

Clique Identification in Signed Graphs: A Balance Theory Based Model

Renjie Sun[✉], Graduate Student Member, IEEE, Yanping Wu[✉], Xiaoyang Wang[✉], Chen Chen[✉],
Wenjie Zhang[✉], Senior Member, IEEE, and Xuemin Lin[✉], Fellow, IEEE

Abstract—Clique, as a fundamental model for graph analysis, is widely investigated in the literature. However, with the emergence of various graph data, such as signed graph, novel clique model is desired to better capture the cohesiveness within these graphs. Different from unsigned graphs, where only one type of edge exists, in signed graphs, nodes can be connected either positively or negatively (e.g., friend or enemy). In this article, we propose a novel clique model, called signed k -clique, which aims to find cohesive subgraphs in signed networks based on the classic clique model and balance theory. Given a signed graph G , an induced subgraph S is a signed k -clique if $|S| \geq k$ and S is a clique without any unbalanced triangle. Moreover, we propose and investigate two fundamental problems, i.e., maximal signed k -clique enumeration and maximum signed k -clique identification, both of which are shown to be NP-hard. For maximal signed k -clique enumeration, novel balance graph based search framework and optimization techniques are proposed to eliminate the limitations in the developed baseline. For maximum signed k -clique identification, different upper bound based techniques are developed to early terminate the search. Furthermore, the support of finding top- γ results is also discussed. Finally, comprehensive experiments on seven real-world datasets are conducted to demonstrate the efficiency and effectiveness of the proposed techniques. Compared with the baseline, the optimized algorithm can achieve up to four orders of magnitude speedup.

Index Terms—Signed graph, balance theory, signed clique, maximal clique, maximum clique.

I. INTRODUCTION

WITH the surge of data, graphs have been widely used to model the complex relationships among different entities, such as social network [1], [2], road network [3], [4], financial network [1], [5] and protein-protein interaction network [6],

Manuscript received 13 May 2022; revised 12 March 2023; accepted 29 April 2023. Date of publication 3 May 2023; date of current version 8 November 2023. This work was supported in part by ARC under Grant DP230101445, in part by ZJNSF under Grant LY21F020012, and in part by ARC under Grants DP210101393, DP200101116, and FT210100303. Recommended for acceptance by S. S. Bhowmick. (Corresponding author: Chen Chen.)

Renjie Sun is with the East China Normal University, Shanghai 200050, China (e-mail: renjie.sun@stu.ecnu.edu.cn).

Yanping Wu is with the Zhejiang Gongshang University, Hangzhou, Zhejiang 314423, China (e-mail: yanpingw.zjgsu@gmail.com).

Xiaoyang Wang and Wenjie Zhang are with the University of New South Wales, Sydney, NSW 2052, Australia (e-mail: xiaoyangw.unsw@gmail.com; zhangw@cse.unsw.edu.au).

Chen Chen is with the University of Wollongong, Wollongong, NSW 2522, Australia (e-mail: chenc@zjgsu.edu.cn).

Xuemin Lin is with the Shanghai Jiao Tong University, Shanghai 200240, China, and also with University of New South Wales, Sydney, NSW 2052, Australia (e-mail: lxue@cse.unsw.edu.au).

Digital Object Identifier 10.1109/TKDE.2023.3272636

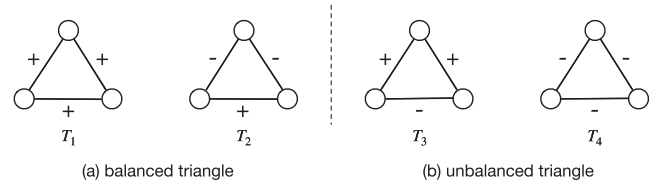


Fig. 1. Example of balanced and unbalanced triangles.

[7]. Finding cohesive subgraphs is a fundamental task in graph analysis, and different cohesive subgraph models are proposed in the literature, such as k -core [8], k -truss [9] and clique [10]. Most existing research about cohesive subgraph mining focuses on unsigned graphs, i.e., treat all connections between users as positive relationships. However, entity interactions involve both positive relationship (e.g., friend) and negative relationship (e.g., enemy). For example, in Epinions¹ online rating platform, users can express positive or negative rate towards others. Therefore, ignoring the signed information may fail to characterize the cohesiveness of subgraphs in signed networks.

For signed graph analysis, balance theory, formulated by Heider in the 1940s [11], is widely adopted and investigated in different domains, such as social psychology [12], complex systems [13] and data clustering [14]. In balance theory, many observations are based on the concept of the balanced triangle, which serves as a fundamental role. In a signed network, a triangle is balanced, if there are odd number of positive edges in the triangle [15]. The balanced triangle is defined based on the intuition that “the friend of my friend is my friend, and the enemy of my enemy is my friend”. As shown in Fig. 1, T_1 and T_2 are balanced triangles, while T_3 and T_4 are unbalanced triangles. In a real social network, a community with more balanced triangles and fewer unbalanced triangles tends to be more stable [11].

Recently, there are some works that try to identify cohesive subgraphs from signed graphs based on different models (e.g., [10], [16], [17], [18], [19], [20], [21]), and characterize the signed information utilizing balance theory or signed degree constraint. For example, in [16], authors leverage balance theory on k -core model for stable community detection. Giatsidis et al. [17] propose an s -core model in signed graphs, which requires each node in the subgraph should have a sufficient number of positive and negative neighbors. In [10], [18], [22], authors

¹www.epinions.com

set hard constraints over the number of positive and negative neighbors to better describe the clique model. However, these works either neglect the impact of unbalanced structures, including partial results, or obtain subgraphs that do not support clique model.

Intuitively, in signed graphs, a stable cohesive subgraph should be densely balanced, i.e., nodes are densely connected and the subgraph is free from unbalanced triangles. Based on this intuition, in this article, we propose a novel cohesive subgraph model for signed networks, named signed k -clique, which meets the following two criteria: (i) it is a clique where the number of nodes in it is no less than k , and (ii) it does not contain any unbalanced triangle. Moreover, we focus on two fundamental problems: *maximal signed k -clique enumeration* (i.e., enumerate all the maximal signed k -cliques) and *maximum signed k -clique identification* (i.e., find the signed k -clique with the largest size). In addition, the extension of finding the top- γ results is also discussed. The found cliques can be very important for many applications, such as discovering balanced communities or cooperative groups in signed social networks and helping to find protein complexes in signed PPI networks. For example, in United States Congress, senators usually have a tendency to support or oppose each other, which can form a signed network. There are two main political parties in United States Congress, i.e., Democratic Party and Republican Party. Members of both parties usually support the senator of its own Party and oppose each other's, which can be characterized by our proposed model. In a protein-protein interaction (PPI) network [23], certain interactions represent positive relationships, indicating activation relationships, while others may represent negative relationships, such as inhibition relationships. By leveraging the balance properties of our model, it may be feasible to avoid the interference caused by negative interactions and enhance the quality of the protein complexes.

Challenges and Contributions. The main challenges of the proposed problems lie in the following aspects. First, we show that both problems are NP-hard, implying it is non-trivial to identify the results. Second, it is challenging to support both clique and balance constraints simultaneously, since existing methods usually focus on one type of the constraints. Third, in real applications, social networks are large in size. Thus, efficient algorithms are needed to process large networks. Our principal contributions in the paper are summarized as follows.

- *Novel models:* Considering the health trend of social networks and the compactness of communities, we propose a novel signed k -clique model, which represents a densely-balanced cohesive subgraph based on clique and balance theory in signed graphs. We formally define the maximal signed k -clique enumeration and maximum signed k -clique identification problems, and prove their hardness.
- *Powerful pruning strategies:* Novel pruning methods are developed by leveraging the properties of the balanced triangle and balanced graph to safely skip unpromising nodes and edges for maximal signed k -clique enumeration. For the maximum signed k -clique problem, several novel upper bound based pruning strategies are developed in order to early terminate the search.

- *Efficient algorithms:* For the maximal signed k -clique enumeration problem, a reasonable baseline algorithm is firstly proposed based on the classic Bron-Kerbosch framework [24]. Then, we develop a more efficient balance based algorithm by considering that signed k -clique is a balanced graph. Finally, equipped with the proposed pruning strategies, we propose SKCE algorithm to enumerate all the maximal signed k -cliques. In addition, we adopt the pivoting technique and two node access orders to further speed up the computation. Furthermore, we present the MSKCE framework, which extends SKCE algorithm with the developed novel upper bound based pruning strategies, to find the maximum signed k -clique, and discuss the extension of finding top- γ results.
- *Comprehensive experimental evaluation:* To demonstrate the performance of proposed techniques, we conduct extensive experiments on seven large real-world datasets. We show that the proposed algorithms and pruning strategies can significantly accelerate the search. For example, on the largest dataset (1,632,804 nodes and 22,301,965 edges), our SKCE algorithm takes only 0.4 seconds to enumerate all the maximal signed k -cliques while the baseline consumes more than 100 seconds.

II. PRELIMINARIES

A. Problem Definition

We consider a signed network $G = (V, E)$ as an undirected graph, where V ($n = |V|$) and E ($m = |E|$) are the sets of nodes and edges in G , respectively. Each edge $(u, v) \in E$ is associated with a label $l(u, v)$ either “+” or “-”. An edge with the label “+” denotes a positive edge implying these two users are friends, while an edge with the label “-” is a negative edge denoting the hostile relationship. Given a graph G , a subgraph $S = (V(S), E(S))$ is an induced subgraph of G , if $V(S) \subseteq V$ and $E(S) = E \cap (V(S) \times V(S))$. Given a subgraph S , let $N_S(u) = \{v | (u, v) \in E(S)\}$ be the set of neighbors of u in S , $N_S^+(u) = \{v | (u, v) \in E(S) \wedge l(u, v) = "+" \}$ be the set of positive neighbors of u , and $N_S^-(u) = \{v | (u, v) \in E(S) \wedge l(u, v) = "-" \}$ be the set of negative neighbors of u . We utilize $d_S(u) = |N_S(u)|$, $d_S^+(u) = |N_S^+(u)|$, $d_S^-(u) = |N_S^-(u)|$ to denote the degree, the positive degree, and the negative degree of u in S , respectively.

Given an unweighted graph G , a triangle Δ is a cycle of length three. Suggested by balance theory, in signed networks, balanced triangle is very essential to keep the stability of community, which is defined as follows.

Definition 1 (balanced triangle). Given a signed graph G , we say a triangle is balanced, denoted by Δ^+ , if it has an odd number of positive edges.

As shown in Fig. 1, for a signed graph, there exist four types of triangles. T_1 and T_2 are balanced triangles, while T_3 and T_4 are unbalanced triangles. To model the cohesiveness of a subgraph, we employ the clique model.

Definition 2 (clique). Given a graph G , an induced subgraph S is a clique, if all pairs of nodes in S are mutually connected.

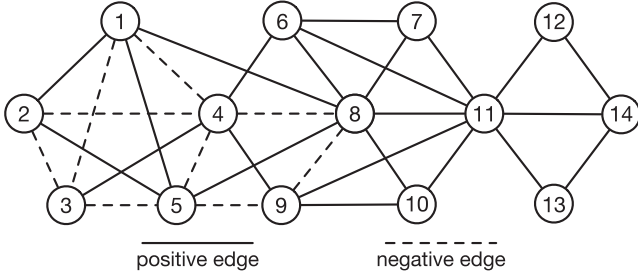


Fig. 2. Running example of signed k -clique.

