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

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

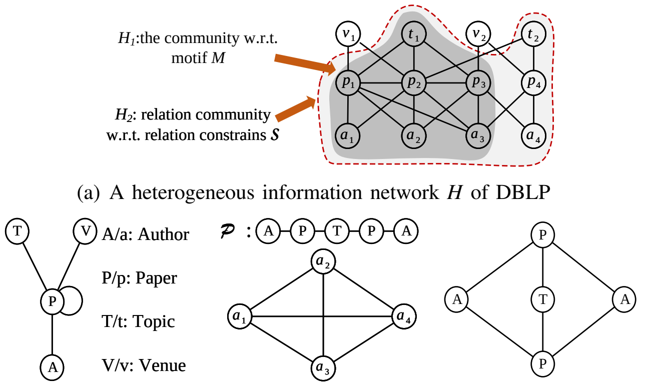
***Abstract*—In this paper, we investigate the problem of motif-based community search over heterogeneous information networks (MOCHI). We introduce a novel motif density modularity (MDM) to measure the *motif cohesiveness* of communities. Based on MDM, we define the MOCHI problem as follows: given a heterogeneous information network (HIN) H, a motif M, and a query vertex set Q, the objective is to identify the 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 M-*ratio* to remove vertices. Extensive experiments on real-world HINs demonstrate the effectiveness and efficiency of our proposed methods.**

***Index Terms*—Community Search, Modularity, Heterogeneous Information Network, Motif**

I. 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]. Fig. 1(a) and Fig. 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 $(P-T)$.

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 $\mathcal{P}$. Fig. 1(c) shows an example of $(k, \mathcal{P})$-core for the HIN H in Fig. 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 relational community, which must satisfy user-specified relational constraints. The relational 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 Fig. 1(a) represents a relational community with $\mathcal{S} = \{\langle A, P, 2\rangle, \langle P, A, 2\rangle, \langle P, T, 1\rangle, \langle T, P, 2\rangle\}$.



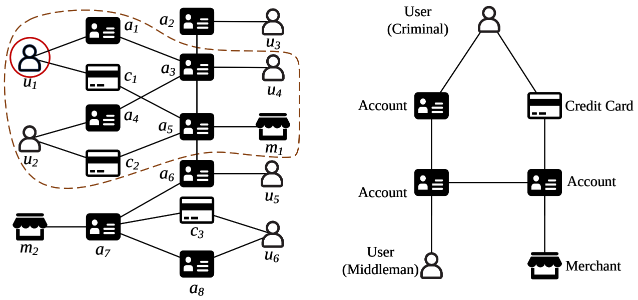
(a) A heterogeneous information network H of DBLP

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

Fig. 1. A motivating example.

While HINs and their communities inherently encapsulate rich semantics that can offer valuable insights, most existing models fall short of fully capturing these semantics. For instance, in Fig. 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 Fig. 1(d). Although existing models, such as $(k, \mathcal{P})$-core and relational 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 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



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

Fig. 2. An application of the MOCHI problem.

called *motif density modularity* (MDM). 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 subgraph of H that is connected by motif instances, contains Q, and maximizes MDM. The advantages of MOCHI include not requiring additional numerical structural parameters, supporting multiple query vertices of different types, and enabling 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, Fig. 2(a) depicts a credit card transaction network consisting of four types of vertices: user, credit card, account, and merchant. The motif M in Fig. 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 Fig. 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 vertex removal 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, 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 accesses.

To efficiently handle HINs with numerous motif instances and vertices, we propose a third algorithm named *motif-distance-based algorithm*, which introduces a novel metric *motif distance* to remove irrelevant vertices. 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 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 II reviews related work. We formulate and analyze the MOCHI problem in Section III. Section IV introduces algorithms to address the MOCHI problem. Section V reports the experimental results. We conclude this paper in Section VI.

II. Related Work

Community Search. Community search (CS) aims to find query-dependent communities within graphs [17]. It has been extensively studied in homogeneous networks since it was first proposed by Sozio and Gionis [18], and various models have been employed for community search, such as k-core [18], [19], k-truss [20], [21], k-edge connected component [22], [23], and clique [24], [25]. Recently, Kim et al. studied density-modularity-based CS [14], which is defined as the density of the community minus the expected density in the random graph. However, it focuses on edges rather than motifs, and its random graph model is tailored for homogeneous networks, making it unsuitable for HINs. Moreover, [26]–[30] explored learning-based CS, which does not require predefined community structures.

CS over graphs with richer semantics has also received significant attention. For example, [31], [32] investigated CS over bipartite graphs, which focus on interactions between two types of vertices. Other works [33], [34] incorporated attributes to search for communities over attributed graphs. However, attributed graphs consist of vertices with the same type and neglect the explicit relationships between attributes. To address these limitations, several studies have explored CS over HINs [6]–[11], where different types of entities are explicitly connected. For example, Jian et al. [6] introduced the concept of a relational community, which allows users to specify relational 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, [35], [36] 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 [37]–[40]. For example, [41], [42] designed bipartite modularity for bipartite graphs. [15], [43] investigated hypergraph modularity to incorporate higher-order information of structures. Arenas et al. [13] first proposed motif modularity, which is defined as the normalized fraction of motif instances within communities minus the expected fraction in a random network. Unlike our motif density modularity, it focuses on the homogeneous graphs and relies on a random graph model that preserves degree distribution but ignores motif structures, thereby restricting its ability to capture semantically rich communities. Recently, [44], [45] 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.

III. Problem Formulation

The HIN can be modeled 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 $\ell_H: V_H \rightarrow L_H$ 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, Fig. 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, $\texttt{deg}_H(v)$ denotes the degree of vertex v in H. Table I summarizes the frequent notations in this paper.

TABLE I
Frequently used 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, 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, 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
$\texttt{deg}_H(v)$ / $\texttt{Mdeg}_H(v)$	the degree / motif degree of v in H
$\mathcal{N}_H(v)$	the neighbors of v in H
$\texttt{vol}_H(S)$ / $\texttt{Mvol}_H(S)$	sum of degree / motif degree of S in H
$\mathcal{B}_M$	motif-based bipartite graph of M and H
$Eq^M_{H[S]}(v)$	the equivalent vertices of v in $H[S]$
H_M	the MW-HIN of M and H
V_{H_M}	vertices with motif instances of M in H
Θ_v^S	the M-ratio of the vertex v in $H[S]$
$V_H(\ell_H(v))$	the vertices with the type of v in H

A. Motif Density Modularity

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

Motif. The motif is a fundamental building block of a graph [46], [47]. 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 the schema of H.

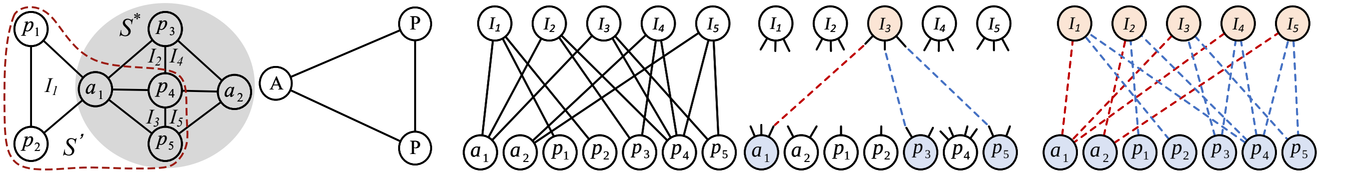
Definition 1. *(Motif Instance). Given an HIN H, a motif M, and a subgraph $H' \subseteq H$, H' is a motif instance of M iff H' is isomorphic to M, i.e., there exists a bijective mapping $\phi: V_M \rightarrow V_{H'}$ such that (1) $\forall u \in V_M, \ell_M(u)=\ell_H(\phi(u))$ and (2) $\forall (u,v)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

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., $\texttt{Mvol}_H(S)=\sum_{v\in S}\texttt{Mdeg}_H(v)$.

Generalized Motif Modularity. The modularity is widely used to evaluate the community quality, which is defined as the differences in graph structures from an expected random graph [48]. Higher modularity indicates well-partitioned communities with more internal connections and fewer external connections. However, as mentioned in Section II, directly applying existing modularity to HINs suffers from the loss of the higher-order structure as well as the heterogeneous information. Hence, we propose a new 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*

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



(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

Fig. 3. An example of MGM.

where $Exp[|\mathcal{I}^M_{H[S]}|]$ *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 $Exp[|\mathcal{I}^M_{H[S]}|]$. The classic modularity [48] leverages the random graph model that preserves the degree distribution of the original graph to compute $Exp[|E_{G[S]}|]$. However, it is specifically designed for homogeneous networks and does not consider motifs, making it unsuitable for HINs. Even if a vertex does not belong to any motif instance in the original HIN, it may still form a motif instance in the generated random graph. 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. $\mathcal{R}_{\mathcal{B}_M} = \{v : \exists I \in \mathcal{I}^M_H, v$ represents $I\}$ and $|\mathcal{R}_{\mathcal{B}_M}| = |\mathcal{I}^M_H|$. 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, Fig. 3(c) depicts the motif-based bipartite graph of H and M shown in Fig. 3(a) and Fig. 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 Fig. 3(a), since p_1 is within the motif instance I_1, there is an edge between I_1 and p_1 in Fig. 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) $\forall v \in \mathcal{L}_{\mathcal{B}_M}$, $\texttt{deg}_{\mathcal{B}_M}(v) = \texttt{Mdeg}_H(v)$*;*

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

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

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 $p_v = \frac{\texttt{deg}_{\mathcal{B}_M}(v)}{\texttt{vol}_{\mathcal{B}_M}(V_H(\ell_H(v)))}$. 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 Fig. 3(d), after breaking all edges of $\mathcal{B}_M$ in Fig. 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 Fig. 3(e).

