SpecGreedy: Unified Dense Subgraph Detection

Wenjie Feng[1,*] (✉), Shenghua Liu[1,*](✉), Danai Koutra[2],
Huawei Shen[1,*], and Xueqi Cheng[1,*]

[1] Institute of Computing Technology, Chinese Academy of Sciences, 100190
[2] University of Michigan, Ann Arbor, MI, USA
wenchiehfeng.us@gmail.com, liushenghua@ict.ac.cn, dkoutra@umich.edu,
{shenhuawei,cxq}@ict.ac.cn

Abstract. How can we effectively detect fake reviews or fraudulent connections on a website? How can we spot communities that suddenly appear based on users' interaction? And how can we efficiently find the minimum cut in a big graph? All of these are related to the problem of finding dense subgraphs, an important primitive problem in graph data analysis with extensive applications across various domains.
We focus on formulating the problem of detecting the densest subgraph in real-world large graphs, and we theoretically compare and contrast several closely related problems. Moreover, we propose a unified framework for the densest subgraph detection (GenDS) and devise a simple and computationally efficient algorithm, SpecGreedy, to solve it by leveraging the graph spectral properties with a greedy approach. We conduct thorough experiments on 40 real-world networks with up to 1.47 billion edges from various domains, and demonstrate that our algorithm yields up to 58.6× speedup and achieves better or approximately equal-quality solutions for the densest subgraph detection compared to the baselines. Moreover, SpecGreedy scales linearly with the graph size and is proved effective in applications, such as finding collaborations that appear suddenly in a big, time-evolving co-authorship network.

Keywords: Dense subgraph detection; Graph pattern mining; Algorithm

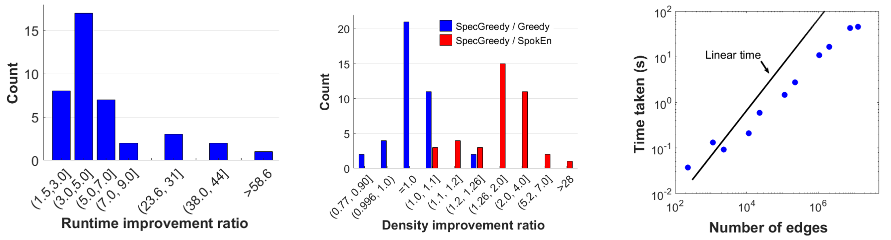
1 Introduction

How can we capture the most contrast groups or communities in temporal or dynamic graphs — e.g. hot-topics or collaborations in the research community that appear suddenly? How can we efficiently determine the minimum cut for a large graph? How can we find the most suspicious users based on their behaviors or spot the largest group with consensus opinion on controversial issues? All these real-world problems are related to the densest subgraph detection task.

Dense pattern mining in graphs is a key primitive task for extracting useful information and capturing underlying principles in relational data. It has benefited

Corresponding to Wenjie Feng and Shenghua Liu.

[*] They are also with CAS Key Laboratory of Network Data Science & Technology, CAS, and University of Chinese Academy of Sciences, Beijing 100049, China.



(a) Speedup statistic for the densest subgraph detection (b) Optimal densest subgraph density quality comparison (c) The linear scalability of SpecGreedy

Fig. 1. Proposed algorithm SpecGreedy is fast, effective, and scalable. (a) Our method detects the densest subgraphs (qualities in Fig. 1(b)) up to 58.6× faster than the widely-used Greedy algorithm for various real-world datasets. (b) SpecGreedy has better or comparable density quality compared with Greedy and SpokEn algorithm in the densest subgraph detection. It consistently outperforms SpokEn for all graphs and finds up to 28× denser subgraph; it obtains the same or denser (more than 1.26×) optimal density for most graphs compared with Greedy, and 4 graphs with very close densities ($\geq$ 0.996×) and only 2 graphs with less than 0.9 density improvement. (c) The time taken of SpecGreedy grows linearly with the size of graph.

various application domains [16], such as capturing the functional groups in biology [30], traffic patterns in human behaviors and interactions [20], communities in social networks [25], anomaly detection in financial and other networks [1], and more. The densest subgraph problem has garnered significant interest in practice because it can be solved exactly in polynomial-time and has an adequate approximation in almost linear time. Goldberg's maximum flow algorithm [13] and Charikar's LP-based algorithm [7] provide the exact solution, and Charikar [7] proved that the simple greedy algorithm is guaranteed to find a result of quality better than the factor 2-approx. with linear time to the graph size. However, these algorithms still incur a prohibitive computational cost for the massive graphs that arise in modern data science applications, without considering the properties of real-world data.

To the best of our knowledge, there is no related work to study the connection of the above problems. Here we summarize the differences and relations for some well-known related problems, including detecting community with sparse cut and suspicious dense subgraphs. We also propose a unified formulation, *generalized densest subgraph* (GenDS) *problem*, which subsumes various application problems. This unification explicitly highlights those relations in a formal way and leads to a consistent method for solving these problems. We thus devise an efficient detection algorithm, SpecGreedy, that leverages the graph spectral properties and greedy peeling strategy to solve the generalized problem. With thorough experiments using 40 diverse real-world networks, we demonstrate that our algorithm is fast, highly effective, and scalable (linear to the number of edges, as shown in Fig. 1); it yields 58.6× speedup, and achieves almost equal or better quality, even for a very large graph with $1.47B$ edges. We also find interesting patterns, such as contrast collaboration dense patterns in DBLP co-authorship data.

Our main contributions include:

- **Theory & Correspondences:** We propose the generalized densest subgraph detection formulation, GenDS, to unify several related problems, and analyze the optimization in the principle of the spectral theory;
- **Algorithm:** We devise SpecGreedy, a fast and scalable algorithm to solve the unified GenDS problem;
- **Experiment:** We conduct thorough empirical analyses of various real-world graphs to verify the efficiency and linear-scalability of SpecGreedy. We also find some large contrast dense subgraphs in co-authorship relations.

Reproducibility: Our open-sourced code, the data used, and the supplement document are available at `https://github.com/wenchieh/specgreedy`.

2 Related Work

In this section, we summarize the related work on the densest subgraph problem and various methods for detecting dense subgraphs in different applications.

Finding the densest subgraph in the large input graph is a widely studied problem [16]. Generally speaking, the goal of such a problem is to find a set of nodes of a given input graph to maximize some notion of density. The so called *densest subgraph problem* (DSP) aims to find a subgraph that maximize the degree density, which is the average of the weights of all its edges. When the edge weights are non-negative, the densest subgraph can be identified optimally in polynomial time by using maximum flow algorithms [13]. However, obtaining the exact solution with maximum flow requires expensive computations despite the theoretical progress achieved in recent years, thus making it prohibitive for large graphs. Charikar [7] introduces a linear-programming formulation of the problem and shows that the greedy algorithm proposed by Asashiro et al. [4] produces a $1/2$-approximation of the optimum density in linear time. [21] proposes an optimization model for local community detection by extending the densest subgraph problem. A recent study [5] proposes a Greedy++ algorithm to improve the output quality of the subgraph over Charikar's greedy peeling algorithm [7] by drawing insights from the iterative approaches from convex optimization. However, when the edge weight can be negative, the above problem becomes NP-hard [27]. When restrictions on the size lower bound are specified, the *densest k-subgraph problem* (DkS) becomes NP-complete [2] and there does not exist any PTAS under a reasonable complexity assumption.