Intuitively, a stable community or cohesive subgraph in signed networks should be: (i) densely connected, i.e., clique; and (ii) free from unbalanced structures, i.e., unbalanced triangles. In addition, for network analysis, an important community should have a sufficient number of entities. Motivated by this, we propose the signed k -clique model to characterize the cohesive subgraphs in signed networks.

Definition 3 (signed k -clique). Given a signed graph G and a positive integer k , an induced subgraph S is a signed k -clique if it satisfies:

- Cohesiveness constraint: S is a clique in G ;
- Size constraint: the number of nodes in S is no less than k , i.e., $|S| \geq k$;
- Balance theory constraint: S does not contain any unbalanced triangle.

Based on the definition of signed k -clique, we define the maximal signed k -clique and the maximum signed k -clique as follows.

Definition 4 (maximal signed k -clique). An induced subgraph S is a maximal signed k -clique, if S is a signed k -clique and there is no supergraph of S that is a signed k -clique.

Definition 5 (maximum signed k -clique). An induced subgraph S is the maximum signed k -clique, if S is the largest signed k -clique of G , i.e., the one with the largest size.

Example 1. Fig. 2 is a signed graph with fourteen nodes. Suppose $k = 4$. There are four maximal signed 4-cliques, where three of them have four nodes and the other one has 5 nodes, i.e., $S_1 = \{u_1, u_4, u_5, u_8\}$, $S_2 = \{u_1, u_2, u_3, u_4, u_5\}$, $S_3 = \{u_4, u_5, u_8, u_9\}$ and $S_4 = \{u_6, u_7, u_8, u_{11}\}$. S_2 is the largest one, i.e., the maximum signed 4-clique.

Problem Statement. Given a signed graph G and a positive integer k , in this article, we aim to develop efficient algorithms to solve the following two problems:

- **Maximal signed k -clique enumeration:** enumerate all the maximal signed k -cliques in G ;
- **Maximum signed k -clique identification:** find the maximum signed k -clique in G .

If we can efficiently enumerate all the maximal signed k -cliques, then we can easily find the maximum signed k -clique by returning the largest one. Therefore, in this article, we mainly focus on solving the maximal signed k -clique enumeration problem. For the maximum signed k -clique identification problem, there may exist multiple signed k -cliques with the same largest size. In this article, we only focus on returning one maximum result. When the context is clear, we use clique, signed clique and signed k -clique interchangeably.

B. Problem Properties

As shown in Theorem 1, the problem of maximal (resp. maximum) signed k -clique enumeration (resp. identification) is NP-hard. Moreover, based on Theorem 2, we further show that the problem is still NP-hard even if the input graph is a clique itself.

Theorem 1. Given a signed graph G , the problem of maximal (resp. maximum) signed k -clique enumeration (resp. identification) is NP-hard.

Proof. We prove the hardness of the studied problems by considering a special case. When there are only positive edges in the graph, the maximal (resp. maximum) signed k -clique problem is reduced to the maximal (resp. maximum) k -clique problem in unsigned graphs, which is NP-hard [25]. Therefore, the problems studied in this article are also NP-hard. \square

Theorem 2. The problem of maximal (resp. maximum) signed k -clique enumeration (resp. identification) is still NP-hard, even if G itself is a clique.

Proof. We prove the theorem by reducing the maximal (resp. maximum) independent set problem [26] to the maximal (resp. maximum) signed k -clique enumeration (resp. identification) problem. An independent set in a graph is a set of nodes, no two of which are adjacent. Given an unsigned graph G' , the maximal independent set problem is to enumerate all independent sets that cannot be contained in the other independent set. The maximum independent set problem is to find an independent set with the largest size. Then, we construct a signed graph $G = (V, E)$ from an unsigned graph $G' = (V', E')$. Let $V = V'$ and $E = V \times V$. The edges in $E \setminus E'$ are positive. The edges in $E \cap E'$ are negative. With the above construction procedure, we can enumerate (resp. identify) the maximal (resp. maximum) signed k -clique iff the maximal (resp. maximum) independent set problem is solved. Hence, the theorem is correct. \square

III. MAXIMAL SIGNED K-CLIQUE ENUMERATION

In this section, we first present a reasonable baseline approach to enumerate all the maximal signed k -cliques by extending the Bron and Kerbosch framework. Then, we leverage the balance property to further accelerate the computation. In addition, novel pruning strategies are proposed to filter the unpromising nodes and edges that are certainly not contained in any maximal signed k -clique. Finally, pivot node selection and heuristic node selection optimizations are presented to speed up the processing.

A. Baseline Approach

Naively, to deal with the problem, we can first use the existing clique enumeration algorithms to compute all the cliques by treating negative and positive edges equally. Then, we can find all the maximal signed k -cliques according to the following three steps: (i) remove nodes from the cliques to break down all unbalanced triangles inside; (ii) discard those signed cliques whose size is less than k ; (iii) for the remaining signed k -cliques, get rid of the ones that are contained by others to meet the maximal constraint. However, this method is quite laborious, and it cannot be finalized even on small graphs. The main reason is that, in the first step, to eliminate the unbalanced triangles in a clique,

Algorithm 1: BASELINE SOLUTION.

Input : G : signed graph, k : size constraint
Output : \mathcal{R} : all maximal signed k -cliques

```

1  $\mathcal{R} \leftarrow \emptyset$ ;
2 ENUMBASE( $\emptyset, V, \emptyset$ );
3 return  $\mathcal{R}$ ;

4 Procedure ENUMBASE( $S, C, X$ )
5 if  $C = \emptyset$  then
6   if  $X = \emptyset$  and  $|S| \geq k$  then
7      $\mathcal{R} \leftarrow \mathcal{R} \cup \{S\}$ ;
8   return;
9 for each  $v \in C$  do
10    $C' \leftarrow C \cap N_G(v)$ ;
11    $X' \leftarrow X \cap N_G(v)$ ;
12   for each  $u \in C'$  do
13     if  $\exists w \in S$  with  $\Delta_{uvw}$  is unbalanced then
14        $C' \leftarrow C' \setminus \{u\}$ ;
15   for each  $u \in X'$  do
16     if  $\exists w \in S$  with  $\Delta_{uvw}$  is unbalanced then
17        $X' \leftarrow X' \setminus \{u\}$ ;
18   ENUMBASE( $S \cup \{v\}, C', X'$ );
19    $C \leftarrow C \setminus \{v\}$ ;
20    $X \leftarrow X \cup \{v\}$ ;

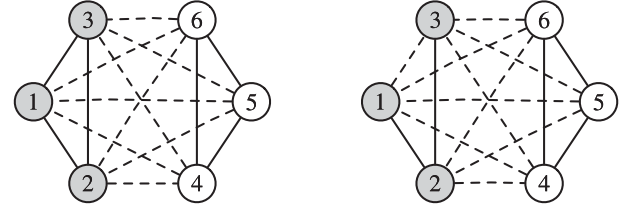
```

there will be large numbers of combinations in terms of node deletion. Eventually, it will lead to many non-maximal signed cliques and burden the subsequent steps.

To relieve the problems above, we propose a baseline approach by extending the classic Bron and Kerbosch (BK) algorithm [27]. The general idea is that, following the BK framework, when adding a new candidate node to the current result, we check whether it will form an unbalanced triangle. In addition, we will remove those non-maximal results in the process of generating signed k -cliques. Algorithm 1 presents the details of the method.

In Line 1, we initialize a set \mathcal{R} to store the results. Then, we try to enumerate all the maximal signed k -cliques by extending the BK framework from Line 2. The details of ENUMBASE procedure are presented in Lines 4–20. Note that, it admits three input parameters $\{S, C, X\}$, where S is the temporary result set, C is the set of possible candidate nodes and X is the excluded set to make sure the returned result is maximal. At the beginning of this procedure, if C is empty, which means that no node can be added into S to form a larger signed clique. Then, if X is empty and the number of nodes in S is no smaller than k , S is a maximal signed k -clique (Lines 5–7). Otherwise, we will traverse the nodes in C , add it into S and update C and X (Lines 9–20). In the loop, we first process C and X based on clique constraints in Lines 10–11. Then, according to the balanced constraint, if a node u in C can form an unbalanced triangle with the currently visited node and any node in S , we remove u from C to prevent it from forming the unbalanced triangle in the subsequent generated S (Lines 12–14). We do a similar treatment for X (Lines 15–17). In Line 18, we use the new S , C and X to enter the recursion and repeat the process until C is empty. When that recursion ends we move the currently visited node from C to X (Lines 19–20).

Discussion. Although the baseline algorithm can enumerate all the maximal signed k -cliques within reasonable time. However, in Lines 12–17 of the algorithm, it needs to visit



(a) balanced graph (b) unbalanced graph

Fig. 3. Example of balanced and unbalanced graph.

all the nodes in C , X and S in each traversal to determine whether an unbalanced triangle will be formed, which can be time-consuming. Moreover, the search space of the baseline algorithm is large, involving a lot of unpromising nodes and edges. In the subsequent sections, to scale for large graphs and speed up the computation, we first propose a novel framework by leveraging the nice property of the signed k -clique model introduced, and then develop several pruning strategies and acceleration methods.

Complexity Analysis. Given a signed graph G , n and m denote the number of nodes and edges, respectively. In the worst case, Algorithm 1 may need to perform 2^n recursions, i.e., all the subsets of V . During each recursion, its time consumption is mainly within Lines 10–17. Lines 10–11 takes $O(nd_{avg})$ time, where d_{avg} is the average degree of the nodes in V . For unbalanced triangle detection in Lines 12–17, the time cost of it is $O(n^2)$. Therefore, the time complexity of Algorithm 1 is $O(2^n n^2)$.

B. Balance Based Approach

To reduce the cost of baseline, a new framework is proposed in this section. Before introducing the detailed method, we first present some related concepts and properties.

Definition 6 (Balanced graph). Given a signed graph $G = (V, E)$. We say a subgraph S of G is balanced, iff it can be divided into two opposite subgraphs S_T and S_D , such that all edges between nodes in the same subgraph are positive and all edges between nodes in the two different subgraphs are negative.

Note that, in Definition 6, when S_T or S_D is empty, S is a balanced graph with only positive edges.

Example 2. As shown in Fig. 3, Fig. 3(a) is a balanced graph, which can be divided into two opposite subgraphs $S_T = \{u_1, u_2, u_3\}$ and $S_D = \{u_4, u_5, u_6\}$. All the edges in S_T are associated with the positive label while all the edges between different subgraphs are negative. However, Fig. 3(b) is an unbalanced graph. Recall that, in this article, we use dashed edges to denote negative edges.

According to the definition of signed k -clique, we can get the following theorem.

Theorem 3. A signed k -clique is a balanced graph.

Proof. Given a signed graph G , a signed k -clique S in G and two empty graphs S_T and S_D , we try to prove the theorem by transforming the signed k -clique into a balanced graph. First, we randomly select a node u from S and put it into S_T . Since S is a clique, the remaining nodes in S are either positive neighbors of

Algorithm 2: BALANCE BASED ALGORITHM.