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*

$$Exp[|\mathcal{I}^M_{H[S]}|] = |\mathcal{I}^M_H| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}, \tag{2}$$

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

The detailed proof can be found in [49]. According to Theorem 1, the generalized motif modularity is

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

However, since GMM inherits the form of classic modularity, it also suffers from the resolution limit [50] and the free-rider effect [51]. Specifically, the resolution limit hinders the identification of small but dense communities, whereas the free-rider effect results in the participation of irrelevant vertices. To address these issues, we introduce an improved version of GMM, named *motif density modularity*.

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:*

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

[1] Due to space limitations, some lemmas and proofs of this paper are provided in the technical report [49].

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. Intuitively, the MDM not only leverages the structural cohesiveness of density modularity [14] and the semantic expressiveness of motif modularity [13], but also incorporates our novel motif-based random graph model tailored for motifs in HINs. Therefore, MDM can effectively capture both the structure and semantic cohesiveness of communities. Furthermore, we theoretically demonstrate that MDM alleviates the free-rider effect and the resolution limit compared to GMM in [49].

B. 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 two vertices $u, v \in V_H$, u and v are M-connected, denoted by $u \leftrightarrow v$, if (1) u and v belong to the same motif instance, or (2) $\exists I_s, I_t \in \mathcal{I}_H^M$, such that $u \in V_{I_s}$, $v \in V_{I_t}$, and $I_s \leftrightarrow I_t$. The subgraph $H[S] \subseteq H$ is M-connected if $\forall u, v \in S$, $u \leftrightarrow v$. 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 a subgraph $H[S] \subseteq H$ satisfying:*

(1) $Q \subseteq S$;

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

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

Take the HIN H in Fig. 3(a) and the motif M in Fig. 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.

For the MOCHI problem, we would like to highlight two issues. (1) Why should the community be M-connected? If an HIN is M-connected, it must be connected, but not vice versa. Therefore, we apply M-connectivity to ensure that the community is densely connected via motif instances. (2) How to select motifs? (i) *Predefined semantic motifs*: employ known motifs with specific semantics, e.g., the credit card fraud motif [16] in Fig. 2(b). (ii) *Custom motifs*: design motifs mutually or specify semantics in natural language to identify relevant motifs by leveraging the large language model. (iii) *Discovered motifs*: apply motif discovery methods [52]–[54] to extract statistically significant motifs w.r.t. structure automatically.

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

$$DM(G, S) = \frac{1}{|S|}(|E_{G[S]}| - \frac{\texttt{vol}_G(S)^2}{4|E_G|}), \tag{3}$$

where $\texttt{vol}_G(S)$ is the sum of the degree of all vertices in S over G. 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 [49]. □

Due to the hardness of the MOCHI problem, obtaining an exact global solution is computationally expensive. Moreover, the existing learning-based methods for combinational optimization struggle to outperform tailored heuristic approaches on large graphs [55]. Hence, we propose heuristic algorithms to tackle this problem.

IV. Algorithms

In this section, we first present a basic algorithm to solve the problem and then propose two more efficient algorithms.

A. Basic Algorithm

A naive solution is to iteratively remove the vertex that maximizes MDM. Algorithm 1 outlines the pseudo-code of the basic algorithm. Specifically, Algorithm 1 first finds all motif instances of M in H and the M-connected vertices that contains Q (lines 1–3). Then, in each round (lines 4-10), Algorithm 1 deletes each vertex that maximizes MDM after its removal to get a subgraph $H[S_i]$, which is M-connected and contains Q. Finally, $H[S^*]$ with the maximal MDM among these identified subgraphs is returned (lines 11-12).

Notably, the computation of MDM relies on the number of motif instances in the whole HIN, which makes enumerating all motif instances necessary. Many subgraph matching methods [56], [57] can be applied to compute motif instances. Given that motif sizes are typically small in practice [58], [59], we adopt *RapidMatch* [60], a join-based subgraph matching algorithm, which enumerates these small motifs efficiently.

Theorem 3. *The time complexity of Algorithm 1 is $O(\alpha_t(H, M) + |V_H|^3 \times \texttt{Mdeg}_H^{max} \times |V_M|)$, where $\alpha_t(H, M)$ is the time cost of enumerating motif instances and $\texttt{Mdeg}_H^{max}$ is the maximum motif degree of H. The space complexity of Algorithm 1 is $O(\max(\alpha_s(H, M), |\mathcal{I}_H^M| \times |V_M|))$, where $\alpha_s(H, M)$ is the space cost of enumerating motif instances. [49]*

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

B. MW-HIN-based Algorithm

The basic algorithm incurs significant overhead as it evaluates all vertices in each iteration and repeatedly traverses the HIN to maintain M-connectivity. To address these inefficiencies, we propose an efficient *vertex selection strategy* and a compact data structure called *motif-based weighted HIN* (MW-HIN). Based on these, we design our second algorithm.

1) Vertex Selection:

The basic algorithm evaluates a vertex by removing it along with the vertices that violate M-connectivity, and then computing the MDM of the remaining HIN. We observe that when only a single vertex is removed, the MDM of the remaining HIN can be computed much more easily.

Lemma 2. *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*

$$MDM(H,S\setminus\{v\},M)=\frac{|\mathcal{I}^M_{H[S]}|-\texttt{Mdeg}_{H[S]}(v)}{|S|-1} - \frac{|\mathcal{I}^M_H|\prod_{u\in V_M}\frac{\texttt{Mvol}_H(S(\ell_M(u)))-\texttt{Mdeg}_H(v)\eta(u,v)}{\texttt{Mvol}_H(V_H(\ell_M(u)))}}{|S|-1}. \qquad (4)$$

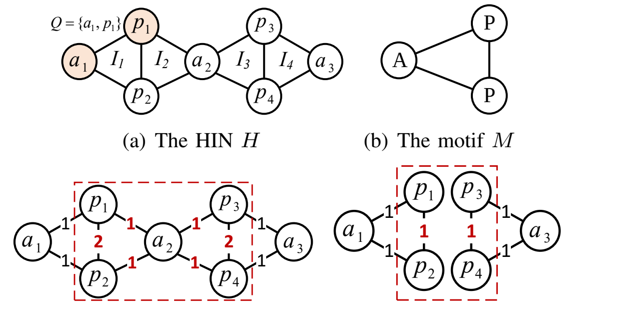
If $\ell_M(u) = \ell_H(v)$, $\eta(u,v) = 1$; otherwise, $\eta(u,v) = 0$.

Lemma 2 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 2, 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: Removable 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.*

For example, in Fig. 4(a) and Fig. 4(b), since $Q = \{a_1, p_1\}$ and $H[S \setminus \{a_3\}]$ is M-connected, the removable vertex is a_3. After removing a_3, there is no removable vertex in the HIN. But, if we remove some vertices together, e.g., $\{p_3, p_4\}$, the remaining vertices are still M-connected. Based on the above observation, we introduce the second case.



(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

Fig. 4. An example of the MW-HIN.

Case II: Removable equivalent vertices

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 $Eq^M_H(v) = \{u \mid u \in V_H, \mathcal{I}^M_H(u) = \mathcal{I}^M_H(v)\}$. Here, $\mathcal{I}^M_H(u)$ 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) $Eq^M_H(v)$ contains v and (2) if v is a removable vertex, $Eq^M_H(v) = \{v\}$. If $Eq^M_H(v)$ has more than one vertex, v cannot be a removable vertex since its removal will disrupt all the motif instances of other vertices in $Eq^M_H(v)$. However, if we remove all the vertices in $Eq^M_H(v)$ together and the remaining HIN is still M-connected, v can also be removable with its equivalent 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 $Eq^M_H(v)$, the remaining HIN is still M-connected, and (2) $MDM(H,S \setminus \{v\},M)$ is maximal.

2) Motif-based Weighted HIN:

After deleting $Eq^M_H(v)$, 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*, 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 $\mathcal{I}^M_H$, 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, $V_{H_M} = \{v \mid v \in V_H, \mathcal{I}^M_H(v) \neq \emptyset\}$, $E_{H_M} = \{(u,v) \mid u,v \in V_{H_M}, \mathcal{I}^M_H(u) \cap \mathcal{I}^M_H(v) \neq \emptyset\}$, $L_{H_M} = L_H$, 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., $\forall(u,v) \in E_{H_M}$, $\omega_{H_M}(u,v) = |\mathcal{I}^M_H(u) \cap \mathcal{I}^M_H(v)|$.*

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, Fig. 4(c) shows the MW-HIN of HIN and motif in Fig. 4(a) and Fig. 4(b), respectively. Since p_1 and p_2 share two common motif instances I_1 and I_2, the edge weight of (p_1, p_2) in MW-HIN is 2.

Lemma 3. *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.*

We can easily check the M-connectivity of the HIN by verifying the connectivity of the MW-HIN via expanding from Q. Next, we describe the construction of MW-HIN.

Construction of MW-HIN. The MW-HIN can be constructed by traversing 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.

