Mining Large Quasi-cliques with Quality Guarantees from Vertex Neighborhoods

Aritra Konar
University of Virginia
Charlottesville, Virginia, USA
aritra@virginia.edu

Nicholas D. Sidiropoulos
University of Virginia
Charlottesville, Virginia, USA
nikos@virginia.edu

ABSTRACT

Mining dense subgraphs is an important primitive across a spectrum of graph-mining tasks. In this work, we formally establish that two recurring characteristics of real-world graphs, namely heavy-tailed degree distributions and large clustering coefficients, imply the existence of substantially large vertex neighborhoods with high edge-density. This observation suggests a very simple approach for extracting large quasi-cliques: simply scan the vertex neighborhoods, compute the clustering coefficient of each vertex, and output the best such subgraph. The implementation of such a method requires counting the triangles in a graph, which is a well-studied problem in graph mining. When empirically tested across a number of real-world graphs, this approach reveals a surprise: vertex neighborhoods include maximal cliques of non-trivial sizes, and the density of the best neighborhood often compares favorably to subgraphs produced by dedicated algorithms for maximizing subgraph density. For graphs with small clustering coefficients, we demonstrate that small vertex neighborhoods can be refined using a local-search method to "grow" larger cliques and near-cliques. Our results indicate that contrary to worst-case theoretical results, mining cliques and quasi-cliques of non-trivial sizes from real-world graphs is often not a difficult problem, and provides motivation for further work geared towards a better explanation of these empirical successes.

CCS CONCEPTS

• **Mathematics of computing → Discrete mathematics**; **Graph theory**; *Combinatorics*;

KEYWORDS

Quasi-cliques; clustering coefficients; triangles; neighborhoods

ACM Reference Format:
Aritra Konar and Nicholas D. Sidiropoulos. 2020. Mining Large Quasi-cliques with Quality Guarantees from Vertex Neighborhoods. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3394486.3403100

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403100

1 INTRODUCTION

Motivation and Overview: The task of extracting dense subgraphs from a given graph constitutes a key primitive in graph mining, with applications ranging from graph compression [8], to discovering protein complexes in protein-protein interaction networks [4, 28], to identifying spam farms in Web graphs [18, 22], and event detection in network streams [3, 9].

Depending on the particular metric employed for quantifying subgraph density, various formulations have been proposed for extracting different classes of dense subgraphs. The archetypal dense subgraph is a clique, i.e., a subgraph where every pair of vertices share an edge. A clique is said to be maximal if it isn't included within a larger clique, and the largest such clique is the maximum clique of a graph. The set of all maximal cliques in a graph can be listed using the classic Bron-Kerbosch algorithm [7], albeit at exponential worst-case complexity. Meanwhile, the problem of extracting the maximum clique is NP–hard [17]– even for power-law graphs [16].

Consequently, a different line of work has focused on developing less stringent, polynomial-time formulations for mining dense subgraphs. The seminal work of Goldberg [20] established that the problem of finding the subgraph with maximum average degree (widely known as the DensestSubgraph problem) can be solved via a sequence of maximum-flow problems. Follow-up work by Charikar [10] showed that a simple greedy vertex-peeling algorithm, that runs in linear-time, provides a $1/2$ approximation for the problem and is near-optimal in practice. However, it was pointed out in [32] that adopting such a metric in practice can potentially yield the entire graph as the densest subgraph. As a result, Tsourakakis [31] introduced the more general problem of finding the subgraph which maximizes the average number of induced k-cliques (known as the k-CliqueDensestSubgraph problem), and provided exact flow-based algorithms and greedy approximation algorithms for the task. It was also shown that this approach yields smaller, denser subgraphs compared to DensestSubgraph.

Another line of work utilizes a different relaxation of the notion of a clique, known as quasi-cliques, to find dense subgraphs. Formally, a α-quasi-clique is a subgraph with edges greater than a fixed fraction $\alpha \in (0, 1)$ of the edges in a clique of the same size. Recently, Tsourakakis *et al.* introduced the OptimalQuasi-Clique (OQC) formulation in [32] for mining quasi-cliques possessing a large number of edges with respect to a random null model. The OQC problem is not known to be NP–hard; however, to the best of our knowledge, it does not admit an exact solution in polynomial-time either. Tsourakakis *et al.* [32] proposed a simple greedy vertex-peeling algorithm (GreedyOQC) and a local-search method (LocalSearchOQC) for extracting approximate solutions

for the problem, and demonstrated that they can work well in practice. Later, Cadena *et al.* [9] applied semidefinite relaxation (SDR) [25] to the problem, and provided sufficient conditions under which SDR can guarantee a high-quality approximate solution. However, the high complexity incurred in solving the semidefinite program is a limitation of the approach.

Approach and Contributions: In this paper, we study the general problem of mining dense subgraphs from undirected graphs. In contrast to the prevailing approaches outlined above, we advocate a very simple method which can be summarized as follows: visit every vertex in the graph, compute the edge-density of the subgraph induced by its one-hop neighbors, and output the "best" (in a certain sense). This simply entails computing the local clustering coefficient [27] of every vertex, which can be accomplished by enumerating all triangles in the graph – a task for which there exist several efficient algorithms [23, 33].

While the approach may seem *apriori* naive (it only considers one-hop neighborhoods), we provide theoretical justification for it by establishing the following result: if a graph possesses a large global clustering coefficient [35] and a heavy-tailed degree distribution [5] (two recurring traits of real-world networks [15, 35]), then it includes large and dense vertex neighborhoods. Our work is motivated by the result of [19], which established that the aforementioned properties of real-world networks imply that neighborhood subgraphs form communities with low conductance scores. However, to the best of our knowledge, the question of whether these properties also imply that vertex neighborhoods themselves constitute large and dense subgraphs (in the sense of being quasi-cliques) has not been studied prior to our present work. More specifically, our result differs from that of [19] in the following aspects.

- The authors of [19] use a probabilistic existence argument to show that high global clustering coefficients and power-law degree distributions imply that there exists a vertex neighborhood with *low conductance*. While we utilize the same probabilistic argument and the same twin graph characteristics, our result formally shows the existence of neighborhoods of non-trivial sizes possessing *high edge-density*, which is a very different metric than conductance, and necessitates a different line of analysis compared to that used in [19].
- In [19], it is also shown that the aforementioned properties of a graph imply the existence of a k-core[1], which is a particular type of dense subgraph. Here, we restrict our attention to vertex neighborhoods, and adopt the edge-density of a subgraph as our notion of density. In general, these two notions of density are not directly comparable. Moreover, the result of [19] relies on an argument that requires the graph to grow asymptotically in size. In contrast, we provide a non-asymptotic analysis to establish our result, albeit at the expense of making an explicit assumption on the power-law exponent of the degree distribution.

