returned as the result (line 26).

Complexity analysis. Algorithm 3 consists of a coarse-grained phase and a fine-grained phase. In the coarse-grained phase, Algorithm 3 takes $O(\texttt{Mdeg}_H^{max} \times |V_M|)$ time to remove each vertex and update the motif instance sets of influenced vertices. This phase runs for $|V_{H_M}|$ iterations. In the fine-grained phase, after removing each vertex with the maximum M-ratio, it takes $O(\texttt{Mdeg}_H^{max} \times |V_M|)$ time to update the M-ratios of the influenced vertices. For each influenced vertex, it takes $O(\log |V_{H_M}|)$ to maintain the order of M-ratio set. In total, the time complexity of the fine-grained phase is $O(|V_{H_M}| \times \texttt{Mdeg}_H^{max} \times |V_M| \times \log |V_{H_M}|)$. Overall, the time complexity of Algorithm 3 is $O(|V_{H_M}| \times \log |V_{H_M}| \times \texttt{Mdeg}_H^{max} \times |V_M|)$. A detailed analysis is provided in our technical report [57].

5 EXPERIMENTS

In this section, we evaluate the effectiveness and efficiency of our proposed model and algorithms on real-world HINs. All algorithms are implemented in C++ and the experiments are conducted on a Linux machine with 2.2GHz CPU and 128 GB memory.

5.1 Experimental Setup

Datasets. We use four real-world HINs in experiments. Specifically, *CIDeRplus* [72] is a biological network including chemicals, genes, diseases, etc. *TMDB* [10] is a movie network consisting of movies, directors, and countries. *Freebase* [73] and *DBpedia* [70] are knowledge graphs. The statistics of these datasets are summarized in Table 1.

Competitors. We compare a set of methods in experiments.

- MM [13]: a higher-order clustering method with traditional random graph model. To apply MM to our problem, we replace the MDM with MM in the MW-HIN-based algorithm.
- RC [6]: a heterogeneous community model that finds the community satisfying relation constraints. To apply RC to our problem, we decompose each motif into the relational constraints and return the maximal relation community containing Q.
- GMM, MDM: our proposed generalized motif modularity and motif density modularity.
- Basic, MW, and MD: our proposed three algorithms, i.e., the basic algorithm, the MW-HIN-based algorithm, and the motif-distance-based algorithm.

Parameters. The parameters include motif size $|V_M|$, query vertex set size $|Q|$, and graph cardinality $|V_H|$. The ranges of $|V_M|$, $|Q|$, and $|V_H|$% are {3, <u>4</u>, 5, 6, 7}, {<u>1</u>, 2, 4, 6, 8}, and {60%, 70%, 80%, 90%, <u>100%</u>}, respectively, where the underlined numbers denote the default values. Following previous works [65, 74], we generate random motifs with sizes ranging from 3 to 7 by performing random walks on each dataset. The structures of the found motifs are in line with [74]. In addition, the query vertex sets are randomly generated while ensuring that the query vertex sets are M-connected such that the result is non-empty. Overall, in each experiment, we generate 200 queries and report the average performance.

Table 1: Dataset statistics (*dens*: density, *deg*: average degree)

Dataset	$\|V_H\|$	$\|E_H\|$	$\|L_H\|$	$dens$	deg
CIDeRplus	13,924	83,916	15	6.03	12.05
TMDB	71,978	113,581	7	1.58	3.16
Freebase	3,993,552	10,998,482	3,238	2.75	5.51
DBpedia	4,521,912	14,039,200	439	3.10	6.21

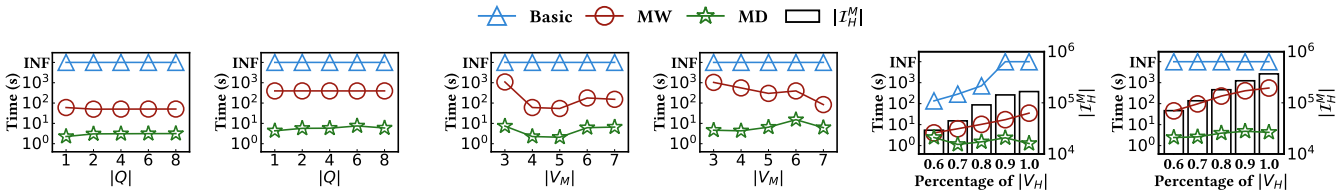
Table 2: Quality comparison of different models

Dataset	Method	Diam	Sim	Coh	$\|C\|$
CIDeRplus	RC	6.15	0.4225	0.0023	1020.10
	MM	5.36	0.4042	0.0019	1120.00
	GMM	5.06	0.4340	0.0039	690.16
	MW	**3.49**	**0.5035**	<u>0.02177</u>	**136.65**
	MD	<u>4.42</u>	<u>0.4644</u>	**0.0377**	<u>416.80</u>
TMDB	RC	11.00	0.6178	0.0002	15070.45
	MM	9.70	0.6145	0.0005	6944.10
	GMM	9.49	0.6567	0.0012	4609.96
	MW	<u>7.85</u>	**0.7788**	**0.0261**	**212.10**
	MD	**5.94**	<u>0.7585</u>	<u>0.0217</u>	<u>563.29</u>
Freebase	RC	7.51	0.8024	0.0224	5152.41
	MM	5.38	0.8089	0.0250	3823.56
	GMM	4.64	0.8287	0.0260	1420.02
	MW	<u>3.52</u>	<u>0.8668</u>	<u>0.0496</u>	**171.48**
	MD	**3.05**	**0.8692**	**0.1323**	<u>645.14</u>
DBpedia	RC	11.38	0.7728	0.0224	12520.06
	MM	9.32	0.7868	0.0259	9537.76
	GMM	8.11	0.8066	0.0264	4050.69
	MW	<u>6.75</u>	<u>0.8418</u>	<u>0.0594</u>	**198.66**
	MD	**4.29**	**0.8446**	**0.1994**	<u>1379.91</u>

5.2 Effectiveness Evaluation

In this section, we evaluate the quality of different models. Due to the lack of ground-truth communities over HINs, we follow the existing works [6–10] to evaluate the quality of communities in terms of *diameter*, *similarity*, *cohesiveness*, and *community size*.

- **Diameter**: the maximum shortest distance between two vertices in the community C, i.e., $\texttt{Diam}(C) = max_{u,v \in C}\{\texttt{dist}_H(u, v)\}$, where $\texttt{dist}_H(u, v)$ is the shortest distance between vertices u, v in the HIN H.
- **Similarity**: the average similarity of vertices in the community C. The similarity of two vertices u and v with the same type is $\texttt{Sim}(u, v) = \frac{|\texttt{NL}(u) \cap \texttt{NL}(v)|}{|\texttt{NL}(u) \cup \texttt{NL}(v)|}$. For a vertex u, $\texttt{NL}(u)$ denotes a multi-set of its neighbors' types.
- **Cohesiveness**: $\texttt{Coh}(C) = \frac{1}{|C|^2} \sum_{u,v \in C} \frac{|\mathcal{I}^M_{H[C]}(u) \cap \mathcal{I}^M_{H[C]}(v)|}{|\mathcal{I}^M_{H[C]}(u) \cup \mathcal{I}^M_{H[C]}(v)|}$.
- **Community size**: the number of vertices in the community.



(a) Freebase (b) DBpedia

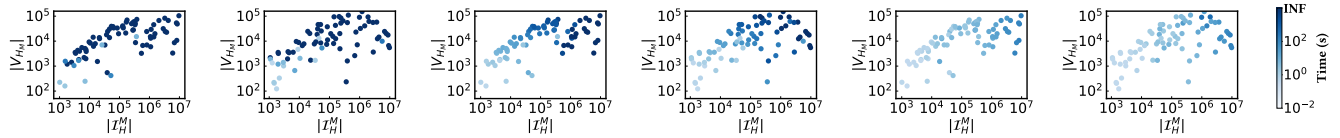
Figure 6: Effect of $|Q|$

(a) Freebase (b) DBpedia

Figure 7: Effect of $|V_M|$

(a) Freebase (b) DBpedia

Figure 8: Effect of $|V_H|$



(a) Basic (Freebase) (b) Basic (DBpedia) (c) MW (Freebase) (d) MW (DBpedia) (e) MD (Freebase) (f) MD (DBpedia)

Figure 9: Effect of $|\mathcal{I}_H^M|$ and $|V_{H_M}|$

Table 3: Ablation study on MW

Models	Running time (sec)				Average speedup
	CIDeRplus	TMDB	Freebase	DBpedia	
MW	0.15	0.92	2.26	3.02	+94%
w/o V	5.83	43.95	46.41	16.51	+17%
w/o M	0.34	4.90	2.37	5.56	+88%
w/o V&M	6.15	70.78	52.69	18.75	-

Table 4: Ablation study on MD

Models	Running time (sec)				Average speedup
	CIDeRplus	TMDB	Freebase	DBpedia	
MD	2.04	6.52	2.10	5.86	+68%
w/o C	1.82	14.51	2.89	14.67	+62%
w/o R	2.20	11.51	2.99	7.14	+61%
w/o C&R	2.26	62.57	9.22	95.50	-

Exp-1: Community quality comparison. Firstly, we compare the community quality returned by RC [6], MM [13], GMM, MW, and MD. Since Basic always fails to finish within the time limit (10^4 seconds), we do not report its results here. Table 2 shows the results over four HINs. We can see that our proposed methods MW and MD outperform other competitors w.r.t. all metrics. The good performance of MW and MD can be attributed to the design of MDM, which integrates the higher-order structure and heterogeneous information within the motif-based random graph model. Besides, the MDM incorporates motif density to enhance the cohesiveness of the community, which effectively alleviates the free-rider effect as well as the resolution limit. Additionally, we observe that MD achieves performance comparable to, or even better than, MW. This is because MD incorporates the motif distance to eliminate vertices that are irrelevant to the query vertices, thereby enhancing the quality of the resulting community.