3) MW-HIN-based Algorithm:

Based on the vertex selection and MW-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 4. *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: $\omega_{H_M}(u, w) - \omega_{H'_M}(u, w) \in [0, |\mathcal{I}_H^M(v)|]$.

Lemma 4 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 Fig. 4(c) and Fig. 4(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 4, 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, we delete the equivalent vertices of v and their incident edges.

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 should also 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 $R(v) \cup S$. In particular, since removing $Eq_H^M(v)$ may break the M-connectivity among the vertices in Q, query vertices might be included in 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 $R()$, enumerates motif instances of M in H, constructs the MW-HIN H_M, and finds connected vertices containing 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 maximal $MDM(H, S_i \setminus \{v\}, M)$ (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). Oth-

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 maximal 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 connected vertices containing $Q$ 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 $Eq^M_{H[S_i]}(v)$;
11:        $H_M[S_i \setminus Eq^M_{H[S_i]}(v)] \leftarrow$ remove $Eq^M_{H[S_i]}(v)$ from $H_M[S_i]$;
12:        lazily update edges' weights of $H_M[S_i \setminus Eq^M_{H[S_i]}(v)]$;
13:        if $H_M[S_i \setminus Eq^M_{H[S_i]}(v)]$ is connected then
14:            $S_{i+1} \leftarrow S_i \setminus Eq^M_{H[S_i]}(v)$; $R() \leftarrow R() \setminus Eq^M_{H[S_i]}(v)$;
15:            break;
16:        else
17:            $R(v) \leftarrow R(v) \cup$ (the vertices disconnected from $Q$ in $H_M[S_i \setminus Eq^M_{H[S_i]}(v)]$);
18:    $i \leftarrow i + 1$;
19: $S^* \leftarrow \text{argmax}_{S \in \{S_0, S_1, ..., S_{i-1}\}} MDM(H, S, M)$;
20: return $H[S^*]$;
```

erwise, Algorithm 2 computes equivalent vertices $Eq^M_{H[S_i]}(v)$ of v, removes them from $H_M[S_i]$, and lazily updates edges' weights of $H_M[S_i \setminus Eq^M_{H[S_i]}(v)]$ (lines 10-12). Then, it checks the connectivity of the updated $H_M[S_i \setminus Eq^M_{H[S_i]}(v)]$ by performing the BFS starting from query vertices. If the updated $H_M[S_i \setminus Eq^M_{H[S_i]}(v)]$ is connected, v is selected for deletion in this round. Algorithm 2 sets S_{i+1} by $S_i \setminus Eq^M_{H[S_i]}(v)$ and removes $Eq^M_{H[S_i]}(v)$ from the reuse set for the next round (lines 13-15). Otherwise, v cannot be removed in this round and Algorithm 2 adds the vertices disconnected from Q in $H_M[S_i \setminus Eq^M_{H[S_i]}(v)]$ into $R(v)$ for reuse (lines 16-17). The vertex deletion continues until no more vertices can be deleted. Finally, $H[S_i]$ with the maximal MDM among the identified subgraphs is returned (lines 19-20).

Theorem 4. *The time complexity of Algorithm 2 is $O(\alpha_t(H, M) + |V_{H_M}| \times \sigma_{\text{check}} \times (\beta_{\text{update}} \times \texttt{Mdeg}_H^{max} \times \log \texttt{Mdeg}_H^{max} + |V_{H_M}| + |E_{H_M}|))$, where σ_{check} is the number of vertex checks performed per round to determine candidate vertices and β_{update} is the number of exact edge weight updates. The space complexity of Algorithm 2 is $O(\max(\alpha_s(H, M), |\mathcal{I}_H^M| \times |V_M| + |V_{H_M}| \times |R| + |E_{H_M}|))$, where $|R|$ is the size of the reuse set. [49]*

C. Motif-distance-based Algorithm

The MW-HIN-based algorithm improves the basic algorithm through vertex selection and MW-HIN, but its efficiency declines on large HINs with numerous motif instances and vertices. To address this problem, we propose a third algorithm that introduces a new metric, *motif distance*, to guide vertex selection, rather than relying solely on MDM.

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 $\texttt{Mdist}_H(u, v)$, is*

(1) if $u \leftrightarrow v$, $\texttt{Mdist}_H(u, v) = \min\{n \mid \exists I_1, I_2, ..., I_n \in \mathcal{I}_H^M, u \in V_{I_1}, v \in V_{I_n}$, 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 $u \not\leftrightarrow v$, $\texttt{Mdist}_H(u, v) = +\infty$.

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 $\texttt{Mdist}_H(S, u) = \min_{v \in S}\{\texttt{Mdist}_H(u, v)\}$. 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 $\texttt{dist}_{H_M}(u, v)$ to denote the unweighted shortest path distance between u and v in H_M.

Lemma 5. *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, $\texttt{Mdist}_H(u, v) = +\infty$.*

Lemma 5 indicates that we can compute the motif distance by computing the distance between vertices in the MW-HIN. 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 observation that the vertex close to the query vertices tends to be within the query vertices' corresponding community [21], [61]. Therefore, we try to remove the vertices farthest from the 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 6. *Given a motif M, an M-connected HIN H. Let $S_0 \subseteq V_H$, and $S' = \{v \in V_H \mid \texttt{Mdist}_H(S_0, v) = \max_{u \in V_H} \texttt{Mdist}_H(S_0, u)\}$. If $H[S_0]$ is M-connected and $S_0 \subseteq V_H \setminus S'$, $H[V_H \setminus S']$ is M-connected.*

To find an M-connected subgraph $H[S_0]$ containing Q as small as possible, we adopt the well-known approximate Steiner tree algorithm [62]. 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,

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 maximal 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: $H[S_0] \leftarrow$ 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 connected vertices containing $S_0$ 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 $\Theta_v^S$;
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 $\Theta_v^S$;
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 $H[S^*]$;
```

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 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 maximal 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 the current HIN, leading to a huge update overhead. Hence, we design a lightweight goodness function called *M-ratio*. Note that the M-ratio is motivated by the density ratio in homogeneous networks, which has been used to identify the vertex removal order effectively and efficiently to find a community with the maximal density modularity [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*

$$\Theta_v^S = \frac{\texttt{Mdeg}_H(v)}{\texttt{Mdeg}_{H[S]}(v)}. \tag{5}$$

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 7. *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 7 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, coarse-grained and fine-grained deletions form the motif-distance-based algorithm, outlined in Algorithm 3. It first computes motif instances and builds the MW-HIN (lines 1–2). Then, it expands Q via an approximate Steiner tree to obtain an M-connected subgraph $H[S_0]$, computes motif distances, collects the connected vertices containing S_0 in H_M, and initializes S^* (lines 3–7). Next, coarse-grained deletion removes vertices farthest from S_0 in batches to identify a subgraph with the maximal MDM (lines 8–12). Fine-grained deletion then computes M-ratios of vertices, iteratively removes the farthest vertices in descending order of their M-ratios, updates the affected M-ratios, and records the optimal S^* (lines 13–25). Finally, the algorithm terminates when $S = S_0$ and returns $H[S^*]$ (line 26).

Theorem 5. *The time complexity of Algorithm 3 is $O(\alpha_t(H, M) + |V_{H_M}| \times \log|V_{H_M}| \times \texttt{Mdeg}_H^{max} \times |V_M|)$. The space complexity of Algorithm 3 is $O(\max(\alpha_s(H, M), |\mathcal{I}_H^M| \times |V_M| + |V_{H_M}| + |E_{H_M}|))$. [49]*

V. Experiments

In this section, we evaluate the effectiveness and efficiency of our proposed methods 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.

A. Experimental Setup

Datasets. We use five real-world HINs in experiments. Specifically, *CIDeRplus* [63] is a biological network including chemicals, genes, diseases, etc. *DBLP* [64] is a bibliographic network consisting of authors, papers, venues, and topics. *TMDB* [10] is a movie network comprising movies, directors, and countries, etc. *Freebase* [65] and *DBpedia* [61] are knowledge graphs. Table II summarizes statistics of these datasets.

Competitors. We compare a set of methods in experiments.

- RC [6]: a heterogeneous community model that finds the community satisfying relational constraints. To apply RC to our problem, we decompose each motif into the relational constraints and return the maximal relational community containing Q.

TABLE II

DATASET STATISTICS (Dens: DENSITY, Deg: AVERAGE DEGREE)

Dataset	$\lvert V_H \rvert$	$\lvert E_H \rvert$	$\lvert L_H \rvert$	Dens	Deg
CIDeRplus(CR)	13,924	83,916	15	6.03	12.05
DBLP(DL)	36,138	170,794	4	4.73	9.45
TMDB(TD)	71,978	113,581	7	1.58	3.16
Freebase(FB)	3,993,552	10,998,482	3,238	2.75	5.51
DBpedia(DP)	4,521,912	14,039,200	439	3.10	6.21

- MM [13]: a higher-order clustering method based on the motif modularity. To apply MM to our problem, we replace the MDM with MM in the MW-HIN-based algorithm.
- HM [15]: a higher-order clustering method based on the hypergraph modularity. To apply HM to our problem, we treat motif instances as hyperedges to construct hypergraphs.
- 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 $\{3, \underline{4}, 5, 6, 7\}$, $\{\underline{1}, 2, 4, 6, 8\}$, and $\{60\%, 70\%, 80\%, 90\%, \underline{100\%}\}$, respectively, where the underlined numbers denote the default values. Following previous works [58], [59], we generate random motifs with sizes ranging from 3 to 7 by performing random walks on each dataset. In addition, the query vertex sets are randomly generated while ensuring that they are M-connected such that the result is non-empty. Overall, in each experiment, we generate 200 queries and report the average performance.