Another line of related research includes contrast graph pattern mining, which aims to discover subgraphs that manifest drastic differences between graphs. Yang et al. [31] proposed to detect the density contrast subgraphs which is equivalent to mining the densest subgraph from a "difference" graph, and employed a local search algorithm to find the solution. Tsourakakis et al. [27] focused on the risk-aversion dense subgraph pattern for a graph with small negative weights and extended the greedy algorithm for this case. [9] detects the k-oppositive

Table 1. Symbols and Definitions

Symbol	Definition
$\mathcal{G} = (V, E)$	Undirected graph with node set V and edge set $E \subseteq V \times V$
$\hat{\mathcal{G}} = (L \cup R, E)$	Bipartite graph with node set, L and R, and edge set $E \subseteq L \times R$
$\mathcal{G}_r = (V, E_r)$	Positive residual graph with node set V and residual edge set E_r
$\boldsymbol{x}, \boldsymbol{y}$	Indicator vector for the selected subset of nodes
$\boldsymbol{u}, \boldsymbol{v}$	Eigenvector or singular vector
$\mathbf{A}, \mathbf{L}$	Adjacency and Laplacian matrix of a graph
$\boldsymbol{d}, \mathbf{D}$	Node degree vector and its diagonal matrix, $d_i = \sum_j a_{ij}$
$\mathbf{I}$	Identity matrix of size $n \times n$
$\mathbf{D}_{\boldsymbol{x}}$	Diagonal matrix for the vector $\boldsymbol{x}$

cohesive groups by solving a quadratic optimization problem for signed networks. Also, dense subgraphs are used to detect communities [8, 30] and anomaly [15, 24]. Fraudar [15] proposed to use the greedy method that incorporates the suspiciousness of nodes and edges during optimization. SpokEn [24] utilizes the "eigenspokes" pattern of community in the EE-plots produced by pairs of eigenvectors of a graph, which is applied to fraud detection.

Besides, there are many works that utilize the spectral properties of the graph to detect communities [25] and dense subgraphs [22, 3], and to partition the input graph [10].

3 Problem and Correspondences

Preliminaries and Definitions. Throughout the paper, vectors are denoted by boldface lowercase letters (e.g. $\boldsymbol{x}$), matrices are denoted by boldface uppercase letters (e.g. $\mathbf{A}$), and sets are denoted by uppercase letters(e.g. S, V). The operator $|\cdot|$ denotes the cardinality of a set or the number of non-zero (`nnz`) elements in a vector and $\|\cdot\|$ is the l_2 norm of a vector, $\lceil x \rceil \equiv \{1, \ldots, x\}$ for brevity. Table 1 gives the complete list of symbols we use in the paper.

Consider an undirected graph $\mathcal{G} = (V, E)$ with $|V| = n$. Let $S \subseteq V$ and $E(S)$ be the edges of subgraph $\mathcal{G}(S)$ induced by the subset S, i.e. $E(S) = \{e_{ij} : v_i, v_j \in S \wedge e_{ij} \in E\}$. Let $\mathbf{A} = (a_{ij}) \in \mathbb{R}^{n \times n}$ be the adjacency matrix of $\mathcal{G}$ and $a_{ij} \geq 0$.

Given an indicator vector $\boldsymbol{x}$ of size n for the subset S, the average degree density of the subgraph $\mathcal{G}(S)$, being the mostly used density measure for the densest subgraph problem, is defined by Charikar [7] as

$$g(S) = \frac{|E(S)|}{|S|} = \frac{1}{2} \cdot \frac{\boldsymbol{x}^T \mathbf{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}, \ \boldsymbol{x} \in \{0, 1\}^n, \tag{1}$$

and avoids the trivial solution by limiting $|\boldsymbol{x}| \geq 1$. Generally, Hooi et al [15] proposed to consider the node weight (some constant for each node) for the total

Table 2. Summary for correspondence to problem GenDS

	Method	matrix $\mathbf{P}$	matrix $\mathbf{Q}$	Constraint
1	MinQuotientCut [10]	$\mathbf{A} - \mathbf{D} = -\mathbf{L}$	$\mathbf{I}$	$\lvert\boldsymbol{x}\rvert < n$
2	Charikar [7]	$\mathbf{A}$	$\mathbf{I}$	$\lvert\boldsymbol{x}\rvert \geq 1$
3	Fraudar [15]	$\mathbf{A} + 2\mathbf{D}_{\boldsymbol{w}}$	$\mathbf{I}$	
4	SparseCutDS[1] [21]	$\mathbf{A} - \frac{2\cdot\alpha}{2\alpha+1}\mathbf{D}$	$\mathbf{I}$	
5	TempDS [30]	$\mathbf{A}_t$	$\mathbf{A}_{t-1} + 2\,\mathbf{I} = \tilde{\mathbf{A}}_{t-1}$	
6	Risk-averse DS [27]	$\mathbf{A}^+ + \lambda_1\mathbf{I} = \tilde{\mathbf{A}}^+$	$\mathbf{A}^- + \lambda_2\mathbf{I} = \tilde{\mathbf{A}}^-$	
	GenDS[2]	$\mathbf{A} + 2\mathbf{D}_{\boldsymbol{c}}$	$\mathbf{A}' + \gamma\,\mathbf{I} = \tilde{\mathbf{A}}'$	

[1] The contrast subgraph pattern [19] equals to set $\alpha = 1$, and $\alpha = \frac{1}{2}$ is considered in [18] for community detection.

[2] Bipartite graphs can be transformed into an undirected graph as Lemma 9.

mass, the density of $\mathcal{G}(S)$ is

$$g(S) = \frac{|E(S)| + \sum_{i\in V} c_i}{|S|} = \frac{\boldsymbol{x}^T\mathbf{A}\boldsymbol{x}}{2\cdot\boldsymbol{x}^T\boldsymbol{x}} + \frac{\boldsymbol{x}^T\mathbf{D}_{\boldsymbol{c}}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = \frac{1}{2}\frac{\boldsymbol{x}^T(\mathbf{A}+2\mathbf{D}_{\boldsymbol{c}})\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}},\ \boldsymbol{x}\in\{0,1\}^n, \tag{2}$$

where $c_i \in \mathbb{R}^+$ is the weight of node i and $\mathbf{D}_{\boldsymbol{c}}$ is the diagonal matrix of the weight vector $\boldsymbol{c} = [c_1, \ldots, c_n]$.

In addition to dense subgraphs within a single graph, we also consider the "contrast" patterns of cross-graphs, i.e., a subset of nodes that have significantly different edges or edge weights in two given graphs of the same nodeset, like the different snapshots of a dynamic graph.

