Detecting Critical Collapsed Nodes in Social Networks: A Cognitive Model of Resilience under Spatial Constraints

Wei Li, Chenghao Li, Yidan Chen, Ziyang Zhang, Weihe Yan, Xiang Li*
({wei.li,chli,chenyidan,ziyang.zhang,yanweihe,lxlx}@hrbeu.edu.cn)
College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China
Moustafa Youssef (moustafa-youssef@aucegypt.edu)
American University in Cairo, Egypt

Abstract

The resilience of social networks hinges on identifying users whose departure causes cascading collapse, influenced by both topology and social cognition, such as spatial relationship constraints. Existing studies often overlook how cognitive and behavioral factors shape network fragility. This paper introduces a cognitive-computational framework to detect critical collapsed nodes under spatial constraints, using the (k,σ)-*core model* to integrate social cohesion (*k-core*) and spatial thresholds (σ). We propose a pruning algorithm leveraging spatial locality for efficient querying of collapsed nodes and formalize the problem of finding optimal collapsed nodes as an NP-hard task. Our greedy heuristic prioritizes nodes with the most significant cascading impact, similar to human strategies in crises. Experiments on eight real-world networks show our model outperforms topology-only baselines in predicting collapse patterns, especially in spatially-embedded communities. Our findings highlight how spatial constraints and social cohesion amplify systemic fragility, providing insights for designing cognitively-aligned interventions to boost network resilience, bridging computational analysis with cognitive science.

Keywords: Collapsed nodes; social network resilience; spatial cognition; cognitive-computational modeling

Introduction

Social networks are not merely topological structures but dynamic systems shaped by human cognition and behavior. The resilience of these networks—their ability to withstand node failures and recover functionality—is deeply tied to how individuals perceive spatial constraints (*e.g.*, geographic proximity or social distance thresholds) and adapt collectively to disruptions. A critical challenge lies in identifying *collapsed nodes* (Tao et al., 2024; Y. Zhang, Lin, Yuan, & Jin, 2022; Zhou, Lv, Wang, Zhang, & Xuan, 2022), whose removal triggers cascading failures, threatening the network's ability to sustain cognitive processes such as information sharing and collective decision-making (Cai et al., 2022; He, Wang, Zhang, Lin, & Zhang, 2023; Yang, Li, Zhang, Luo, & Li, 2022). Thus, it is critical to identify the members who cause the community collapsed and protect them to improve the stability of the network structure. For example, Friendster, a popular social network with over 115 million users, was forced to suspend operations due to a mass exodus of users and a decline in user engagement.

Existing computational approaches primarily focus on graph-theoretic metrics (*e.g.*, centrality (Bugedo, Riveros, & Salas, 2024), *k-core* decomposition) to detect vulnerable nodes. Bhawalkar et al. (Bhawalkar, Kleinberg, Lewi, Roughgarden, & Sharma, 2015) first propose the anchored *k*-core problem to prevent the dissolution of social networks, attract as many users as possible to stay, and enable the maximum number of users to continue to participate in social networks. Zhang et al. (F. Zhang, Zhang, Qin, Zhang, & Lin, 2017) also propose the collapsed *k*-core problem, which is to remove *b* nodes from a *k*-core subgraph, such that the remaining *k*-core subgraph becomes minimum, in order to find collapsed users. After that, Zhang et al. (F. Zhang, Li, Zhang, Qin, & Zhang, 2018) extend this problem to the *k*-truss model and propose the collapsed *k*-truss. Due to the low efficiency of collapsed users finding algorithms in (F. Zhang et al., 2017), Zhang et al. (F. Zhang, Xie, Wang, Yang, & Jiang, 2022) propose the collapsed coreness problem. They introduce the core forest pruning technique, candidate nodes are pruned based on sub-trees containing nodes, and the graph is divided into different tree nodes to limit the calculation cost.

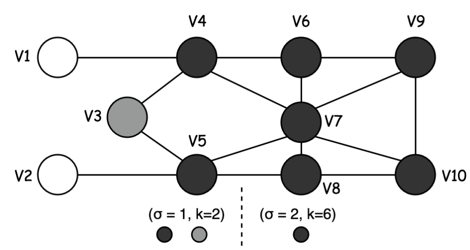


Figure 1: A *collapsed nodes* set in different distance.

Motivation. Although there are some studies on network collapse problem currently, these methods often overlook two key cognitive factors: (1) the *spatial constraints* inherent in human interactions (*e.g.*, people prioritize geographically proximate connections due to bounded attention), and (2) the *adaptive resilience mechanisms* communities employ to mitigate collapse (*e.g.*, redistributing tasks to nearby nodes). This gap limits their ability to model real-world social networks, where spatial and cognitive dynamics co-evolve. For example, in Figure 1, there are two node sets marked with different colors, corresponding to different social relation distances. We can observe that in different relation distance constraints, different sets of collapse nodes with varying neighbor structures can be obtained (*i.e.*, considering *more structural information* within node neighborhoods) and find more *fine-grained* subgraphs like $\{v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$ compared to

In D. Barner, N.R. Bramley, A. Ruggeri and C.M. Walker (Eds.), *Proceedings of the 47th Annual Conference of the Cognitive Science Society*

nodes set $\{v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$ on a larger social distance.