It has further been shown [21] that irrespective of the degree distribution, graphs with high global clustering coefficients admit a decomposition as a union of vertex disjoint subgraphs, each of which is guaranteed to possess a certain minimum edge and triangle density. We point out that where neighborhoods are concerned, high edge and triangle density are necessary, but not sufficient to guarantee the presence of dense neighborhoods of non-trivial sizes. As a counter-example, consider a graph which is a union of disjoint 4-cliques. In this case, the global clustering coefficient is the maximum possible value 1, and each vertex neighborhood is simply a triangle, which also attains maximum edge and triangle density. To rule out such unfavorable cases, we employ the power-law degree assumption, which is commonly observed in many real-world networks.

In order to test our hypothesis regarding the existence of such large neighborhood subgraphs with high edge-density, and to gauge the empirical efficacy of our approach, we carried out a series of experiments on 15 different publicly available datasets, with the GreedyOQC algorithm of [32] and the sophisticated maximum flow-based algorithm of [26] for computing the triangle-densest subgraph [31] used as benchmarks. We point out that these baselines are not neighborhood based, and constitute dedicated algorithms for dense subgraph discovery. Our main empirical findings can be summarized as follows:

- For graphs which obey our sufficient conditions, we discovered that neighborhoods can surprisingly form *maximal cliques* and quasi-cliques of non-trivial sizes. Furthermore, the quality of these neighborhood subgraphs is comparable, or even better compared to the baselines. While these results validate the essence of our theoretical argument, they also reveal the conservative nature of our analysis, as we obtain better results in practice.
- For graphs with low global clustering coefficients, neighborhoods with high local clustering coefficients can be small in size. However, we demonstrate that they can serve as good seed sets for a local-search algorithm proposed in [32]. We provide empirical justification for our choice by demonstrating that it is consistently better in terms of size and edge-density compared to subgraphs obtained via other simple seeding strategies such as the core decomposition and selecting neighborhoods with high average degree. Refining our neighborhoods via this algorithm allows us to obtain cliques and near-cliques of even better quality compared to the baselines.

Finally, we note that, while the scope of our algorithmic contributions is limited, our main purpose is to highlight the fact that substantially large dense neighborhoods exist in real-world graphs. On the theoretical side, we provide practical sufficient conditions on the graph characteristics (in terms of power-law degree distributions and clustering coefficients) to quantify the existence of such large, dense neighborhoods. On the practical side, via extensive experiments, we verify that such neighborhoods are of comparable, or even better quality, compared to a range of baselines, and when refined using a local search algorithm they yield state-of-the-art results. Our findings suggest that contrary to worst-case complexity results [13, 16, 17], it is possible to extract large cliques and near-cliques from real-world graphs using a *very simple approach* – and this is quite remarkable.

[1] A k-core is the maximal subgraph of a graph where every vertex is connected to at least k other vertices.

2 PRELIMINARIES

Given a simple, unweighted, undirected graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ on n vertices, the *neighborhood* of a vertex $v \in \mathcal{V}$ is the subset of vertices $\mathcal{N}_v \subseteq \mathcal{V}$ that share an edge with v. This can be expressed as

$$\mathcal{N}_v := \{u \in \mathcal{V} : (u, v) \in \mathcal{E}\}, \forall\, v \in \mathcal{V}. \tag{1}$$

The degree of vertex $v \in \mathcal{V}$ is $d_v := |\mathcal{N}_v|$. A *wedge* is a path of length 2 formed by an unordered pair of edges $\{(s, v), (v, t)\}$ that share a common vertex v. A wedge is said to be *closed* if its end points (s, t) are connected by an edge. Let $w_v := \binom{d_v}{2}$ denote the number of wedges centered at vertex v and $w_v^{(c)}$ denote the corresponding number of closed wedges. The *local clustering coefficient* of v is then the fraction of wedges centered at v that are closed, i.e.,

$$C_v := \frac{w_v^{(c)}}{w_v}, \forall\, v \in \mathcal{V}. \tag{2}$$

Let $w := \sum_{v \in \mathcal{V}} w_v$ be the total number of wedges in $\mathcal{G}$. The *global clustering coefficient* of $\mathcal{G}$ is the overall fraction of wedges in $\mathcal{G}$ that are closed, i.e.,

$$C_g := \frac{1}{w} \sum_{v \in \mathcal{V}} w_v^{(c)}. \tag{3}$$

Define a probability mass function p on the vertices of $\mathcal{G}$ that assigns each vertex $v \in \mathcal{V}$ a probability equal to the fraction of overall wedges centered at v, i.e.,

$$p_v := \frac{w_v}{w}, \forall\, v \in \mathcal{V}. \tag{4}$$

It is known [19, Claim 4.2] that the above twin definitions of clustering coefficients obey the following relation with respect to (w.r.t.) the distribution p.

$$\mathbb{E}_p[C_v] = C_g \tag{5}$$

Given a subset of vertices $\mathcal{S} \subseteq \mathcal{V}$, define $\mathcal{E}(\mathcal{S})$ as the subset of $\mathcal{E}$ containing edges only between the vertices in $\mathcal{S}$. For the subgraph $\mathcal{G}_\mathcal{S} := (\mathcal{S}, \mathcal{E}(\mathcal{S}))$ induced by $\mathcal{S}$, let $e(\mathcal{S}) := |\mathcal{E}(\mathcal{S})|$ denote the number of edges in $\mathcal{G}_\mathcal{S}$. The density of a subgraph is measured via its edge-density

$$\delta(\mathcal{S}) := \frac{e(\mathcal{S})}{\binom{|\mathcal{S}|}{2}}, \tag{6}$$

which quantifies how closely $\mathcal{G}_\mathcal{S}$ resembles a clique on $|\mathcal{S}|$ vertices in terms of edges, i.e., $\delta(\mathcal{S}) = 1$ when $\mathcal{S}$ is a clique. Given a parameter $\alpha \in (0, 1)$, a subgraph $\mathcal{G}_\mathcal{S}$ is said to be a $\alpha-$quasi-clique if $\delta(\mathcal{S}) \geq \alpha$, i.e., if the number of its edges is at least as large as a fixed fraction α of the edges in a clique on $|\mathcal{S}|$ vertices.