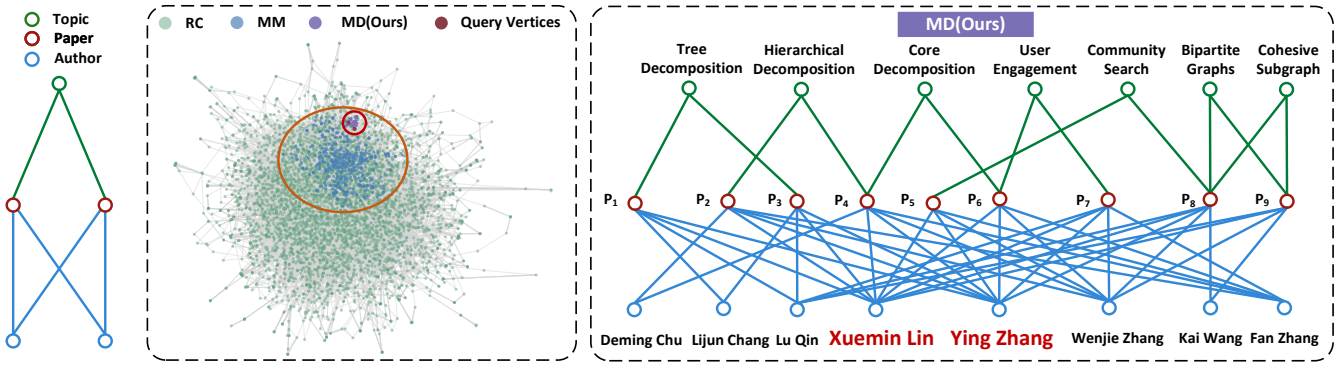
5.3 Efficiency Evaluation

In this section, we evaluate the efficiency of our proposed algorithms in terms of running time over Freebase and DBpedia. Due to space limitations, the results of other datasets can be found in our technical report [57]. Note that if a query does not complete within 10^4 seconds, we denote it by "INF".

Exp-2: Effect of $|Q|$. We vary the number of query vertices $|Q|$ from 1 to 8 to test its effect on algorithms. The running times of the three algorithms are shown in Figure 6. We can observe that Basic fails to finish within the running time limit, while MW and MD are at least 1 - 3 orders of magnitude faster than Basic. This is because MW incorporates the vertex selection, MW-HIN, and optimizations to improve the efficiency, and MD removes vertices in batches with a lightweight goodness function M-ratio. In addition, the running time of MW is generally stable when varying $|Q|$, while the running time of MD increases slightly. It is because for larger $|Q|$, MD takes more time to construct a Steiner tree to find a small M-connected subgraph containing Q.

Exp-3: Effect of $|V_M|$. We study the effect of motif size $|V_M|$ on three algorithms. Figure 7 reports the running time of three algorithms w.r.t. different motif sizes. The running time of MW and MD fluctuates without a clear trend. This is because the running time of algorithms is mainly determined by the number of motif instances $|\mathcal{I}_H^M|$ and the number of vertices associated with motif instances $|V_{H_M}|$. Even for the motifs with the same size, their structures may vary greatly, leading to different $|\mathcal{I}_H^M|$ and $|V_{H_M}|$. Additionally, the MD is more stable than MW because it narrows down the search space of the community by coarse-grained deletion, therefore influenced less by the change of $|\mathcal{I}_H^M|$ and $|V_{H_M}|$.

Exp-4: Scalability. To evaluate the scalability of our proposed algorithms, we extract different fractions of vertices from the original HINs to generate the induced subgraphs with different sizes. Figure 8 shows the results. For Freebase, the running time of Basic increases rapidly. For DBpedia, the Basic fails to finish within the time limit. In contrast, the running time of MW increases stably when varying the fraction of vertices, while MD fluctuates slightly. This is because when HIN becomes larger, $|\mathcal{I}_H^M|$ and $|V_{H_M}|$ also increase, resulting in a larger and denser MW-HIN. However, when $|\mathcal{I}_H^M|$ and $|V_{H_M}|$ increases, a subgraph with a larger MDM may be formed in the neighborhood of the query vertices, which can be located by the coarse-grained deletion instead of removing vertices iteratively. Therefore, the running time of MD may even decrease as shown in Figure 8(a).



(a) Motif

(b) Communities of RC, MM, and MD

Figure 10: A case study on DBLP.

Exp-5: Effect of $|\mathcal{I}_H^M|$ and $|V_{H_M}|$. As mentioned in previous experiments, the running time of algorithms is mainly determined by the number of motif instances $|\mathcal{I}_H^M|$ and the number of vertices associated with motif instances $|V_{H_M}|$. In this experiment, we explore the effect of $|\mathcal{I}_H^M|$ and $|V_{H_M}|$ on three algorithms. However, $|\mathcal{I}_H^M|$ and $|V_{H_M}|$ cannot be set directly, which are determined by the motif structure. To this end, we generate a set of different motifs. In each experiment, we record $|\mathcal{I}_H^M|$, $|V_{H_M}|$, and the corresponding running time. Figure 9 shows the results. Note that the colored points in the figure represent the running time. The darker the color, the longer the running time. We can observe that the running time of the three algorithms increases with the growth of $|\mathcal{I}_H^M|$ and $|V_{H_M}|$. In addition, Basic is unable to complete the community search within the running time limit when $|\mathcal{I}_H^M| > 10^5$ and $|V_{H_M}| > 10^4$, and MW fails to finish when $|\mathcal{I}_H^M| > 10^6$ and $|V_{H_M}| > 10^5$. However, MD can still efficiently finish the community search for ten million motif instances, demonstrating the efficiency of MD.

5.4 Ablation Studies

Exp-6: Ablation study of MW. In this experiment, we evaluate the effectiveness of the vertex selection strategy and MW-HIN proposed in Section 4.2. Table 3 shows the results. Specifically, w/o V refers to MW without the vertex skipping strategy, w/o M refers to MW without applying MW-HIN, and w/o V&M denotes MW without both of these components. The results show that the vertex selection strategy and MW-HIN can improve the average performance of w/o V&M by 88% and 17%, respectively. When these optimizations are combined, they improve the overall performance of w/o V&M by 94%. The results confirm the effectiveness of the vertex selection strategy and MW-HIN.

Exp-7: Ablation study of MD. In this experiment, we evaluate the effectiveness of the coarse-grained deletion and M-ratio proposed in Section 4.3. Table 4 shows the results. Specifically, w/o C refers to MD without the coarse-grained deletion, and w/o R refers to MD without applying the M-ratio, but still utilizing $MDM(H, S\setminus\{v\}, M)$ as the vertex goodness function. Meanwhile, w/o C&R denotes MD without both of these components. The results show that the coarse-grained deletion and M-ratio can improve the average performance by 61% and 62%, respectively. Furthermore, when both techniques are combined, the overall performance of w/o C&R can be improved by 68%. These results demonstrate the effectiveness of coarse-grained deletion and M-ratio in enhancing MOCHI's community search performance.