Contributions. As far as we are aware, this paper proposes a cognitive-computational framework that integrates spatial constraints and human adaptability into collapsed node detection. Our main contributions can be outlined as follows:

- We introduce a novel model linking spatial constraints to network resilience, grounded in cognitive science principles. Then, we propose an efficient algorithm leveraging spatial locality heuristics observed in human problem-solving.
- For the issue of too many collapsed nodes in large networks, we define the optimal collapsed nodes finding problem based on (k,σ)-*core model*. Similarly, we give theoretical proofs (NP-hardness and inapproximability) mirroring the computational complexity of human decision-making under resource constraints and a greedy heuristic algorithm.
- We conduct comprehensive empirical evaluations on large real and synthetic graphs. The empirical results validate the efficiency of our models and algorithms.

Preliminaries

In our paper, we give an *undirected* graph (network) $G = (V,E)$, where $V(G)$ is the node set, and $E(G)$ denotes the edge set. We denote the count of nodes and edges by n and m, respectively. The $N_G(u)$ is defined as $\{v \in V(G) | (u,v) \in E(G)\}$ and the degree of u is denoted as $d_G(u) = |N_G(u)|$. Before giving detailed problem statements, we first introduce the social distance between two nodes as follows:

Definition 1. Social Relation Path. *Given a graph G, the social relation path between node u and v, denoted as $P_G(u,v)$, is a reachability paths set between u and v. Suppose p_j is a path between u and v, $p_j = \{u,\ldots,u_i,\ldots,v\}$ and $P_G(u,v) = \bigcup_1^N p_j$, where N is the number of reachability paths and $1 \leq j \leq N$.*

Definition 2. Social Relation Distance. *Given a graph G, we denote social relation distance between u and v as $\sigma_G(u,v)$ and $\sigma_G(u,v) = min\{|p| p \in P_G(u,v)|\}$.*

Definition 3. σ-*distance* neighbor. *Given a graph G and an integer σ, we denote $\sigma-distance$ neighbors for node u as $F^\sigma(u,G)$. The social relation distance between nodes in $F^\sigma(u,G)$ and u is less than or equal to σ. i.e., $F^\sigma(u,G) = \{v \in N_G(u) | \sigma_G(u,v) \leq \sigma\}$.*

Definition 4. (k,σ)**-core.** *Given a graph G, two integers k and σ, the (k,σ)-core is a subgraph S which is denoted as $C_{k,\sigma}(G)$, such that: (i) all nodes $u \in S$ satisfy $|F^\sigma(u,S)| \geq k$, (ii) S is connected, (iii) There is no superset $S \subset S'$ and S' satisfies above conditions.*

In the (k,σ)-core model, a node that has more neighbors in a given distance means it is more closely connected to the network it belongs to. Therefore, we give the following definition:

Definition 5. Relationship Closeness. *Given a graph G and an integer σ, we denote the relationship closeness of node u as $K^\sigma(u,G)$. It is equal to a maximal k which satisfies $K^\sigma(u,G) = \{k | u \in C_{k,\sigma}(G)$ and $u \notin C_{k+1,\sigma}(G)\}$ in given σ.*

If all nodes with maximal relationship closeness leave the community, the entire social network will collapse, and no cohesive subgraph satisfying structural constraints will exist (*i.e.*, in Figure1, the nodes set $\{v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$ leaves network in distance 2). Therefore, we refer to them as *collapsed nodes*. Conversely, the departure of other nodes will not have a significant impact on network structure (*i.e.*, the v_1, v_2 and v_3 in distance 2).

Definition 6. Collapsed Node. *A node v whose removal causes the (k,σ)-core to disintegrate, i.e., the size of the remaining (k,σ)-core drops below a critical threshold $\alpha|H|$. This mirrors the cascade effect in cognitive systems, where the failure of a critical node disrupts collective information flow.*

Problem Statement 1. Given a graph G and an integer σ ($\sigma \geq 1$), in this paper, we investigate the efficient computation for the *collapsed node* set S_c.

However, in real social networks, due to the complex social relationships, *Problem 1* may identify a large number of collapsed nodes. To address this issue, we want to limit the size and give the definition of *optimal collapsed nodes*. The detailed problem definition is as follows:

Problem Statement 2. Given a graph G, two integers k ($k \geq 1$), σ ($\sigma \geq 1$) and a collapsed nodes set S_c, in this paper we investigate the algorithms about finding an *optimal collapsed nodes* set S_o efficiently, whose size is b.