Generalized Densest Subgraph Problem. Therefore, we propose a generalized densest subgraph problem which subsumes various well-known formulations:

Problem 1 (GenDS: Generalized Densest Subgraph detection). **Given** a graph $\mathcal{G} = (V, E)$ and its contrast $\mathcal{G}' = (V, E')$ with $|V| = n$ nodes; **find** the optimal subset $S^* \subseteq V$ and $|S^*| \geq 1$ **such that**

$$S^* = \underset{S\subseteq V,\,|S|\geq 1}{\arg\max}\ g(S;\mathbf{P},\mathbf{Q}) = \underset{\boldsymbol{x}\in\{0,1\}^n,|\boldsymbol{x}|\geq 1}{\arg\max}\ \frac{\boldsymbol{x}^T\mathbf{P}\boldsymbol{x}}{\boldsymbol{x}^T\mathbf{Q}\boldsymbol{x}}, \tag{3}$$

where matrices $\mathbf{P}$ and $\mathbf{Q}$ are related to $\mathcal{G}$ and $\mathcal{G}'$, that is, $\mathbf{P} = \mathbf{A} + 2\mathbf{D}_{\boldsymbol{c}}$ and $\mathbf{Q} = \mathbf{A}' + \gamma\mathbf{I}$.

Here we define $\tilde{\mathbf{A}}' = \mathbf{A}' + \gamma\mathbf{I}$ as the *augmented adjacency matrix* of graph $\mathcal{G}'$. The denominator in Eq. (3) simultaneously considers the size of the node subset and the connections in the subgraph $\mathcal{G}(S)$. Specifically, if the contrast $\mathcal{G}'$ is an empty graph, $\mathbf{Q}$ degenerates to be a γ-scale identity matrix with only considering the size of the subgraph in GenDS. Note that $\mathbf{P}$ also becomes an augmented adjacency matrix of $\mathcal{G}$ as well if the node weights are equal, i.e., $c_i = c_1 > 0$.

As we show in Theorem 2, our proposed GenDS problem is more general and many dense subgraph-based formulations are special cases of it.

Theorem 2. GenDS *is a general framework for the MinQuotientCut, the densest subgraph detection (Charikar), Fraudar (suspicious dense subgraph),* SparseCutDS *(dense community with sparse cut),* TempDS *(temporal dense subgraph), and Risk-averse DS (consensus dense subgraph), and more.*

The following remarks provide detailed instantiations of GenDS for several problems. Table 2 summarizes the setting and provides the corresponding equation carefully aligned to highlight the correspondences to GenDS.

Remark 3. [MinQuotientCut] The optimal quotient cut ratio problem aims at partitioning the graph into two parts with minimum cut. Let the set of cut edges for S be $cut(S) = \{(u, v) \in E | u \in S, v \in V \setminus S\}$, its size can be formulated as

$$|cut(S)| = \sum_{e_{ij} \in E} a_{ij}(\boldsymbol{x}_i - \boldsymbol{x}_j)^2 = \boldsymbol{x}^T(\mathbf{D} - \mathbf{A})\boldsymbol{x} = \boldsymbol{x}^T\mathbf{L}\boldsymbol{x},$$

where $\boldsymbol{x}_i = 1$ if $i \in S$, and $\boldsymbol{x}_i = 0$ otherwise. The cut ratio of S is $\frac{|cut(S)|}{\min\{|S|, |V \setminus S|\}}$. Without loss of generality, assuming S is the smaller set compared with its complement, we have the minimum cut ratio by maximizing $-\frac{\boldsymbol{x}^T\mathbf{L}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}}$, which corresponds to $\mathbf{P} = -\mathbf{L}$ with $\boldsymbol{c} = -\frac{\boldsymbol{d}}{2}$ and $\mathbf{Q} = \mathbf{I}$ with $\mathbf{A}' = \mathbf{0}$ and $\gamma = 1$. [1]

Remark 4. [Charikar] The densest subgraph detection problem as formulated in Eq. (1) corresponds to $\mathbf{P} = \mathbf{A}$ and $\mathbf{Q} = \mathbf{I}$ with ignoring the constant factor.[2]

Remark 5. [Fraudar] The suspicious densest group detection problem treats the weights of nodes and edges as their suspiciousness score of nodes and edges, i.e. c_u and a_{ij} measure how individually suspicious the particular node u and edge e_{ij} is (can be determined by other information, like profile and text of content) resp. As Eq. (2) shows, it corresponds to $\mathbf{P} = \mathbf{A} + 2\mathbf{D}_{\boldsymbol{c}}$ and $\mathbf{Q} = \mathbf{I}$, where the numerator $\boldsymbol{x}^T\mathbf{P}\boldsymbol{x}$ is the total suspiciousness of the subgraph ignoring the constant factor.

Remark 6. [SparseCutDS] SparseCutDS finds a community that is densely connected internally but sparsely connected to the rest of the graph, and it is optimized by maximizing the density while minimizing the average cut size [21]. With the formulation of the cut size in remark 3, the objective to be maximized by SparseCutDS is denoted as

$$g_\alpha(S) = \frac{|E(S)| - \alpha \cdot |cut(S)|}{|S|} = \frac{\boldsymbol{x}^T\left((\frac{1}{2} + \alpha)\mathbf{A} - \alpha\mathbf{D}\right)\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = c \cdot \frac{\boldsymbol{x}^T\left(\mathbf{A} - \frac{2\alpha}{2\alpha+1}\mathbf{D}\right)\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}},$$

where α controls the weight of the $|cut(S)|$ term and $c = \frac{1}{2} + \alpha$ is a constant. Thus, it corresponds to $\mathbf{P} = \mathbf{A} + 2\mathbf{D}_{\boldsymbol{c}}$ with $\mathbf{D}_{\boldsymbol{c}} = -\frac{\alpha}{2\alpha+1}\mathbf{D}$ and $\mathbf{Q} = \mathbf{I}$.

[1] In the other setting with $\mathbf{Q} = \mathbf{D}$, this problem is also equivalent to set $\mathbf{P} = -\mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$, i.e., the normalized Laplacian matrix of $\mathcal{G}$, and $\mathbf{Q} = \mathbf{I}$.

[2] [29, 23] used $\tilde{\mathbf{A}}$ with different γ to explore the trade-off between density and size of final dense subgraphs with the domain-set based optimization method.

Remark 7. [TempDS] TempDS detects dense subgraphs with nodes S appearing at time t suddenly while having very few edges at time $t-1$ [30]. Let $\mathbf{A}_t$ and $\mathbf{A}_{t-1}$ be adjacency matrices of the snapshots of a temporal graph. Thus, $\boldsymbol{x}^T\mathbf{A}_t\boldsymbol{x}$ and $\boldsymbol{x}^T\mathbf{A}_{t-1}\boldsymbol{x}$ are twice the numbers of edges in corresponding subgraphs. By considering the size of subset S, the objective of TempDS can be formulated as:

$$g(S) = \frac{\boldsymbol{x}^T\mathbf{A}_t\boldsymbol{x}}{\boldsymbol{x}^T(\mathbf{A}_{t-1}+2\mathbf{I})\boldsymbol{x}} = \frac{\boldsymbol{x}^T\mathbf{A}_t\boldsymbol{x}}{\boldsymbol{x}^T\tilde{\mathbf{A}}_{t-1}\boldsymbol{x}}.$$

Remark 8. [Risk-averse DS] Given a graph $\mathcal{G}$, the positive entry a_{ij} of its adjacency matrix $\mathbf{A}$ represents the expected *reward* of the edge (u_i, u_j) and the negative entry is opposite to the *risk* of the edge, the absolute value $|a_{ij}|$ measures the strength. Then $\mathbf{A}$ can be written into $\mathbf{A} = \mathbf{A}^+ - \mathbf{A}^-$, where $\mathbf{A}^+$ is the *reward network* and composed of all positive edges in $\mathbf{A}$, that is, its entry $\mathbf{A}^+_{i,j} = \max(a_{ij}, 0)$; and $\mathbf{A}^-$ is the *opposition risk network* and its entry $\mathbf{A}^-_{i,j} = |\min(a_{ij}, 0)|$.