5.5 Case Study

In this section, we conduct a case study on a DBLP HIN, which is extracted from the DBLP network [1]. Specifically, we select publications from top-tier conferences in the fields of Database, Information Retrieval, Artificial Intelligence, Data Mining, and Computer Vision from 2020 to 2022. The resulting HIN contains 64,891 vertices and 122,111 edges. We aim to identify an academic collaboration community involving Prof. Xuemin Lin and Prof. Ying Zhang, two prominent database researchers. This community consists of authors who have co-authored at least two papers on the same topic with other members, along with their collaborative papers and associated research topics. To this end, we use Lin and Zhang as query vertices and set the motif in Figure 10(a). Then we apply RC [6], MM [13], and our motif-distance-based method MD to search the community, respectively. Figure 10(b) shows the community returned by each method. The communities from RC and MM are oversized, lacking structural and semantic cohesiveness. In contrast, MD identifies a cohesive group of researchers frequently collaborating with Lin and Zhang, along with their papers, and key topics (e.g., core decomposition, community search, cohesive subgraph). This demonstrates MD's effectiveness for real-world HIN community search.

6 CONCLUSIONS

This paper studies a new motif-based community search (MOCHI) problem over large HINs, which aims to find a cohesive heterogeneous community satisfying the semantics of a specified motif. To capture structure and semantic cohesion, we propose a novel community model named motif density modularity (MDM). Based on MDM, we formulate the MOCHI problem and prove its NP-hardness. To tackle this problem, we propose three algorithms: a basic method, an MW-HIN-based method, and a motif-distance-based method. Extensive experiments on real-world HINs demonstrate that MOCHI efficiently finds communities with high similarity and motif cohesiveness. In the future, we will explore community search over more complex HINs, such as knowledge graphs.

REFERENCES

[1] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998, 2008.

[2] https://www.imdb.com/.

[3] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. Heterogeneous network representation learning: A unified framework with survey and benchmark. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4854–4873, 2020.

[4] Thomas Rebele, Fabian Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames. In *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference*, pages 177–185, 2016.

[5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[6] Xun Jian, Yue Wang, and Lei Chen. Effective and efficient relational community detection and search in large dynamic heterogeneous information networks. *Proceedings of the VLDB Endowment*, 13(10):1723–1736, 2020.

[7] Yixiang Fang, Yixing Yang, Wenjie Zhang, Xuemin Lin, and Xin Cao. Effective and efficient community search over large heterogeneous information networks. *Proceedings of the VLDB Endowment*, 13(6):854–867, 2020.

[8] Yixing Yang, Yixiang Fang, Xuemin Lin, and Wenjie Zhang. Effective and efficient truss computation over large heterogeneous information networks. In *2020 IEEE 36th international conference on data engineering (ICDE)*, pages 901–912, 2020.

[9] Yangqin Jiang, Yixiang Fang, Chenhao Ma, Xin Cao, and Chunshan Li. Effective community search over large star-schema heterogeneous information networks. *Proceedings of the VLDB Endowment*, 15(11):2307–2320, 2022.

[10] Yingli Zhou, Yixiang Fang, Wensheng Luo, and Yunming Ye. Influential community search over large heterogeneous information networks. *Proceedings of the VLDB Endowment*, 16(8):2047–2060, 2023.

[11] Yuqi Li, Guosheng Zang, Chunyao Song, Xiaojie Yuan, and Tingjian Ge. Leveraging semantic information for enhanced community search in heterogeneous graphs. *Data Science and Engineering*, pages 1–18, 2024.

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

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

[14] Junghoon Kim, Siqiang Luo, Gao Cong, and Wenyuan Yu. Dmcs: Density modularity based community search. In *Proceedings of the 2022 International Conference on Management of Data*, pages 889–903, 2022.

[15] Zijin Feng, Miao Qiao, and Hong Cheng. Modularity-based hypergraph clustering: Random hypergraph model, hyperedge-cluster relation, and computation. *Proceedings of the ACM on Management of Data*, 1(3):1–25, 2023.

[16] Xiafei Qiu, Wubin Cen, Zhengping Qian, You Peng, Ying Zhang, Xuemin Lin, and Jingren Zhou. Real-time constrained cycle detection in large dynamic graphs. *Proceedings of the VLDB Endowment*, 11(12):1876–1888, 2018.

[17] Yixiang Fang, Xin Huang, Lu Qin, Ying Zhang, Wenjie Zhang, Reynold Cheng, and Xuemin Lin. A survey of community search over big graphs. *The VLDB Journal*, 29:353–392, 2020.

[18] Xin Huang, Laks V. S. Lakshmanan, and Jianliang Xu. *Community Search over Big Graphs*. Synthesis Lectures on Data Management. 2019.

[19] Mauro Sozio and Aristides Gionis. The community-search problem and how to plan a successful cocktail party. In *SIGKDD*, pages 939–948, 2010.

[20] Wanyun Cui, Yanghua Xiao, Haixun Wang, and Wei Wang. Local search of communities in large graphs. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 991–1002, 2014.

[21] Esra Akbas and Peixiang Zhao. Truss-based community search: a truss-equivalence based indexing approach. *Proceedings of the VLDB Endowment*, 10(11):1298–1309, 2017.

[22] Xin Huang, Laks VS Lakshmanan, Jeffrey Xu Yu, and Hong Cheng. Approximate closest community search in networks. *Proceedings of the VLDB Endowment*, 9(4), 2015.

[23] Lijun Chang, Xuemin Lin, Lu Qin, Jeffrey Xu Yu, and Wenjie Zhang. Index-based optimal algorithms for computing steiner components with maximum connectivity. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 459–474, 2015.

[24] Jiafeng Hu, Xiaowei Wu, Reynold Cheng, Siqiang Luo, and Yixiang Fang. On minimal steiner maximum-connected subgraph queries. *IEEE Transactions on Knowledge and Data Engineering*, 29(11):2455–2469, 2017.

[25] Wanyun Cui, Yanghua Xiao, Haixun Wang, Yiqi Lu, and Wei Wang. Online search of overlapping communities. In *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, pages 277–288, 2013.

[26] Long Yuan, Lu Qin, Wenjie Zhang, Lijun Chang, and Jianye Yang. Index-based densest clique percolation community search in networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(5):922–935, 2017.

[27] Jun Gao, Jiazun Chen, Zhao Li, and Ji Zhang. Ics-gnn: lightweight interactive community search via graph neural network. *Proceedings of the VLDB Endowment*, 14(6):1006–1018, 2021.

[28] Yuli Jiang, Yu Rong, Hong Cheng, Xin Huang, Kangfei Zhao, and Junzhou Huang. Query driven-graph neural networks for community search: from non-attributed, attributed, to interactive attributed. *Proceedings of the VLDB Endowment*, 15(6):1243–1255, 2022.

[29] Ling Li, Siqiang Luo, Yuhai Zhao, Caihua Shan, Zhengkui Wang, and Lu Qin. Coclep: Contrastive learning-based semi-supervised community search. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2483–2495, 2023.

[30] Jianwei Wang, Kai Wang, Xuemin Lin, Wenjie Zhang, and Ying Zhang. Neural attributed community search at billion scale. *Proceedings of the ACM on Management of Data*, 1(4):1–25, 2024.

[31] Yuxiang Wang, Xiaoxuan Gou, Xiaoliang Xu, Yuxia Geng, Xiangyu Ke, Tianxing Wu, Zhiyuan Yu, Runhuai Chen, and Xiangying Wu. Scalable community search over large-scale graphs based on graph transformer. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1680–1690, 2024.

[32] Qing Liu, Minjun Zhao, Xin Huang, Jianliang Xu, and Yunjun Gao. Truss-based community search over large directed graphs. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2183–2197, 2020.

[33] Yixiang Fang, Zhongran Wang, Reynold Cheng, Hongzhi Wang, and Jiafeng Hu. Effective and efficient community search over large directed graphs. *IEEE Transactions on Knowledge and Data Engineering*, 31(11):2093–2107, 2018.

[34] Xin Huang and Laks VS Lakshmanan. Attribute-driven community search. *Proceedings of the VLDB Endowment*, 10(9):949–960, 2017.

[35] Qing Liu, Yifan Zhu, Minjun Zhao, Xin Huang, Jianliang Xu, and Yunjun Gao. Vac: vertex-centric attributed community search. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 937–948, 2020.

[36] Kai Wang, Wenjie Zhang, Xuemin Lin, Ying Zhang, Lu Qin, and Yuting Zhang. Efficient and effective community search on large-scale bipartite graphs. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 85–96, 2021.

[37] Zi Chen, Yiwei Zhao, Long Yuan, Xuemin Lin, and Kai Wang. Index-based biclique percolation communities search on bipartite graphs. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 2699–2712, 2023.

[38] Xiaoye Miao, Yue Liu, Lu Chen, Yunjun Gao, and Jianwei Yin. Reliable community search on uncertain graphs. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1166–1179, 2022.

[39] Wensheng Luo, Xu Zhou, Kenli Li, Yunjun Gao, and Keqin Li. Efficient influential community search in large uncertain graphs. *IEEE Transactions on Knowledge and Data Engineering*, 35(4):3779–3793, 2021.