3 VERTEX NEIGHBORHOODS AS DENSE SUBGRAPHS

In this section, we analyze whether vertex neighborhoods themselves can be potential candidates for dense subgraphs in real-world graphs. Our starting point is the following simple observation which states that the edge-density of a vertex neighborhood equals its local clustering coefficient.

Lemma 3.1. *For all $\mathcal{S} = \mathcal{N}_v$, $\delta(\mathcal{S}) = C_v$.*

Proof. Observe that every edge in $\mathcal{N}_v$ induces a closed wedge centered at v, which implies that $e(\mathcal{N}_v) = w_v^{(c)}$. Furthermore, as $d_v = |\mathcal{N}_v|$, we have $\binom{|\mathcal{N}_v|}{2} = w_v$. □

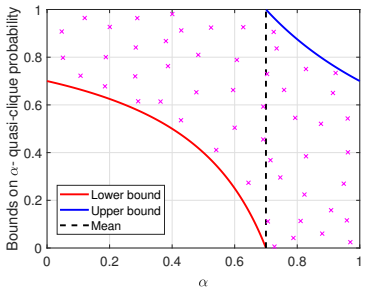

Figure 1: Illustration of the upper bound (8) and lower bound (9) on the probability of a vertex neighborhood being a $\alpha-$quasi-clique for $C_g = 0.7$. The purple crosses mark the feasible region.

If we treat C_v as a random variable with distribution p, an immediate consequence of the above lemma and (5) is the following equation

$$\mathbb{E}_p[\delta(\mathcal{N}_v)] = C_g, \tag{7}$$

which implies that for graphs with large global clustering coefficients, the edge-density of a vertex neighborhood is also large on average. If a vertex $v \in \mathcal{V}$ is sampled with probability p_v, we can establish the following bounds on the probability of $\mathcal{N}_v$ being an $\alpha-$quasi-clique.

Lemma 3.2. *For all $\alpha > C_g$,*

$$Pr\{\delta(\mathcal{N}_v) \geq \alpha\} \leq \frac{C_g}{\alpha}, \forall\, v \in \mathcal{V}. \tag{8}$$

Meanwhile, for $\alpha < C_g$,

$$Pr\{\delta(\mathcal{N}_v) \geq \alpha\} \geq \frac{C_g - \alpha}{1 - \alpha}, \forall\, v \in \mathcal{V}. \tag{9}$$

Proof. The upper bound (8) follows as a simple consequence of Markov's inequality. To establish the lower bound (9), we use the following result extracted from [19, Theorem 4.6]

$$\Pr\{C_v \leq \alpha\} \leq \frac{1 - C_g}{1 - \alpha}. \tag{10}$$

Combining the above inequality with Lemma 3.1 yields the desired claim. □

Clearly, the lower bound (9) is more informative compared to the upper bound (8), as Markov's inequality typically yields a loose bound on the tail probability. Note that for large C_g, the lower bound (9) can yield a non-trivial result. This can be observed from Figure 1, which illustrates the bounds as a function of α for $C_g = 0.7$. For example, when $\alpha = 2/3$, observe that the probability of a vertex neighborhood $\mathcal{N}_v$ being a $2/3-$quasi-clique is at least 10%. It is also evident that the bounds diverge as α approaches the mean C_g. It is only in the extreme case of $C_g = 1$, that the bounds coincide to yield $\Pr\{\delta(\mathcal{N}_v) \geq \alpha\} = 1$. This result can be explained by the fact that for $C_g = 1$, the graph $\mathcal{G}$ is a union of disjoint cliques. Consequently, any vertex neighborhood is also a clique (being the subgraph of a clique), which is always a quasi-clique for every choice of α.

Additional insight regarding the behavior of the distribution about the mean C_g can be obtained by analyzing the variance of $\delta(\mathcal{N}_v)$. To this end, we will require the following result.

Lemma 3.3. $\mathbb{V}_p[\delta(\mathcal{N}_v)] \leq C_g(1 - C_g)$

Proof. Note that the second-order moment of the random variable C_v can be bounded as

$$\mathbb{E}_p[C_v^2] = \sum_{v \in \mathcal{V}} p_v C_v^2 \leq \sum_{v \in \mathcal{V}} p_v C_v = C_g, \tag{11a}$$

where the inequality stems from the fact that $C_v \in [0, 1], \forall\ v \in \mathcal{V}$. Combining the result with (5) and Lemma 1, we obtain

$$\begin{aligned} \mathbb{V}_p[\delta(\mathcal{N}_v)] &= \mathbb{E}_p[C_v^2] - (\mathbb{E}_p[C_v])^2 \\ &\leq C_g(1 - C_g), \end{aligned} \tag{12}$$

which establishes the desired claim. □

The result implies that for low C_g, the variance is small, and thus the values of $\delta(\mathcal{N}_v)$ are likely to be "close" to the mean C_g. In other words, it is unlikely that many neighborhoods exhibit high edge-density. Conversely, as the obtained bound is symmetric about $C_g = 1/2$, for large C_g, the vertex neighborhoods with edge-density close to C_g are likely candidates for being dense subgraphs.

While the aforementioned results suggest that graphs with high global clustering coefficients harbor potentially many dense vertex neighborhoods, as pointed out in the introduction, high edge-density alone is a necessary, but not sufficient condition for the existence of large, dense vertex neighborhoods.

Thus far, our analysis has only been reliant on the clustering coefficient of a graph. We now attempt to incorporate another salient characteristic of real-world graphs into our analysis: heavily-skewed degree distributions. It is well known that the degree distribution of many graphs can be well approximated by a power-law [35]. Let $(d_{\min}, d_{\max})$ denote the smallest degree > 1 and the largest degree of a graph respectively, and let $\mathcal{D} := \{d_{\min}, \cdots, d_{\max}\}$ denote the set of unique degrees in $\mathcal{G}$. For a given degree $d \in \mathcal{D}$, let n_d denote the number of times a vertex $v \in \mathcal{V}$ takes value d. In order to facilitate analysis, we make the following simplifying assumptions:

(C1) The power law exponent of the degree distribution of $\mathcal{G}$ is $\gamma = 2$, which is fairly reasonable as γ typically takes values in the range $[1.75, 3]$ for real world networks [2]. This enables us to express

$$n_d = cnd^{-2}, \forall\ d \in \mathcal{D}, \tag{13}$$

where $c \in \mathbb{R}$ denotes the normalization constant of the degree distribution.