Finding *Collapsed Node* Set

Human navigation in social networks often follows *spatial locality heuristics*—individuals prioritize nearby connections due to bounded attention and cognitive load. Inspired by this behavior, our algorithm dynamically prunes nodes beyond spatial threshold σ, mimicking the natural tendency to focus on proximate relationships during crisis response. Our algorithm iteratively removes nodes violating these constraints, mirroring the adaptive pruning observed in human group reorganization under stress. The detail about how to find σ-distance neighbors for each node is shown in Algorithm 1 and the time cost is $O(m+n)$. However, in this solution, the most time-consuming step is that we need to recompute the number of σ-distance neighbors for all remaining nodes after node deletion. Such repeated calculation is extraordinarily inefficient and cannot be applied to find results in large networks. Therefore, we give some lemmas to reduce the size of the node set that needs to be updated.

Lemma 1. *Given a graph G, an integer σ and node $w \in F^\sigma(u,G)$. If $\sigma_G(w,u) = l$ and $\mathbb{P} = \{v | v \in F^{\sigma-l}(u,G) \wedge \sigma_{G\setminus\{u\}}(w,v) > \sigma\}$, we have $F^\sigma(w,G) \setminus F^{\sigma-l}(w,G\setminus\{u\}) = \mathbb{P} \bigcup \{u\}$.*

Based on Lemma 1, we can get $|F^\sigma(w,G)| - |F^{\sigma-1}(w,G\setminus\{u\})| = |\mathbb{P}| + 1$. Therefore, after deleting node u, the main step of updating $|F^{\sigma}(w, G\backslash\{u\})|$ is how to calculate the nodes set $\mathbb{P}$ efficiently and we can get $F^{\sigma-l}(u,G)$ from $F^{\sigma}(u,G)$. Except for this step, we need to find nodes v which satisfies $\sigma_{G\backslash\{u\}}(w,v) \leq \sigma$ because we can recompute their social relation distance in subgraph $F^{\sigma}(u,G)$ instead of the whole graph G.

Algorithm 1: COMPUTESIGMAFN

Input: A graph G, an integer σ and node v
Output: $F^{\sigma}(v)$

1 Initialize Q, Set, $visited[|V(G)|]$ and $dist[|V(S)|]$;
2 **if** $\sigma = 1$ **then** return $|N_G(v)|$;
3 Q.push(v), $visited[v] = true$, $dist[v] = 0$;
4 **while** $Q \neq \emptyset$ **do**
5 $\quad u \leftarrow Q.pop()$;
6 $\quad$ **if** $dist[u] > \sigma$ **then** break;
7 $\quad Set.add(u)$;
8 $\quad$ **foreach** $w \in N_G(u)$ **do**
9 $\quad\quad$ **if** $visited[w] = false$ **then**
10 $\quad\quad\quad visited[w] = true, dist[w] = dist[u] + 1, Q.push(w)$
11 $Set \leftarrow Set\backslash\{v\}, F^{\sigma}(v) = Set$;
12 **return** $F^{\sigma}(v)$;

Lemma 2. *Given an integer σ and $G' = G\backslash\{u\}$. If the social relation distance between two nodes satisfies $\sigma_{G'}(v,w) \leq \sigma$ ($w \in F^{\sigma}(u,G), v \in F^{\sigma-l}(u,G)$), the all social relation paths between u and v are contained in subgraph $F^{\sigma}(u,G)$ ($\sigma_G(u,v) = l$). (i.e., $\forall u_j \in P_G(v,w), u_j \in F^{\sigma}(u,G)$)*

With Lemma 2, we can get the $\mathbb{P}$ in the induced subgraph S formed based on $F^{\sigma}(u,G)$. When we delete a node u from graph G, we only need to consider the node v, which satisfies the above two lemmas in S.

Algorithm 2: OFCU

Input: A graph G and an integer σ
Output: The collapsed nodes set S_c

1 Initialize S_c and array $T[0\ldots|V(G)|]$ is empty;
2 **foreach** $u \in V(G)$ **do**
3 $\quad |F^{\sigma}(u,G)| \leftarrow$COMPUTESIGMAFN (G, σ, u);
4 $\quad T[|F^{\sigma}(u,G)|] \leftarrow T[|F^{\sigma}(u,G)|] \cup \{u\}$;
5 **while** $G \neq \emptyset$ **do**
6 $\quad k = \min\{|F^{\sigma}(u,G)| | u \in V(G)\}$;
7 $\quad$ **while** $u \in T[k]$ **do**
8 $\quad\quad T[k] \leftarrow T[k]\backslash\{u\}, K^{\sigma}(u) = k$;
9 $\quad\quad$ UPDATESIGMAFN (G, u, σ);
10 $\quad\quad$ **foreach** $v \in F^{\sigma}(u,G)$ **do**
11 $\quad\quad\quad$ **if** $|F^{\sigma}(v, G\backslash\{u\})| \leq k$ *and* $v \notin T[k]$ **then**
12 $\quad\quad\quad\quad T[|F^{\sigma}(v,G\backslash\{u\})|] \leftarrow T[|F^{\sigma}(v,G\backslash\{u\})|]\backslash\{v\}$;
13 $\quad\quad\quad\quad T[k] \leftarrow T[k] \cup \{v\}$;
14 $\quad\quad G \leftarrow G\backslash\{u \cup E(u)\}$;
15 $S_c \leftarrow \bigcup_{\max\{K^{\sigma}(v)|v \in V(G)\}}\{v\}$;
16 **return** S_c;