[40] Michael Yu, Dong Wen, Lu Qin, Ying Zhang, Wenjie Zhang, and Xuemin Lin. On querying historical k-cores. *Proceedings of the VLDB Endowment*, 2021.

[41] Junyong Yang, Ming Zhong, Yuanyuan Zhu, Tieyun Qian, Mengchi Liu, and Jeffrey Xu Yu. Scalable time-range k-core query on temporal graphs. *Proceedings of the VLDB Endowment*, 16(5):1168–1180, 2023.

[42] Yuxiang Wang, Chengjie Gu, Xiaoliang Xu, Xinjun Zeng, Xiangyu Ke, and Tianxing Wu. Efficient and effective (k, p)-core-based community search over attributed heterogeneous information networks. *Information Sciences*, 661:120076, 2024.

[43] Lianpeng Qiao, Zhiwei Zhang, Ye Yuan, Chen Chen, and Guoren Wang. Keyword-centric community search over large heterogeneous information networks. In *International Conference on Database Systems for Advanced Applications*, pages 158–173, 2021.

[44] Thang N Dinh, Xiang Li, and My T Thai. Network clustering via maximizing modularity: Approximation algorithms and theoretical limits. In *2015 IEEE International Conference on Data Mining*, pages 101–110, 2015.

[45] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical review E*, 70(6):066111, 2004.

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

[47] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

[48] Michael J Barber. Modularity and community detection in bipartite networks. *Physical Review E*, 76(6):066102, 2007.

[49] Tsuyoshi Murata. Detecting communities from bipartite networks based on bipartite modularities. In *2009 International Conference on Computational Science and Engineering*, volume 4, pages 50–57, 2009.

[50] Nicolas Neubauer and Klaus Obermayer. Towards community detection in k-partite k-uniform hypergraphs. In *Proceedings of the NIPS 2009 Workshop on Analyzing Networks and Learning with Graphs*, pages 1–9, 2009.

[51] Tsuyoshi Murata. Modularity for heterogeneous networks. In *Proceedings of the 21st ACM Conference on Hypertext and Hypermedia*, pages 129–134, 2010.

[52] Ling Huang, Chang-Dong Wang, and Hong-Yang Chao. Hm-modularity: A harmonic motif modularity approach for multi-layer network community detection. *IEEE Transactions on Knowledge and Data Engineering*, 33(6):2520–2533, 2019.

[53] Yafang Liu, Aiwen Li, An Zeng, Jianlin Zhou, Ying Fan, and Zengru Di. Motif-based community detection in heterogeneous multilayer networks. *Scientific Reports*, 14(1):8769, 2024.

[54] Ron Milo, Shai Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri Chklovskii, and Uri Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[55] Nataša Pržulj and Noël Malod-Dognin. Network analytics in the age of big data. *Science*, 353(6295):123–124, 2016.

[56] Mark EJ Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104, 2006.

[57] Yuhan Zhou, Qing Liu, Xin Huang, Jianliang Xu, and Yunjun Gao. Mochi: Motif-based community search over large heterogeneous information networks (technical report), 2025. https://github.com/ZJU-DAILY/MOCHI/blob/main/MOCHI_TechnicalReport.pdf.

[58] Santo Fortunato and Marc Barthelemy. Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1):36–41, 2007.

[59] Yubao Wu, Ruoming Jin, Jing Li, and Xiang Zhang. Robust local community detection: on free rider effect and its elimination. *Proceedings of the VLDB Endowment*, 8(7):798–809, 2015.

[60] Emanuele Martorana, Roberto Grasso, Giovanni Micale, Salvatore Alaimo, Dennis E. Shasha, Rosalba Giugno, and Alfredo Pulvirenti. Motif finding algorithms: A performance comparison. In *From Computational Logic to Computational Biology*, volume 14070, pages 250–267, 2024.

[61] Shuo Yu, Yufan Feng, Da Zhang, Hayat Dino Bedru, Bo Xu, and Feng Xia. Motif discovery in networks: A survey. *Computer Science Review*, 37:100267, 2020.

[62] Ali Jazayeri and Christopher C. Yang. Motif discovery algorithms in static and temporal networks: A survey. *J. Complex Networks*, 8(4), 2020.

[63] Zhijie Zhang, Yujie Lu, Weiguo Zheng, and Xuemin Lin. A comprehensive survey and experimental study of subgraph matching: Trends, unbiasedness, and interaction. *Proc. ACM Manag. Data*, 2(1):60:1–60:29, 2024.

[64] Xi Wang, Qianzhen Zhang, Deke Guo, and Xiang Zhao. A survey of continuous subgraph matching for dynamic graphs. *Knowl. Inf. Syst.*, 65(3):945–989, 2023.

[65] Jiafeng Hu, Reynold Cheng, Kevin Chen-Chuan Chang, Aravind Sankar, Yixiang Fang, and Brian YH Lam. Discovering maximal motif cliques in large heterogeneous information networks. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 746–757, 2019.

[66] Saket Gurukar, Sayan Ranu, and Balaraman Ravindran. Commit: A scalable approach to mining communication motifs from dynamic networks. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 475–489, 2015.

[67] Shixuan Sun, Xibo Sun, Yulin Che, Qiong Luo, and Bingsheng He. Rapidmatch: A holistic approach to subgraph query processing. *Proceedings of the VLDB Endowment*, 14(2):176–188, 2020.

[68] Jon M Kleinberg. Navigation in a small world. *Nature*, 406(6798):845–845, 2000.

[69] Yu A Malkov and Dmitry A Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836, 2018.

[70] Yuxiang Wang, Shuzhan Ye, Xiaoliang Xu, Yuxia Geng, Zhenghe Zhao, Xiangyu Ke, and Tianxing Wu. Scalable community search with accuracy guarantee on attributed graphs. In *40th IEEE International Conference on Data Engineering, ICDE 2024*, pages 2737–2750, 2024.

[71] Kurt Mehlhorn. A faster approximation algorithm for the steiner problem in graphs. *Information Processing Letters*, 27(3):125–128, 1988.

[72] https://mips.helmholtz-muenchen.de/CIDeRplus/.

[73] Hannah Bast, Florian Bäurle, Björn Buchhold, and Elmar Haußmann. Easy access to the freebase dataset. In *23rd International World Wide Web Conference, WWW '14, , Companion Volume*, pages 95–98. ACM, 2014.

[74] Yingli Zhou, Yixiang Fang, Chenhao Ma, Tianci Hou, and Xin Huang. Efficient maximal motif-clique enumeration over large heterogeneous information networks. *Proceedings of the VLDB Endowment*, 17(11):2946–2959, 2024.

APPENDIX

A NOTATIONS

Table 5 summarizes the notations frequently used in this paper.

Table 5: Notations

Notation	Description
$H = (V_H, E_H, L_H, \ell_H)$	an HIN with vertex set V_H, edge set E_H, type set L_H, and type mapping function ℓ_H
$M = (V_M, E_M, L_M, \ell_M)$	a motif with vertex set V_M, edge set E_M, type set L_M, and type mapping function ℓ_M
$H[S]$	a subgraph of H induced by $S \subseteq V_H$
$\mathcal{I}_H^M$	a set of all motif instances of M in H
$\mathtt{deg}_H(v)$	the degree of the vertex v in H
$\mathtt{Mdeg}_H(v)$	the motif degree of the vertex v in H
$\mathcal{N}_H(v)$	the neighbors of v in H
$\mathtt{vol}_H(S)$	the sum of degree of S over H
$\mathtt{Mvol}_H(S)$	the sum of motif degree of S over H
$\mathcal{B}_M$	a motif-based bipartite graph of M
$Eq^M_{H[S]}(v)$	the equivalent vertices of v in $H[S]$
H_M	the MW-HIN of the motif M and HIN H
V_{H_M}	the vertices with motif instances of M in H
Θ_v^S	the M-ratio of the vertex v in $H[S]$

B TIME COMPLEXITY ANALYSIS

Time complexity of Algorithm 1. Let $\mathtt{Mdeg}_H^{max}$ be the maximum motif degree. The time complexity of collecting motif instances with *RapidMatch* is $O(|E_M| \times |E_H| + \alpha(M) + |V_M|^2)$, where $\alpha(\cdot)$ is the cost of nucleus decomposition [67]. Then, it takes $O(\mathtt{Mdeg}_H^{max} \times |V_M| \times |V_H|)$ time to find all vertices in H being M-connected to Q. In each round, it takes $O(|V_H| \times (\mathtt{Mdeg}_H^{max} \times |V_M| \times |V_H|))$ time to remove each vertex, maintain M-connectivity, and compute MDM. The process is repeated up to $|V_H|$ times. Since the cost of vertex removal is the dominant term of the overall time complexity, the time complexity of Algorithm 1 can be simplified to $O(|V_H|^3 \times \mathtt{Mdeg}_H^{max} \times |V_M|)$.