(C2) The set $\mathcal{D}$ does not contain any "missing" degrees, i.e, there exists a vertex of degree d for every possible choice of d satisfying $d_{\min} \leq d \leq d_{\max}$.

Our objective is now to combine both aspects (skewed degree distributions and high clustering coefficients) to formally establish the existence of vertex neighborhoods of non-trivial sizes with high edge density. In order to do so, we take recourse to the probabilistic method [2], a classical and powerful technique in combinatorics for certifying the existence of combinatorial objects possessing certain properties within a probability space. We proceed by first defining the following pair of "bad" events:

[2]This choice is made for convenience and brevity of exposition; we can handle other values of $\gamma > 2$ as well, but the derivations are more cumbersome, see Remark 1.

(A) a vertex sampled with probability p_v has a neighborhood with "low" edge-density,

(B) a vertex sampled with probability p_v has a "small" degree, i.e., a neighborhood of small size.

If we can establish that the probability of either event occurring is strictly less than 1, then it implies the existence of a vertex neighborhood which simultaneously possesses high edge-density and non-trivial size. The exact notions of "low" edge-density and "small" neighborhood size will be quantified next.

Note that (10) already provides an upper bound on the probability of event A occurring. We now seek to establish an upper bound on the probability of event B. For a given parameter $\beta \in \left(\frac{d_{\min}}{d_{\max}}, 1\right)$, define $\bar{d} := \beta d_{\max}$ and let $\mathcal{S}_{\bar{d}}$ denote the set of all vertices having degree greater than 1 but lesser than equal to $\bar{d}$, i.e.,

$$\mathcal{S}_{\bar{d}} := \{v \in \mathcal{V} : d_{\min} \leq d_v \leq \bar{d}\}. \tag{14}$$

We also define a set $\bar{\mathcal{D}} \subseteq \mathcal{D}$ to be the subset of all unique degrees of $\mathcal{G}$ not exceeding $\bar{d}$, i.e.,

$$\bar{\mathcal{D}} := \{d \in \mathcal{D} : d_{\min} \leq d \leq \bar{d}\}. \tag{15}$$

Armed with these definitions, we can derive the following upper bound on the probability of sampling a vertex with a degree smaller than a fraction β of the largest degree $d_{\max}$.

Lemma 3.4. $Pr\{v \in \mathcal{S}_{\bar{d}}\} < \frac{\beta d_{\max} - \log \beta}{d_{\max} - \log\left(\frac{d_{\min}}{d_{\min}-1}\right)}$

Proof. The probability of event B can be expressed as

$$\Pr\{v \in \mathcal{S}_{\bar{d}}\} = \sum_{v \in \mathcal{S}_{\bar{d}}} p_v = \sum_{v \in \mathcal{S}_{\bar{d}}} \frac{w_v}{w} = \frac{\sum_{v \in \mathcal{S}_{\bar{d}}} w_v}{\sum_{v \in \mathcal{V}} w_v}. \tag{16}$$

Exploiting the twin facts that $w_v = \binom{d_v}{2}, \forall\ v \in \mathcal{V}$ and that the degree distribution of $\mathcal{G}$ obeys a power law of the form (13), we obtain the following expressions for the numerator and denominator of (16)

$$\begin{aligned} \sum_{v \in \mathcal{S}_{\bar{d}}} w_v &= \frac{cn}{2} \sum_{d \in \bar{\mathcal{D}}} d(d-1)d^{-2} = \frac{cn}{2} \sum_{d \in \bar{\mathcal{D}}} \left(1 - \frac{1}{d}\right), \\ \sum_{v \in \mathcal{V}} w_v &= \frac{cn}{2} \sum_{d \in \mathcal{D}} d(d-1)d^{-2} = \frac{cn}{2} \sum_{d \in \mathcal{D}} \left(1 - \frac{1}{d}\right). \end{aligned} \tag{17}$$

This allows us to further simplify (16) to

$$\Pr\{v \in \mathcal{S}_{\bar{d}}\} = \frac{|\bar{\mathcal{D}}| - \sum_{d_{\min}}^{\bar{d}} 1/d}{|\mathcal{D}| - \sum_{d_{\min}}^{d_{\max}} 1/d}. \tag{18}$$

In order to derive an upper bound on (18), we exploit the following general fact regarding partial harmonic sums (see [11, Appendix A, p. 1154])

$$\int_l^{u+1} \frac{dx}{x} \leq \sum_{n=l}^{u} \frac{1}{n} \leq \int_{u-1}^{l} \frac{dx}{x} \tag{19}$$

where (l, u) are integers that obey $1 < l < u$ and denote the lower and upper limits of the sum respectively. On computing the integrals, we obtain the approximation bounds

$$\log\left(\frac{u+1}{l}\right) \leq \sum_{n=l}^{u} \frac{1}{n} \leq \log\left(\frac{u}{l-1}\right) \tag{20}$$

Applying the lower bound to the partial harmonic sum appearing in the numerator and the upper bound to the one in the denominator of (18), we obtain

$$\Pr\{v \in \mathcal{S}_{\bar{d}}\} \leq \frac{|\bar{\mathcal{D}}| - \log\left(\frac{\bar{d}+1}{d_{\min}}\right)}{|\mathcal{D}| - \log\left(\frac{d_{\max}}{d_{\min}-1}\right)} \tag{21}$$

The upper bound obtained above can be further bounded by applying the following chain of (strict) inequalities

$$\begin{aligned}
\frac{|\bar{\mathcal{D}}| - \log\left(\frac{\bar{d}+1}{d_{\min}}\right)}{|\mathcal{D}| - \log\left(\frac{d_{\max}}{d_{\min}-1}\right)} &< \frac{|\bar{\mathcal{D}}| - \log\left(\frac{\bar{d}}{d_{\min}}\right)}{|\mathcal{D}| - \log\left(\frac{d_{\max}}{d_{\min}-1}\right)} \\
&= \frac{|\bar{\mathcal{D}}| - \log\beta - \log\left(\frac{d_{\max}}{d_{\min}}\right)}{|\mathcal{D}| - \log\left(\frac{d_{\max}}{d_{\min}-1}\right)} \\
&= \frac{|\bar{\mathcal{D}}| - \log\beta - \log\left(\frac{d_{\max}}{d_{\min}}\right)}{|\mathcal{D}| - \log\left(\frac{d_{\min}}{d_{\min}-1}\right) - \log\left(\frac{d_{\max}}{d_{\min}}\right)} \\
&< \frac{|\bar{\mathcal{D}}| - \log\beta}{|\mathcal{D}| - \log\left(\frac{d_{\min}}{d_{\min}-1}\right)}.
\end{aligned} \tag{22}$$