Input : G : signed graph, k : size constraint
Output : \mathcal{R} : all maximal signed k -cliques

```

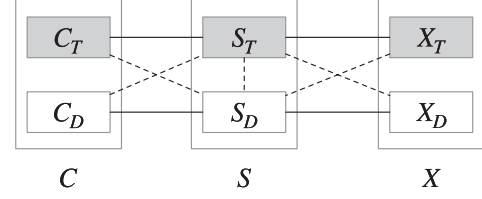
1  $Empty \leftarrow 1$ ;
2  $\mathcal{R} \leftarrow \emptyset$ ;
3  $ENUMBB(\emptyset, \emptyset, V, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ ;
4 return  $\mathcal{R}$ ;

5 Procedure  $ENUMBB(S_T, S_D, C, C_T, C_D, X, X_T, X_D)$ 
6 if  $C = \emptyset$  then
7   if  $|X| = \emptyset$  and  $|S_T| + |S_D| \geq k$  then
8      $\mathcal{R} \leftarrow \mathcal{R} \cup \{S_T \cup S_D\}$ ;
9   return;
10 if  $Empty = 1$  then
11    $Empty \leftarrow 0$ ;
12   for each  $v \in C$  do
13      $ENUMBB(\{v\}, \emptyset, N_C^+(v), N_C^-(v), N_X(v),$ 
14        $N_X^+(v), N_X^-(v))$ ;
15      $C \leftarrow C \setminus \{v\}$ ;
16      $X \leftarrow X \cup \{v\}$ ;
17 else
18   for each  $v \in C_T$  do
19      $ENUMBB(S_T \cup \{v\}, S_D, N_{C_T}^+(v) \cup N_{C_D}^-(v), N_{C_T}^-(v),$ 
20        $N_{C_D}^+(v), N_{X_T}^+(v) \cup N_{X_D}^-(v), N_{X_T}^-(v), N_{X_D}^-(v))$ ;
21      $C_T \leftarrow C_T \setminus \{v\}$ ;
22      $X_T \leftarrow X_T \cup \{v\}$ ;
23   for each  $v \in C_D$  do
24      $ENUMBB(S_T, S_D \cup \{v\}, N_{C_T}^-(v) \cup N_{C_D}^+(v), N_{C_T}^-(v),$ 
25        $N_{C_D}^-(v), N_{X_T}^-(v) \cup N_{X_D}^+(v), N_{X_T}^-(v), N_{X_D}^+(v))$ ;
26      $C_D \leftarrow C_D \setminus \{v\}$ ;
27      $X_D \leftarrow X_D \cup \{v\}$ ;

```

u or negative neighbors of u . We put all the positive neighbors of u in S_T and the remaining nodes in S_D . Then, S_T and S_D contain all the nodes in S . We can get the following three inferences: (i) All nodes in S_T are positively connected. Suppose there are two nodes v and w in S_T that are negatively connected. Since v and w are both positive neighbors of u , then they will form an unbalanced triangle with u , which violates the balance constraint of the signed k -clique model. (ii) All nodes in S_D are positive connected. Suppose the edge between two nodes v and w in S_D is negative. Since v and w are both negative neighbors of u , then the triangle formed by u, v and w is unbalanced, which violates the balance constraint of the signed k -clique model. (iii) The edges between S_T and S_D are negative. Suppose there are two nodes v in S_T and w in S_D that are positively connected. Since v is the positive neighbor of u and w is the negative neighbor of u , then they will form an unbalanced triangle with u . Therefore, a signed k -clique is a balanced graph. \square

Balance Based Algorithm. According to Theorem 3, a signed k -clique is a balanced graph. Since all nodes in a clique are mutually connected, a signed k -clique can be split into two sub-cliques, such that all the nodes in the same sub-clique are positively connected, and nodes between different sub-cliques are negatively connected. Thus, the temporary result S in Algorithm 1 can be partitioned into two subgraphs S_T and S_D with mutually exclusive sets of nodes. To enlarge S , we need to continually add a node u from C to S . If u is added into S_T , it must be positively connected to all nodes in S_T and negatively connected to all nodes in S_D , otherwise unbalanced triangles will be generated in S . Similarly, it holds if u is added to S_D .

Fig. 4. Example of S, C and X .

Therefore, we can partition the nodes in C and X based on the properties of S_T and S_D , so that nodes can be selected from the corresponding sets to be added to S_T or S_D and avoid the generation of unbalanced triangles. Then, we can save the cost of keeping checking the appearance of unbalanced triangles.

For candidate node set C , we divide it into two sets C_T and C_D , such that each node in C_T (resp. C_D) is a common positive and negative neighbor of all nodes in S_T (resp. S_D) and S_D (resp. S_T), respectively. For excluded node set X , the node sets X_T and X_D are generated in the similar manner as C . Through this way, the relationships between the nodes in S, C and X are shown in Fig. 4, where (i) they are all divided into two parts, and (ii) the solid (resp. dotted) lines indicate that all nodes between two parts are positive (resp. negative) connected. Thus, when we add a new node into S , we can safely prune the nodes from C and X if they can no longer form the same relationships with the nodes in S as in Fig. 4.

The detailed BALANCE BASED algorithm is shown in Algorithm 2. The main differences between Algorithms 2 and 1 are in Lines 10–24. At the beginning of the algorithm, S_T, S_D, C_T, C_D, X_T and X_D are empty sets, and we use a flag $Empty$ to record it (Lines 1 and 10). If $Empty$ is 1, we traverse all the nodes v in C , and initialize S_T, S_D, C_T, C_D, X_T and X_D as $\{v\}, \emptyset, N_C^+(v), N_C^-(v), N_X^+(v)$ and $N_X^-(v)$, respectively (Lines 13). Otherwise, we put the nodes in C_T and C_D into S_T and S_D , respectively, and update C_T, C_D, X_T and X_D accordingly (Lines 17–24). Through this way, we can get rid of the procedure of detecting unbalanced triangles during the enumeration.

Complexity Analysis. As mention before, the primary difference between Algorithms 2 and 1 is that S, C and X are split into two sets, respectively. This division eliminates the need for unbalanced triangle detection, i.e., Lines 12–17 of Algorithm 1. Therefore, the time complexity of Algorithm 2 is $O(2^n nd_{avg})$.

C. Pruning Strategies

In this section, to reduce the search space, we propose several effective rules to filter the unpromising nodes and edges that are certainly not contained in any maximal signed k -clique.

Definition 7 (k -core). Given a graph G and a positive integer k , an induced subgraph $S \subseteq G$ is the k -core of G , denoted by $C_k(G)$, if (i) S satisfies the minimum degree constraint, i.e., $d_S(u) \geq k$ for each node $u \in S$; and (ii) S is maximal, i.e., any supergraph of S cannot be a k -core.

Lemma 1 (k -core based pruning rule). Given a signed graph G , a subgraph $S \subseteq G$ cannot be a signed k -clique, if S does not belong to $(k-1)$ -core.

Proof. Based on the definition of signed k -clique, any node in a signed k -clique should have at least $k-1$ neighbors, which meets the definition of $(k-1)$ -core. Thus, the lemma holds. \square

Algorithm 3: COMPUTECORE(G, k).

Input : G : a graph, k : degree constraint
Output : $C_k(G)$: k -core of G
1 **while** $\exists u \in G$ with $d_G(u) < k$ **do**
2 $G \leftarrow G \setminus \{u\}$;
3 **return** G ;

Algorithm 4: NEPRUNING (G, k).

Input : G : signed graph, k : size constraint
Output : G : new graph
1 $\Delta_u^+, \Delta_{(u,v)}^+ \leftarrow$ Compute balanced triangles for each node $u \in G$
and edge $(u, v) \in G$;
2 **while** $\exists u \in G$ with $|\Delta_u^+| < \frac{(k-1)(k-2)}{2}$ **or** $\exists (u, v) \in G$ with
 $|\Delta_{(u,v)}^+| < k - 2$ **do**
3 **for each** $\Delta_{uvw} \in \Delta_u^+$ **do**
4 $\Delta_v^+ \leftarrow \Delta_v^+ \setminus \Delta_{uvw}$;
5 $\Delta_w^+ \leftarrow \Delta_w^+ \setminus \Delta_{uvw}$;
6 $\Delta_{(v,w)}^+ \leftarrow \Delta_{(v,w)}^+ \setminus \Delta_{uvw}$;
7 **for each** $\Delta_{uvw} \in \Delta_{(u,v)}^+$ **do**
8 $\Delta_u^+ \leftarrow \Delta_u^+ \setminus \Delta_{uvw}$;
9 $\Delta_v^+ \leftarrow \Delta_v^+ \setminus \Delta_{uvw}$;
10 $\Delta_w^+ \leftarrow \Delta_w^+ \setminus \Delta_{uvw}$;
11 $\Delta_{(u,w)}^+ \leftarrow \Delta_{(u,w)}^+ \setminus \Delta_{uvw}$;
12 $\Delta_{(v,w)}^+ \leftarrow \Delta_{(v,w)}^+ \setminus \Delta_{uvw}$;
13 $G \leftarrow G \setminus u$;
14 $G \leftarrow G \setminus (u, v)$;
15 **return** G ;

To compute the k -core, as shown in Algorithm 3, we iteratively remove the nodes whose degree is less than k until all the remained nodes satisfy the degree constraint. The time cost of computation k -core is $O(m)$ [28].

Given a node u , we use Δ_u^+ and $|\Delta_u^+|$ to denote the balanced triangles that contain u and the cardinality of Δ_u^+ , respectively. Similarly, we utilize $\Delta_{(u,v)}^+$ and $|\Delta_{(u,v)}^+|$ to represent the balanced triangles that contain the edge (u, v) and its corresponding cardinality, respectively. Then, based on the properties of the triangle and the signed k -clique model, we have the following two pruning rules held.

Lemma 2 (Node based pruning rule). For any node u in a signed k -clique, the number of the balanced triangles containing u must be no less than $\frac{(k-1)(k-2)}{2}$.

Proof. Given a signed k -clique S with size $|S| \geq k$, for any node $u \in S$, the number of neighbors of u (i.e., $|S| - 1$) is no less than $k - 1$ in terms of the properties of the clique. For any v in those $k - 1$ neighbors of u , the nodes in $S \setminus \{u, v\}$ can form $k - 2$ balanced triangles with u and v since $|S \setminus \{u, v\}| \geq k - 2$. Thus, there are at least $\frac{(k-1)(k-2)}{2}$ balanced triangles containing u , as each triangle will be counted twice. Thus, if node u involves in less than $\frac{(k-1)(k-2)}{2}$ balanced triangles, it cannot be contained in any signed k -clique. Therefore, the lemma holds. \square

Lemma 3 (Edge based pruning rule). For any edge e in a signed k -clique, the number of the balanced triangles containing e is no less than $k - 2$.