Time complexity of Algorithm 2. Let $\sigma_{\mathtt{check}}$ denote the number of vertex checks performed per round to determine candidate vertices, and $\beta_{\mathtt{update}}$ denote the number of exact edge weight updates after vertex removal. Algorithm 2 takes $O(|E_M| \times |E_H| + \alpha(M) + |V_M|^2)$ to collect motif instances with *RapidMatch*, where $\alpha(\cdot)$ is the cost of nucleus decomposition [67]. Then, it takes $O(|\mathcal{I}_H^M| \times |V_M|^2)$ time to construct the MW-HIN. Based on MW-HIN, it takes $O(|V_{H_M}|+|E_{H_M}|)$ time to find the vertices connected to Q in H_M. To maximize MDM, it iteratively removes the candidate vertex with the maximum $MDM(H, S_i \setminus \{v\}, M)$ while maintaining M-connectivity and updating H_M. The process repeats up to $|V_{H_M}|$ times. In each loop of selecting vertex, it first computes $MDM(H, S_i \setminus \{v\}, M)$ for each vertex and sort them with $O(|V_{H_M}| \times \log |V_{H_M}|)$ time. Then it checks whether each vertex is a candidate vertex from top to bottom according to its $MDM(H, S_i \setminus \{v\}, M)$. For each vertex, it takes $O(\mathtt{Mdeg}_H^{max} \times |V_M| \times \log \mathtt{Mdeg}_H^{max})$ time to compute equivalent vertices of the selected vertex. Next, it takes $O(\beta_{\mathtt{update}} \times \mathtt{Mdeg}_H^{max} \times \log \mathtt{Mdeg}_H^{max})$ time to update the edge weight of MW-HIN after vertex removal. Finally, it takes $O(|V_{H_M}|+|E_{H_M}|)$ time to check the connectivity of the updated MW-HIN. Overall, the time complexity of Algorithm 2 can be simplified to $O(|V_{H_M}| \times \sigma_{\mathtt{check}} \times (\beta_{\mathtt{update}} \times \mathtt{Mdeg}_H^{max} \times \log \mathtt{Mdeg}_H^{max} + |V_{H_M}| + |E_{H_M}|))$.

Time complexity of Algorithm 3. Algorithm 3 takes $O(|E_M| \times |E_H| + \alpha(M) + |V_M|^2)$ to collect motif instances with *RapidMatch*, where $\alpha(\cdot)$ is the cost of nucleus decomposition [67]. Then, it takes $O(|\mathcal{I}_H^M| \times |V_M|^2)$ time to construct the MW-HIN. The time complexity of expanding Q to find a small M-connected subgraph $H[S_0]$ with approximate Steiner tree algorithm is $O(|E_{H_M}| + |V_{H_M}| \times \log |V_{H_M}|)$. Based on MW-HIN, it takes $O(|E_{H_M}|+|V_{H_M}| \times \log |V_{H_M}|)$ time to compute motif distance between S_0 and other vertices in H_M. The time complexity of finding the vertices connected to S_0 is $O(|V_{H_M}| + |E_{H_M}|)$. Next, it takes $O(|V_{H_M}| \times \mathtt{Mdeg}_H^{max} \times |V_M|)$ time to execute coarse-grained deletion. After that, it takes $O(|V_{H_M}| \times \log |V_{H_M}|)$ time to compute the M-ratio for each vertex and sort them. The process of fine-grained deletion runs for $|V_{H_M}|$ iterations. In each iteration, it selects the vertex with the maximum M-ratio from an ordered set. After removing the selected vertex, it takes $O(\mathtt{Mdeg}_H^{max} \times |V_M| \times \log |V_{H_M}|)$ time to update the M-ratios of the influenced vertices and maintain the order of the M-ratio set. Therefore, the time complexity of the fine-grained deletion is $O(|V_{H_M}| \times \mathtt{Mdeg}_H^{max} \times |V_M| \times \log |V_{H_M}|)$. Overall, the time complexity of Algorithm 3 can be simplified to $O(|V_{H_M}| \times \log |V_{H_M}| \times \mathtt{Mdeg}_H^{max} \times |V_M|)$.

C THE PROOF OF LEMMA 1

Lemma 1. *Given an HIN H, a motif M, and a vertex set $S \subseteq V_H$, the motif-based bipartite graph $\mathcal{B}_M$ of M and H satisfies:*

(1) $\forall v \in \mathcal{L}_{\mathcal{B}_M}$, $\mathtt{deg}_{\mathcal{B}_M}(v) = \mathtt{Mdeg}_H(v)$;

(2) For a motif instance $I \in \mathcal{I}^M_{H[S]}$, the vertex $v_I \in \mathcal{R}_{\mathcal{B}_M}$ satisfies that $\mathcal{N}_{\mathcal{B}_M}(v_I) \subseteq S$;

(3) $\forall l \in L_M$, $\mathtt{vol}_{\mathcal{B}_M}(V_H(l)) = \mathtt{Mvol}_H(V_H(l))$.

Here, $\mathtt{deg}_{\mathcal{B}_M}(v)$ is the degree of v in $\mathcal{B}_M$, $\mathcal{N}_{\mathcal{B}_M}(v_I)$ is the neighbors of v_I in $\mathcal{B}_M$, and $\mathtt{vol}_{\mathcal{B}_M}(V_H(l)) = \sum_{v \in V_H(l)} \mathtt{deg}_{\mathcal{B}_M}(v)$.

Proof. For (1), the degree of vertex $v \in \mathcal{L}_{\mathcal{B}_M}$ in $\mathcal{B}_M$ corresponds to the number of motif instances that includes v, which is exactly the motif degree of v. Therefore, $\mathtt{deg}_{\mathcal{B}_M}(v) = \mathtt{Mdeg}_H(v)$.

For (2), for each motif instance $I \in \mathcal{I}^M_{H[S]}$, all the vertices of I are included in $H[S]$. Therefore, the neighbors of $v_I \in \mathcal{R}_{\mathcal{B}_M}$, which corresponds to the motif instance I, are included in S.

For (3), since $\mathtt{deg}_{\mathcal{B}_M}(v) = \mathtt{Mdeg}_H(v)$, we have $\mathtt{vol}_{\mathcal{B}_M}(V_H(l)) = \sum_{v \in V_H(l)} \mathtt{deg}_{\mathcal{B}_M}(v) = \sum_{v \in V_H(l)} \mathtt{Mdeg}_H(v) = \mathtt{Mvol}_H(V_H(l))$. □

D THE PROOF OF LEMMA 2

Lemma 2. *Given an HIN H, a motif M, the corresponding motif-based bipartite graph $\mathcal{B}_M$ of M and H, and the random motif-based bipartite graph $\mathcal{B}'_M$ generated by MGM, $\mathcal{B}_M$ and $\mathcal{B}'_M$ have the following releationships.*

(1) $\forall v \in \mathcal{L}_{\mathcal{B}_M}$, $Exp[\mathtt{deg}_{\mathcal{B}'_M}(v)] = \mathtt{deg}_{\mathcal{B}_M}(v) = \mathtt{Mdeg}_H(v)$.

(2) $\forall l \in L_M$, $Exp[\mathtt{vol}_{\mathcal{B}'_M}(V_H(l))] = \mathtt{Mvol}_H(V_H(l))$.

Note that $Exp[\mathtt{deg}_{\mathcal{B}'_M}(v)]$ denotes the expected degree of v in $\mathcal{B}'_M$ and $Exp[\mathtt{vol}_{\mathcal{B}'_M}(V_H(l))]$ denotes the expected sum of degree of vertices with type l in $\mathcal{B}'_M$.

Proof. Let S be a multi-set of vertices where a vertex can appear multiple times in S. For each motif instance $I \in \mathcal{I}_H^M$, we add the vertices with type l within I into S. The number of vertices with type

l in I is $|V_M(l)|$. Since the number of motif instances in H is $|\mathcal{I}_H^M|$, we have $|S| = |\mathcal{I}_H^M||V_M(l)|$. For any vertex $v \in V_H(l)$, the number of duplications of v in S is exactly the motif degree of v. Therefore, $\texttt{Mvol}_H(V_H(l)) = \sum_{v \in V_H(l)} \texttt{Mdeg}_H(v) = |S| = |\mathcal{I}_H^M||V_M(l)|$.