The Risk-averse dense subgraph detection problem finds a subgraph that has a large positive average degree and small negative average degree [27], it is formulated in GenDS format by setting $\mathbf{P} = \mathbf{A}^+ + 2\mathbf{D}_{\boldsymbol{c}}$ with $\boldsymbol{c} = \frac{\gamma_1}{2}\mathbf{1}$ and $\mathbf{Q} = \mathbf{A}^- + \gamma_2\mathbf{I}$, where $\gamma_1, \gamma_2 \geq 0$ control the size of the subgraph by considering the contribution of the size of the subset S.

As for the densest subgraph detection in a bipartite graph $\hat{\mathcal{G}}$, it can be reduced to the GenDS framework by converting $\hat{\mathcal{G}}$ to be a monopartite graph as following.

Lemma 9. *Given a bipartite graph $\hat{\mathcal{G}} = (L \cup R, E)$ with $|L| + |R| = n$, the densest bipartite subgraph detection problem over $\hat{\mathcal{G}}$ corresponds to the setting that $\boldsymbol{x} = [\boldsymbol{y}, \boldsymbol{z}]$, where $\boldsymbol{y} \in \{0,1\}^{|L|}, \boldsymbol{z} \in \{0,1\}^{|R|}$, and $\mathbf{P}$, $\mathbf{Q} \in \mathbb{R}^{n\times n}$,*

$$\mathbf{P} = \begin{bmatrix} \mathbf{D}_{\boldsymbol{c}_L} & \frac{\mathbf{A}}{2} \\ \frac{\mathbf{A}^T}{2} & \mathbf{D}_{\boldsymbol{c}_R} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{D}_{\boldsymbol{c}_L} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_{\boldsymbol{c}_R} \end{bmatrix}, \ \mathbf{Q} = \begin{bmatrix} \mathbf{I}_{|L|} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{|R|} \end{bmatrix} \tag{4}$$

where $\boldsymbol{c}_L$ and $\boldsymbol{c}_R$ are the node weight vectors for the nodesets L and R respectively, $\mathbf{I}_{|L|}$ is the identity matrix of size $|L| \times |L|$, and $\mathbf{I}_{|R|}$ is similar.

To avoid the trivial solution for the weighted graph (single edge with heavy weight), we can introduce column weights as $\mathbf{A} \cdot diag(\frac{1}{h(\mathbf{1}^T\mathbf{A})})$ for some function h, e.g. $h(x) = x^\alpha$ with $\alpha \in \mathbb{R}^+$ or $h(x) = \log(x + c)$ (c is a small constant to prevent the denominator becoming zero). Besides, we can use the motif-based high-order graphs [32] to recognize more complex and interesting dense patterns.

4 Theoretical Analysis

In this section, we connect the optimization of GenDS to the graph spectral theory, showing that we can efficiently approximate the solution by the skewness properties of the spectrum in real-world graphs, thus guide our algorithm design.

Given the graph $\mathcal{G}$ and its contrast $\mathcal{G}'$, we construct a *"positive residual"* graph $\mathcal{G}_r = (V, E_r)$ with $E_r = \{(u,v) | (u,v) \in E \wedge (u,v) \notin E'\}$, and its adjacency matrix is denoted as $\mathbf{A}_r = (\mathbf{P} - \mathbf{Q})^+$. Then the densest subgraph detection in $\mathcal{G}_r$ means that it maximizes the density in $\mathcal{G}$ while minimizes the connection in $\mathcal{G}'$. Thus, the objective function in Eq. (3) is reformulated as

$$S^* = \mathop{\arg\max}_{S \subseteq V, |S| \geq 1} g(S; \mathbf{P}, \mathbf{Q}) = \mathop{\arg\max}_{\boldsymbol{x} \in \{0,1\}^n, |\boldsymbol{x}| \geq 1} \frac{\boldsymbol{x}^T (\mathbf{P} - \mathbf{Q})^+ \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \mathop{\arg\max}_{\boldsymbol{x} \in \{0,1\}^n, |\boldsymbol{x}| \geq 1} \frac{\boldsymbol{x}^T \mathbf{A}_r \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}. \tag{5}$$

We will use this transformation in the following theoretical optimality analysis.

Consider the optimization problem with a similar form as Eq. (5) defined in the real domain, which is formulated in the *Rayleigh quotient* manner, that is

$$R(\mathbf{A}_r, \boldsymbol{x}) = \frac{\boldsymbol{x}^T \mathbf{A}_r \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}, \boldsymbol{x} \in \mathbb{R}^n,\ \boldsymbol{x} \neq \mathbf{0}, \tag{6}$$

where $\mathbf{A}_r \in \mathbb{R}^{n \times n}$ is a symmetric matrix; $R(\mathbf{A}_r, c\boldsymbol{x}) = R(\mathbf{A}_r, \boldsymbol{x})$ for any non-zero scalar c. The objective of GenDS in Eq. (5) is a binary-variable special case.

The Rayleigh–Ritz Theorem [10] in the spectral theory gives the optimality of Eq. (6) with eigenvalues of $\mathbf{A}_r \in \mathbb{R}^{n \times n}$, that is,

Theorem 10 (Rayleigh–Ritz Theorem[3]). *Let $\mathbf{A}_r$ be a symmetric matrix with eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ and corresponding eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n$. Then*

$$\begin{aligned} \lambda_1 &= \max_{\boldsymbol{x} \neq \mathbf{0}} R(\mathbf{A}_r, \boldsymbol{x}) = \max_{\boldsymbol{x} \in \mathbb{R}^n, \|\boldsymbol{x}\| = 1} \boldsymbol{x}^T \mathbf{A}_r \boldsymbol{x} \Longrightarrow \boldsymbol{x} = \boldsymbol{u}_1 \\ \lambda_n &= \min_{\boldsymbol{x} \neq \mathbf{0}} R(\mathbf{A}_r, \boldsymbol{x}) = \min_{\boldsymbol{x} \in \mathbb{R}^n, \|\boldsymbol{x}\| = 1} \boldsymbol{x}^T \mathbf{A}_r \boldsymbol{x} \Longrightarrow \boldsymbol{x} = \boldsymbol{u}_n. \end{aligned} \tag{7}$$

In general, for $1 \leq k \leq n$, let $\mathcal{S}_k$ denote the span of $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ (with $\mathcal{S}_0 = \mathbf{0}$), and let $\mathcal{S}_k^\perp$ denote the orthogonal complement of $\mathcal{S}_k$. Then

$$\lambda_k = \max_{\boldsymbol{x} \neq \mathbf{0},\, \boldsymbol{x} \in \mathcal{S}_{k-1}^\perp} R(\mathbf{A}_r, \boldsymbol{x}) = \max_{\|\boldsymbol{x}\| = 1,\, \boldsymbol{x} \in \mathcal{S}_{k-1}^\perp} \boldsymbol{x}^T \mathbf{A}_r \boldsymbol{x} \Longrightarrow \boldsymbol{x} = \boldsymbol{u}_k, \tag{8}$$

which means λ_k is the largest value of $R(\mathbf{A}_r, \boldsymbol{x})$ over the complement space $\mathcal{S}_{k-1}^\perp$.