Proof. Given a signed k -clique S with size $|S| \geq k$, for any edge $e(u, v)$ is contained in S , each node in $S \setminus \{u, v\}$ is

connected to u and v in terms of the properties of clique. Thus, any node in $S \setminus \{u, v\}$ can form a balanced triangle with u and v . Since $|S \setminus \{u, v\}| \geq k - 2$, the number of the balanced triangles containing e in S is no less than $k - 2$. Thus, the lemma holds. \square

Armed with both of the node and edge based pruning strategies, we devise the NEPRUNING algorithm to filter the unpromising nodes and edges. The details of the procedure are shown in Algorithm 4. At first, we compute the balanced triangles for each node u and edge (u, v) in G , i.e., $\Delta_u^+, \Delta_{(u,v)}^+$ (Line 1). According to Lemma 2, if there exist a node u in G with $|\Delta_u^+|$ less than $(k - 1)(k - 2)/2$, we can safely prune u (Line 13). The removal of u destroys the balanced triangle containing it, and the information of other nodes and edges in these balanced triangles needs to be updated (Lines 3–6). For the edge (u, v) in G with $|\Delta_{(u,v)}^+|$ smaller than $k - 2$, we also remove it (Line 14) and update the balanced triangles of its associated nodes and edges (Lines 7–12), in terms of Lemma 3. This process terminates until no node or edge violates Lemmas 2 and 3.

Complexity Analysis. In Line 1, the time cost of counting all balanced triangles within G is $O(m^{3/2})$. Let $|\Delta^+|$ be the number of balanced triangles in G . Every balanced triangle in G will only be broken once. Therefore, the time complexity of Algorithm 4 is $O(m^{3/2} + |\Delta^+|)$.

Example 3. In Fig. 2, recall that nodes u_{12} , u_{13} and u_{14} have been removed after computing 3-core and all the remained nodes in G have at least 3 neighbors. Then, we can skip node u_{10} as it is contained in less than 3 balanced triangles (Lemma 2). Based on Lemma 3, we straightly eliminate edges (u_4, u_6) and (u_9, u_{11}) as $|\Delta_{(u_4, u_6)}^+| = |\Delta_{(u_9, u_{11})}^+| = 0 < 2$.

D. Pivot Node Based Optimization

In this section, we present a pivot based optimization to accelerate the computation. Recall that, in Algorithm 2, Lines 17 and 21 need to iterate over each node in C_T and C_D , and each iteration will generate a new search branch, i.e., Lines 18 and 22. However, not all branches can lead to a maximal signed k -clique. If we can skip some branches that do not generate maximal results, we can greatly reduce the computation cost. Inspired by this, to improve the performance, we extend the pivoting technique [29] for our balance based search framework to prune some search branches. For example, when selecting a node u from C_T as the pivot node, we can pass over the positive neighbors of u in the iteration of C_T and the negative neighbors of u in the iteration of C_D . Actually, the pivot node can also be selected from C_D , X_T and X_D .

Before presenting the detailed pivot node selection strategy, we prove that all the maximal signed k -cliques can still be enumerated, though some search branches are skipped. Suppose the parameters of current recursion in Algorithm 2 is $\{S_T, S_D, C, C_T, C_D, X, X_T, X_D\}$. Note that, $S_T \cup S_D$ is the temporary result set and will be contained in any signed k -clique generated in the current search branch. First, assume the pivot node u is selected from C_T . For any positive (resp. negative) neighbor $v \in C_T$ (resp. $v \in C_D$) of u , there are two types of the maximal signed k -clique containing v and $S_T \cup S_D$, i.e., one including u and the other excluding u . No doubt that all

the maximal signed k -cliques that contain u, v and $S_T \cup S_D$ can be enumerated in the search branch after adding u into $S_T \cup S_D$. For any maximal signed k -cliques that contain v and $S_T \cup S_D$ but excluding u , it must contain a node w that is not the neighbor of u , otherwise it is not maximal. These maximal signed k -cliques can be found in the search branch after adding w into $S_T \cup S_D$. Thus, there is no need to waste the search branch for v . When the pivot node u is selected from C_D , the proof is similar. Second, assume the pivot node u is selected from $X = X_T \cup X_D$. Since X is the excluded node set, all the maximal signed k -cliques that contain u and $S_T \cup S_D$ have been enumerated. The following proof is similar as above. Therefore, the correctness of the pivoting technique is demonstrated and the details of the pivot node selection method are shown as follows.

Naively, the nodes with a larger degree are more likely to be selected, as more neighbor nodes can be filtered. Considering the balance constraint in our model, we introduce the pivot node selection scheme based on the concept of the balanced graph. Note that, in Algorithm 2, the sets C and X can be divided into four sets C_T, C_D, X_T and X_D , respectively. To skip more unpromising branches, let $p1$ (resp. $p2$) be the representative node in C_T and X_T (resp. C_D and X_D), i.e., $p1 = \arg \max_{u \in C_T \cup X_T} |N_{C_T}^+(u) \cup N_{C_D}^-(u)|$ (resp. $p2 = \arg \max_{u \in C_D \cup X_D} |N_{C_D}^+(u) \cup N_{C_T}^-(u)|$). Then, we select $p1$ as pivot node, if $|N_{C_D}^+(p2) \cup N_{C_T}^-(p2)|$ smaller than $|N_{C_T}^+(p1) \cup N_{C_D}^-(p1)|$, otherwise $p2$ is pivot node.

E. Node Access Order Optimization

In the ENUMBB procedure of Algorithm 2, we need to select a node v from C_T and C_D , and add it to the temporary result set S_T and S_D in each iteration, respectively. Naively, we could randomly choose a node in each iteration. However, it may be inefficient in practice. For example, it may choose a node with large numbers of neighbors first which could lead to many search branches and affect the performance of proposed strategies. Hence, we propose two node access orders to accelerate the computation.

Degree-Based Strategy. Obviously, the smaller the degree of a node is, the less likely it is to form a signed k -clique, which implies the search branch can be early terminated. Motivated by it, we propose a heuristic node selection method, i.e., the degree-based order, to strengthen the pruning performance. Specifically, in each iteration, we pick the node from C_T and C_D with the smallest degree in C . It may cause many other nodes that are non-neighbors of v removed from C_T and C_D . Thus, the number of nodes in C_T and C_D is reduced and the ENUMBB procedure can be accelerated. In the experiments, we show that the degree-based order significantly outperforms the random node selection strategy.

Example 4. In Fig. 2, suppose the current signed k -clique is $S = \{u_1\}$, we have $C = \{u_2, \dots, u_5, u_8\}$ where $d(u_2) = 3$, $d(u_3) = 3$, $d(u_4) = 4$, $d(u_5) = 4$ and $d(u_8) = 2$. In the iteration, if we first add u_8 to C , we can directly prune two nodes, i.e., u_2, u_3 . Otherwise, no candidate can be filtered when selecting u_4, u_5 , or only one node can be pruned if we choose u_2, u_3 .

Balance-Based Strategy. Recall that, a signed k -clique is a balanced graph and can be split into two sub-cliques, where all nodes in the same sub-clique are positively connected, and nodes between different sub-cliques are negatively connected. Although the degree-based order can accelerate the ENUMBB procedure, it ignores the balance property of our signed k -clique model. Considering the motivation of degree-based order and the balance property of our model, we develop a novel balance-based order to further speed up the search. For a node $v \in C_T$, we use $ed(v) = |N_{C_T}^+(v)| + |N_{C_D}^-(v)|$ to denote the effective degree of node v , where $N_{C_T}^+(v)$ represents the positive neighbors of v in C_T and $N_{C_D}^-(v)$ represents the negative neighbors of v in C_D . Similarly, for a node $u \in C_D$, $ed(u) = |N_{C_T}^-(u)| + |N_{C_D}^+(u)|$, denoting the effective degree of node u .

Specifically, in each iteration, we select the node from C_T and C_D with the smallest effective degree. The smaller the effective degree is, the less likely it is to form a signed k -clique. Then, the search branch can be early terminated. The reason why efficient degree is better than degree is that, after adding v from C_T (resp. C_D) to S_T (resp. S_D), for a node $u \in C_T$ (resp. $u \in C_D$) which is negatively connected to node v , node u will be deleted from the candidate set C , since u, v and nodes in S_T (resp. S_D) will form unbalanced triangles. Similarly, so does it to the node $u \in C_D$ (resp. $u \in C_T$) that has a positive connection with v . Hence, selecting the node with the smaller effective degree for processing can filter more nodes from C_T and C_D compared to degree-based order. Our experimental results also verify its advantage.

Example 5. In Fig. 2, suppose the current signed k -clique $S_T = \{u_1\}$ and $S_D = \emptyset$, we have $C_T = \{u_2, u_5, u_8\}$ and $C_D = \{u_3, u_4\}$. Based on the concept of effective degree, we can obtain $ed(u_2) = 3$, $ed(u_3) = 3$, $ed(u_4) = 4$, $ed(u_5) = 4$ and $ed(u_8) = 2$. Obviously, node u_8 with the minimum effective degree is more likely to be selected in the next iteration, as it can remove more nodes similar to Example 4.

F. SKCE Algorithm

By integrating all the proposed techniques, we present the maximal signed k -clique enumeration (SKCE) algorithm, whose details are shown in Algorithm 5. In Lines 1–2, it first invokes the COMPUTECORE algorithm to derive the $(k-1)$ -core based on Lemma 1 and then utilizes node/edge based pruning rules to reduce the candidate space until all nodes and edges in the left subgraph satisfy those two pruning rules based on Lemmas 2 and 3. After filtering the search space, we use ENUMSKCE procedure to enumerate all the maximal signed k -cliques. The details of ENUMSKCE procedure are shown in Lines 7–37. We first check whether the maximal possible size of the signed clique which can be generated in this search branch is no less than k (Line 8), and if it is not, we can terminate the current branch. The pseudo-code in Lines 10–19 is the same as Lines 6–15 in Algorithm 2. According to the pivot node selection scheme in Section III-D, we choose the best node from C_T, C_D, X_T and X_D as the pivot node (Lines 22–24), and use it to get rid of some invalid search branches (Lines 25–29). In Lines 30 and 34, we use the balance-based order which is proposed in Section III-E as the

Algorithm 5: SKCE ALGORITHM.