B. Effectiveness Evaluation

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

- **Diameter**: $\texttt{Diam}(C) = max_{u,v \in C}\{\texttt{dist}_{H[C]}(u,v)\}$, where $\texttt{dist}_{H[C]}(u,v)$ is the shortest distance of u, v in $H[C]$.
- **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.

Exp-1: Overall performance evaluation. We compare the community quality and running time returned by different methods. Since Basic always fails to finish within the time limit (10^4 seconds), we do not report its results here. Table III shows the results. All the baselines except RC depend on motif instances to adapt to our problem, thus achieving the same enumeration time. Our proposed methods MW and MD outperform other competitors in most cases w.r.t. all metrics except the search time. Although RC and HM are generally more efficient, they often produce communities of lower quality. In contrast, MD identifies higher-quality communities within a reasonable response time. 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.

TABLE III

OVERALL COMPARISON OF DIFFERENT MODELS

Dataset	Models	Diam	Sim	Coh	$\lvert C \rvert$	T_1 [1]	T_2 [1]
CR	RC	6.2	0.42	0.0023	1020	-	**0.02**
	MM	5.4	0.40	0.0019	1120	0.41	128.71
	HM	6.6	0.41	0.0024	1490	0.41	$\underline{0.69}$
	GMM	5.1	0.43	0.0039	690	0.41	123.12
	MW	**3.5**	**0.50**	$\underline{0.0218}$	**137**	0.41	121.36
	MD	$\underline{4.4}$	$\underline{0.46}$	**0.0377**	$\underline{417}$	0.41	0.81
DL	RC	23.1	0.67	0.0101	13765	-	**0.12**
	MM	16.7	0.67	0.0255	6486	0.62	236.49
	HM	$\underline{12.3}$	0.67	0.0327	7220	0.62	$\underline{1.44}$
	GMM	20.4	0.69	0.0266	3779	0.62	222.47
	MW	15.8	$\underline{0.70}$	**0.0498**	**227**	0.62	223.61
	MD	**8.1**	**0.71**	$\underline{0.0456}$	$\underline{1064}$	0.62	2.30
TD	RC	11.0	0.62	0.0002	15070	-	**0.06**
	MM	9.7	0.62	0.0005	6944	1.26	494.23
	HM	10.6	0.60	0.0005	12529	1.26	$\underline{2.54}$
	GMM	9.5	0.66	0.0012	4610	1.26	494.42
	MW	$\underline{7.9}$	**0.78**	**0.0261**	**212**	1.26	498.34
	MD	**5.9**	$\underline{0.76}$	$\underline{0.0217}$	$\underline{563}$	1.26	2.80
FB	RC	7.5	0.80	0.0224	5152	-	$\underline{1.05}$
	MM	5.4	0.81	0.0250	3824	0.73	57.56
	HM	4.2	0.83	0.0273	1432	0.73	2.42
	GMM	4.6	0.83	0.0260	1420	0.73	57.11
	MW	$\underline{3.5}$	$\underline{0.87}$	$\underline{0.0496}$	**171**	0.73	56.84
	MD	**3.1**	**0.87**	**0.1323**	$\underline{645}$	0.73	**0.51**
DP	RC	11.4	0.77	0.0225	12520	**-**	**0.74**
	MM	9.3	0.79	0.0259	9538	1.87	576.93
	HM	6.8	0.79	0.0272	7241	1.87	4.88
	GMM	8.1	0.81	0.0264	4051	1.87	559.97
	MW	$\underline{6.8}$	$\underline{0.84}$	$\underline{0.0594}$	**199**	1.87	567.44
	MD	**4.3**	**0.85**	**0.1994**	$\underline{1380}$	1.87	$\underline{2.59}$

[1] T_1: motif instances enumeration time (sec); T_2: community search time (sec).

Exp-2: Sensitivity analysis. To examine the sensitivity of our methods w.r.t. motifs, we vary motif edges from 3 to 6 and report the results of MW and MD on DBpedia in Fig. 5. Both methods are motif-sensitive, allowing the discovery of distinct semantic communities by varying the motifs. Specifically, as $|E_M|$ increases, $|L_M|$ decreases since generating motifs that are both type-rich and structurally cohesive becomes more challenging, resulting in communities with fewer types, denser connections, and smaller diameters. Meanwhile, a larger $|E_M|$ imposes stricter structural constraints, whereas a smaller $|L_M|$ implies looser type constraints. This trade-off increases the uncertainty of motif instances and relevant vertices, causing the community size, similarity, cohesiveness, and running time to fluctuate without a consistent trend. Compared with MW, MD is more stable w.r.t. diameter and running time by pruning the search space using motif distance, whereas MW achieves comparable or even better stability across other quality metrics due to its finer-grained optimization of MDM.

C. Efficiency Evaluation

In this section, we evaluate the efficiency of our proposed algorithms from running time and memory. Note that if a query

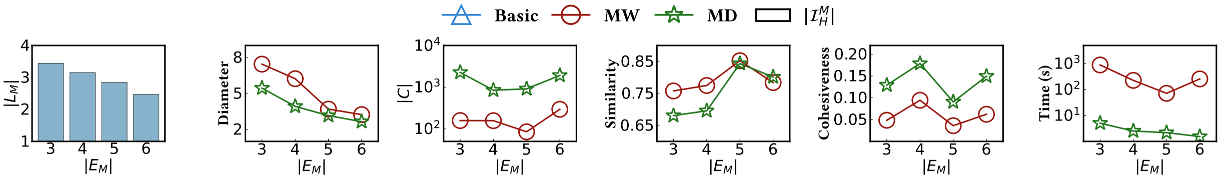
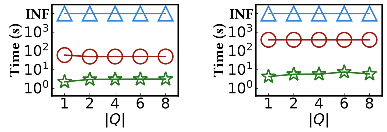
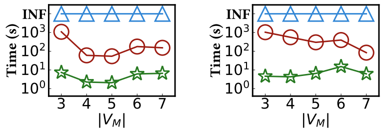


Fig. 5. Sensitivity analysis



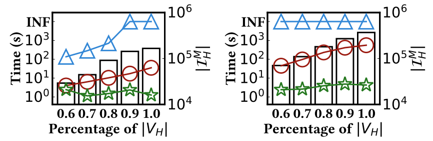
(a) Freebase (b) DBpedia

Fig. 6. Effect of $|Q|$



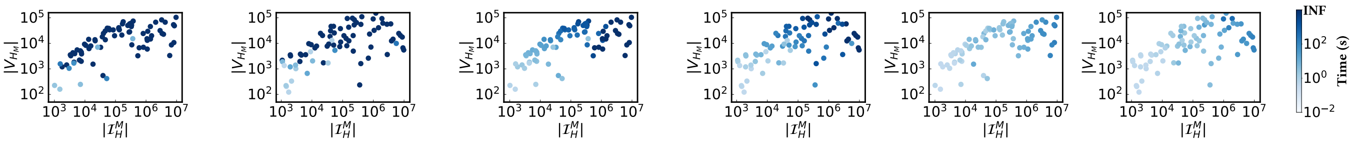
(a) Freebase (b) DBpedia

Fig. 7. Effect of $|V_M|$



(a) Freebase (b) DBpedia

Fig. 8. Effect of $|V_H|$



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

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

does not complete within 10^4 seconds, we denote it by ”INF”.

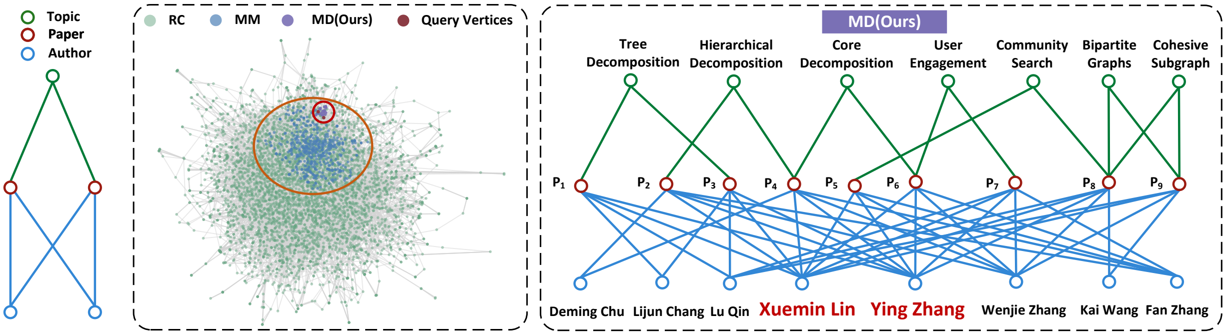
Exp-3: Effect of $|Q|$. We vary the number of query vertices $|Q|$ from 1 to 8 to test its effect on algorithms. The running times of the three algorithms are shown in Fig. 6. We observe that Basic fails to complete within the time limit, whereas MW and MD are faster by at least one to three orders of magnitude, owing to the incorporation of optimization strategies. In addition, the running time of MW decreases slightly when varying $|Q|$ on Freebase, while the running time of MD increases slightly. It is because for larger $|Q|$, MW can skip more vertices whose reuse set contains query vertices as these vertices will never be removed, while MD takes more time to construct an approximate Steiner tree to find a small M-connected subgraph containing Q.