With the analogy of eigenvalues and singular values of matrices, the latter achieve the optimality property that resembles those of Rayleigh quotient matrices [11]. To avoid the large magnitude negative eigenvalues for the real graphs [26], here we utilize the singular values and singular vectors instead in the following.

Let $\mathbf{A}_r = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T = \sum_{i=1}^r \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T$ be the singular value decomposition of the matrix $\mathbf{A}_r$, the columns of $\mathbf{U}$ and $\mathbf{V}$ are called the left- and right- singular vectors respectively, i.e., $\mathbf{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_r]$ and $\mathbf{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_r]$. $\mathbf{\Sigma} = diag(\sigma_1, \ldots, \sigma_r)$ for singular values $\sigma_1 \geq \cdots \geq \sigma_r > 0$. Then, we also have the following representation regard to the GenDS problem,

[3] The proof details of the theorem refer to [10].

Lemma 11. *The optimal solution for the* GenDS *in Eq.* (3) *can be written as*

$$S^* = \operatorname*{arg\,max}_{\boldsymbol{x}\in\{0,1\}^n, |\boldsymbol{x}|\geq 1} \frac{\boldsymbol{x}^T\mathbf{A}_r\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = \operatorname*{arg\,max}_{|S|\geq 1} \frac{1}{|S|}\sum_{i=1}^{n}\sigma_i\left(\sum_{j\in S}\boldsymbol{u}_{ij}\right)\left(\sum_{j\in S}\boldsymbol{v}_{ij}\right) \tag{9}$$

where $\boldsymbol{u}_{ij}$ and $\boldsymbol{v}_{ij}$ denote the j-th element of the singular vector $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ corresponding to the singular value σ_i resp. The optimal density value $g_{opt} \leq \sigma_1$.

As for the bipartite graph case, given an asymmetric matrix $\mathbf{A}_r \in \mathbb{R}^{m\times n}$, we define the related quadratic optimization problem as

$$R(\mathbf{A}_r; \boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x}^T\mathbf{A}_r\boldsymbol{y}}{\boldsymbol{x}^T\boldsymbol{x}+\boldsymbol{y}^T\boldsymbol{y}}, \ \boldsymbol{x}\in\mathbb{R}^m, \boldsymbol{y}\in\mathbb{R}^n, \ \boldsymbol{x}\neq\mathbf{0}, \ \boldsymbol{y}\neq\mathbf{0}. \tag{10}$$

And we also obtain the following theorem that leads to a similar statement as Theorem 10. Thus, it helps to avoid constructing the big matrix ($\mathbb{R}^{(m+n)\times(m+n)}$) for the bipartite graph. The detailed proof is given in the supplement.

Theorem 12 (Bigraph Spectral). *Suppose $\mathbf{A}_r$ is an $m\times n$ matrix, $\mathbf{A}_r = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ is its singular value decomposition. For any vector $\boldsymbol{x}\in\mathbb{R}^m, \boldsymbol{y}\in\mathbb{R}^n$,*

$$\sigma_1 = \max_{\|\boldsymbol{x}\|=\|\boldsymbol{y}\|=1} \boldsymbol{x}^T\mathbf{A}_r\boldsymbol{y} \geq \max_{\boldsymbol{x}\neq\mathbf{0},\boldsymbol{y}\neq\mathbf{0}} 2\cdot R(\mathbf{A}_r,\boldsymbol{x},\boldsymbol{y}) \Longrightarrow \begin{matrix}\boldsymbol{x}=\boldsymbol{u}_1\\ \boldsymbol{y}=\boldsymbol{v}_1\end{matrix}. \tag{11}$$

In general, for $1\leq k\leq r$, let $\mathcal{S}_k^U$, $\mathcal{S}_k^V$ denote the span of $\boldsymbol{u}_1,\ldots,\boldsymbol{u}_k$ and $\boldsymbol{v}_1,\ldots,\boldsymbol{v}_k$ (with $\mathcal{S}_0^U=\mathbf{0}, \mathcal{S}_0^V=\mathbf{0}$), then

$$\sigma_k = \max_{\substack{\|\boldsymbol{x}\|=\|\boldsymbol{y}\|=1\\ \boldsymbol{x}\perp\mathcal{S}_{k-1}^U,\, \boldsymbol{y}\perp\mathcal{S}_{k-1}^V}} \boldsymbol{x}^T\mathbf{A}_r\boldsymbol{y} \geq \max_{\substack{\boldsymbol{x}\neq\mathbf{0},\boldsymbol{y}\neq\mathbf{0}\\ \boldsymbol{x}\perp\mathcal{S}_{k-1}^U,\, \boldsymbol{y}\perp\mathcal{S}_{k-1}^V}} 2\cdot R(\mathbf{A}_r,\boldsymbol{x},\boldsymbol{y}) \Longrightarrow \begin{matrix}\boldsymbol{x}=\boldsymbol{u}_k\\ \boldsymbol{y}=\boldsymbol{v}_k\end{matrix}.$$

Therefore, given a bipartite graph $\hat{\mathcal{G}} = (L\cup R, E)$ with the adjacency matrix $\mathbf{A}\in\mathbb{R}^{|L|\times|R|}$, we will have the similar properties as Lemma 11 as

Lemma 13. *For the densest bipartite subgraph detection in Fraudar with $\mathbf{P} = diag(\left[\mathbf{A}/2, \mathbf{A}^T/2\right])$ and $\boldsymbol{x}^T\mathbf{P}\boldsymbol{x} = |E(S)|$, the optimal solution can be written as*

$$\begin{aligned} S^* &= \operatorname*{arg\,max}_{\boldsymbol{x}\in\{0,1\}^n,|\boldsymbol{x}|\geq 1} \frac{\boldsymbol{x}^T\mathbf{P}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} = \operatorname*{arg\,max}_{\boldsymbol{y}\in\{0,1\}^{|L|},\, \boldsymbol{z}\in\{0,1\}^{|R|}, |\boldsymbol{y}|>0, |\boldsymbol{z}|>0} R(\mathbf{A}_r,\boldsymbol{y},\boldsymbol{z}) \\ &\leq \operatorname*{arg\,max}_{S=\delta(\boldsymbol{y})\cup\delta(\boldsymbol{z}),\, |S|\geq 1} \frac{1}{|S|}\sum_{i=1}\sigma_i\left(\sum_{j\in\delta(\boldsymbol{y})}\boldsymbol{u}_{ij}\right)\left(\sum_{j\in\delta(\boldsymbol{z})}\boldsymbol{v}_{ij}\right), \end{aligned} \tag{12}$$

where $\boldsymbol{u}_{ij}, \boldsymbol{v}_{ij}$ denote the j-th element of the singular vector $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, and the optimal density value $g_{opt}\leq\sigma_1$.

Moreover, if the matrix $\mathbf{Q}$ is positive definite (i.e., $\boldsymbol{x}\mathbf{Q}\boldsymbol{x}^T > 0$ for any $\boldsymbol{x} \neq \mathbf{0}$) in GenDS, the Eq. (3) under the relaxation $\boldsymbol{x} \in \mathbb{R}^n$ is equivalent to the *generalized Rayleigh quotient*, its optimization reduces to the *generalized eigenvalue decomposition* problem; the *min-max principle* provides result about the optimality similar to Theorem 10. Due to the singularity of $\mathbf{Q}$ in the real scenario, we take the residual graph form $\mathcal{G}_r$ for approximation as discussed above.