Upon defining $\Delta := \frac{d_{\min}}{d_{\min}-1}$, and using the fact that

$$|\bar{\mathcal{D}}| = \bar{d} - d_{\min} + 1 = \beta d_{\max} - d_{\min} + 1,$$
$$|\mathcal{D}| = d_{\max} - d_{\min} + 1,$$

it simply remains to apply the chain of inequalities derived in (21) and (22) to finally obtain the claimed upper bound on the probability of event B

$$\begin{aligned}
\Pr\{v \in \mathcal{S}_{\bar{d}}\} &< \frac{\beta d_{\max} - d_{\min} + 1 - \log\beta}{d_{\max} - d_{\min} + 1 - \log\Delta} \\
&< \frac{\beta d_{\max} - \log\beta}{d_{\max} - \log\Delta}
\end{aligned} \tag{23}$$

□

Remark 1: Our assumption regarding the value of the power-law exponent can be relaxed to any value $\gamma > 2$ to obtain a result of a similar flavor, at the expense of a more cumbersome analysis. Owing to space constraints, we only sketch the requisite modifications. The key difference for $\gamma > 2$ is that the functions being summed in the numerator and denominator of (18) are now $d^{2-\gamma}$ and $d^{1-\gamma}$, which are non-increasing in d for $\gamma > 2$. For such functions, the integral approximation trick borrowed from [11, Appendix A, p. 1154] still applies, and consequently, can again be used to derive an upper bound on (18). The exact form of the bound is dependent on the specific value of γ used, as this determines the form that the integrals ultimately take.

Back to our present case of $\gamma = 2$, define the quantities $\eta := \frac{\beta d_{\max} - \log\beta}{d_{\max} - \log\Delta}$, and $\beta_{\max}$ to be the largest value of β that satisfies $\eta < C_g$. With Lemma 3.4 in hand, we can establish the following theorem.

Theorem 3.5. *Under assumptions (C1) and (C2), there exists a vertex neighborhood of size $|\mathcal{N}_v| \geq \beta d_{\max}$ and edge-density $\delta(\mathcal{N}_v) \geq \frac{C_g - \eta}{1-\eta}$, for every choice of $\beta \in \left(\frac{d_{\min}}{d_{\max}}, \beta_{\max}\right)$.*

Proof. Since $|\mathcal{N}_v| = d_v, \forall\, v \in \mathcal{V}$, from Lemma 7 we obtain

$$\Pr\{v \in \mathcal{S}_{\bar{d}}\} = \Pr\{d_{\min} \leq |\mathcal{N}_v| \leq \beta d_{\max}\} < \eta. \tag{24}$$

Meanwhile, on setting $\alpha := \frac{C_g - \eta}{1-\eta}$ in (10), we obtain

$$\Pr\{\delta(\mathcal{N}_v) \leq \alpha\} \leq 1 - \eta. \tag{25}$$

A simple application of the union bound then reveals that the probability of either of the above events occurring is strictly less than 1, thus implying that the complement "good" event occurs with positive probability. Hence, there exists a vertex neighborhood of size $|\mathcal{N}_v| \geq \beta d_{\max}$ which is at least a $\frac{C_g - \eta}{1-\eta}$ quasi-clique. □

When $d_{\max}$ is large, then $\eta \approx \beta$, and thus the quasi-clique value (roughly) varies like $\frac{C_g - \beta}{1-\beta}$. In this case, $\beta_{\max} \approx C_g$, with the result that the allowable range of β is the interval $\left(\frac{d_{\min}}{d_{\max}}, C_g\right)$. A limitation of our result is that it does not allow us obtain results for $\beta > C_g$. However, for large C_g, we obtain a non-trivial lower bound on the size of $\mathcal{N}_v$ and its edge-density. As an illustration of our lower bound for a real graph, please refer to Figure 5 in the supplement.

Additionally, we point out an interesting fact about vertex neighborhoods: if a neighborhood $\mathcal{N}_v$ forms a clique on k-vertices, then $\mathcal{N}_v \cup \{v\}$ is a clique on $(k+1)$-vertices, which we designate as an *ego-clique*. The following result asserts that such ego-cliques must be maximal.

Theorem 3.6. *Let $\mathcal{N}_v$ be a clique on k-vertices and $C_{k+1}(v) := \mathcal{N}_v \cup \{v\}$ be an ego-clique on $(k+1)$-vertices. Every such ego-clique is maximal.*

Proof. Assume the contrary, i.e., that there exists a clique $C_\ell \subset \mathcal{V}$ on ℓ-vertices such that $\ell > k+1$ and $C_{k+1}(v) \subset C_\ell$. Then, there exists a vertex $u \in C_\ell \setminus C_{k+1}(v)$ which is one-hop away from v, since $v \in C_\ell$. This implies that $u \in \mathcal{N}(v) \subset C_{k+1}(v)$, which is a contradiction. □

4 EXPERIMENTAL EVALUATION

In this section, we devise a series of experiments on a variety of datasets that aims to address the following questions: (a) Do dense vertex neighborhoods of non-trivial sizes exist in real-world graphs? (b) How does the approach fare in comparison to dedicated algorithms for dense subgraph discovery?

4.1 Datasets

The list of datasets used and a summary of their statistics are presented in Table 1. If the original graph is directed, a symmetrization step is first performed. Unless specified, the datasets were obtained from [24], and can be classified as follows:

(A) **Co-authorship graphs:** The vertices denote scientists, and the edges represent collaborations between co-authors of a scientific publication. The datasets include co-authorship graphs constructed from arXiv submissions in three different scientific disciplines (arXiv-HepPh, arXiv-AstroPh and arXiv-CondMat), as well as larger graphs comprising the largest connected component of the arXiv and DBLP co-authorship graphs (arXiv [14] and dblp respectively).

(B) **Social networks:** The vertices are people, and the edges indicate "friend" relationships. The datasets include two different snapshots of the Facebook friendship graph (Facebook-A and Facebook-B [34]), friendship networks obtained from a blogging website (blogCatalog3 [30]), and a location-based social networking website (loc-Gowalla).

(C) **Web graphs:** Vertices are web pages, while the edges denote symmetrized hyperlinks (web-Stanford and web-Google).

(D) **Miscellaneous:** An assortment of graphs drawn from different domains: a protein-protein interaction network (ppi-Human), an email communications network (email-Enron), a router graph (router-Caida [12]), and an item-item co-purchase network (Amazon).