Exp-4: Effect of $|V_M|$. We study the effect of motif size $|V_M|$ on three algorithms. Fig. 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-5: 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. Fig. 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 $|V_{H_M}|$ increase, 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 Fig. 8(a).

Exp-6: Effect of $|\mathcal{I}_H^M|$ and $|V_{H_M}|$. As mentioned in previous experiments, the running time of algorithms is mainly determined by $|\mathcal{I}_H^M|$ and $|V_{H_M}|$. In this experiment, we explore the effect of them on three algorithms. Fig. 9 shows the results. The darker the color of the points, 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 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.

Exp-7: Memory analysis. In this experiment, we evaluate the memory usage of our proposed methods. Since Basic fails to complete within the time limit, we only report the results of MW and MD. As shown in Table IV, we have two observations. (1) On small HINs such as CR, DL, and TD, both MW and MD consume much more memory than the HIN, with MD usually less than MW. This is because these type-sparse HINs contain many motif instances (on average 288,095), and MW incurs extra overhead to maintain reuse sets. (2) On large HINs such as FB and DP, the memory usages of MW and MD are similar, only slightly higher than the HINs. This is because in these type-rich HINs, only a small fraction of vertices are motif-related, leading to relatively few motif instances. In such cases, memory consumption is usually dominated by motif



(a) Motif

(b) Communities of RC, MM, and MD

Fig. 10. A case study on DBLP.

TABLE IV
MEMORY ANALYSIS (MB)

Dataset	HIN	MW	MD
CR	41	215	198
DL	86	420	277
TD	63	602	487
FB	5546	5846	5846
DP	6982	7633	7646

TABLE V
ABLATION STUDY ON MW

Models	Running time (sec)					Average speedup
	CR	DL	TD	FB	DP	
MW	0.2	0.16	0.9	2.3	3.0	+94%
w/o V	5.8	2.5	44.0	46.4	16.5	+16%
w/o M	0.3	0.6	4.9	2.4	5.6	+87%
w/o V&M	6.2	2.9	70.8	52.7	18.8	-

TABLE VI
ABLATION STUDY ON MD

Models	Running time (sec)					Average speedup
	CR	DL	TD	FB	DP	
MD	2.0	2.8	6.5	2.1	5.9	+73%
w/o C	1.8	7.6	14.5	2.9	14.7	+67%
w/o R	2.2	3.8	11.5	3.0	7.1	+67%
w/o C&R	2.3	41.5	62.6	9.2	95.5	-

enumeration, while the reuse set in MW may be small. As a result, MW's memory usage becomes comparable to, or even slightly smaller than, that of MD.

D. Ablation Studies

Exp-8: Ablation study of MW. In this experiment, we evaluate the effectiveness of the vertex selection strategy and MW-HIN in MW. Since MW without any optimizations always fail to finish within the time limit, we use the motifs with $|\mathcal{I}_H^M| < 10^4$. Table V shows the results. Specifically, w/o V denotes MW without the vertex selection strategy, w/o M denotes MW without MW-HIN, and w/o V&M removes both. The results show that the vertex selection strategy and MW-HIN can improve the average performance of w/o V&M by 87% and 16%, 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-9: Ablation study of MD. In this experiment, we evaluate the effectiveness of the coarse-grained deletion and M-ratio in MD. Table VI reports the results, where w/o C denotes MD without coarse-grained deletion, w/o R refers to MD without M-ratio (while still using $MDM(H, S \setminus \{v\}, M)$), and w/o C&R removes both. The results show that both the coarse-grained deletion and M-ratio can improve the average performance of w/o C&R by 67%. Furthermore, when both techniques are combined, the overall performance of w/o C&R can be improved by 73%. These results demonstrate the effectiveness of coarse-grained deletion and M-ratio.

E. 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 Fig. 10(a). Fig. 10(b) shows the community returned by different methods. 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., community search, cohesive subgraph). This demonstrates MD's effectiveness for real-world HIN community search.

VI. 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 devise learning-based methods to address the MOCHI problem.

References

[1] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 990–998, 2008.

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

[3] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han, "Heterogeneous network representation learning: A unified framework with survey and benchmark," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10, pp. 4854–4873, 2020.

[4] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum, "Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames," in *The Semantic Web–ISWC 2016: 15th International Semantic Web Conference*, pp. 177–185, 2016.

[5] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008.

[6] X. Jian, Y. Wang, and L. Chen, "Effective and efficient relational community detection and search in large dynamic heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 13, no. 10, pp. 1723–1736, 2020.

[7] Y. Fang, Y. Yang, W. Zhang, X. Lin, and X. Cao, "Effective and efficient community search over large heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 13, no. 6, pp. 854–867, 2020.

[8] Y. Yang, Y. Fang, X. Lin, and W. Zhang, "Effective and efficient truss computation over large heterogeneous information networks," in *2020 IEEE 36th international conference on data engineering (ICDE)*, pp. 901–912, 2020.

[9] Y. Jiang, Y. Fang, C. Ma, X. Cao, and C. Li, "Effective community search over large star-schema heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2307–2320, 2022.

[10] Y. Zhou, Y. Fang, W. Luo, and Y. Ye, "Influential community search over large heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 16, no. 8, pp. 2047–2060, 2023.

[11] Y. Li, G. Zang, C. Song, X. Yuan, and T. Ge, "Leveraging semantic information for enhanced community search in heterogeneous graphs," *Data Science and Engineering*, pp. 1–18, 2024.

[12] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[13] A. Arenas, A. Fernandez, S. Fortunato, and S. Gomez, "Motif-based communities in complex networks," *Journal of Physics A: Mathematical and Theoretical*, vol. 41, no. 22, p. 224001, 2008.

[14] J. Kim, S. Luo, G. Cong, and W. Yu, "Dmcs: Density modularity based community search," in *Proceedings of the 2022 International Conference on Management of Data*, pp. 889–903, 2022.

[15] Z. Feng, M. Qiao, and H. Cheng, "Modularity-based hypergraph clustering: Random hypergraph model, hyperedge-cluster relation, and computation," *Proceedings of the ACM on Management of Data*, vol. 1, no. 3, pp. 1–25, 2023.

[16] X. Qiu, W. Cen, Z. Qian, Y. Peng, Y. Zhang, X. Lin, and J. Zhou, "Real-time constrained cycle detection in large dynamic graphs," *Proceedings of the VLDB Endowment*, vol. 11, no. 12, pp. 1876–1888, 2018.

[17] Y. Fang, X. Huang, L. Qin, Y. Zhang, W. Zhang, R. Cheng, and X. Lin, "A survey of community search over big graphs," *The VLDB Journal*, vol. 29, pp. 353–392, 2020.

[18] M. Sozio and A. Gionis, "The community-search problem and how to plan a successful cocktail party," in *SIGKDD*, pp. 939–948, 2010.

[19] W. Cui, Y. Xiao, H. Wang, and W. Wang, "Local search of communities in large graphs," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 991–1002, 2014.

[20] E. Akbas and P. Zhao, "Truss-based community search: a truss-equivalence based indexing approach," *Proceedings of the VLDB Endowment*, vol. 10, no. 11, pp. 1298–1309, 2017.

[21] X. Huang, L. V. Lakshmanan, J. X. Yu, and H. Cheng, "Approximate closest community search in networks," *Proceedings of the VLDB Endowment*, vol. 9, no. 4, 2015.

[22] L. Chang, X. Lin, L. Qin, J. X. Yu, and W. 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*, pp. 459–474, 2015.

[23] J. Hu, X. Wu, R. Cheng, S. Luo, and Y. Fang, "On minimal steiner maximum-connected subgraph queries," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 11, pp. 2455–2469, 2017.

[24] W. Cui, Y. Xiao, H. Wang, Y. Lu, and W. Wang, "Online search of overlapping communities," in *Proceedings of the 2013 ACM SIGMOD international conference on Management of data*, pp. 277–288, 2013.

[25] L. Yuan, L. Qin, W. Zhang, L. Chang, and J. Yang, "Index-based densest clique percolation community search in networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 5, pp. 922–935, 2017.

[26] J. Gao, J. Chen, Z. Li, and J. Zhang, "Ics-gnn: lightweight interactive community search via graph neural network," *Proceedings of the VLDB Endowment*, vol. 14, no. 6, pp. 1006–1018, 2021.

[27] Y. Jiang, Y. Rong, H. Cheng, X. Huang, K. Zhao, and J. Huang, "Query driven-graph neural networks for community search: from non-attributed, attributed, to interactive attributed," *Proceedings of the VLDB Endowment*, vol. 15, no. 6, pp. 1243–1255, 2022.

[28] L. Li, S. Luo, Y. Zhao, C. Shan, Z. Wang, and L. Qin, "Coclep: Contrastive learning-based semi-supervised community search," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 2483–2495, 2023.

[29] J. Wang, K. Wang, X. Lin, W. Zhang, and Y. Zhang, "Neural attributed community search at billion scale," *Proceedings of the ACM on Management of Data*, vol. 1, no. 4, pp. 1–25, 2024.

[30] Y. Wang, X. Gou, X. Xu, Y. Geng, X. Ke, T. Wu, Z. Yu, R. Chen, and X. 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*, pp. 1680–1690, 2024.