Real-world Graph Properties. The sparsity and various power-laws are key components of the real-world networks gathered from the world-wide-web, social networks, E-commerce, on-line reviews, recommend systems, and more. Those primary properties contribute to the time and space-efficient computing or storage, and synthetically modeling the realistic networks. Various studies [12, 17] have shown that most real-world graphs have a statistically significant power-law distribution with degree distribution, the distribution of "bipartite cores" ($\approx$ communities), a cutoff in the eigenvalue or singular values of the adjacency matrix and the Laplacian matrix, etc. Also, the distribution of eigenvector elements (indicators of "network value") associated with the top-ranked eigenvalues of the graph adjacency matrix is skewed [6].

Thus, based on the spectral formulation of GenDS, the skewness of singular values and components in singular vectors of real-world graphs guarantees that we can simply consider the top singular vectors and use a few of top-rank elements in them to efficiently construct the candidates for dense subgraphs and detect the optimal result, We will introduce this in more details in the following algorithm.

5 Algorithms & Complexity Analysis

In this section, we present our proposed method SpecGreedy for the generalized densest subgraph detection problem GenDS and provide analysis for its property.

We first review the related Charikar's peeling algorithm. It takes the entire original graph as the starting point, then greedily removes the node with the smallest degree from the graph, and returns the densest one among the shrinking sequence of subgraphs created by the procedure. It is guaranteed to return a solution of at least half of the optimum density, i.e., $g^* \geq \frac{1}{2} g_{opt}$. In addition, using the priority tree to manage the nodes in the peeling process, the complexity of the greedy algorithm is $O(|E| \log |V|)$.

However, the densest subgraphs usually have small sizes and are embedded in a large graph (background), which leads to many searches and update steps to obtain an approximation solution or even the candidates for Charikar's algorithm.

Implications of Theoretical Analysis: Lemma 11 and 13 show the upper bound of the optimal density, i.e., $g_{opt} \leq \sigma_1$, and the σ_k is the optimal value for the real space orthogonal to $\mathcal{S}_{k-1}$ $(k > 1)$ as Theorem 10 and 12; the formulation of S^* highlights that the real-value singular vectors provide some insight to find the optimal densest subgraph. Thus, these nodes in S^* will have higher importance in the singular vectors associating with the top-ranked singular values.

Considering the skewed distribution of the elements in a singular vector, we can construct some small nodeset candidates, which derive some subgraphs,

Algorithm 1 SpecGreedy: General dense subgraph detection

```
Input: Matrix $\mathbf{A}_r$ of the positive residual $\mathcal{G}_r$; density metric $g$; top approx. rank $k$.
Output: The densest subgraph.
 1: $S = \emptyset$
 2: $[\mathbf{U}, \Sigma, \mathbf{V}] = \mathbf{SVD}(\mathbf{A}_r, k)$          ▷ Top-$k$ spectral decomposition of $\mathbf{A}_r$
 3: for $r \leftarrow 1, \ldots, k$ do
 4:     Construct the candidate node subset $S_r$ based on $\boldsymbol{u}_r$ and $\boldsymbol{v}_r$, i.e.
        $S_r = \{i : \boldsymbol{u}_{ri} > \frac{1}{\sqrt{|L|}}, i \in L\} \cup \{j : \boldsymbol{v}_{rj} > \frac{1}{\sqrt{|R|}}, j \in R\}$
 5:     $S_r^* \leftarrow$ Greedy($\mathcal{G}(S_r)$, $g$)   ▷ Greedily remove nodes to maximize the metric $g$.
 6:     if $g(S_r^*) > g(S)$ then                          ▷ $g(S) = g_{cur}^*$
 7:         $S \leftarrow S_r^*$
 8:     if $g(S) > \sigma_{r+1}$ then                       ▷ Spectral early-stopping condition
 9:         break
10: return $\mathcal{G}(S)$.
```

with the top-ranked nodes based on the singular vectors to avoid detecting the densest subgraph from the whole graph, that is, $S_C = \{S_1, \ldots, S_k\}$ for some $1 \leq k < n$, where the candidate $S_i = \{j; \boldsymbol{u}_{ij} > \Delta_L, j \in \lceil |L| \rceil\} \cup \{j; \boldsymbol{v}_{ij} > \Delta_R, j \in \lceil |R| \rceil\}$ for the singular vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$, Δ_L and Δ_R are some pre-defined truncation thresholds; the optimal density for $\mathcal{G}(S_i)$ is $g_i \leq \sigma_i$. Here we determine the selection thresholds as $\Delta_L = 1/\sqrt{|L|}$ and $\Delta_R = 1/\sqrt{|R|}$ [4] based on the re-formulation of the optimal solution in the Eq. (9) and Eq. (12).

Proposed Algorithm. Therefore, we propose SpecGreedy, which utilizes graph spectral properties and the greedy peeling strategy to solve the GenDS problem. Algorithm 1 summarizes our approach.

Given the adjacency matrix $\mathbf{A}_r$ of the positive residual graph $\mathcal{G}_r$, density metric g, and the top approximation rank k which controls the maximum size of the candidate set. SpecGreedy finds the top-k spectral decomposition of the matrix at first (Line 2) , then detects the possible densest subgraphs based on the top singular vectors. In each round, it constructs the candidate subset S_r based on the truncated singular vectors $\boldsymbol{u}_r$ and $\boldsymbol{v}_r$, then uses the *greedy algorithm* to search the densest subgraph for $\mathcal{G}(S_r)$ to maximize the density metric g. It checks the stop condition based on the next singular value for the current optimal result in Line 8 for early stopping.

How many subgraph candidates do we need to check? Let g_{cur}^* be the current detected optimal density with some off-the-shelf detection approaches, if there is some $1 < j \leq k$ satisfied that $g_{cur}^* \geq \sigma_j$, the optimal density then can be achieved based on the singular vectors is g_{cur}^* due to the decreasing-order of singular values ($\sigma_j > \sigma_{j+1}$) and the aforementioned upper-bound ($g_i \leq \sigma_i$). Finally, the subgraph with the optimal density is returned. It is worth mentioning that the power-law distribution nature of the eigenvalues and singular values of real-world graphs and the theoretical bounds of solutions (the exact or 1/2-approx. result) for detection approaches guarantee that the size of candidates will be very small.

[4] If $\mathbf{A}_r$ is the symmetric matrix as in Eq. (9), $|L| = |R| = n$ and $\Delta_L = \Delta_R = 1/\sqrt{n}$.

Besides the pre-computing top-k spectral decomposition strategy in Line 2, we can use a lazy or online way to compute the $(r+1)$-th largest spectral decomposition result with the power method or the efficient Krylov subspace methods such as the Lanczos method [14]. In the experiment, we adopt an incremental decomposition way which gets the top-l singular values and singular vectors first, and if the stop condition in Line 8 is not satisfied, then get the further top-$(l+s)$ singular values and vectors with step-size s. This stepwise increasing-decomposition will continue until $l+s \geq k$ or the early-stopping condition holds. Moreover, we can use other densest subgraph detection approaches in Line 5 considering the enhancement of solution, e.g. Greedy++ [5] or the LP method.