4.2 Assessing the Quality of Neighborhood Subgraphs

Given a dataset, we first compute the edge-density of all vertex neighborhoods. This requires calculating the local clustering coefficient of every vertex, which can be accomplished by triangle counting - a task that incurs a worst-case complexity of $O(m^{3/2})$ for a graph with m edges. For our purposes, we employed the MAximal Clique Enumerator (MACE) algorithm (the C code of which is publicly available at [33]) to obtain triangle counts.

Next, for every unique degree in the graph, we compute the highest neighborhood edge-density score over all vertices of that degree and display this information on a plot versus the log of the unique degrees. We designate such a plot as the *neighborhood density profile* (NDP) of a graph, which is shown for six datasets in Figure 2. The NDP plots in the first column represent graphs with high global clustering coefficients, which serve as good test beds for our working hypothesis that vertex neighborhoods are dense subgraphs. Meanwhile, the graphs in the second column possess very low global clustering coefficients, and illustrate the outcome when our sufficient conditions for high neighborhood edge-density are not met. In each NDP plot, we mark the largest degree $d_{\max}$ by a vertical magenta line, the size of the largest clique discovered by the greedyOQC algorithm (for comparison) using a vertical red line, while the global clustering coefficient is highlighted using a black horizontal line. A feature common to all NDP plots is that the neighborhood edge-density decreases with increase in degree, which follows from the fact that the local clustering coefficient of a vertex is inversely proportional to the square of its degree. However, when the global clustering coefficient of the graph is large, from the NDP plots in the first column, it is evident that vertex neighborhoods themselves constitute large (relative to the largest degree $d_{\max}$), dense subgraphs. In fact, it can be observed that several neighborhoods $\mathcal{N}(v)$ attain an edge-density equal to 1, i.e., they form a clique. Recalling the result of Theorem 3.6, it then follows that for the arXiv-HepPh, and dblp datasets, inspecting vertex neighborhoods alone surprisingly reveals maximal cliques of non-trivial sizes. Furthermore, for these datasets, the size of the largest such ego-clique matches the result obtained using the greedyOQC algorithm. On the other hand, for the facebook-A dataset, the size of the largest ego-clique is roughly 6−times smaller than that obtained by greedyOQC. However, it can be seen that there do exist vertex neighborhoods of size comparable to that of the clique discovered by greedyOQC, which are 0.9-quasi-cliques, and thus, are also substantially dense. Taken together, the NDP plots in the first column of Figure 2 provide empirical validation of our hypothesis that graphs with power-law degree distributions and high global clustering coefficients harbor large, dense neighborhood subgraphs.

We now turn our attention to the second column, where C_g is small. In this case, note that the size of the densest neighborhood subgraph is small with respect to the largest degree $d_{\max}$. In particular, for the blogCatalog3 graph, the NDP reveals that the neighborhood edge-density decays quickly with the degree. This represents the worst-case scenario, where the vertex neighborhoods themselves are not appealing candidates for being dense subgraphs of non-trivial sizes. On the other hand, for the graphs loc-Gowalla, and web-Stanford, there are a few dense vertex neighborhoods which form small (relative to $d_{\max}$) subgraphs of non-trivial sizes, and represent atypical or "anomalous" regions of the graph. Note that in terms of quality, on the loc-Gowalla graph, the densest vertex neighborhood is near-optimal in terms of size and edge-density compared to the solution returned by greedyOQC, while on the web-Stanford graph, the largest ego-clique is 4 times larger in size compared to the clique computed by greedyOQC.

In summary, the NDP of a graph is very informative in assessing the edge-density of neighborhood subgraphs. It reveals the presence of large, dense neighborhood subgraphs in real-world graphs

Table 1: Summary of graph statistics: the number of vertices (n), the number of edges (m), the largest degree ($d_{\max}$), the global clustering coefficient (C_g), and the mean local clustering coefficient $\bar{C}$.

Graph	n	m	$d_{\max}$	C_g	$\bar{C}$
arXiv-HepPh	12,008	112K	491	0.659	0.612
arXiv-AstroPh	18,772	198K	504	0.318	0.677
arXiv-CondMat	23,133	93,497	279	0.264	0.633
arXiv	86,376	517K	1,253	0.560	0.678
dblp	317K	1.05M	343	0.306	0.632
Facebook-A	4,039	88,234	1,045	0.519	0.605
blogCatalog3	10,312	333K	3,992	0.091	0.463
Facebook-B	63,731	817K	1,098	0.148	0.221
loc-Gowalla	196K	950K	14,730	0.023	0.237
web-Stanford	281K	2.31M	38,625	0.008	0.598
web-Google	875K	5.10M	6,332	0.055	0.514
ppi-Human	21,557	342K	2,130	0.119	0.207
email-Enron	36,692	183K	1,383	0.085	0.497
router-Caida	192K	609K	1,071	0.061	0.157
Amazon	334K	923K	549	0.205	0.397

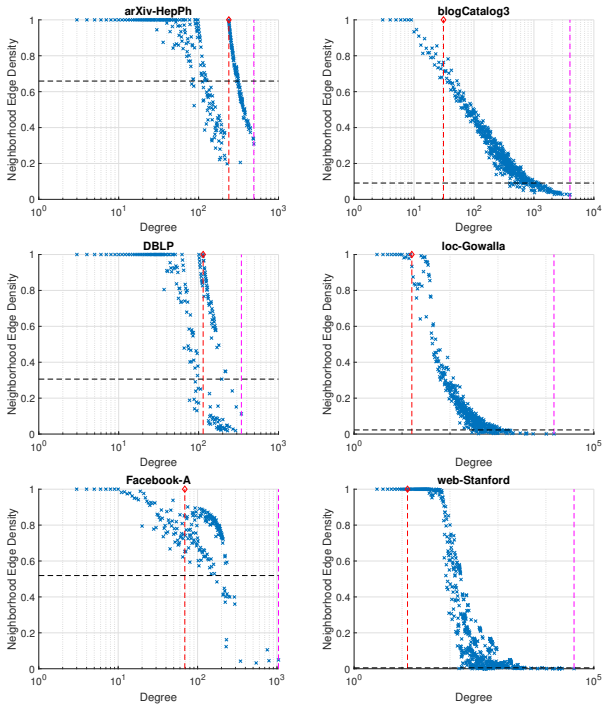

Figure 2: The Neighborhood Density Profile of six real-world graphs. Each plot depicts the maximum of the edge-density of vertex neighborhoods of a given degree versus the log of the degrees. Horizontal black line – global clustering coefficient (C_g), red vertical line – densest subgraph returned by the GREEDYOQC algorithm, and the magenta vertical line – largest degree $d_{\max}$. The graphs in the first column figures have high C_g values, while the ones in the second column have small C_g values.