[31] K. Wang, W. Zhang, X. Lin, Y. Zhang, L. Qin, and Y. Zhang, "Efficient and effective community search on large-scale bipartite graphs," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 85–96, 2021.

[32] Z. Chen, Y. Zhao, L. Yuan, X. Lin, and K. Wang, "Index-based biclique percolation communities search on bipartite graphs," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pp. 2699–2712, 2023.

[33] X. Huang and L. V. Lakshmanan, "Attribute-driven community search," *Proceedings of the VLDB Endowment*, vol. 10, no. 9, pp. 949–960, 2017.

[34] Q. Liu, Y. Zhu, M. Zhao, X. Huang, J. Xu, and Y. Gao, "Vac: vertex-centric attributed community search," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pp. 937–948, 2020.

[35] Y. Wang, C. Gu, X. Xu, X. Zeng, X. Ke, and T. Wu, "Efficient and effective (k, p)-core-based community search over attributed heterogeneous information networks," *Information Sciences*, vol. 661, p. 120076, 2024.

[36] L. Qiao, Z. Zhang, Y. Yuan, C. Chen, and G. Wang, "Keyword-centric community search over large heterogeneous information networks," in *International Conference on Database Systems for Advanced Applications*, pp. 158–173, 2021.

[37] T. N. Dinh, X. Li, and M. T. Thai, "Network clustering via maximizing modularity: Approximation algorithms and theoretical limits," in *2015 IEEE International Conference on Data Mining*, pp. 101–110, 2015.

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

[39] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of statistical mechanics: theory and experiment*, vol. 2008, no. 10, p. P10008, 2008.

[40] V. A. Traag, L. Waltman, and N. J. Van Eck, "From louvain to leiden: guaranteeing well-connected communities," *Scientific reports*, vol. 9, no. 1, p. 5233, 2019.

[41] M. J. Barber, "Modularity and community detection in bipartite networks," *Physical Review E*, vol. 76, no. 6, p. 066102, 2007.

[42] T. Murata, "Detecting communities from bipartite networks based on bipartite modularities," in *2009 International Conference on Computational Science and Engineering*, vol. 4, pp. 50–57, 2009.

[43] B. Kaminski, P. Misiorek, P. Pralat, and F. Théberge, "Modularity based community detection in hypergraphs," *J. Complex Networks*, vol. 12, no. 5, 2024.

[44] L. Huang, C.-D. Wang, and H.-Y. Chao, "Hm-modularity: A harmonic motif modularity approach for multi-layer network community detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2520–2533, 2019.

[45] Y. Liu, A. Li, A. Zeng, J. Zhou, Y. Fan, and Z. Di, "Motif-based community detection in heterogeneous multilayer networks," *Scientific Reports*, vol. 14, no. 1, p. 8769, 2024.

[46] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[47] N. Pržulj and N. Malod-Dognin, "Network analytics in the age of big data," *Science*, vol. 353, no. 6295, pp. 123–124, 2016.

[48] M. E. Newman, "Finding community structure in networks using the eigenvectors of matrices," *Physical review E*, vol. 74, no. 3, p. 036104, 2006.

[49] Y. Zhou, Q. Liu, X. Huang, J. Xu, and Y. 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.

[50] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the national academy of sciences*, vol. 104, no. 1, pp. 36–41, 2007.

[51] Y. Wu, R. Jin, J. Li, and X. Zhang, "Robust local community detection: on free rider effect and its elimination," *Proceedings of the VLDB Endowment*, vol. 8, no. 7, pp. 798–809, 2015.

[52] E. Martorana, R. Grasso, G. Micale, S. Alaimo, D. E. Shasha, R. Giugno, and A. Pulvirenti, "Motif finding algorithms: A performance comparison," in *From Computational Logic to Computational Biology*, vol. 14070, pp. 250–267, 2024.

[53] S. Yu, Y. Feng, D. Zhang, H. D. Bedru, B. Xu, and F. Xia, "Motif discovery in networks: A survey," *Computer Science Review*, vol. 37, p. 100267, 2020.

[54] A. Jazayeri and C. C. Yang, "Motif discovery algorithms in static and temporal networks: A survey," *J. Complex Networks*, vol. 8, no. 4, 2020.

[55] Y. Peng, B. Choi, and J. Xu, "Graph learning for combinatorial optimization: A survey of state-of-the-art," *Data Sci. Eng.*, vol. 6, no. 2, pp. 119–141, 2021.

[56] Z. Zhang, Y. Lu, W. Zheng, and X. Lin, "A comprehensive survey and experimental study of subgraph matching: Trends, unbiasedness, and interaction," *Proc. ACM Manag. Data*, vol. 2, no. 1, pp. 60:1–60:29, 2024.

[57] X. Wang, Q. Zhang, D. Guo, and X. Zhao, "A survey of continuous subgraph matching for dynamic graphs," *Knowl. Inf. Syst.*, vol. 65, no. 3, pp. 945–989, 2023.

[58] J. Hu, R. Cheng, K. C.-C. Chang, A. Sankar, Y. Fang, and B. Y. Lam, "Discovering maximal motif cliques in large heterogeneous information networks," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 746–757, 2019.

[59] Y. Zhou, Y. Fang, C. Ma, T. Hou, and X. Huang, "Efficient maximal motif-clique enumeration over large heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 17, no. 11, pp. 2946–2959, 2024.

[60] S. Sun, X. Sun, Y. Che, Q. Luo, and B. He, "Rapidmatch: A holistic approach to subgraph query processing," *Proceedings of the VLDB Endowment*, vol. 14, no. 2, pp. 176–188, 2020.

[61] Y. Wang, S. Ye, X. Xu, Y. Geng, Z. Zhao, X. Ke, and T. Wu, "Scalable community search with accuracy guarantee on attributed graphs," in *40th IEEE International Conference on Data Engineering, ICDE 2024*, pp. 2737–2750, 2024.

[62] K. Mehlhorn, "A faster approximation algorithm for the steiner problem in graphs," *Information Processing Letters*, vol. 27, no. 3, pp. 125–128, 1988.

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

[64] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *WWW*, pp. 2022–2032, 2019.

[65] H. Bast, F. Bäurle, B. Buchhold, and E. Haußmann, "Easy access to the freebase dataset," in *23rd International World Wide Web Conference, WWW '14, , Companion Volume*, pp. 95–98, 2014.

[66] N. Alon, D. Moshkovitz, and S. Safra, "Algorithmic construction of sets for k-restrictions," *ACM Transactions on Algorithms (TALG)*, vol. 2, no. 2, pp. 153–177, 2006.

[67] U. Feige, "A threshold of ln n for approximating set cover," *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.

Appendix

A. 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, $\mathtt{deg}_{\mathcal{B}_M}(v)$ denotes the degree of v in $\mathcal{B}_M$, $\mathcal{N}_{\mathcal{B}_M}(v_I)$ denotes 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), we have $\mathtt{vol}_{\mathcal{B}_M}(V_H(l)) = \sum_{v \in V_H(l)} \mathtt{deg}_{\mathcal{B}_M}(v) = \sum_{v \in V_H(l)} \mathtt{Mdeg}_H(v) = \mathtt{Mvol}_H(V_H(l))$ since $\mathtt{deg}_{\mathcal{B}_M}(v) = \mathtt{Mdeg}_H(v)$. □

B. The Proof of Motif-based Random Graph Model

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

For (1), for each vertex $v_I \in \mathcal{R}_{\mathcal{B}_M}$, it will rewire $|V_M(l)|$ vertices with the 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 $|V_M(\ell_H(v))| \frac{\mathtt{deg}_{\mathcal{B}_M}(v)}{\mathtt{vol}_{\mathcal{B}_M}(V_H(\ell_H(v)))}$. The process is repeated for $|\mathcal{R}_{\mathcal{B}_M}|$ times and each procedure is independent. Since $|\mathcal{R}_{\mathcal{B}_M}| = |\mathcal{I}^M_H|$, then we can get that $Exp[\mathtt{deg}_{\mathcal{B}'_M}(v)] = |\mathcal{I}^M_H||V_M(\ell_H(v))| \frac{\mathtt{deg}_{\mathcal{B}_M}(v)}{\mathtt{vol}_{\mathcal{B}_M}(V_H(\ell_H(v)))}$. By combining with Lemma 1, we can infer that $\mathtt{vol}_{\mathcal{B}_M}(V_H(l)) = \mathtt{Mvol}_H(V_H(l)) = |\mathcal{I}^M_H||V_M(l)|$ and $\mathtt{deg}_{\mathcal{B}_M}(v) = \mathtt{Mdeg}_H(v)$. Hence, $Exp[\mathtt{deg}_{\mathcal{B}'_M}(v)] = \mathtt{deg}_{\mathcal{B}_M}(v) = \mathtt{Mdeg}_H(v)$.

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

C. 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*

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

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

Proof. We estimate $Exp[|\mathcal{I}^M_{H[S]}|]$ by the expected number of motif instances in $\mathcal{B}'_M$, which all consist of vertices in S, i.e., $Exp[|\mathcal{I}^M_{H[S]}|] = Exp[|\{v_I \in \mathcal{R}_{\mathcal{B}'_M} : \forall (v_I, v) \in \mathcal{E}_{\mathcal{B}'_M}, v \in S\}|]$. 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 $\frac{\mathtt{Mvol}_H(S(l))}{\mathtt{Mvol}_H(V_H(l))}$. 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 $|\mathcal{I}^M_H|$ times. Therefore, $Exp[|\mathcal{I}^M_{H[S]}|] = |\mathcal{I}^M_H| \prod_{u \in V_M} \frac{\mathtt{Mvol}_H(S(\ell_M(u)))}{\mathtt{Mvol}_H(V_H(\ell_M(u)))}$. □

D. The Proof of Free-rider Effect and Resolution Limit

Lemma. *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 $g(S \cup S^*) \geqslant g(S)$ [51].