Algorithm 2 shows the pseudocode of finding collapsed nodes set S_c and the lemmas we have discussed. We also define a bucket T to implement the increasing order processing of $|F^{\sigma}(u,G)|$ and can update the nodes in $T[i]$ in $O(1)$ time by using bucket sorting (Khaouid, Barsky, Srinivasan, & Thomo, 2015). Firstly, we initialize set S_c and bucket T in Line 1. And then, we use Algorithm 1 to calculate $|F^{\sigma}(u,G)|$ of each node u, while sorting each node in increasing order of $|F^{\sigma}(u,G)|$ using T (Lines 2-4). We get the minimum $|F^{\sigma}(u,G)|$ in Line 6 and remove nodes in $T[k]$, setting the relationship closeness of u to k in Line 8. After that, we use Algorithm 3 to update $|F^{\sigma}(v, G\backslash\{u\})|$ for each node v in Line 9 and perform a re-bucket sorting (Lines 10-13). Finally, after all nodes are processed, the collapsed nodes can be obtained in Line 15.

Algorithm 3: UPDATESIGMAFN

Input: A graph G, a node u and an integer σ

1 S is an induced subgraph from $F^{\sigma}(u,G)$;
2 Initialize array $A[n][m] \leftarrow \emptyset$ $(1 \leq n \leq 2, 0 \leq m \leq |V(S)|)$;
3 **foreach** $v \in F^{\sigma}(u,G)$ **do**
4 $\quad$ **if** $\sigma_G(u,v) < \sigma$ **then**
5 $\quad\quad A[1][v] \leftarrow \{v\}, A[2][v] \leftarrow \{v\}$;
6 $i = 1, j = 2, t = 1$;
7 **while** $t \leq \sigma$ **do**
8 $\quad i \leftrightarrow j$;
9 $\quad$ **foreach** $e(v,x) \in E(S)$ **do**
10 $\quad\quad A[i][v] \leftarrow A[i][v] \cup A[j][x], A[i][x] \leftarrow A[i][x] \cup A[j][v]$;
11 $\quad t = t + 1$;
12 **foreach** $v \in V(S)$ **do**
13 $\quad l = \sigma_G(u,v)$;
14 $\quad |F^{\sigma}(v, G\backslash\{u\})| = |F^{\sigma}(v,G)| - 1$;
15 $\quad$ **while** $x \in V(S)$ *and* $\sigma_G(u,x) \leq \sigma - l$ **do**
16 $\quad\quad$ **if** $x \notin A[j][v]$ **then**
17 $\quad\quad\quad |F^{\sigma}(v, G\backslash\{u\})| = |F^{\sigma}(v, G\backslash\{u\})| - 1$;

Algorithm 3 shows a process about updating the number of σ-distance neighbors for each remaining node. Firstly, we get the corresponding induced subgraph S formed based on $F^{\sigma}(u,G)$ in Line 1 and initialize the nodes with social relation distance less than σ in A as the nodes themselves, equal to σ as the empty (Lines 3-5). Then, we find all nodes that can be reached from v within distance σ (Lines 6-11). Finally, updating the number of σ-distance neighbors for each node with two lemmas proposed above (Lines 12-17). The time cost of Algorithm 3 is $O(\eta^2 + \theta \cdot \sigma \cdot \eta)$, where η and θ represent the number of nodes and edges in the induced subgraph, respectively.

Theorem 1. *The time complexity of Algorithm 2 is $O(n \cdot (N+M))$, where M and N are the time cost of computing* σ-distance neighbors *in Line 3 and updating the remaining nodes in Line 9.*

Finding *Optimal Collapsed Node* Set

In the previous section, we propose the problem of finding *collapsed nodes* based on σ-distance. However, in most real social networks, we usually find a large number of *collapsed nodes*. Therefore, in this section, we give the definition about *optimal collapsed node* and find a corresponding set with size b.

Definition 7. ***Follower (F. Zhang et al., 2018).*** *Given a graph G and node u, the followers of u is that if we delete* *u from $C_{k,\sigma}(G)$ subgraph, more nodes in $C_{k,\sigma}(G)$ might be deleted as well due to the (k,σ)-core model constraint. These nodes are called followers of u, which is denoted by $Follower^{\sigma}(u, C_{k,\sigma}(G))$.*

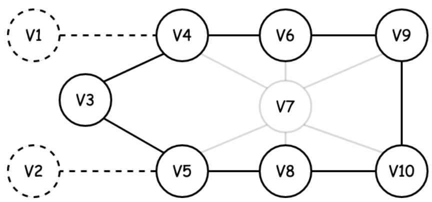


Figure 2: After v_7 leaves $(4,2)$-core subgraph.