Input : G : signed graph, k : size constraint
Output : All Maximal signed k -cliques

```

1  $G \leftarrow \text{COMPUTECORE}(G, k);$  /* Algorithm 3 */
2  $G \leftarrow \text{NEPRUNING}(G, k);$  /* Algorithm 4 */
3  $\text{Empty} \leftarrow 1;$ 
4  $\mathcal{R} \leftarrow \emptyset;$ 
5  $\text{ENUMSKCE}(\emptyset, \emptyset, V, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset);$ 
6 return  $\mathcal{R};$ 

7 Procedure  $\text{ENUMSKCE}(S_T, S_D, C, C_T, C_D, X, X_T, X_D)$ 
8 if  $|S_T| + |S_D| + |C| < k$  then
9   return;
10 if  $C = \emptyset$  then
11   if  $|X| = \emptyset$  then
12      $\mathcal{R} \leftarrow \mathcal{R} \cup \{S_T \cup S_D\};$ 
13   return;
14 if  $\text{Empty} = 1$  then
15    $\text{Empty} \leftarrow 0;$ 
16   for each  $v \in C$  do
17      $\text{ENUMSKCE}(\{v\}, \emptyset, N_C(v), N_C^+(v), N_C^-(v), N_X(v),$ 
18        $N_X^+(v), N_X^-(v));$ 
19      $C \leftarrow C \setminus \{v\};$ 
20      $X \leftarrow X \cup \{v\};$ 
21 else
22    $\text{doingT} \leftarrow \emptyset; \text{doingD} \leftarrow \emptyset;$ 
23    $p1 \leftarrow \arg \max_{u \in C_T \cup X_T} |N_{C_T}^+(u) \cup N_{C_D}^-(u)|;$ 
24    $p2 \leftarrow \arg \max_{u \in C_D \cup X_D} |N_{C_D}^+(u) \cup N_{C_T}^-(u)|;$ 
25   if  $|N_{C_T}^+(p1) \cup N_{C_D}^-(p1)| \geq |N_{C_D}^+(p2) \cup N_{C_T}^-(p2)|$  then
26      $\text{doingT} \leftarrow C_T \setminus N_{C_T}^+(p1);$ 
27      $\text{doingD} \leftarrow C_D \setminus N_{C_D}^-(p1);$ 
28   else
29      $\text{doingT} \leftarrow C_T \setminus N_{C_T}^-(p2);$ 
30      $\text{doingD} \leftarrow C_D \setminus N_{C_D}^+(p2);$ 
31   for each  $v \in \text{doingT}$  with increase order of  $\text{ed}(v)$  do
32      $\text{ENUMSKCE}(S_T \cup \{v\}, S_D, N_{C_T}^+(v) \cup N_{C_D}^-(v), N_{C_T}^+(v),$ 
33        $N_{C_D}^-(v), N_{X_T}^+(v) \cup N_{X_D}^-(v), N_{X_T}^+(v), N_{X_D}^-(v));$ 
34      $C_T \leftarrow C_T \setminus \{v\};$ 
35      $X_T \leftarrow X_T \cup \{v\};$ 
36   for each  $v \in \text{doingD}$  with increase order of  $\text{ed}(v)$  do
37      $\text{ENUMSKCE}(S_T, S_D \cup \{v\}, N_{C_T}^-(v) \cup N_{C_D}^+(v), N_{C_T}^-(v),$ 
38        $N_{C_D}^+(v), N_{X_T}^-(v) \cup N_{X_D}^+(v), N_{X_T}^-(v), N_{X_D}^+(v));$ 
39      $C_D \leftarrow C_D \setminus \{v\};$ 
40      $X_D \leftarrow X_D \cup \{v\};$ 

```

access order of nodes. Finally, we return the results and terminate the algorithm in Line 6.

Algorithm Correctness. In the SKCE algorithm, for the pruning phase, we only filter the nodes and edges that do not satisfy Lemmas 1, 2 and 3, since they will not be in any signed k -clique. In the maximal signed k -clique enumeration phase, we only extend the Algorithm 2, and eliminate the search branch which can not generate maximal signed k -clique based on the pivoting technique. The change in node access order in Lines 30 and 34 does not affect the final result. Therefore, we can obtain all the maximal signed k -cliques correctly.

Complexity Analysis. Compared with Algorithm 2, Algorithm 5 employs several optimization strategies, such as pruning strategies, pivot optimization and order optimization. In Line 1-2, the time cost of pruning strategies is $O(m^{3/2} + |\Delta^+|)$. For pivot optimization, we need to access

all nodes in C_T , C_D , X_T and X_D , and select the best one, which takes $O(nd_{avg})$ time. The order optimization cost $O(n \log n)$. Therefore, the time complexity of Algorithm 5 is $O(m^{3/2} + |\Delta^+| + 2^n(nd_{avg} + n \log n))$.

Example 6. In Fig. 2, for $k = 4$, we can remove u_{12} , u_{13} and u_{14} after computing 3-core and all the remained nodes in G have at least 3 neighbors. Then, we skip node u_{10} as it is contained in less than 3 balanced triangles (Lemma 2). Based on Lemma 3, we straightly eliminate edges (u_4, u_6) and (u_9, u_{11}) as $|\Delta_{(u_4, u_6)}^+| = |\Delta_{(u_9, u_{11})}^+| = 0 < 2$. Finally, we obtain all the maximal signed 4-cliques, i.e., $S_1 = \{u_1, u_4, u_5, u_8\}$, $S_2 = \{u_1, u_2, u_3, u_4, u_5\}$, $S_3 = \{u_4, u_5, u_8, u_9\}$ and $S_4 = \{u_6, u_7, u_8, u_{11}\}$.

IV. MAXIMUM SIGNED K-CLIQUE IDENTIFICATION

In this section, we first present the MSKCE framework developed for finding the maximum signed k -clique. Then, we present the details of deriving upper bounds used in the MSKCE framework to accelerate the computation. Finally, we extend the MSKCE framework to find the top- γ maximal signed k -cliques.

A. MSKCE Framework

Naively, to find the maximum signed k -clique, we can first enumerate all the maximal signed k -cliques and return the optimum one. However, this method will enumerate a lot of unnecessary results. To reduce the computation cost, we propose the MSKCE framework by extending the SKCE algorithm. The difference is that when enumerating the cliques, we try to estimate the upper bound of the current explored signed k -clique size. Then, we can early terminate the search of the certain branch if the upper bound is no larger than the size of the current found optimal one. Therefore, for MSKCE, it is essential to derive a tight upper bound. Specifically, we need to replace the maximal checking in Lines 10–13 of Algorithm 5 by the maximum checking procedure, i.e., the current best result will be replaced if a larger signed k -clique is generated. We can add the upper bound pruning method between Lines 9 and 10 to terminate some unnecessary search branches. To keep it concise, we omit the detailed pseudocode of MSKCE. The correctness of the algorithm is easily verifiable if we obtain the upper bound correctly. In the worst case, the time complexity of MSKCE is the same as that of SKCE.

B. Upper Bound Estimation

In this subsection, different upper bound estimation techniques are developed to enhance the performance.

Size-Based Upper Bound. In the ENUMSKCE procedure of Algorithm 5, $S_T \cup S_D$ is the current clique found and C is the candidate node set for the supplement of S . Clearly, we can use $|S| + |C|$ to be the bound of signed k -clique size in the current search branch.

Color-Based Upper Bound. However, the size-based upper bound could be loose in practice and lead to unsatisfied pruning power. To derive a tighter upper bound, we employ the concept

of graph coloring [30]. Specifically, for each search space (S, C, X) , we assign a color to each node in C such that each pair of adjacent nodes will have different colors. We will use a greedy coloring algorithm based on degree-ordering to conduct the coloring phase [31]. Obviously, all the nodes in a signed k -clique should have different colors. When searching the maximum signed k -clique, we can use $|S| + \alpha$ to serve as an upper bound of the size of the signed k -clique in the current search branch, where α is the number of colors in C . Note that, α will be much tighter than $|C|$, and lead to better performance compared with the size-based upper bound.

Recall that, the signed k -clique model has three constraints, i.e., cohesiveness constraint, size constraint and balance constraint. However, the above two upper bounds only focus on the first two constraints and neglect the balance constraint. Therefore, to obtain tighter bounds, we present the following two methods.

Size-Based Upper Bound.* In the size-based upper bound, we assume that any connected nodes in C can be added to the signed k -clique at the same time. However, it is not always the fact in practice, since they may form unbalanced triangles with the nodes in S . Motivated by this, we try to remove as few nodes from C as possible so that they will not form unbalanced triangles after they are merged into S . For a node u in C_T (resp. C_D), let $ud(u) = |N_{C_T}^-(u)| + |N_{C_D}^+(u)|$ (resp. $|N_{C_D}^-(u)| + |N_{C_T}^+(u)|$) be the unbalance degree of a node u in C . It corresponds to the number of nodes that can form unbalanced triangles with u in C , when C is added into S . We can remove the nodes from C by the decreasing order of their unbalance degree until the unbalance degree becomes 0 for all the remained nodes in C . Then, we can get a new upper bound with the updated $|S| + |C|$.

Color-Based Upper Bound.* In color-based upper bound, we assign different colors for any two nodes in C if they are adjacent, because they may exist in a signed k -clique in the search space (S, C, X) . However, if there are two nodes in C such that the triangle formed by these two nodes with each node in S is unbalanced, these two nodes in C cannot be preserved simultaneously in a signed k -clique, due to the balance constraint. Thus, we can assign them the same color to get a smaller color set. As mentioned before, all the nodes in C can be split into two disjoint sets C_T and C_D , we can further enhance the upper bound by adopting the concept of the balanced graph to reduce the color set size. Specifically, for two nodes $u, v \in C_T$ (C_D), if they are negatively connected, we can assign them a same color, because they can only form unbalanced triangle with nodes in S . Similarly, $u \in C_T$ and $v \in C_D$ can also be set with an identical color if they are positively connected. Then, after all the nodes in C are colored, we can get the color set size as the new upper bound. Note that, for three nodes $u, v \in C_T$ and $w \in C_D$, u, w are positive neighbors, and v is the negative neighbor for both u and w . If u is assigned with red color, we cannot assign both v and w red, even though v is a negative neighbor of u and w is a positive neighbor of u . This is because v and w are negatively connected, and they can exist in a signed k -clique simultaneously.

TABLE I
SUMMARY OF EVALUATED TECHNIQUES AND ALGORITHMS

Techniques	Description
CPruning	the k -core based pruning technique (Lemma 1)
NEPruning	the node and edge based pruning techniques (Lemmas 2 and 3)
RO	order strategy: randomly selects the node in Lines 30 and 34 of Algorithm 5
DO	order strategy: selects the node with the minimum degree in Lines 30 and 34 of Algorithm 5
BO	order strategy: selects the node with the minimum effective degree in Lines 30 and 34 of Algorithm 5
S/S*	the size/size*-based upper bound for finding the maximum signed k -clique
C/C*	the color/color*-based upper bound for finding the maximum signed k -clique

Algorithms	Description
Base	the baseline algorithm of enumerating all maximal signed k -cliques, i.e., Algorithm 1
Base+	the balance based algorithm proposed in Section 3.2, i.e., Algorithm 2
SKC	the baseline algorithm equipped with pruning strategies proposed in Section 3.3, i.e., conference version [32]
SKCE	equip all techniques proposed in Sections III-C, III-D and III-E with Base+, i.e., Algorithm 5
MSKCE-S/S*	equip S/S* in the MSKCE framework
MSKCE-C/C*	equip C/C* in the MSKCE framework
TSKCE-S/S*	equip S/S* in the TSKCE framework
TSKCE-C/C*	equip C/C* in the TSKCE framework

C. Find Top- γ Maximal Signed K-Cliques

For identifying the top- γ maximal signed k -cliques, the algorithm is similar and the upper bounds derived above can also be applied to accelerate the computation. Specifically, during the computation, the algorithm maintains γ current largest results. Assume the size of the current γ th result is η . For each search branch visited, we can terminate this branch if the upper bound size in the current branch is smaller than η . We denote this algorithm as TSKCE.

V. EXPERIMENTS

A. Experiment Setup

Algorithms. In this article, different optimization techniques and search methods are proposed to tackle the maximal signed k -clique enumeration problem and (top- γ) maximum signed k -clique identification problem. To evaluate the advantage of different approaches, the techniques and algorithms evaluated in the paper are summarized in Table I.