For (1), for each $v_I \in \mathcal{R}_{\mathcal{B}_M}$, it rewires $|V_M(l)|$ vertices with type $l \in L_M$. Therefore, the probability that each $v \in \mathcal{L}_{\mathcal{B}_M}$ with type $\ell_H(v)$ selected to link v_I is $|V_M(\ell_H(v))|\frac{\texttt{deg}_{\mathcal{B}_M}(v)}{\texttt{vol}_{\mathcal{B}_M}(V_H(\ell_H(v)))}$. The process is repeated for $|\mathcal{R}_{\mathcal{B}_M}|$ times and each procedure is independent. Since $|\mathcal{R}_{\mathcal{B}_M}| = |\mathcal{I}_H^M|$, then we can get that $Exp[\texttt{deg}_{\mathcal{B}'_M}(v)] = |\mathcal{I}_H^M||V_M(\ell_H(v))|\frac{\texttt{deg}_{\mathcal{B}_M}(v)}{\texttt{vol}_{\mathcal{B}_M}(V_H(\ell_H(v)))}$. Combining Lemma 1, we have $\texttt{vol}_{\mathcal{B}_M}(V_H(l)) = \texttt{Mvol}_H(V_H(l)) = |\mathcal{I}_H^M||V_M(l)|$ and $\texttt{deg}_{\mathcal{B}_M}(v) = \texttt{Mdeg}_H(v)$. Hence, $Exp[\texttt{deg}_{\mathcal{B}'_M}(v)] = \texttt{deg}_{\mathcal{B}_M}(v) = \texttt{Mdeg}_H(v)$.

For (2), since $Exp[\texttt{vol}_{\mathcal{B}'_M}(V_H(l))] = \sum_{v \in V_H(l)} Exp[\texttt{deg}_{\mathcal{B}'_M}(v)]$ and $Exp[\texttt{deg}_{\mathcal{B}'_M}(v)] = \texttt{Mdeg}_H(v)$, we have $Exp[\texttt{vol}_{\mathcal{B}'_M}(V_H(l))] = \texttt{Mvol}_H(V_H(l))$. □

E THE PROOF OF THEOREM 1

Theorem 1. *Given an HIN H, a motif M, and a vertex set $S \subseteq V_H$, the expected number of motif instances in $H[S]$ under MGM is*

$$Exp[|\mathcal{I}_{H[S]}^M|] = |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))},$$

where $S(\ell_M(u))$ is a set of vertices in S with type $\ell_M(u)$.

Proof. We estimate $Exp[|\mathcal{I}_{H[S]}^M|]$ by the expected number of motif instances in $\mathcal{B}'_M$, which all consist of vertices in S, i.e., $Exp[|\mathcal{I}_{H[S]}^M|] = Exp[|\{v_I \in \mathcal{R}_{\mathcal{B}'_M} : \forall (v_I, v) \in \mathcal{E}_{\mathcal{B}'_M}, v \in S\}|]$. Given a motif instance I in $\mathcal{B}'_M$, the MGM selects different types for vertices according to M. The probability of selecting a vertex with the type l from S is $\frac{\texttt{Mvol}_H(S(l))}{\texttt{Mvol}_H(V_H(l))}$. Hence, the probability that all selected vertices of a motif instance are from S is $\prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}$. Since MGM needs to select vertices for all the motif instances, the process will be repeated for $|\mathcal{I}_H^M|$ times. Therefore, $Exp[|\mathcal{I}_{H[S]}^M|] = |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}$. □

F THE PROOF OF LEMMA 3

Lemma 3. *Whenever MDM suffers from the free-rider effect and resolution limit, GMM suffers from the free-rider effect and resolution limit as well.*

Proof. Given a graph $G = (V_G, E_G)$, a goodness function $g(\cdot)$ and a query vertex set Q, let $G[S]$ and $G[S^*]$ be solutions of community search based on $g(\cdot)$ with $Q \neq \emptyset$ and $Q = \emptyset$, respectively. We say that $g(\cdot)$ suffers from the free-rider effect for the community search if $g(S \cup S^*) \geqslant g(S)$ [59].

Let S and S^* be the identified communities with $Q \neq \emptyset$ and $Q = \emptyset$, respectively. For $Q = \emptyset$, the MOCHI problem finds the maximum M-connected subgraph $H[S^*] \subseteq H$ that maximizes MDM. Suppose that MDM suffers from the free-rider effect. Let $S^+ = S \cup S^*$. According to the definition of the free-rider effect, we can get that $MDM(H, S^+, M) \geqslant MDM(H, S, M)$, i.e.,

$$\frac{1}{|S^+|}(|\mathcal{I}_{H[S^+]}^M| - |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S^+(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}) \geqslant \frac{1}{|S|}(|\mathcal{I}_{H[S]}^M| - |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}).$$

We multiply both sides of the inequality by $\frac{|S^+|}{|\mathcal{I}_H^M|}$ and we have

$$\frac{1}{|\mathcal{I}_H^M|}(|\mathcal{I}_{H[S^+]}^M| - |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S^+(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}) \geqslant \frac{|S^+|}{|\mathcal{I}_H^M||S|}(|\mathcal{I}_{H[S]}^M| - |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}).$$

Note that we do not consider the communities with MDM smaller than 0 as they are meaningless. Therefore, we can get that $|\mathcal{I}_{H[S]}^M| - |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))} > 0$. Since $|S^+| \geqslant |S|$, we can infer that

$$\frac{1}{|\mathcal{I}_H^M|}(|\mathcal{I}_{H[S^+]}^M| - |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S^+(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}) \geqslant \frac{1}{|\mathcal{I}_H^M|}(|\mathcal{I}_{H[S]}^M| - |\mathcal{I}_H^M| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}),$$

which is the same as $GMM(H, S^+, M) \geqslant GMM(H, S, M)$. Therefore, whenever MDM suffers from the free-rider effect, GMM suffers from the free-rider effect as well.

Given a graph $G = (V_G, E_G)$, a query vertex set Q , an objective function $g(\cdot)$ and a community constrain $\mathcal{S}$, let $G[S]$ be the community satisfying $\mathcal{S}$ and containing Q, and $G[S']$ be any subgraph of G satisfying $\mathcal{S}$ such that $G[S \cup S']$ is connected and $S \cap S' = \emptyset$. If there exists the subgraph $G[S']$ such that $g(S \cup S') \geqslant g(S)$ and $G[S \cup S']$ satisfies $\mathcal{S}$, we say that the objective function $g(\cdot)$ suffers from the resolution limit for community search [14, 58].

Suppose that $\mathcal{S}$ is the constraint of the MOCHI problem, $H[S]$ is the community satisfying $\mathcal{S}$ and containing Q, and $H[S']$ is a subgraph satisfying $\mathcal{S}$ such that $H[S \cup S']$ is connected, $S \cap S' = \emptyset$, and $H[S \cup S']$ satisfies $\mathcal{S}$. We use S' to replace S^* in the proof of the free-rider effect. Similarly, if $MDM(H, S \cup S', M) \geqslant MDM(H, S, M)$, we can get that $GMM(H, S \cup S', M) \geqslant GMM(H, S, M)$ because $|S \cup S'| \geqslant |S|$. Therefore, whenever MDM suffers from the resolution limit, GMM suffers from the resolution limit as well. □

G THE PROOF OF THEOREM 2

Theorem 2. *The MOCHI problem is NP-hard.*

Proof. We reduce the DMCS problem [14], which has been proved to be NP-hard, to the MOCHI problem. In particular, given a homogeneous network $G = (V_G, E_G)$, a query vertex set Q, the DMCS problem is to find a connected subgraph $G[S] \subseteq G$ that contains Q and has the maximum density modularity. Here, the density modularity is

$$DM(G, S) = \frac{1}{|S|}(|E_{G[S]}| - \frac{\texttt{vol}_G(S)^2}{4|E_G|}).$$

We show that the DMCS problem is a special case of the MOCHI problem. Specifically, for the MOCHI problem, we set that (1) the

vertices of HIN H are of the same type l and (2) the query motif M is an edge between vertices with type l. Then we have $\mathcal{I}_H^M = E_H$, $\mathcal{I}_{H[S]}^M = E_{H[S]}$, $\texttt{Mvol}_H(S) = \texttt{vol}_H(S)$ and $\texttt{Mvol}_H(V_H) = 2|\mathcal{I}_H^M| = 2|E_H|$. Therefore,

$$\begin{aligned} MDM(H,S,M) &= \frac{1}{|S|}(|\mathcal{I}_{H[S]}^M| - |\mathcal{I}_H^M| \prod_{u\in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}) \\ &= \frac{1}{|S|}(|E_{H[S]}| - |E_H|(\frac{\texttt{vol}_H(S)}{2|E_H|})^2) \\ &= \frac{1}{|S|}(|E_{H[S]}| - \frac{\texttt{vol}_H(S)^2}{4|E_H|}) \\ &= DM(H,S). \end{aligned}$$

Note that, for any $(u,v) \in E_H$, u and v are M-adjacent. Hence, M-connectivity degenerates into the classic connectivity under the above settings. Then, we can get that finding a solution to the MOCHI problem equals finding a solution to the DMCS problem. Since the DMCS problem is NP-hard, the MOCHI problem is NP-hard as well. □

H THE PROOF OF LEMMA 4

Lemma 4. *Given an HIN H, a motif M, and a subgraph $H[S] \subseteq H$. For a vertex $v \in S$, if we delete v from $H[S]$, the MDM of $H[S \setminus \{v\}]$ is*

$$MDM(H, S\setminus\{v\}, M) = \frac{|\mathcal{I}_{H[S]}^M| - \texttt{Mdeg}_{H[S]}(v)}{|S|-1} - \frac{|\mathcal{I}_H^M| \prod_{u\in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u))) - \texttt{Mdeg}_H(v)\eta(u,v)}{\texttt{Mvol}_H(V_H(\ell_M(u)))}}{|S|-1}.$$

If $\ell_M(u) = \ell_H(v)$, $\eta(u,v) = 1$; otherwise, $\eta(u,v) = 0$.