Theorem 2. *Given a graph G, two integers k and σ, finding optimal collapsed nodes based (k,σ)-core is* NP*-hard.*

Proving the NP-hardness of optimal node selection (Theorem 2) reflects the computational intractability faced by humans when making resource-constrained decisions—a phenomenon aligned with *bounded rationality* theory. Just as humans resort to heuristics under time pressure, our greedy algorithm prioritizes nodes with the highest *cascading impact*, defined as the product of topological centrality and spatial influence decay. Although we always choose the *collapsed node* with most *followers* in each iteration as *optimal one*, in the worst case, finding *follower* set is also time-consuming. For example, the *followers* of a candidate node are all the nodes remaining in the (k,σ)-core subgraph, and the deletion of it will cause the entire network to break down. Therefore, we need to compute the *follower* set by checking all nodes in the (k,σ)-core subgraph. To improve the efficiency of the basic greedy algorithm, we first give several lemmas and pruning strategies as follows:

Lemma 3. *Given a graph G, a node $u \in C_{k,\sigma}(G)$ and $H = \{v | v \in F^{\sigma}(u) \wedge |F^{\sigma}(v)| \leq k\}$. If u has at least one follower, then the follower comes from set H.*

Lemma 4. *Given a graph G, a node $u \in C_{k,\sigma}(G)$, if v is the follower of u, we have $v \in F^{\sigma}(u) \wedge K^{\sigma}(v) \leq K^{\sigma}(u)$.*

Lemma 5. *Given a graph G, two nodes $u, v \in C_{k,\sigma}(G)$, if v is the follower of u, we have $Follower^{\sigma}(v, C_{k,\sigma}(G)) \subset Follower^{\sigma}(u, C_{k,\sigma}(G))$.*

Therefore, according to the above lemmas, if a *collapsed node* in $C_{k,\sigma}(G)$ subgraph is a *follower* of another *collapsed node*, it may not be an *optimal one*. If a *collapsed node* has the most number of *followers* before iteration begins, it is likely to be the *optimal one*. We can use these lemmas to further reduce the candidate checking space and improve the efficiency of the basic algorithm. It should be noted that the *b optimal collapsed nodes* are obtained from the *collapsed ones*. If the number of *collapsed nodes* is less than or equal to *b*, that is, all the *collapsed nodes* are optimal, and we can directly use Algorithm 2 to obtain results. Otherwise, we use Algorithm 4 to get *b optimal collapsed nodes*, and the pseudocode is shown as follows.

Firstly, we check the maximum relationship closeness and the size of the collapsed node-set. If one of them does not meet the conditions, we directly return the S_o(Lines 1-3). Otherwise, we perform *b* iterations, where in each iteration, we calculate *followers* for each current candidate node and select one with the most *followers* as *optimal collapsed node*(Lines 5-12). And then, we add the node *s* to set S_o, remove node *s* and corresponding *followers* from *collapsed node* set S_c in Line 14. After that, we update the remaining nodes in (k,σ)-core subgraph in Line 15. Finally, we can get the result in set S_o after the *b* iterations are completed.

Algorithm 4: OFBCU

Input: A graph $G(V,E)$, collapsed node set S_c, size constraint b and k, σ
Output: The optimal collapsed node set S_o

1 $S_o \leftarrow \emptyset$;
2 **if** *maximal relationship closeness* $< k$ **then return** S_o ;
3 **if** $S_c.size() \leq b$ **then** $S_o \leftarrow S_o \cup S_c$;
4 **else**
5     $i = b$;
6   **while** $i > 0$ **do**
7         $fk = 0$;
8   **foreach** $u \in S_c$ **do**
9             $F \leftarrow$ OptimizedComputeFollowers$(u, C_{k,\sigma}(G))$, $t \leftarrow |F|$;
10   **if** $fk < t$ **then**
11                 $fk = t, u^* = u$;
12         $s \leftarrow$optimal collapsed node u^* in this iteration;
13         $S_o \cup \{s\}$;
14         $S_c \leftarrow S_c \backslash \{s\}, S_c \leftarrow S_c \backslash F$;
15   update $C_{k,\sigma}(S_o, G), i = i - 1$;

Algorithm 5: OptimizedComputeFollowers

Input: $C_{k,\sigma}(G)$, candidate collapsed node x
Output: F (the followers of node x)