Datasets. In the experiment, we employ seven datasets. Table II shows the statistic details of datasets. Bitcoin², WordNet³, Slashdot², Epinion², Wiki⁴, Youtube² and Pokec² are all real-world networks. Bitcoin and Epinion are who-trusts-whom networks, where each edge represents the reputation, i.e., positive edge represents the trust while negative edge represents the

²<http://snap.stanford.edu>

³<https://wordnet.princeton.edu>

⁴<http://konect.uni-koblenz.de>

TABLE II
STATISTICS OF DATASETS

Dataset	$n = V $	$m = E $	$ E^+ $	$ E^- $	Parameters (k)
Bitcoin	7,605	14,125	12,973	1,152	(1, 3, <u>5</u> , 7, 9)
WordNet	19,668	165,367	142,404	22,963	(1, 3, <u>5</u> , 7, 9)
Slashdot	82,144	500,481	382,915	117,566	(5, 10, <u>15</u> , 20, 25)
Epinion	131,828	711,210	592,592	118,618	(5, 10, <u>15</u> , 20, 25)
Wiki	138,593	717,573	631,547	86,026	(5, 10, <u>15</u> , 20, 25)
Youtube	1,157,828	2,987,625	2,091,338	896,287	(1, 3, <u>5</u> , 7, 9)
Pokec	1,632,804	22,301,965	15,611,376	6,690,589	(1, 3, <u>5</u> , 7, 9)

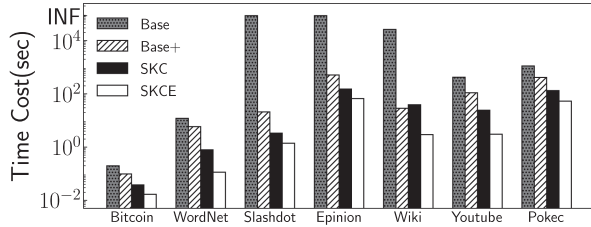


Fig. 5. Evaluation of SKCE on all the datasets.

distrust. WordNet is a network that represents the relationship of synonyms (i.e., positive edge) and antonyms (i.e., negative edge) between words. The Slashdot dataset, collected from Slashdot, contains friend and foe links between the users. Wiki contains interpreted interactions between the users of the English Wikipedia that have edited pages about politics, where each interaction is given a positive or negative value. Both Youtube and Pokec are large real social networks that are used to evaluate the scalability of algorithms. Following the procedure in [18], we generate the signed labels by randomly picking 70% of the edges as the positive edges and the remained edges as negative edges. All the datasets are publicly available.

Parameters and Workloads. Due to the unique feature and distribution of each graph, we set different parameters for datasets. The parameters of each dataset are shown in the last column of Table II. Specifically, we vary k from 1 to 9 for Bitcoin, WordNet, Youtube and Pokec with the default value 5, while for the rest datasets, we vary k from 5 to 25 and set 15 as the default value. For each setting, we run the algorithm ten times and report the average value. For those experiments that cannot finish within 24 hours, we set them as **INF**. Note that, for some experiments, we only report the result on the largest six datasets. All the programs are implemented in standard C++. All the experiments are performed on a PC with an Intel i5-9600KF 3.7 GHz CPU and 64 GB main memory.

B. Performance Evaluation for SKCE

Efficiency Evaluation. To evaluate the efficiency, we first conduct the experiments on all datasets and report the response time of Base, Base+, SKC and SKCE with the default settings. The results are shown in Fig. 5. Clearly, SKCE significantly outperforms Base on the Slashdot and Wiki datasets by four orders of magnitude faster. For example, in Wiki, SKCE needs 2.95063 seconds to enumerate all maximal signed 15-cliques, while Base takes 26368.7 seconds. For Slashdot and Epinion datasets, Base

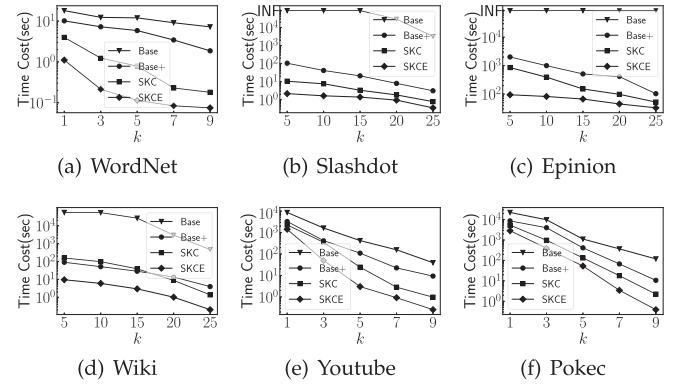


Fig. 6. Efficiency evaluation of SKCE by varying k .

cannot even finish in a reasonable time, i.e., 24 hours. We can find that the response time of SKC is smaller than Base and Base+. Moreover, SKCE consistently outperforms SKC, which verifies the effectiveness of newly proposed techniques in the extension. This is because SKCE takes into account the balance property of the model, i.e., Theorem 3, and applies the pivoting method and optimized access order for the candidate set. Besides, Base+ is much faster than Base, which further validates the novelty of the search framework. Based on the experimental results, it can be inferred that SKCE exhibits better performance on dense graphs. This is because, for dense graphs, a significant amount of time is dedicated to identifying the formation of unbalanced triangles during the search process. SKCE, on the other hand, circumvents this operation, resulting in improved performance. Furthermore, SKCE is insensitive to the distribution of positive and negative edges within the dataset.

In Fig. 6, we further report the response time of different algorithms by varying k . As shown, when k increases, the response time decreases for all algorithms. This is because the number of maximal k -cliques becomes smaller for larger k , i.e., smaller search space. Additionally, the gap between SKCE and Base widens as k increases, primarily because the search process retains denser subgraphs.

Effectiveness Evaluation. To evaluate the effectiveness of the proposed model, we report the number of maximal cliques found based on the traditional k -clique model and the proposed signed k -clique model. For the traditional k -clique model, we ignore the labels on the edges and enumerate the k -cliques. In Fig. 7, we present the results conducted on all datasets by varying k . As observed, the number of normal k -cliques is larger than that of our model, which indicates that there are many normal k -cliques containing unbalanced triangles in the signed networks. For both models, the number of cliques found decreases when k increases, since fewer cliques will survive for larger k .

Scalability Testing. To test the scalability of proposed algorithms, we generate five subgraphs by randomly sampling 20–100% of the nodes from the Epinion, Wiki and Youtube datasets, and report the response time on these subgraphs. The results are shown in Fig. 8. By varying $|V|$, the response time gradually increases, as more search space is required to explore. In addition, we find that all the algorithms show near-linear

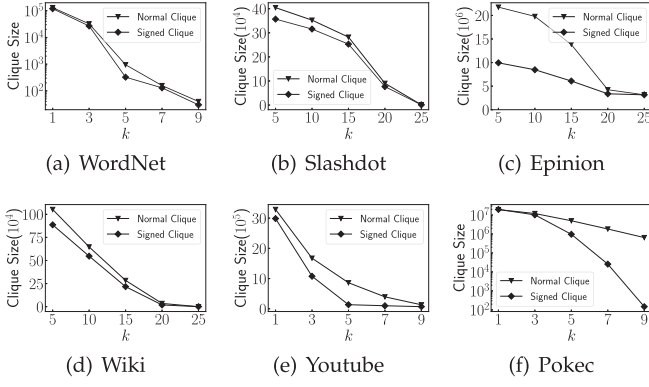
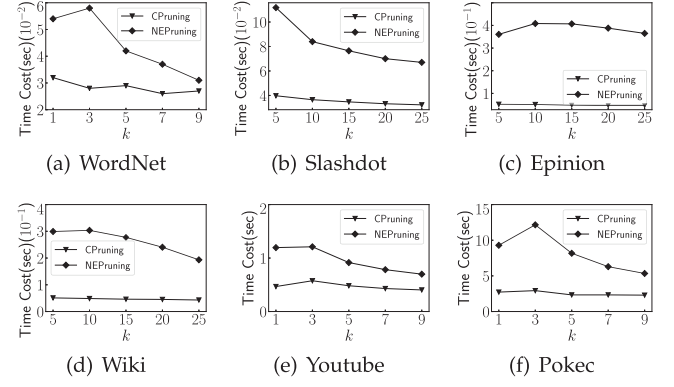
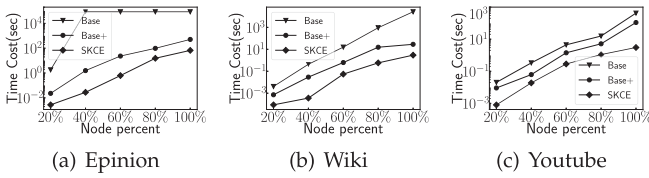
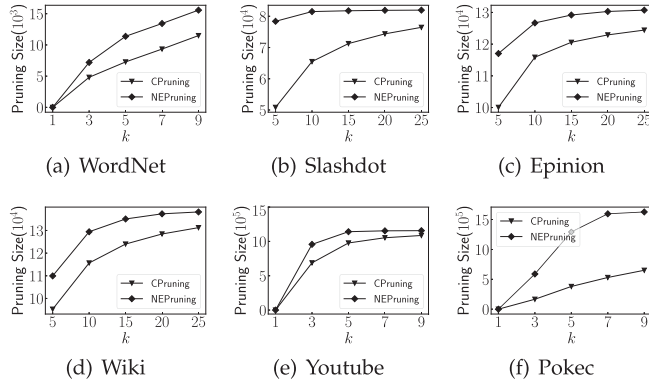
Fig. 7. Effectiveness evaluation of model by varying k .Fig. 10. Time cost of pruning rules by varying k .

Fig. 8. Scalability evaluation of SKCE.

Fig. 9. Evaluation of pruning rules by varying k .

scalability in the enumeration task, which verifies the scalability of proposed techniques.

Evaluation of Pruning Rule. In Fig. 9, we report the number of nodes pruned by using CPruning (i.e., Lemma 1) and NEPruning (i.e., Lemmas 2 and 3) on all datasets. For the edge based pruning rule, we count the number of nodes that become isolated after the removal of edges. As we can observe, though CPruning can greatly reduce the search space, more nodes can be further filtered by NEPruning. This is because CPruning only considers the size constraint, while NEPruning takes the balance constraint into account as well. For most cases, when k increases, the gap between the number of nodes pruned by the two algorithms becomes smaller. This is because, the majority of nodes do not have sufficient neighbors, i.e., $k - 1$ neighbors, when k is large.

In Fig. 10, we report the response time of CPruning and NEPruning on all the datasets. NEPruning is slower than CPruning, since NEPruning needs to take into account the

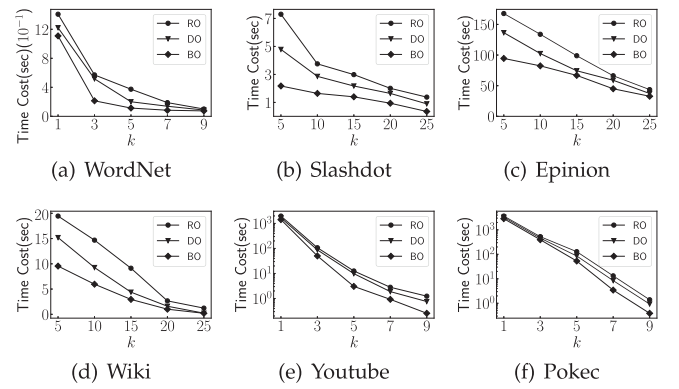


Fig. 11. Evaluation of node access order for SKCE.

triangle information and it is more complicated than CPruning. In addition, removing nodes will lead to some edges violating the edge based pruning rule and vice versa. While, the cost of both methods can be negligible compared to that of enumerating all the maximal signed k -cliques, and can significantly reduce the search space.