Theorem 14 (Time Complexity). *The complexity of* SpecGreedy *algorithm is $O(K \cdot |E| + K \cdot |E(\tilde{S})| \log |\tilde{S}|)$ where $\tilde{S} = \max_{|S_i|} S_i$ and K is the top approximation rank.*

Ideally, $K = \min\{k, r_{opt}+1\}$ where k is the input parameter and r_{opt} is the rank with optimal resultant density g^*. The complexity of computing a top eigenvector/singular vector in sparse graphs is linear, i.e., $O(|E(V)|)$, and the total complexity of the greedy algorithm in Line 5 is $O(|E(S)| \log |S|)$ for $\mathcal{G}(S)$. Given the skewness of the top singular vectors in real-world graphs, we usually have $|\tilde{S}| \ll |V|$, making SpecGreedy a linear algorithm in the number of edges.

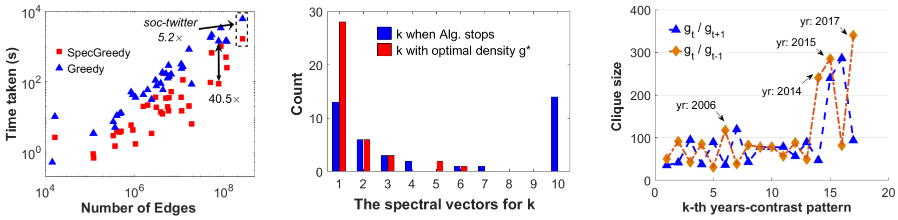
6 Experiments

We design experiments to answer the following questions:

1. **Q1. Efficiency:** How does our SpecGreedy compare to the state-of-the-art greedy algorithm for detecting the densest subgraph?
2. **Q2. Effectiveness:** How well does SpecGreedy work on real data, and perform on detecting the contrast dense subgraph and injected subgraphs?
3. **Q3. Scalability:** How does our method scale with the input graph size?

Data: We used a variety of datasets (40 in total) obtained from 5 popular network repositories, including 32 monopartite graphs and 8 bipartite ones, and 5 of them also have edge weights; the largest unweighted graph is the `soc-twitter` graph with roughly $1.47B$ edges, while the smallest unweighted graph has roughly $14.5K$ edges. Multiple edges, self-loops are removed, and directionality is ignored for directed graphs. The detailed information about those real-world networks is provided in the supplement.

Implementations: We implemented efficient dense subgraph detection algorithms for comparison. We implemented our algorithm, Greedy [7], SpokEn [24], and Fraudar [15] in Python; SpokEn actually detects the densest subgraph only based on the truncation of the singular vectors like our method. In all the experiments, we set the parameter of top approximation rank $k = 10$ and $l = s = 3$ for SpecGreedy. We ran all experiments on a machine with 2.4GHz Intel(R) Xeon(R) CPU, 64GB of main memory.



(a) The densest subgraph detection for all graphs.

(b) The statistic of spectral vectors for k.

(c) Contrast patterns for the DBLP co-authorship graphs.

Fig. 2. The performance of SpecGreedy for the real-world graphs. (a) SpecGreedy runs faster than Greedy in all graphs for detecting the densest subgraph with the same or comparable density, achieves 58.6× speedup for *ca-DBLP2012* and about 3× for the largest graph *soc-twitter*. (b) The statistic information about k for spectral vectors. The densest subgraphs with optimal density $g*$ are achieved in the first singular vector for most of the datasets. The blue bars show the statistics of k when algorithm stops given the parameter $k = 10$. (c) The contrast patterns for DBLP co-authorship data in $2000 - 2017$ with the positive residual $\mathcal{G}_r$ (very large cliques in 2017, 2015, and 2014).

6.1 Q1. Efficiency

To answer Q1, we apply our method SpecGreedy and the baseline Greedy on 40 unweighted networks and compare their runtime.

Fig.1(a) shows the statistical information about the runtime improvement ratio of SpecGreedy compared with the Greedy algorithm for detecting the densest subgraphs; Fig.2(a) illustrates more detailed information about the time taken of the two methods: for each network dataset, it provides the runtime of the two methods and the network size.

Observation: Our method runs faster than Greedy and achieves the same or comparable optimal densities as shown in Fig.1(b). Among these varied-size datasets, SpecGreedy achieves 3.0-5.0× speedup for 17 of them, 1.5-3.0× for 8, and 5.0-7.0× for 7 graphs, and more than 58.6× for the `ca-DBLP2012` graph. As we can see, SpecGreedy is efficient for large graphs, e.g. 30× for `ca-DBLP-NET`, 25× for `cit-Patents`, and 3× speedup for `soc-twitter`.

For the 5 weighted graphs, we observe similar results as above. SpecGreedy achieves 24-39× speedup for 3 of them and 11-17× for the rest. Greedy will have poor performance for the graph dominated by few edges with heavy weights due to it needs to peel each edge of the whole graph.

Fig. 2(b) summarizes the statistics about spectral vectors k for obtaining the optimal density g^* and actual k when the algorithm stops. Larger k means taking more time for SVD and detection candidate subgraphs. We can see that the densest subgraphs with optimal density g^* are achieved in the first spectral vector for most of the datasets, the second one for 6 of the graphs, and only 3 graphs need to check more than 5 singular vectors. There are 26 graphs where SpecGreedy stops for the early-stopping condition, while the rest need to check all 10 singular vectors due to the small optimal density or flat power-law factor of singular values. Besides, we find that some subgraphs detected based on the top $k-1$ vectors also cliques with a smaller size than the optimal one. So, the above heuristic observation and the power-law distribution of singular values contribute to the efficiency of SpecGreedy, and the small k is enough for good results.

6.2 Q2. Effectiveness

In this section, we verify that SpecGreedy detects high-quality densest subgraphs in real-world graphs and accurately detects injected subgraphs with different injection density. Moreover, focusing on a large-scale collaboration network, we show that SpecGreedy also finds significant contrast dense subgraphs.

Density Improvement. Following the setup we described in Q1, Fig. 1(b) shows the improvement ratio of optimal densities found by SpecGreedy compared to the Greedy and SpokEn algorithm. As we can see, SpecGreedy consistently outperforms SpokEn by detecting denser densest subgraphs for all real-world datasets. It even achieves more than 28.3× higher density for the `soc-twitter` graph, Also, SpecGreedy obtains the same or denser (more than 1.26×) optimal density for most graphs compared with Greedy; there are 4 graphs that the optimal densities detected by SpecGreedy have less than but very close ($\geq 0.996\times$) densities as detected by Greedy, and 2 graphs with less than 0.9 density improvement. So, utilizing the spectral distribution of the densest subgraph, SpecGreedy can improve the quality of solution of Greedy in most cases due to avoid arbitrary ties-break in graphs for removing in Greedy to some extent.