Proof. Let $S' = S \setminus \{v\}$. Then, $|\mathcal{I}_{H[S']}^M| = |\mathcal{I}_{H[S]}^M| - \texttt{Mdeg}_{H[S]}(v)$ and $|S'| = |S| - 1$. For each $u \in V_M$, if $\ell_M(u) = \ell_H(v)$, then $\texttt{Mvol}_H(S'(\ell_M(u))) = \texttt{Mvol}_H(S(\ell_M(u))) - \texttt{Mdeg}_H(v)$; otherwise, $\texttt{Mvol}_H(S'(\ell_M(u))) = \texttt{Mvol}_H(S(\ell_M(u)))$. Therefore, we can get that $Exp[|\mathcal{I}_{H[S']}^M|] = |\mathcal{I}_H^M| \prod_{u\in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u))) - \texttt{Mdeg}_H(v)\eta(u,v)}{\texttt{Mvol}_H(V_H(\ell_M(u)))}$. By substituting $|S'|$, $|\mathcal{I}_{H[S']}^M|$, and $Exp[|\mathcal{I}_{H[S']}^M|]$ into the definition of MDM in Definition 3, the proof is completed. □

I THE PROOF OF LEMMA 5

Lemma 5. *Given an HIN H, a motif M, and an MW-HIN H_M of M and H, H is M-connected iff (1) $\forall v \in V_H$, $\texttt{Mdeg}_H(v) \geqslant 1$, and (2) H_M is connected.*

Proof. ($\Rightarrow$) Suppose that H is M-connected, then $\forall v \in V_H$, $\texttt{Mdeg}_H(v) \geqslant 1$, otherwise H is not M-connected. Next, we prove that H_M is connected by contradiction. Assume that $\exists v_s, v_t \in V_H$, v_s and v_t are not connected in H_M. Since H is M-connected, there exists a sequence of motif instances $I_1, I_2, ..., I_n$ in H such that $v_s \in V_{I_1}$,$v_t \in V_{I_n}$ and $V_{I_i} \cap V_{I_{i+1}} \neq \emptyset (1 \leqslant i < n)$. Let $v_i \in V_{I_i} \cap V_{I_{i+1}}$. Then we have a sequence of adjacent vertices $v_1, v_2, ..., v_{n-1}$ in H_M, s.t. , $(v_j, v_{j+1}) \in E_{H_M} (1 \leqslant j < n-1)$ because both v_j and v_{j+1} belong to the same motif instance I_{j+1} in H. Note that $v_s, v_1 \in V_{I_1}$ and $v_t, v_{n-1} \in V_{I_n}$. Therefore, $v_s = v_1$ or $v_s \neq v_1, (v_s, v_1) \in E_{H_M}$. Similarly, $v_t = v_{n-1}$ or $v_t \neq v_{n-1}, (v_{n-1}, v_t) \in E_{H_M}$. Hence, we can find a sequence of adjacent vertices $v_1, v_2, ..., v_{n-1}$ in H_M to connect v_s and v_t, which is contrary to the assumption. To this end, H_M is connected.

($\Leftarrow$) Suppose that $\forall v \in V_H$, $\texttt{Mdeg}_H(v) \geqslant 1$ and H_M is connected. Next, we will prove that H is M-connected by contradiction. Assume H is not M-connected, then $\exists v_s, v_t \in V_H$, v_s and v_t are not M-connected in H. Since H_M is connected, there exists a path formed by a sequence of vertices $v_1, v_2, ..., v_n$ in H_M such that $(v_i, v_{i+1}) \in E_{H_M} (1 \leqslant i < n)$ and $v_s = v_1, v_t = v_n$. There are two cases:

(1) If $n = 2$, v_s and v_t have the same motif instances.

(2) If $n > 2$, let $I_i \in \mathcal{I}_H^M(v_i) \cap \mathcal{I}_H^M(v_{i+1})$, there exists a sequence of motif instances $I_1, I_2, ..., I_{n-1}$ in H such that $v_s \in V_{I_1}, v_t \in V_{I_{n-1}}$, and $V_{I_j} \cap V_{I_{j+1}} = v_{j+1} (1 \leqslant j < n-1)$.

Therefore, v_s and v_t are M-connected in H, which is contrary to the assumption. Then we can get that H is M-connected. □

J THE PROOF OF LEMMA 6

Lemma 6. *Given an HIN H, a motif M, an MW-HIN H_M of M and H, a vertex $v \in V_{H_M}$, and an edge $(u,w) \in E_{H_M}$ with $u, w \neq v$. Let H'_M be the MW-HIN obtained by deleting v from H_M, and $\mathcal{N}_{H_M}(v)$ be the neighbor set of v in H_M.*

(1) Limited Influence Scope: If $u \notin \mathcal{N}_{H_M}(v)$ or $w \notin \mathcal{N}_{H_M}(v)$, $\omega_{H_M}(u,w) = \omega_{H'_M}(u,w)$.

(2) Bounded Influence Strength: $0 \leq \omega_{H_M}(u,w) - \omega_{H'_M}(u,w) \leq |\mathcal{I}_H^M(v)|$.

Proof. For (1), there are three cases, i.e., (i) $u, w \notin \mathcal{N}_{H_M}(v)$; (ii) $u \notin \mathcal{N}_{H_M}(v)$ and $w \in \mathcal{N}_{H_M}(v)$; (iii) $u \in \mathcal{N}_{H_M}(v)$ and $w \notin \mathcal{N}_{H_M}(v)$. The case (iii) is the same as the case (ii) by exchanging u and w. Therefore, we prove the first two cases.

(i) If $u, w \notin \mathcal{N}_{H_M}(v)$, $\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(v) = \emptyset$ and $\mathcal{I}_H^M(w) \cap \mathcal{I}_H^M(v) = \emptyset$. The updated edge weight of (u,w) in H'_M can be computed as $\omega_{H'_M}(u,w) = |(\mathcal{I}_H^M(u) \setminus \mathcal{I}_H^M(v) \cap (\mathcal{I}_H^M(w) \setminus \mathcal{I}_H^M(v))| = |\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)| = \omega_{H_M}(u,w)$.

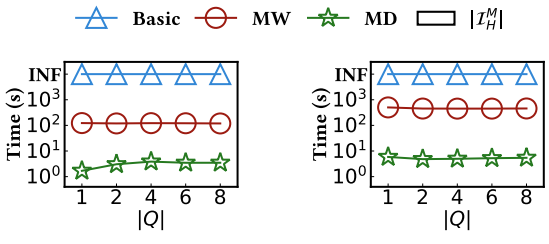
(ii) If $u \notin \mathcal{N}_{H_M}(v)$ and $w \in \mathcal{N}_{H_M}(v)$, $\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(v) = \emptyset$ and $\mathcal{I}_H^M(w) \cap \mathcal{I}_H^M(v) \neq \emptyset$. The updated edge weight of (u,w) in H'_M can be computed as $\omega_{H'_M}(u,w) = |(\mathcal{I}_H^M(u) \setminus \mathcal{I}_H^M(v)) \cap (\mathcal{I}_H^M(w) \setminus \mathcal{I}_H^M(v))| = |(\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)) \setminus \mathcal{I}_H^M(v)|$. Since $(\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)) \subseteq \mathcal{I}_H^M(u)$ and $\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(v) = \emptyset$, we have $\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w) \setminus \mathcal{I}_H^M(v) = \mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)$. Hence, $\omega_{H'_M}(u,w) = \omega_{H_M}(u,w)$.