with power-law degree distributions and high global clustering coefficients, thereby confirming the essence of the result provided by Theorem 3.5. Moreover, it illustrates that graphs exhibiting the aforementioned traits often feature the surprising attribute that neighborhoods themselves constitute maximal cliques of non-trivial sizes, with the size of the largest clique being the same as that determined by GREEDYOQC, which is a non-neighborhood based method. On the other hand, it also showcases that when C_g is small, then neighborhood subgraphs may still form small, dense subgraphs of non-trivial sizes. That being said, there also exist unfavorable instances where neighborhood subgraphs are dense only on a very small scale. The following section explores ways of using such neighborhood subgraphs as seeds for a local-search method in order to grow dense subgraphs of larger sizes.

4.3 Growing Dense Subgraphs from Vertex Neighborhoods

In this section, we describe how the LOCALSEARCHOQC algorithm of [32] can be used to refine the quality of vertex neighborhoods. Given an initial seed set $\mathcal{S}_0 \subseteq \mathcal{V}$, the LOCALSEARCHOQC algorithm aims to maximize the edge-surplus objective function

$$f_\alpha(\mathcal{S}) := e(\mathcal{S}) - \alpha\binom{|\mathcal{S}|}{2} \tag{26}$$

by searching for vertices, which when added or deleted from the current solution set, yields an improvement in the objective function. The procedure is continued until a locally optimal solution is found, (i.e., until addition or deletion of a single vertex from the solution set does not lead to an improvement in the objective), or a maximum number of iterations $T_{\max}$ are reached. While the algorithm has a low run-time complexity of $O(mT_{\max})$, its performance is particularly sensitive to the choice of initialization $\mathcal{S}_0 \subseteq \mathcal{V}$ as the objective function $f_\alpha(\mathcal{S})$ is difficult to maximize (globally). In that regard, we provide compelling empirical evidence that selecting vertex neighborhoods (on the basis of their clustering coefficients) constitutes good seeds for LOCALSEARCHOQC. We devised a pair of simple strategies for judiciously selecting seed sets via this metric - owing to space limitations, the full details are provided in the supplement (see strategies **(S1)** and **(S2)**).

In order to provide empirical justification for our choice, we performed a comparison against a pair of low-complexity alternatives for obtaining seed sets. These are (i) computing the core decomposition of the graph [29], and (ii) selecting vertex neighborhoods on the basis of their average degree. The first choice is motivated by the fact that under the same general assumptions made in Theorem 3.5, a result of a similar flavor has been established in [19] regarding the existence of a dense core, and that the core decomposition can be computed efficiently in linear-time [6]. As the procedure generates a hierarchy of nested subgraphs, we used the final subgraph in the hierarchy (which is the smallest in size and the densest) as a candidate seed. The second choice was proposed in [32] to initialize LOCALSEARCHOQC; i.e., the neighborhood with the highest average degree is selected as the seed. Note that computing the average degree of a vertex neighborhood incurs the same complexity as computing clustering coefficients. However, selecting neighborhoods via this metric presently lacks theoretical justification, in contrast to ours. The quality of the best seeds obtained by the alternatives is depicted in Table 2 – these results are representative of both the best and worst outcomes. Meanwhile, the quality of the best neighborhood obtained obtained by employing strategy **(S2)** is depicted in Table 3 (see columns under Quasi-cliques with heading "NB"). It is evident that our neighborhood selection strategy consistently yields seeds that are of considerably higher quality compared to those obtained via the alternatives (in terms of both size and edge-density). We conclude that our mechanism of generating seeds is well suited for providing high-quality initializations for LOCALSEARCHOQC on real-world data compared to the prevailing

Table 2: Quality of subgraphs obtained via core decomposition and selecting neighborhoods based on average degree in terms of their size $|\mathcal{S}|$ and edge-density $|\delta(\mathcal{S})|$.

	Core decomposition		Avg. degree	
Graph	$\lvert\mathcal{S}\rvert$	$\delta(\mathcal{S})$	$\lvert\mathcal{S}\rvert$	$\delta(\mathcal{S})$
ARXIV-ASTROPH	57	1	81	0.75
ARXIV	146	0.49	147	0.52
BLOGCATALOG3	447	0.4	1550	0.08
FACEBOOK-B	699	0.12	723	0.07
LOC-GOWALLA	183	0.41	162	0.27
WEB-STANFORD	387	0.29	694	0.17
ROUTER-CAIDA	92	0.45	91	0.31
AMAZON	497	0.013	47	0.20

baselines. Following the suggestion of [32], we set the maximum number of iterations $T_{\max} = 50$ in our experiments.Apart from the choice of the initial seed set $\mathcal{S}_0$, the performance of the algorithm is also dependent on the choice of the parameter $\alpha \in (0, 1]$. The recommendation of [32] is to set $\alpha = 1/3$. However, we observed that in practice, on many graphs, the performance of the algorithm with neighborhood seeding can be significantly improved by simply increasing α to much larger values. For a more thorough discussion on selecting α, please refer to the supplement.

4.4 Main Results and Discussion

We compared our approach against two non-neighborhood based methods – the greedyOQC algorithm of [32] and a sophisticated flow-based algorithm proposed in [26] for efficently computing the k-clique densest subgraph [31]. For the former algorithm, which employs greedy vertex peeling to maximize the OQC function (26) and runs in linear time $O(m + n)$, we used a value of $\alpha > 1/3$, as it substantially improves upon the performance reported in [32] (see supplement for an example). Meanwhile, for a given integer $k \geq 3$, the latter method requires a list of all k-cliques in the graph as input. For fair comparison, we used $k = 3$, which reduces to listing triangles, that we already obtained using MACE for computing clustering coefficients. Note that for this choice of k, the algorithm aims to compute the triangle-densest subgraph (TDS). We used the C- based implementation developed by the authors of [26] that is publicly available at [1] to obtain our results.