$F \leftarrow \emptyset, C_{k,\sigma}(G) \backslash \{x\}$;
while $w \in C_{k,\sigma}(G)$ **do**
    if $\exists w \in F^{\sigma}(x) \wedge |F^{\sigma}(w)| \leq k$ **then**
        $F \leftarrow F \cup \{w\}, C_{k,\sigma}(G) \leftarrow C_{k,\sigma}(G) \backslash \{w\}$;
    else
        $m \leftarrow K^{\sigma}(w, C_{k,\sigma}(G))$;
        if $m < k$ **then**
            $F \leftarrow F \cup \{w\}, C_{k,\sigma}(G) \leftarrow C_{k,\sigma}(G) \backslash \{w\}$;
$C_{k,\sigma} \leftarrow C_{k,\sigma} \cup F$;

Theorem 3. *The time complexity of Algorithm 4 is $O(b \cdot n \cdot c)$, where n is the number of candidate collapsed nodes and c is the time cost of calculating followers during each iteration. In the worst case, although the time complexity of Algorithm 4 is the same as the basic one, it also reduces the computation of candidate collapsed nodes.*

Experiments and Results

We have carried out empirical studies to evaluate the effectiveness of our newly developed model, as well as the efficiency of our proposed algorithms. The following algorithms were specifically evaluated.

- FCU: Baseline for *Problem 1*, which iteratively deletes nodes with the weakest relationship closeness in given σ like core decomposition.

- OFCU: Algorithm 2.
- BITFCU: We use bitwise-or operator in bitmaps (Dai et al., 2021) to replace the traditional union operation in Algorithm 3 Line 9.
- FBCU: Baseline for *Problem 2*, which selects the current most *followers collapsed node* of each iteration and adds it to the *optimal collapsed node* set until b nodes are found.
- OFBCU: Algorithm 4.
- RFBCU: Randomly select a *collapsed node* of each iteration and add it to the *optimal collapsed node* set until b nodes are found.
- CKC: The *optimal collapsed node* discovery algorithm based on k-core proposed by Zhang et al. (F. Zhang et al., 2018).

All the algorithms are implemented in C++, and all the experiments are conducted on a Windows server with an Intel(R) Core(TM) 2.9GHz CPU and 64GB memory.

Table 1: GRAPH STATISTICS

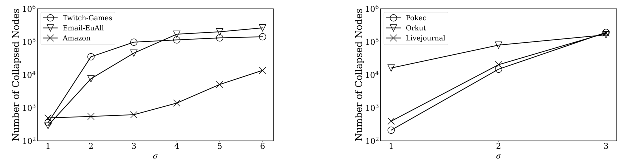
Graph	$\|V\|$	$\|E\|$	$\|\hat{p}\|$
Facebook	4,039	88,234	4.7
CA-AstroPh	18,772	198,050	5.1
Twitch-Games	168,114	1,048,575	6.3
Email-EuAll	265,214	364,481	4.5
Amazon	548,552	925,872	5.8
Pokec	1,632,804	22,301,964	5.2
Orkut	2,997,167	106,349,209	4.8
Livejournal	4,847,571	42,851,237	6.5

Datasets. We use 8 public real networks and $|\hat{p}|$ means average *social relation distance* in graph G. Orkut are downloaded from Network Repository[1] and other datasets are downloaded from SNAP[2].

Finding *Collapsed Nodes* Experiments

Eval-I: Effectiveness Testing. In this test, we set σ to 1 through 6 in the medium dataset and 1 through 3 in the large dataset, with 2 as default. If σ is too large, it will cause each participant to be associated with all others. Two figures in Figure 3 show that the larger the σ, the more *collapsed nodes* are found. And then, we typically obtain a larger *collapsed node* set in a social network that has higher density and larger size. For example, in Figure 3b, when setting σ to 3, the number of *collapsed nodes* found in three large datasets exceeds 120,000. And then, we also compare our model with other traditional models in Figure 4. Due to σ, we can find more *collapsed nodes* than other models in all social networks because the social relation distance of nodes is not limited to just their neighbors but instead include all nodes within σ. With σ, we can better focus on all neighbors within a social relation distance of a node.

Eval-II: Efficiency Testing. In Figure 5, we can find that the time cost of three proposed algorithms increases slightly when σ is increased by 1 to 6. Because a larger σ means that we need to check more neighbors for each node, our improved algorithms OFBCU and BITFCU are more efficient than baseline FCU. Especially on a large network like Orkut and Livejournal, the efficiency improvement of two improved algorithms is more significant, and we early terminated the algorithm when the running time exceeded 10^5s.



(a) Medium dataset (b) Large dataset

Figure 3: (Eval-I) Number of collapsed nodes for varying σ.