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 $S^+ = S \cup S^*$, we can get that $MDM(H, S^+, M) \geqslant MDM(H, S, M)$, i.e.,

$$\begin{aligned}&\frac{1}{|S^+|}(|\mathcal{I}^M_{H[S^+]}| - |\mathcal{I}^M_H| \prod_{u \in V_M} \frac{\mathtt{Mvol}_H(S^+(\ell_M(u)))}{\mathtt{Mvol}_H(V_H(\ell_M(u)))})\\ &\geqslant \frac{1}{|S|}(|\mathcal{I}^M_{H[S]}| - |\mathcal{I}^M_H| \prod_{u \in V_M} \frac{\mathtt{Mvol}_H(S(\ell_M(u)))}{\mathtt{Mvol}_H(V_H(\ell_M(u)))}).\end{aligned}$$

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

$$\begin{aligned}&\frac{1}{|\mathcal{I}^M_H|}(|\mathcal{I}^M_{H[S^+]}| - |\mathcal{I}^M_H| \prod_{u \in V_M} \frac{\mathtt{Mvol}_H(S^+(\ell_M(u)))}{\mathtt{Mvol}_H(V_H(\ell_M(u)))})\\ &\geqslant \frac{|S^+|}{|\mathcal{I}^M_H||S|}(|\mathcal{I}^M_{H[S]}| - |\mathcal{I}^M_H| \prod_{u \in V_M} \frac{\mathtt{Mvol}_H(S(\ell_M(u)))}{\mathtt{Mvol}_H(V_H(\ell_M(u)))}).\end{aligned}$$

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

$$
\begin{aligned}
&\frac{1}{|\mathcal{I}^M_H|}(|\mathcal{I}^M_{H[S^+]}| - |\mathcal{I}^M_H| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S^+(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}) \\
\geqslant\ &\frac{1}{|\mathcal{I}^M_H|}(|\mathcal{I}^M_{H[S]}| - |\mathcal{I}^M_H| \prod_{u \in V_M} \frac{\texttt{Mvol}_H(S(\ell_M(u)))}{\texttt{Mvol}_H(V_H(\ell_M(u)))}),
\end{aligned}
$$

which is the same as $GMM(H, S^+, M) \geqslant GMM(H, S, M)$. Therefore, 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], [50].

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, GMM suffers from the resolution limit as well. □

E. 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

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

where $\texttt{vol}_G(S)$ is the sum of the degree of all vertices in S over 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 $\mathcal{I}^M_H = E_H, \mathcal{I}^M_{H[S]} = E_{H[S]}$, $\texttt{Mvol}_H(S) = \texttt{vol}_H(S)$ and $\texttt{Mvol}_H(V_H) = 2|\mathcal{I}^M_H| = 2|E_H|$. Therefore,

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

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. □

F. The Proof of Theorem 3

Theorem 3. *The time complexity of Algorithm 1 is $O(\alpha_t(H, M) + |V_H|^3 \times \texttt{Mdeg}^{max}_H \times |V_M|)$, where $\alpha_t(H, M)$ is the time cost of enumerating motif instances and $\texttt{Mdeg}^{max}_H$ is the maximum motif degree of H. The space complexity of Algorithm 1 is $O(\max(\alpha_s(H, M), |\mathcal{I}^M_H| \times |V_M|))$, where $\alpha_s(H, M)$ is the space cost of enumerating motif instances.*

Proof. The time complexity of enumerating motif instances is $O(\alpha_t(H, M))$. Then, Algorithm 1 takes $O(\texttt{Mdeg}^{max}_H \times |V_M| \times |V_H|)$ time to find all vertices M-connected to Q in H. In each round, it takes $O(|V_H| \times (\texttt{Mdeg}^{max}_H \times |V_M| \times |V_H|))$ time to remove each vertex, maintain M-connectivity, and compute MDM. The process is repeated up to $|V_H|$ times. Therefore, the time complexity of Algorithm 1 can be simplified to $O(\alpha_t(H, M) + |V_H|^3 \times \texttt{Mdeg}^{max}_H \times |V_M|)$.

The space complexity of enumerating motif instances is $O(\alpha_s(H, M))$. Algorithm 1 stores the mapping between vertices and motif instances in $\mathcal{I}^M_H$. Therefore, the space complexity of Algorithm 1 is $O(\max(\alpha_s(H, M), |\mathcal{I}^M_H| \times |V_M|))$. □

G. The Proof of Lemma 2

Lemma 2. *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*

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

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

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

H. The Proof of Lemma 3

Lemma 3. *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. □

I. The Proof of Lemma 4

Lemma 4. *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: $\omega_{H_M}(u, w) - \omega_{H'_M}(u, w) \in [0, |\mathcal{I}_H^M(v)|]$.

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

J. The Proof of Theorem 4

Theorem 4. *The time complexity of Algorithm 2 is $O(\alpha_t(H, M) + |V_{H_M}| \times \sigma_{\texttt{check}} \times (\beta_{\texttt{update}} \times \texttt{Mdeg}_H^{max} \times \log \texttt{Mdeg}_H^{max} + |V_{H_M}| + |E_{H_M}|))$, where $\sigma_{\texttt{check}}$ is the number of vertex checks performed per round to determine candidate vertices and $\beta_{\texttt{update}}$ is the number of exact edge weight updates. The space complexity of Algorithm 2 is $O(\max(\alpha_s(H, M), |\mathcal{I}_H^M| \times |V_M| + |V_{H_M}| \times |R| + |E_{H_M}|))$, where $|R|$ is the size of the reuse set.*

Proof. Algorithm 2 takes $O(\alpha_t(H, M))$ time to enumerate motif instances and 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 $O(\texttt{Mdeg}_H^{max} \times |V_M| \times \log \texttt{Mdeg}_H^{max})$ time to compute equivalent vertices of the selected vertex. Next, it takes $O(\beta_{\texttt{update}} \times \texttt{Mdeg}_H^{max} \times \log \texttt{Mdeg}_H^{max})$ 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 $O(\alpha_t(H, M) + |V_{H_M}| \times \sigma_{\texttt{check}} \times (\beta_{\texttt{update}} \times \texttt{Mdeg}_H^{max} \times \log \texttt{Mdeg}_H^{max} + |V_{H_M}| + |E_{H_M}|))$.

The space complexity of enumerating motif instances is $O(\alpha_s(H, M))$. The space required to store the mapping between vertices and motif instances is $O(|\mathcal{I}_H^M| \times |V_M|)$. To construct the MW-HIN, $O(|V_{H_M}| + |E_{H_M}|)$ space is needed. Furthermore, maintaining a reuse set for each vertex requires $O(|V_{H_M}| \times |R|)$ space. Therefore, the overall space complexity of Algorithm 2 can be simplified to $O(\max(\alpha_s(H, M), |\mathcal{I}_H^M| \times |V_M| + |V_{H_M}| \times |R| + |E_{H_M}|))$. □

K. The Proof of Lemma 5

Lemma 5. *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, $\texttt{Mdist}_H(u, v) = +\infty$.*

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 6

Lemma 6. *Given a motif M, an M-connected HIN H. Let $S_0 \subseteq V_H$, and $S' = \{v \in V_H \mid \texttt{Mdist}_H(S_0, v) = \max_{u \in V_H} \texttt{Mdist}_H(S_0, u)\}$. If $H[S_0]$ is M-connected and $S_0 \subseteq V_H \setminus S'$, $H[V_H \setminus S']$ is M-connected .*

Proof. Let's merge the vertices of S_0 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_0 and other vertices in H. Since S' consists of the vertices farthest from S_0 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 $H[V_H \setminus S'] = H[S_0]$, $H[V_H \setminus S']$ is M-connected because $H[S_0]$ is M-connected.

(2) If $H[V_H \setminus S'] \neq H[S_0]$, for each $u \in V_H \setminus S'$, we have $u \in S_0$ or u has a father vertex w in $\mathcal{T}$. If $u \in S_0$, then removing each vertex $v \in S'$ will not break the M-connectivity of u because $H[S_0]$ 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 removing v from H_M only influences the edges within its one-hop-induced subgraph in H_M according to Lemma 4. To this end, $H[V_H \setminus S']$ is M-connected. □

M. The Proof of Lemma 7

Lemma 7. *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. The Proof of Theorem 5

Theorem 5. *The time complexity of Algorithm 3 is $O(\alpha_t(H,M) + |V_{H_M}| \times \log|V_{H_M}| \times \texttt{Mdeg}_H^{max} \times |V_M|)$. The space complexity of Algorithm 3 is $O(\max(\alpha_s(H,M), |\mathcal{I}_H^M| \times |V_M| + |V_{H_M}| + |E_{H_M}|))$.*

Proof. Algorithm 3 takes $O(\alpha_t(H,M))$ to collect motif instances. 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 $O(|V_{H_M}| \times \texttt{Mdeg}_H^{max} \times |V_M|)$ 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 $O(\texttt{Mdeg}_H^{max} \times |V_M| \times \log|V_{H_M}|)$ 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 $O(|V_{H_M}| \times \texttt{Mdeg}_H^{max} \times |V_M| \times \log|V_{H_M}|)$. Overall, the time complexity of Algorithm 3 can be simplified to $O(\alpha_t(H,M) + |V_{H_M}| \times \log|V_{H_M}| \times \texttt{Mdeg}_H^{max} \times |V_M|)$.