Injection Detection. We further evaluate the performance of SpecGreedy by performing a synthetic experiment where we inject dense subgraphs as ground truth. For a more realistic setting, we also added extra edges as 'camouflage' between the nodes in the selected injection subgraph and the remaining unselected nodes. We compared SpecGreedy, Greedy and SpokEn in terms of F measure in detecting the injected patterns, and reports the averaged F-score over 5 trials. Specifically, we injected a 600×600 subgraph with different injection densities to an amazon-Art review subgraph of size $4K \times 4K$, and we select the two different cases with background densities 2.7E-5 and 3.4E-5 for comparison. From the result, we observe that SpecGreedy achieves equally high accuracy as Greedy and is better than SpokEn, the detailed figures are provided in the supplement.

Case study. As a case study, we also apply SpecGreedy on the DBLP co-authorship data [28] from 2000 to 2017 to identify interesting contrast dense patterns. Fig. 2(c) shows the contrast dense subgraphs pattern detected by SpecGreedy with constructing the positive residual graphs $\mathcal{G}_r$. Those densest contrast subgraphs are all cliques of different sizes, which means the connections that form a clique only appear in $\mathcal{G}_t$ rather than $\mathcal{G}_{t-1}$ (or $\mathcal{G}_{t+1}$). As we can see, there are 3 extremely large cliques for 2017, 2015, and 2014, related to the publications in 'Brain network and Disease', 'Neurology and Medicine', and 'Physics' from some large collaborative groups of different disciplines.

6.3 Q3. Scalability

Figure 1(c) shows the linear scaling of SpecGreedy's running time in the number of edges of the graph. Here we used the `ca-Patents-AM` graph and randomly subsampled different proportions of the edges from it for detecting the densest subgraph. The slope parallel to the main diagonal indicates linear growth.

7 Conclusions

In this paper, we propose the generalized densest subgraph detection, GenDS, which unifies several well-known instances of related problems. We devise the SpecGreedy algorithm to solve the generalized problem based on graph spectral properties and a greedy peeling approach. Our main contributions are as follows.

- **Theory & Correspondences:** We propose the unified formulation for the densest subgraph detection from different applications, and analyze our proposed optimization problem by leveraging spectral theory.
- **Algorithm:** We devise a fast algorithm, SpecGreedy, to solve the GenDS.
- **Experiments:** The efficiency of SpecGreedy is verified on 40 real-world graphs. SpecGreedy runs linearly with the graph size and is effective in applications, like finding sudden bursts in research co-authorship relationships.

The quality guaranteed detection algorithm design and streaming graphs adaptation are also possible extension directions for this work.

Acknowledgments

This work was upported by the Strategic Priority Research Program of Chinese Academy of Sciences, Grant No. XDA19020400, NSF of China No. 61772498, U1911401, 61872206, 91746301, National Science Foundation under Grant No. IIS 1845491, and Army Young Investigator Award No. W911NF1810397.

References

1. L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description:A survey. *Data Min. Knowl. Discov.*, 29(3):626–688, May 2015.
2. R. Andersen and K. Chellapilla. Finding dense subgraphs with size bounds. In *WAW*, 2009.
3. R. Andersen and S. M. Cioaba. Spectral densest subgraph and independence number of a graph. *J. UCS*, 13(11):1501–1513, 2007.
4. Y. Asahiro, K. Iwama, H. Tamaki, and T. Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000.
5. D. Boob, Y. Gao, R. Peng, S. Sawlani, C. E. Tsourakakis, D. Wang, and J. Wang. Flowless: Extracting densest subgraphs without flow computations. In *WWW'20.*

6. D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-mat: A recursive model for graph mining. In *SDM*, pages 442–446. SIAM, 2004.
7. M. Charikar. Greedy approximation algorithms for finding dense components in a graph. In *APPROX*, 2000.
8. J. Chen and Y. Saad. Dense subgraph extraction with application to community detection. *IEEE TKDE*, 2010.
9. L. Chu, Z. Wang, J. Pei, J. Wang, Z. Zhao, and E. Chen. Finding gangs in war from signed networks. In *KDD*, pages 1505–1514. ACM, 2016.
10. Fan. R. K. Chung. *Spectral Graph Theory. American Mathematical Soc.*, 1996.
11. A. Dax. From eigenvalues to singular values: a review. *APM*, 2013.
12. N. Eikmeier and D. F. Gleich. Revisiting power-law distributions in spectra of real world networks. In *KDD*, pages 817–826, 2017.
13. A. V. Goldberg. Finding a maximum density subgraph. *UCB*, 1984.
14. G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3. JHU press, 2012.
15. B. Hooi, H. A. Song, A. Beutel, N. Shah, K. Shin, and C. Faloutsos. Fraudar: Bounding graph fraud in the face of camouflage. In *SIGKDD*, pages 895–904, 2016.
16. V. E. Lee, N. Ruan, R. Jin, and C. C. Aggarwal. A survey of algorithms for dense subgraph discovery. In *Managing and Mining Graph Data*, 2010.
17. J. Leskovec, D. Chakrabarti, J. Kleinberg, C. Faloutsos, and Z. Ghahramani. Kronecker graphs: An approach to modeling networks. *JMLR*, 11, 2010.
18. Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, and L. Chen. Erratum: Quantitative function for community detection. *Physical Review E*, 91(1):019901, 2015.
19. S. Liu, B. Hooi, and C. Faloutsos. A contrast metric for fraud detection in rich graphs. *TKDE*, 31(12):2235–2248, 2018.
20. Y. Liu, L. Zhu, P. A. Szekely, A. Galstyan, and D. Koutra. Coupled clustering oftime-series and networks. In *SDM*, pages 531–539. SIAM, 2019.
21. A. Miyauchi and N. Kakimura. Finding a dense subgraph with sparse cut. In *CIKM*, 2018.
22. D. Papailiopoulos, I. Mitliagkas, A. Dimakis, and C. Caramanis. Finding dense subgraphs via low-rank bilinear optimization. In *ICML*, pages 1890–1898, 2014.
23. M. Pavan and M. Pelillo. Dominant sets and pairwise clustering. *IEEE Trans, on pattern analysis and machine intelligence*, 29(1):167–172, 2006.
24. B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *PAKDD*, 2010.
25. H.-W. Shen and X.-Q. Cheng. Spectral methods for the detection of network community structure: a comparative analysis. *JSTAT*, 2010(10):P10020, 2010.
26. C. E. Tsourakakis. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *ICDM*, pages 608–617. IEEE, 2008.
27. C. E. Tsourakakis, T. Chen, N. Kakimura, and J. W. Pachocki. Novel dense subgraph discovery primitives: Risk aversion and exclusion queries. *ECML-PKDD'19*
28. H. Wan, Y. Zhang, J. Zhang, and J. Tang. Aminer: Search and mining of academic social networks. *Data Intelligence*, 1(1):58–76, 2019.
29. Z. Wang, L. Chu, J. Pei, A. Al-Barakati, and E. Chen. Tradeoffs between density and size in extracting dense subgraphs: A unified framework. In *ASONAM*, 2016.
30. S. W. Wong, C. Pastrello, M. Kotlyar, C. Faloutsos, and I. Jurisica. Sdregion: Fast spotting of changing communities in biological networks. In *SIGKDD*, 2018.
31. Y. Yang, L. Chu, Y. Zhang, Z. Wang, J. Pei, and E. Chen. Mining density contrast subgraphs. In *ICDE*, pages 221–232. IEEE, 2018.
32. H. Yin, A. R. Benson, J. Leskovec, and D. F. Gleich. Local higher-order graph clustering. In *KDD*, pages 555–564, 2017.