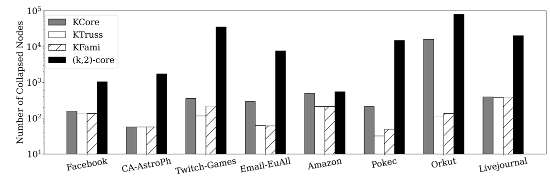
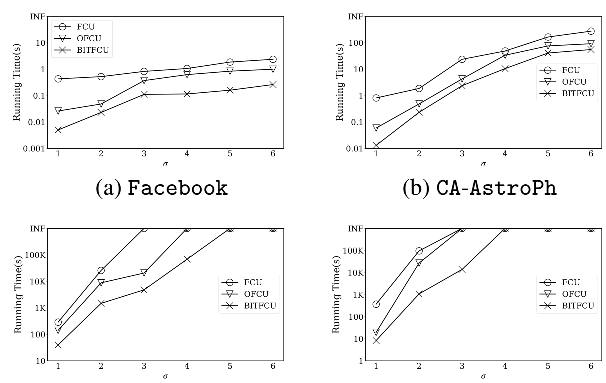


Figure 4: (Eval-I) Number of collapsed nodes in different datasets.



(a) Facebook (b) CA-AstroPh

(c) Orkut (d) Livejournal

Figure 5: (Eval-II) Query time for varying σ.

Finding *Optimal Collapsed Nodes* Experiments

Eval-III: Effectiveness Testing. We set $k = 30$, $b = 15$, $\sigma = 2$ and the Figure 6 shows that the number of *followers* corresponding to *b optimal collapsed nodes* found by our two algorithms with that of a random one, which randomly chooses b nodes from *collapsed nodes*. We find that our greedy approach is effective and can make us obtain *b optimal collapsed nodes* with more *followers* than the random algorithm in all experimental datasets.

In Figure 7a, we can find that different k will affect the number of *followers*, while the number of *followers* fluctuates as k increases. And Figure 7b shows that the number of *followers* increases as the σ increases because larger σ means that, for each node, we should consider more neigh-

[1] https://networkrepository.com/soc-orkut.php
[2] https://snap.stanford.edu/data/

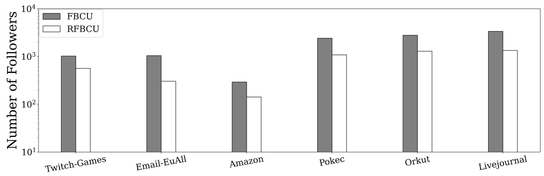
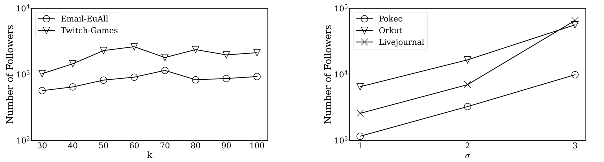


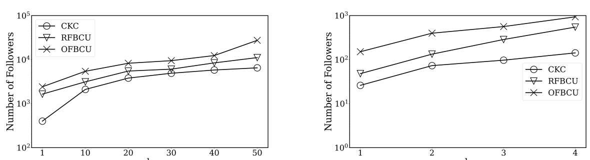
Figure 6: (Eval-III) Number of followers in different datasets.



(a) Varying k ($b = 30$, $\sigma = 2$) (b) Varying σ ($b = 20$, $k = 50$)

Figure 7: (Eval-III) Followers for varying k and σ.

bors, the leave will affect more nodes and make them dissatisfy the structural limitations of the community. In Figure 8, the $k = 20$, $\sigma = 2$, and we can find that if we want to find more *optimal collapsed nodes*, more corresponding *followers* will be removed. In addition, with the same b value, our algorithm OFBCU can always find more *followers* than the other two algorithms because the social relation distance can allow us to find more neighbors at one node.



(a) `Orkut` (b) `Facebook`

Figure 8: (Eval-III) Followers for varying b.

Eval-IV: Efficiency Testing. In Figure 9, we set $b = 15$, $\sigma = 2$, $k = 30$, and it shows the query time of our two proposed algorithms in different datasets. We can find that OFBCU costs less time than FBCU, which shows that our proposed lemmas are effective. For example, in `Twitch-Games` and `Pokec`, the time cost of them is about 6 and 336 seconds with OFBCU while FBCU will need 44 and 169 seconds, respectively.

Figure 10 shows that the time cost of two algorithms both decreases as k increases because a larger k means that more nodes, which dissatisfy the requirement of having more neighbors within σ-distance, will be removed. In contrast, for both Figure 11 and Figure 12, the time cost increases as σ and b increase. Because we need to check more neighbors for each candidate node within σ-distance and find more *optimal collapsed nodes*.

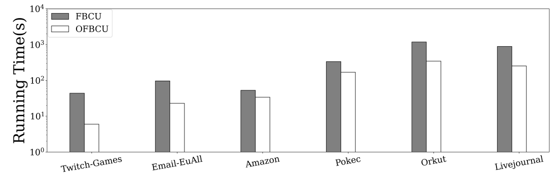
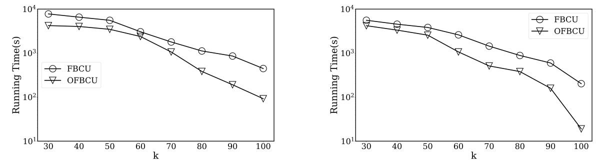
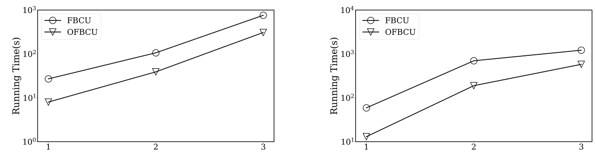


Figure 9: (Eval-IV) Efficiency testing.