The space complexity of enumerating motif instances is $O(\alpha_s(H,M))$. The space required to store the mapping between vertices and motif instances is $O(|\mathcal{I}_H^M| \times |V_M|)$, and the space needed for MW-HIN is $O(|V_{H_M}| + |E_{H_M}|)$. Therefore, the overall space complexity of Algorithm 3 can be simplified to $O(\max(\alpha_s(H,M), |\mathcal{I}_H^M| \times |V_M| + |V_{H_M}| + |E_{H_M}|))$. □

O. Time and Space Complexities

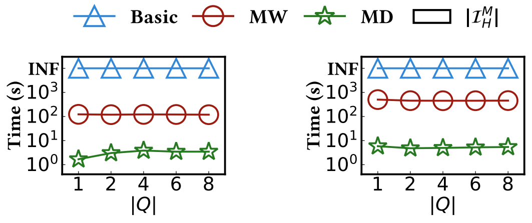
Table VII summarizes the time and space complexities of different baselines and our proposed methods in this paper. The detailed proof of time and space complexities of our proposed methods can be found in Appendix-F, Appendix-J, and Appendix-N. Here, we analyze the time and space complexities of baselines (i.e., RC [6], HM [15], MM [13], and GMM). The notations can refer to Table I.

(1) *Time and space complexities of* RC [6]. The time and space complexities of RC are $O(|V_H| \times |\mathcal{S}| + |E_H|)$ and $O(|V_H| \times |L_{\mathcal{S}}|)$, respectively [6]. Here, $\mathcal{S}$ is a set of relational constraints and $L_{\mathcal{S}}$ is a set of types in $\mathcal{S}$. Since

TABLE VII

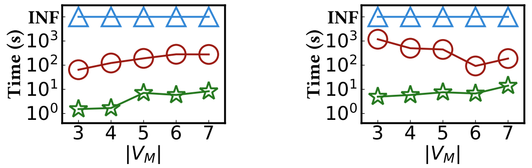
TIME AND SPACE COMPLEXITIES OF DIFFERENT METHODS

	Time	**Space**
RC [6]	$O(\|V_H\| \times \|E_M\| + \|E_H\|)$	$O(\|V_H\| \times \|L_M\|)$
HM [15]	$O(\alpha_t(H,M) + \|V_M\| \times \texttt{Mvol}_H(V_H) + \|\mathcal{I}_H^M\| \times \|V_M\|^2)$	$O(\max(\alpha_s(H,M), \texttt{Mvol}_H(V_H) + \|\mathcal{I}_H^M\| \times \|V_M\|^2))$
MM [13]	$O(\alpha_t(H,M) + \|V_{H_M}\| \times \sigma_{\texttt{check}} \times (\beta_{\texttt{update}} \times$ $\texttt{Mdeg}_H^{max} \times \log \texttt{Mdeg}_H^{max} + \|V_{H_M}\| + \|E_{H_M}\|))$	$O(\max(\alpha_s(H,M), \|\mathcal{I}_H^M\| \times \|V_M\| + \|V_{H_M}\| \times \|R\| + \|E_{H_M}\|))$
GMM	$O(\alpha_t(H,M) + \|V_{H_M}\| \times \sigma_{\texttt{check}} \times (\beta_{\texttt{update}} \times$ $\texttt{Mdeg}_H^{max} \times \log \texttt{Mdeg}_H^{max} + \|V_{H_M}\| + \|E_{H_M}\|))$	$O(\max(\alpha_s(H,M), \|\mathcal{I}_H^M\| \times \|V_M\| + \|V_{H_M}\| \times \|R\| + \|E_{H_M}\|))$
Basic	$O(\alpha_t(H,M) + \|V_H\|^3 \times \texttt{Mdeg}_H^{max} \times \|V_M\|)$	$O(\max(\alpha_s(H,M), \|\mathcal{I}_H^M\| \times \|V_M\|))$
MW	$O(\alpha_t(H,M) + \|V_{H_M}\| \times \sigma_{\texttt{check}} \times (\beta_{\texttt{update}} \times$ $\texttt{Mdeg}_H^{max} \times \log \texttt{Mdeg}_H^{max} + \|V_{H_M}\| + \|E_{H_M}\|))$	$O(\max(\alpha_s(H,M), \|\mathcal{I}_H^M\| \times \|V_M\| + \|V_{H_M}\| \times \|R\| + \|E_{H_M}\|))$
MD	$O(\alpha_t(H,M) + \|V_{H_M}\| \times \log \|V_{H_M}\| \times \texttt{Mdeg}_H^{max} \times \|V_M\|)$	$O(\max(\alpha_s(H,M), \|\mathcal{I}_H^M\| \times \|V_M\| + \|V_{H_M}\| + \|E_{H_M}\|))$



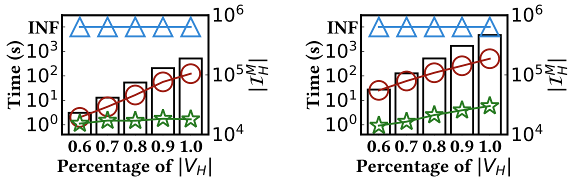
(a) CIDeRplus (b) TMDB

Fig. 11. Effect of $|Q|$



(a) CIDeRplus (b) TMDB

Fig. 12. Effect of $|V_M|$



(a) CIDeRplus (b) TMDB

Fig. 13. Effect of $|V_H|$

we split the motif into a set of relational constraints $\mathcal{S}$, we have $|E_M| = |\mathcal{S}|$ and $|L_M| = |L_{\mathcal{S}}|$. Therefore, the time and space complexities of RC are $O(|V_H| \times |E_M| + |E_H|)$ and $O(|V_H| \times |L_M|)$, respectively.

(2) *Time and space complexities of* HM [15]. Let $G_h = (V_{G_h}, E_{G_h})$ be a hypergraph. The time and space complexities of HM are $O(|\tilde{e}| \times \texttt{vol}(G_h) + \texttt{vol}_2(G_h))$ and $O(\texttt{vol}(G_h) + \texttt{vol}_2(G_h))$, respectively [15]. Here, $|\tilde{e}|$ is the average edge cardinality in G_h, $\texttt{vol}(G_h)$ is the sum of degree of vertices in G_h, and $\texttt{vol}_2(G_h) = \sum_{e \in E_{G_h}} \binom{|e|}{2}$. For each query, we enumerate the motif instances and treat them as the hyperedges to construct the hypergraph G_h. Therefore, we have $|\tilde{e}| = |V_M|$, $\texttt{vol}(G_h) = \texttt{Mvol}_H(V_H)$, and $\texttt{vol}_2(G_h) = \sum_{e \in E_{G_h}} \binom{|e|}{2} = \sum_{I \in \mathcal{I}_H^M} \binom{|V_I|}{2} = |\mathcal{I}_H^M||V_M|(|V_M| - 1)/2$. Overall, the time and space complexities of HM are $O(\alpha_t(H,M) + |V_M| \times \texttt{Mvol}_H(V_H) + |\mathcal{I}_H^M| \times |V_M|^2)$ and $O(\max(\alpha_s(H,M), \texttt{Mvol}_H(V_H) + |\mathcal{I}_H^M| \times |V_M|^2))$, respectively.

(3) *Time and space complexities of* MM [13] and GMM. Since we implement MM [13] and GMM by replacing the modularity function in the MW to solve our problem, their time and space complexity are the same as the MW.

P. The Proof of Inapproximability for MOCHI Problem

Theorem. *Unless P=NP, the MOCHI problem cannot be approximated in polynomial time within a constant factor.*

Proof. Assume that there exists a polynomial-time algorithm $\mathcal{A}$ that achieves a constant-factor approximation for the MOCHI problem. According to the proof of Theorem 2, since the DMCS problem is a special case of the MOCHI problem, applying $\mathcal{A}$ to DMCS instances yields a polynomial-time constant-factor approximation algorithm $\mathcal{A}'$ for the DMCS problem. However, [14] proves that finding a solution to the instance of the DMCS problem is the same as finding a solution to the instance of the set cover problem, which is known to be inapproximable within a factor of $(1-\epsilon)\ln|U|$ for any $\epsilon > 0$ unless P = NP [66], [67]. This contradicts our assumption that the DMCS problem admits a constant-factor approximation. Consequently, no such algorithm $\mathcal{A}$ can exist, implying the MOCHI problem cannot be approximated in polynomial time within a constant factor unless P = NP. □

Q. Additional Experiments

Effect of $|Q|$. Fig. 11 shows the effect of query vertex size on three algorithms by varying $|Q|$ on CIDeRplus and TMDB. **Effect of $|V_M|$.** Fig. 12 shows the effect of motif size on three algorithms by varying $|V_M|$ on CIDeRplus and TMDB. **Effect of $|V_H|$.** Fig. 13 shows the scalability of three algorithms by varying the fraction of $|V_H|$ on CIDeRplus and TMDB.