By combining case (i) and case (ii), if $u \notin \mathcal{N}_{H_M}(v)$ or $w \notin \mathcal{N}_{H_M}(v)$, $\omega_{H'_M}(u,w) = \omega_{H_M}(u,w)$.

For (2), $\omega_{H'_M}(u,w) = |(\mathcal{I}_H^M(u) \setminus \mathcal{I}_H^M(v)) \cap (\mathcal{I}_H^M(w) \setminus \mathcal{I}_H^M(v))| = |\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)) \setminus \mathcal{I}_H^M(v)|$. Then we find that $|\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)| - |\mathcal{I}_H^M(v)| \leqslant |(\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)) \setminus \mathcal{I}_H^M(v)| \leqslant |\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)|$. Since $\omega_{H_M}(u,w) = |\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)|$ and $\omega_{H'_M}(u,w) = |\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w)) \setminus \mathcal{I}_H^M(v)|$, we have $0 \leq \omega_{H_M}(u,w) - \omega_{H'_M}(u,w) \leq |\mathcal{I}_H^M(v)|$. □

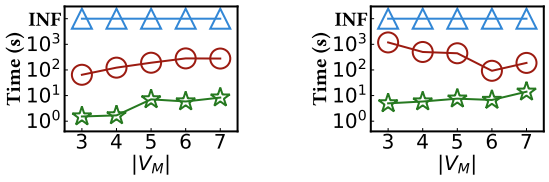
K THE PROOF OF LEMMA 7

Lemma 7. *Given an HIN H, a motif M, and an MW-HIN H_M of M and H. $\forall u, v \in V_H$, if $u, v \in V_{H_M}$, $\texttt{Mdist}_H(u,v) = \texttt{dist}_{H_M}(u,v)$. Otherwise, $\texttt{Mdist}_H(u,v) = +\infty$.*



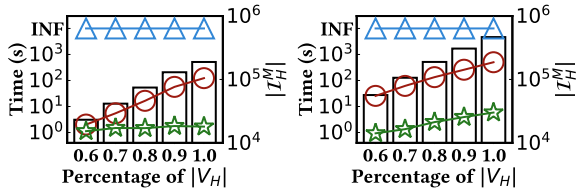
(a) CIDeRplus (b) TMDB

Figure 11: Effect of $|Q|$



(a) CIDeRplus (b) TMDB

Figure 12: Effect of $|V_M|$



(a) CIDeRplus (b) TMDB

Figure 13: Effect of $|V_H|$

Proof. If $u \notin V_{H_M}$ or $v \notin V_{H_M}$, u or v does not have any motif instance. Therefore, u and v are not M-connected in H. Then we can conclude that $\texttt{Mdist}_H(u,v) = +\infty$. Next, we prove that if $u, v \in V_{H_M}$, $\texttt{Mdist}_H(u,v) = \texttt{dist}_{H_M}(u,v)$. There are three cases.

(1) If u and v are not M-connected in H, we have $\texttt{Mdist}_H(u,v) = +\infty$. Suppose that there exists a path consisting of adjacent vertices $v_1, v_2, ..., v_n$ in H_M such that $u = v_1$ and $v = v_n$. Then we can find a sequence of motif instances $I_1, I_2, ..., I_{n-1}$ in H such that $u \in I_1$, $v \in I_{n-1}$, $I_1 = I_{n-1}$ or $V_{I_j} \cap V_{I_{j+1}} \neq \emptyset (1 \leq j < n-1)$, where $I_i \subseteq \mathcal{I}_H^M(v_i) \cap \mathcal{I}_H^M(v_{i+1}) (1 \leq i < n)$. It contradicts the assumption that u and v are not M-connected in H. Therefore, $\texttt{Mdist}_H(u,v) = \texttt{dist}_{H_M}(u,v) = +\infty$.

(2) If u and v are M-connected in H and $\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(v) \neq \emptyset$, $(u,v) \in E_{H_M}$. Therefore, $\texttt{Mdist}_H(u,v) = \texttt{dist}_{H_M}(u,v) = 1$.

(3) If u and v are M-connected in H and $\mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(v) = \emptyset$, $\texttt{Mdist}_H(u,v) > 1$ and $(u,v) \notin E_{H_M}$. Suppose that $\texttt{Mdist}_H(u,v) = n (n \geqslant 2)$ and $I_1, I_2, ..., I_n$ is a sequence of motif instances in H that connects u and v. Then we can find a sequence of adjacent vertices $v_1, v_2, ..., v_{n-1}$ in H_M such that $v_i \subseteq V_{I_i} \cap V_{I_{i+1}} (1 \leq i < n)$. Note that $(u,v) \notin E_{H_M}$, $u, v_1 \in V_{I_1}$, and $v, v_{n-1} \in V_{I_n}$, then we have $(u,v_1), (v_{n-1},v) \in E_{H_M}$. We can get a path consisting of vertices $u, v_1, v_2, ..., v_{n-1}, v$ in H_M and the length of the path is n. It is the shortest path between u, v in H_M. This is because if we can find a shorter path with length $n' < n$, then we can find a sequence of motif instances that makes u, v M-connected in H by selecting a common motif instance between each pair of adjacent vertices in this path and the number of motif instances is n'. It violates the assumption that $\texttt{Mdist}_H(u,v) = n$. Therefore, $\texttt{Mdist}_H(u,v) = \texttt{dist}_{H_M}(u,v) = n$. Combining all three cases, we can conclude that if $u, v \in V_{H_M}$, $\texttt{Mdist}_H(u,v) = \texttt{dist}_{H_M}(u,v)$. □

L THE PROOF OF LEMMA 8

Lemma 8. *Given a motif M, an M-connected HIN H. Let $S \subseteq V_H$, and $S' = \{v \in V_H \mid \texttt{Mdist}_H(S,v) = \max_{u \in V_H} \texttt{Mdist}_H(S,u)\}$. If $H[S]$ is M-connected and $S \subseteq V_H \setminus S'$, $H[V_H \setminus S']$ is M-connected .*

Proof. Let's merge the vertices of S into a super vertex and then construct a BFS tree $\mathcal{T}$ in the MW-HIN H_M of H based on the motif distance between S and other vertices in H. Since S' consists of the vertices farthest from S w.r.t. motif distance, the vertices in S' are the leaf vertices in the BFS tree $\mathcal{T}$. Next, we discuss the following two cases.

(1) If $H[V_H \setminus S'] = H[S]$, $H[V_H \setminus S']$ is M-connected because $H[S]$ is M-connected.

(2) If $H[V_H \setminus S'] \neq H[S]$, for each $u \in V_H \setminus S'$, we have $u \in S$ or u has a father vertex w in $\mathcal{T}$. If $u \in S$, then removing each vertex $v \in S'$ will not break the M-connectivity of u because $H[S]$ is M-connected. If u has a father vertex w in $\mathcal{T}$, removing each vertex $v \in S'$ will not affect the edges between u and its father vertex w in $\mathcal{T}$. This is because $\mathcal{I}_H^M(v) \cap \mathcal{I}_H^M(u) \cap \mathcal{I}_H^M(w) = \emptyset$ and removing v from H_M only influences the edges within its one-hop-induced subgraph in H_M. To this end, $H[V_H \setminus S']$ is M-connected. □

M THE PROOF OF LEMMA 9

Lemma 9. *Given an HIN H, a motif M, an M-connected subgraph $H[S] \subseteq H$, and a vertex $u \in S$. Let $S' = S \setminus \{u\}$. For a vertex $v \in S'$, if $\mathcal{I}_{H[S]}^M(v) \cap \mathcal{I}_{H[S]}^M(u) = \emptyset$, $\Theta_v^S = \Theta_v^{S'}$.*

Proof. Since $\mathcal{I}_{H[S]}^M(v) \cap \mathcal{I}_{H[S]}^M(u) = \emptyset$, removing u does not affect the motif instances of v. Therefore, we have $\texttt{Mdeg}_{H[S]}(v) = \texttt{Mdeg}_{H[S']}(v)$. As $\texttt{Mdeg}_H(v)$ is fixed, it follows that $\Theta_v^S = \Theta_v^{S'}$. □

N ADDITIONAL EXPERIMENTS

Effect of $|Q|$. Figure 11 shows the effect of query vertex size on three algorithms by varying $|Q|$ on CIDeRplus and TMDB.

Effect of $|V_M|$. Figure 12 shows the effect of motif size on three algorithms by varying $|V_M|$ on CIDeRplus and TMDB.

Effect of $|V_H|$. Figure 13 shows the scalability of three algorithms by varying the fraction of $|V_H|$ on CIDeRplus and TMDB.