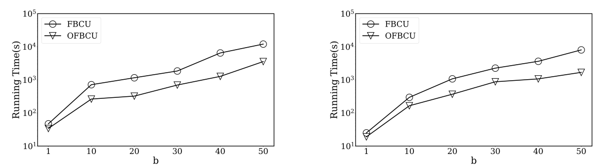
(a) `Orkut` (b) `Livejournal`

Figure 10: (Eval-IV) Query time for varying k ($b = 50$, $\sigma = 2$).



(a) `Pokec` (b) `Livejournal`

Figure 11: (Eval-IV) Query time for varying σ ($k = 75$, $b = 15$).



(a) `Orkut` (b) `Pokec`

Figure 12: (Eval-IV) Query time for varying b ($k = 20$, $\sigma = 2$).

Conclusion

We propose a cognitive-computational framework to detect collapsed nodes under spatial constraints, bridging graph theory with principles of spatial cognition and bounded rationality. First, spatial constraints induce cognitive biases in failure perception. Moreover, the NP-hardness of optimal selection mirrors human decision-making under resource scarcity, validated by our cognitively-aligned greedy heuristic. Extensive empirical studies demonstrate the effectiveness of our model and the efficiency of our techniques. Future work will extend this framework to model dynamic cognitive thresholds (*e.g.*, stress-induced σ reduction) and collaborative recovery strategies.

Acknowledgements

This research was sponsored by the Stable Supporting Fund of National Key Laboratory of Underwater Acoustic Technology, JCKYS2023604SSJS013, and the Natural Science Foundation of Heilongjiang Province, LH2023F020.

References

Bhawalkar, K., Kleinberg, J., Lewi, K., Roughgarden, T., & Sharma, A. (2015). Preventing unraveling in social networks: the anchored k-core problem. *SIAM Journal on Discrete Mathematics*, *29*(3), 1452–1475.

Bugedo, S., Riveros, C., & Salas, J. (2024). A family of centrality measures for graph data based on subgraphs. *ACM Transactions on Database Systems*, *49*(3), 1–45.

Cai, T., Yang, S., Li, J., Sheng, Q. Z., Yang, J., Wang, X., … Gao, L. (2022). Incremental graph computation: Anchored vertex tracking in dynamic social networks. *IEEE Transactions on Knowledge and Data Engineering*.

Dai, Q., Li, R.-H., Qin, L., Wang, G., Yang, W., Zhang, Z., & Yuan, Y. (2021). Scaling up distance-generalized core decomposition. In *Proceedings of the 30th acm international conference on information & knowledge management* (pp. 312–321).

He, Y., Wang, K., Zhang, W., Lin, X., & Zhang, Y. (2023). Scaling up k-clique densest subgraph detection. *Proceedings of the ACM on Management of Data*, *1*(1), 1–26.

Khaouid, W., Barsky, M., Srinivasan, V., & Thomo, A. (2015). K-core decomposition of large networks on a single pc. *Proceedings of the VLDB Endowment*, *9*(1), 13–23.

Tao, Y., Guo, K., Zheng, Y., Pan, S., Cao, X., & Chang, Y. (2024). Breaking the curse of dimensional collapse in graph contrastive learning: A whitening perspective. *Information Sciences*, *657*, 119952.

Yang, Z., Li, X., Zhang, X., Luo, W., & Li, K. (2022). K-truss community most favorites query based on top-t. *World Wide Web*, 1–21.

Zhang, F., Li, C., Zhang, Y., Qin, L., & Zhang, W. (2018). Finding critical users in social communities: The collapsed core and truss problems. *IEEE Transactions on Knowledge and Data Engineering*, *32*(1), 78–91.

Zhang, F., Xie, J., Wang, K., Yang, S., & Jiang, Y. (2022). Discovering key users for defending network structural stability. *World Wide Web*, *25*(2), 679–701.

Zhang, F., Zhang, Y., Qin, L., Zhang, W., & Lin, X. (2017). Finding critical users for social network engagement: The collapsed k-core problem. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 31).

Zhang, Y., Lin, L., Yuan, P., & Jin, H. (2022). Significant engagement community search on temporal networks. In *Database systems for advanced applications: 27th international conference, dasfaa 2022, virtual event, april 11–14, 2022, proceedings, part i* (pp. 250–258).

Zhou, B., Lv, Y., Wang, J., Zhang, J., & Xuan, Q. (2022). Attacking the core structure of complex network. *IEEE Transactions on Computational Social Systems*.