Evaluation of Node Access Order Optimization. To estimate the performance of the node selection methods in SKCE, we report the response time of SKCE equipped with different order optimization techniques, i.e., RO, DO and BO, on all the datasets. The results are shown in Fig. 11. As shown, RO performs worst, since it selects the node in random order without any basis. As we can see, BO constantly outperforms the others. This is because it not only resorts to the degree-based order, but also takes the balance property of our model into account. Moreover, the response time decreases as k increases, since the number of maximal signed k -cliques becomes smaller when k increases.

C. Performance Evaluation of MSKCE

To evaluate the performance of the MSKCE framework, we only report the response time by using different upper bounds, since they always return the same clique, i.e., the maximum one. The results are shown in Fig. 12. Clearly, MSKCE-C* significantly outperforms MSKCE-S, which verifies the advantage of

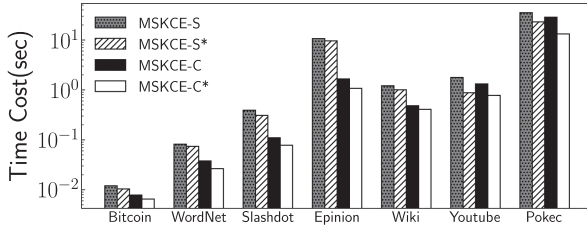
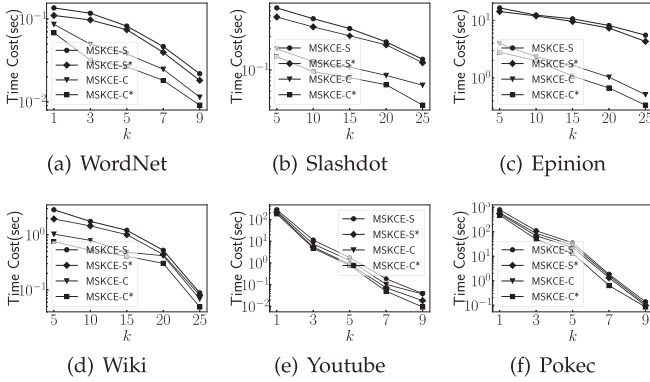


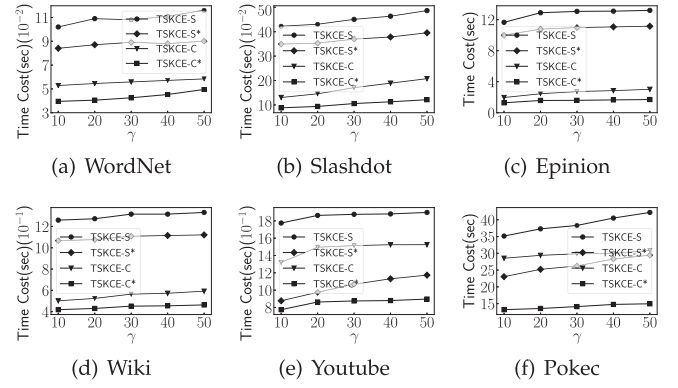
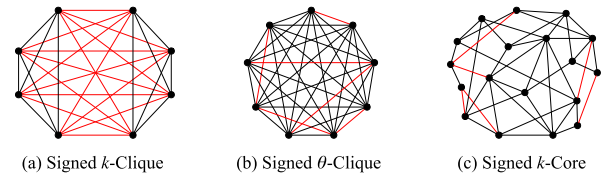
Fig. 12. Efficiency evaluation of MSKCE on all the datasets.

Fig. 13. Efficiency evaluation of MSKCE by varying k .

optimized upper bound. In particular, for Epinion dataset, the gap is up to an order of magnitude. Fig. 13 further shows the response time of MSKCE-S, MSKCE-S*, MSKCE-C and MSKCE-C* by varying k . As observed, the response time decreases when k increases, because we can prune more cliques that violate the size constraint. MSKCE-C* always outperforms the others on all datasets due to the tighter upper bound derived. For example, when $k = 15$, MSKCE-C* only takes 0.07795 seconds to find the maximum signed k -clique in Slashdot, while MSKCE-S needs 0.3925 seconds and MSKCE-S* costs 0.309164 seconds. We can also observe that the gap between the four methods is not particularly obvious in Pokec and Youtube datasets. This is due to the fact that these two datasets have many nodes and edges filtered after CPruning and NEPruning procedures. Thus, the remained graph is tight and leads to a close upper bound.

D. Performance Evaluation of TSKCE

As discussed, we can extend the maximum signed k -clique framework MSKCE to derive the top- γ maximal signed k -cliques. Note that, the upper bound optimization only influences the efficiency of the algorithms, i.e., they always return the same top- γ results. Therefore, we only report the response time here. By varying γ , the results are shown in Fig. 14. TSKCE-C* always outperforms the others on all datasets due to the tighter upper bound. As observed, the response time slightly changes when varying γ . For example, in Wiki, TSKCE-C* takes 4.18636 seconds to find the top-10 results, and it only takes 4.63065 seconds when γ becomes 50.

Fig. 14. Efficiency evaluation of TSKCE by varying γ .Fig. 15. Case study on Slashdot with $k = \theta = 5$.

E. Case Study

To further evaluate the effectiveness of the proposed model, we conduct case studies over two datasets.

Case Study on Slashdot. We first conduct a case study on the Slashdot dataset, where each node represents a user, and positive (resp. negative) edges denote support (resp. opposition) relationships. We compare our model with two recent cohesive subgraph models for signed graph analysis, namely signed θ -clique [10] and signed k -core [16]. Fig. 15 shows the visualized results returned, where red (resp. black) edges represent negative (resp. positive) relationships. In a signed θ -clique, each node in the subgraph has θ more positive neighbors than negative ones. However, this model suffers from erratic divisions within the group, making it difficult to identify groups of users who support each other. On the other hand, a signed k -core is a subgraph without any unbalanced triangle and each node inside has more than k positive neighbors. Only partial results are presented in Fig. 15(c), since the excessive number of nodes in the signed k -core result. However, the connections within the group may be inadequate, i.e., less cohesive. As shown, the signed k -clique model can divide users into two subgroups based on their support/opposition relationships, with intra-group users supporting each other and inter-group users dissenting from each other. This division allows us to extract tightly connected opinion leaders and support further analysis. Thus, the signed k -clique model can preserve unique features and is more stable in the signed network than others.

Case Study on a Signed PPI Network. Second, we utilize Fly-Sign [23], a widely-used signed protein-protein interaction (PPI) network, to verify the effectiveness of the signed k -clique model. The network comprises 3352 proteins and 6094 PPIs, with 4111 interactions positive and the remaining interactions negative.

In [23], the authors demonstrate that unbalanced triangles in signed PPI networks are significantly dynamic and unstable. These unbalanced triangles can act as negative feedback loops or incoherent feed-forward loops, ultimately influencing system controllability. In the case study, we extract the top-50 signed k -clique and signed θ -clique from FlySign with $k = \theta = 5$. The result reveals that out of the 50 signed θ -clique, 37 of them contain unbalanced triangles. As the signed k -clique does not produce any unbalanced triangles, it implies that the protein complexes discovered via our model could possess enhanced stability and controllability.

VI. RELATED WORK

In this section, we introduce the related work from the following two aspects.

Clique. In graph analysis, clique serves as an important model in various applications [24], [33], [34], [35], [36]. There are two fundamental problems for clique computation, i.e., maximal clique enumeration and maximum clique identification. For enumerating all the maximal clique, most algorithms are based on the classic backtracking method, i.e., Bron-Kerbosch algorithm [24], which requires only polynomial storage space and escapes recomputing the same clique. Eppstein et al. [27] improve the classic Bron-Kerbosch algorithm by using a degeneracy ordering. The pivoting strategy [29], [37], [38] has been developed to reduce the redundant search branches in backtracking to accelerate the computation of the Bron-Kerbosch algorithm. For identifying the maximum clique, although the maximal clique with the largest size is the maximum clique, the lack of pruning techniques makes these algorithms inefficient and unsuitable for maximum clique computation. Thus, Carraghan et al. [39] propose the basic branch and bound algorithm to compute the maximum clique by pruning those branches whose further expansion fails to find a larger clique than the current optimal one. In [40], [41], Tomita et al. use graph coloring to obtain a tighter upper bound on the size of the maximum clique for each search branch. Furthermore, there are some works to study the variant of clique on the special graph data [42], [43], [44], [45]. For example, Zhang et al. [42] develop efficient algorithms to enumerate all the maximal spatial cliques with geo-location and distance constraint in spatial graphs. In [43], Li et al. study the maximal (k, τ) -clique search in uncertain graphs. However, the above developed methods cannot solve our maximal (resp. maximum) signed k -clique enumeration (resp. identification) problem, since our model involves signed information and is free from unbalanced triangle.

Signed Graph Analysis. In signed graph analysis, balance theory is a fundamental tool, which is originally proposed by Heider [11] and generalized by Cartwright [15], implying “the friend of my friend is my friend” and “the enemy of my enemy is my friend”. Balance theory has been widely used to investigate the properties of signed networks [16], [46]. In [47], Leskovec et al. use balance theory for positive and negative links prediction. In [48], Bonchi et al. study the problem of discovering polarized communities in signed networks. In particular, any triangle in those polarized communities is balanced triangle. However, it

does not consider the cohesiveness of the community. In recent years, mining signed networks has attracted a lot of attention. Yang et al. [49] propose a framework based on agent-based random walk model to extract communities in signed networks. In [50], Cadena et al. extend the concept of subgraph density to conduct event detection on signed graphs. Giatsidis et al. [17] extend the k -core model for the signed graphs. Hao et al. [51] introduce a k -balanced trusted clique model. It requires the clique size equals k and all the relationships must be positive. While, the model is too strict for real applications and quite different from ours. There are also some recent studies focusing on clique models (e.g., [10], [18], [19], [52]) through different constraints. However, due to the different nature, the proposed techniques cannot be extended to support the problems studied in this article.

VII. CONCLUSION

In this article, we propose and investigate the maximal signed k -clique enumeration and the maximum signed k -clique identification problems based on balance theory in signed graphs. We formalize the problems and show both of them are NP-hard. Novel pruning techniques are proposed to filter the unpromising nodes and edges. A reasonable baseline is first proposed to enumerate all the maximal signed k -clique. Then, by leveraging the balance property of our model, we develop a more efficient approach, i.e., balance based algorithm. Finally, armed with pruning techniques and several optimization strategies, advanced algorithm, i.e., SKCE, is developed to accelerate the enumeration of cliques. In addition, four bounds, size-based, size*-based, color-based and color*-based upper bounds, are developed to further enhance the performance of identifying the maximum signed k -clique. We conduct extensive experiments on seven real-world signed networks to evaluate the effectiveness and efficiency of the proposed techniques.

REFERENCES

- [1] D. Cheng, C. Chen, X. Wang, and S. Xiang, “Efficient top-k vulnerable nodes detection in uncertain graphs,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1460–1472, Feb. 2023.
- [2] C. Chen, Q. Zhu, R. Sun, X. Wang, and Y. Wu, “Edge manipulation approaches for K-core minimization: Metrics and analytics,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 390–403, Jan. 2023.
- [3] E. Kanoulas, Y. Du, T. Xia, and D. Zhang, “Finding fastest paths on a road network with speed patterns,” in *Proc. IEEE Int. Conf. Data Eng.*, 2006, pp. 10–10.
- [4] W. Luo, H. Tan, L. Chen, and L. M. Ni, “Finding time period-based most frequent path in big trajectory data,” in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2013, pp. 713–724.
- [5] D. Cheng, X. Wang, Y. Zhang, and L. Zhang, “Risk guarantee prediction in networked-loans,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2020, pp. 4483–4489.
- [6] U. Stelzl et al., “A human protein-protein interaction network: A resource for annotating the proteome,” *Cell*, vol. 122, pp. 957–968, 2005.
- [7] G. Kar, A. Gursoy, and O. Keskin, “Human cancer protein-protein interaction network: A structural perspective,” *PLoS Comput. Biol.*, vol. 5, 2009, Art. no. e1000601.
- [8] S. B. Seidman, “Network structure and minimum degree,” *Social Netw.*, vol. 3, pp. 269–287, 1983.
- [9] J. Cohen, “Trusses: Cohesive subgraphs for social network analysis,” Nat. Secur. Agency, Tech. Rep. 16, 2008.

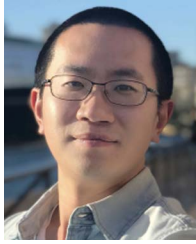
- [10] C. Chen, Y. Wu, R. Sun, and X. Wang, "Maximum signed k -clique identification in large signed graphs," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 1791–1802, Feb. 2023.
- [11] F. Heider, "Attitudes and cognitive organization," *J. Psychol.*, vol. 21, pp. 107–112, 1946.
- [12] T. Antal, P. L. Krapivsky, and S. Redner, "Social balance on networks: The dynamics of friendship and enmity," *Physica*, vol. 224, pp. 130–136, 2006.
- [13] C. J. Puccia and R. Levins, *Qualitative Modeling of Complex Systems: An Introduction to Loop Analysis and Time Averaging*. Cambridge, MA, USA: Harvard Univ. Press, 1985.
- [14] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak, "Spectral analysis of signed graphs for clustering, prediction and visualization," in *Proc. Int. Conf. Data Minings*, 2010, pp. 559–570.
- [15] D. Cartwright and F. Harary, "Structural balance: A generalization of heider's theory," *Psychol. Rev.*, vol. 63, 1956, Art. no. 277.
- [16] R. Sun, C. Chen, X. Wang, Y. Zhang, and X. Wang, "Stable community detection in signed social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 5051–5055, Oct. 2022.
- [17] C. Giatsidis, B. Cautis, S. Maniu, D. M. Thilikos, and M. Vazirgiannis, "Quantifying trust dynamics in signed graphs, the s-cores approach," in *Proc. Int. Conf. Data Minings*, 2014, pp. 668–676.
- [18] R.-H. Li et al., "Efficient signed clique search in signed networks," in *Proc. IEEE 34th Int. Conf. Data Eng.*, 2018, pp. 245–256.
- [19] Z. Chen, L. Yuan, X. Lin, L. Qin, and J. Yang, "Efficient maximal balanced clique enumeration in signed networks," in *Proc. Int. World Wide Web Conf.*, 2020, pp. 339–349.
- [20] M. Gao, E.-P. Lim, D. Lo, and P. K. Prasetyo, "On detecting maximal quasi antagonistic communities in signed graphs," *Data Mining Knowl. Discov.*, vol. 30, pp. 99–146, 2016.
- [21] R. Sun, Y. Wu, C. Chen, X. Wang, W. Zhang, and X. Lin, "Maximal balanced signed biclique enumeration in signed bipartite graphs," in *Proc. IEEE 38th Int. Conf. Data Eng.*, 2022, pp. 1887–1899.
- [22] R. Sun, C. Chen, X. Wang, W. Zhang, Y. Zhang, and X. Lin, "Efficient maximum signed biclique identification," in *Proc. IEEE Int. Conf. Data Eng.*, 2023, pp. 1307–1319.
- [23] A. Vinayagam et al., "Integrating protein-protein interaction networks with phenotypes reveals signs of interactions," *Nature Methods*, vol. 11, pp. 94–99, 2014.
- [24] L. Yuan, L. Qin, X. Lin, L. Chang, and W. Zhang, "Diversified top- k clique search," *VLDB J.*, vol. 25, pp. 171–196, 2016.
- [25] D. S. Johnson and M. R. Garey, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979.
- [26] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou, "On generating all maximal independent sets," *Inf. Process. Lett.*, vol. 27, pp. 119–123, 1988.
- [27] D. Eppstein, M. Löffler, and D. Strash, "Listing all maximal cliques in sparse graphs in near-optimal time," in *Proc. 21st Int. Symp. Algorithms Comput.*, 2010, pp. 403–414.
- [28] F. Zhang, W. Zhang, Y. Zhang, L. Qin, and X. Lin, "OLAK: An efficient algorithm to prevent unraveling in social networks," *Proc. VLDB Endowment*, vol. 10, pp. 649–660, 2017.
- [29] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theor. Comput. Sci.*, vol. 363, pp. 28–42, 2006.
- [30] M. R. Garey and D. S. Johnson, "The complexity of near-optimal graph coloring," *J. ACM*, vol. 23, pp. 43–49, 1976.
- [31] L. Yuan, L. Qin, X. Lin, L. Chang, and W. Zhang, "Effective and efficient dynamic graph coloring," *Proc. VLDB Endowment*, vol. 11, pp. 338–351, 2017.
- [32] R. Sun, Q. Zhu, C. Chen, X. Wang, Y. Zhang, and X. Wang, "Discovering cliques in signed networks based on balance theory," in *Proc. 25th Int. Conf. Database Syst. Adv. Appl.*, 2020, pp. 666–674.
- [33] J. Cheng, L. Zhu, Y. Ke, and S. Chu, "Fast algorithms for maximal clique enumeration with limited memory," in *Proc. Int. Conf. Knowl. Discov. Data Mining*, 2012, pp. 1240–1248.
- [34] L. Chang, "Efficient maximum clique computation over large sparse graphs," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 529–538.
- [35] J. Cheng, Y. Ke, A. W.-C. Fu, J. X. Yu, and L. Zhu, "Finding maximal cliques in massive networks," *ACM Trans. Database Syst.*, vol. 36, pp. 1–34, 2011.
- [36] Y. Gong et al., "Exact- k recommendation via maximal clique optimization," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 617–626.
- [37] I. Koch, "Enumerating all connected maximal common subgraphs in two graphs," *Theor. Comput. Sci.*, vol. 250, pp. 1–30, 2001.
- [38] D. Eppstein, M. Löffler, and D. Strash, "Listing all maximal cliques in large sparse real-world graphs," *J. Exp. Algorithmics*, vol. 18, pp. 3.1–3.21, 2013.
- [39] R. Carraghan and P. M. Pardalos, "An exact algorithm for the maximum clique problem," *Operations Res. Lett.*, vol. 9, pp. 375–382, 1990.
- [40] E. Tomita and T. Kameda, "An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments," *J. Glob. Optim.*, vol. 37, pp. 95–111, 2007.
- [41] E. Tomita, Y. Sutani, T. Higashi, S. Takahashi, and M. Wakatsuki, "A simple and faster branch-and-bound algorithm for finding a maximum clique," in *Proc. 4th Int. Workshop WALCOM Algorithms Comput.*, 2010, pp. 191–203.
- [42] C. Zhang, Y. Zhang, W. Zhang, L. Qin, and J. Yang, "Efficient maximal spatial clique enumeration," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 878–889.
- [43] R.-H. Li, Q. Dai, G. Wang, Z. Ming, L. Qin, and J. X. Yu, "Improved algorithms for maximal clique search in uncertain networks," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 1178–1189.
- [44] H. Qin, R.-H. Li, G. Wang, L. Qin, Y. Cheng, and Y. Yuan, "Mining periodic cliques in temporal networks," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 1130–1141.
- [45] H. Jiang, C.-M. Li, and F. Manyá, "An exact algorithm for the maximum weight clique problem in large graphs," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2017, pp. 830–838.
- [46] Y. Wu, R. Sun, C. Chen, X. Wang, and Q. Zhu, "Maximum signed (k, r) -truss identification in signed networks," in *Proc. Conf. Inf. Knowl. Manage.*, 2020, pp. 3337–3340.
- [47] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Predicting positive and negative links in online social networks," in *Proc. Int. World Wide Web Conf.*, 2010, pp. 641–650.
- [48] F. Bonchi, E. Galimberti, A. Gionis, B. Ordozgoiti, and G. Ruffo, "Discovering polarized communities in signed networks," in *Proc. Conf. Inf. Knowl. Manage.*, 2019, pp. 961–970.
- [49] B. Yang, W. Cheung, and J. Liu, "Community mining from signed social networks," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 10, pp. 1333–1348, Oct. 2007.
- [50] J. Cadena, A. K. Vullikanti, and C. C. Aggarwal, "On dense subgraphs in signed network streams," in *Proc. IEEE 16th Int. Conf. Data Minings*, 2016, pp. 51–60.
- [51] F. Hao, S. S. Yau, G. Min, and L. T. Yang, "Detecting k -balanced trusted cliques in signed social networks," *IEEE Internet Comput.*, vol. 18, no. 2, pp. 24–31, Mar./Apr. 2014.
- [52] R.-H. Li et al., "Signed clique search in signed networks: Concepts and algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 710–727, Feb. 2021.



Renjie Sun (Graduate Student Member, IEEE) received the BSc degree in computer science and technology from Henan University in 2017, and the MSc degree in computer technology from Zhejiang Gongshang University in 2021. He is currently working toward the PhD degree with East China Normal University, China. His current research interests include graph data mining, signed data analysis, and attribute graph processing.



Yanping Wu received the BSc degree in computer science and technology from Zhejiang Gongshang University in 2020. She is currently working toward the postgraduation degree with Zhejiang Gongshang University, China. Her current research interests include massive network analysis and distributed data processing.



Xiaoyang Wang received the BSc and MSc degrees in computer science from Northeastern University, China in 2010 and 2012, respectively, and the PhD degree from the University of New South Wales, Australia in 2016. He is a lecture with the University of New South Wales, Australia and the University of Technology Sydney, Australia. His research interests include query processing on massive graph, spatial data and data stream.



Wenjie Zhang (Senior Member, IEEE) received the PhD degree in computer science and engineering from the University of New South Wales, in 2010. She is currently a professor and ARC future fellow with the School of Computer Science and Engineering, the University of New South Wales, Australia. Since 2008, she has published more than 100 papers in SIGMOD, VLDB, ICDE, *ACM Transactions on Database Systems*, *IEEE Transactions on Knowledge and Data Engineering*, and the *VLDB Journal*.



Chen Chen received the BSc degree from the Dalian University of Technology, the MSc degree from Northeastern University and the PhD degree from the University of New South Wales, Australia. She is a lecturer with the University of Wollongong, Australia. She was a postdoctoral associate with East China Normal University, China. Her research interests include graph analysis and machine learning.



Xuemin Lin (Fellow, IEEE) received the BSc degree in applied math from Fudan University, in 1984, and the PhD degree in computer science from the University of Queensland, in 1992. He is a scientia professor within the School of Computer Science and Engineering, the University of New South Wales. His current research interests include approximate query processing, spatial data analysis, and graph visualization.