We summarize the outcomes of our experiments across all datasets in Table 3, which displays the size of the largest clique obtained by each method on each dataset, along with the "best" quasi-clique (i.e., the densest subgraph that is not a clique). The algorithm of [26] does not have any parameter to tune, and hence, we simply display the obtained results. For greedyOQC, we report the largest clique obtained by setting $\alpha = 1$. Meanwhile, for localSearchOQC, the cliques were obtained using the neighborhood seed sets **(S1)** and $\alpha = 1$, while the quasi-cliques were recovered using the neighborhood seed sets **(S2)**. We compared the quasi-cliques returned by the different methods on the basis of their size, edge-density and triangle-density. For fair comparison, we report the quasi-cliques obtained by each method for $\alpha = 0.9$ – if a method returned a clique for this choice of α, we used the next smaller value of $\alpha \in \{0.7, 0.75, 0.8, 0.85\}$ for which a quasi-clique is obtained. If no quasi-clique is returned by a method for any choice of α, we leave a blank in its corresponding location in the table. Our main findings can be summarized as follows:

(1) The best neighborhood (*without* refinement) is, in general, of much higher quality compared to the TDS computed by [26], which requires triangle-listing as a pre-processing step. Furthermore, there always exists a high quality neighborhood quasi-clique (with $\alpha \geq 0.92$ in all but one case) of substantial size - refinement via localSearchOQC mainly yields a similar sized subgraph with improved triangle-density. Overall, these results provide empirical validation of our hypothesis that real-world graphs contain high-quality dense neighborhood subgraphs of non-trivial sizes.
(2) The greedyOQC algorithm (with appropriate tuning) is well-suited for clique discovery in general. However, on 6/15 datasets, the largest clique discovered by greedyOQC *and* localSearchOQC is no better than the largest ego-clique. On the remaining datasets, while the largest ego-clique can be small relative to greedyOQC, by using neighborhoods as seeds for localSearchOQC, we can discover a clique of comparable, or even larger size.
(3) Regarding the performance of localSearchOQC and greedyOQC, while both methods recover quasi-cliques of high quality, the former algorithm has a tendency to produce "denser" quasi-cliques of higher triangle density compared to the latter method.
(4) On 7/15 datasets (particularly, on collaboration networks), we observed that greedyOQC produces a clique, but not any dense quasi-cliques, with the algorithm becoming "stuck" at the same clique for all choices of α. Such an undesirable behavior was not observed for localSearchOQC.

To conclude, our results indicate that selecting vertex neighborhoods based on their local clustering coefficient reveals dense subgraphs of substantial size, which can be competitive with or even better than those obtained by dedicated methods for dense subgraph discovery. We also demonstrated that such vertex neighborhoods are good seeds for localSearchOQC, being substantially better overall than seeds obtained via other simple alternatives such as the core decomposition or choosing neighborhoods with large average degree. Further refining neighborhoods with this simple algorithm allows us to consistently obtain both cliques and quasi-cliques of even higher quality compared to the baselines across a wide variety of heterogeneous datasets.

5 CONCLUSIONS

Our main aim in this paper was to draw attention to the fact that real-world graphs harbor dense vertex neighborhoods of non-trivial sizes, which are often of comparable or higher quality relative to those discovered by dedicated algorithms for maximizing subgraph density. We provided theoretical justification of this phenomenon, in terms of sufficient conditions (namely, a power-law degree distribution and a large global clustering coefficient) under which such a surprising result can be expected in a real-world graph. In practice, our conditions seem to be conservative. We also provided compelling empirical evidence that refining a judiciously chosen neighborhood via a simple local search algorithm delivers state-of-the-art performance at low complexity. This indicates that discovering large cliques and near-cliques is not always hard for real-world graphs, and provides motivation for future work that provides a more refined analysis of these empirical results.

6 ACKNOWLEDGEMENTS

Supported by the National Science Foundation and the Army Research Office under Grants No. IIS-1908070 and ARO-W911NF1910407 respectively. The authors additionally acknowledge the assistance of Charalampos Tsourakakis and Paris Karakasis in executing [1].

REFERENCES

[1] 2015. Large Near-Clique Detection. http://github.com/tsourolampis/Scalable-Large-Near-Clique-Detection.
[2] Noga Alon and Joel H Spencer. 2016. *The probabilistic method.* John Wiley & Sons.

Table 3: Single best clique and quasi-clique computed by each method. The second column displays the clique size (the larger the better), while the last 3 columns display the quality of quasi-cliques as measured by their size $|\mathcal{S}|$, edge-density $\delta(\mathcal{S}) = e(\mathcal{S})/\binom{|\mathcal{S}|}{2}$ and triangle-density $\tau(\mathcal{S}) = t(\mathcal{S})/\binom{|\mathcal{S}|}{3}$ (here, $t(\mathcal{S})$ is the number of triangles in subgraph $\mathcal{S}$). NB - neighborhood, NB+LS - local search with neighborhood seeds, GRDY - greedyOQC, TDS - flow based algorithm of [26] for computing the triangle densest subgraph.

	Cliques	Quasi-cliques													
	$\lvert\mathcal{S}\rvert$	$\lvert\mathcal{S}\rvert$						$\delta(\mathcal{S})$				$\tau(\mathcal{S})$			
Graph	NB	NB + LS	GRDY	NB	NB + LS	GRDY	TDS	NB	NB + LS	GRDY	TDS	NB	NB + LS	GRDY	TDS
arXiv-HepPh	239	239	239	246	247	-	239	0.95	0.95	-	1	0.92	0.91	-	1
arXiv-AstroPh	57	57	57	48	45	-	76	0.90	0.99	-	0.80	0.83	0.97	-	0.59
arXiv-CondMat	23	**26**	**26**	19	18	-	30	0.86	0.96	-	0.93	0.68	0.89	-	0.72
arXiv	74	74	74	75	60	-	146	0.95	0.98	-	0.49	0.92	0.94	-	0.25
dblp	114	114	114	105	-	-	114	0.95	-	-	1	0.92	-	-	1
Facebook-A	11	32	**69**	50	53	118	195	0.94	0.98	0.97	0.79	0.85	0.94	0.92	0.54
blogCatalog3	10	29	**31**	12	52	52	621	0.95	0.96	0.96	0.31	0.87	0.88	0.88	0.05
Facebook-B	12	**25**	**25**	20	17	36	198	0.95	0.98	0.96	0.36	0.85	0.95	0.89	0.08
loc-Gowalla	15	**28**	16	36	32	23	311	0.94	0.99	0.95	0.27	0.85	0.97	0.86	0.04
web-Stanford	53	**53**	14	71	68	16	684	0.95	0.99	0.96	0.17	0.89	0.97	0.88	0.02
web-Google	25	43	**44**	54	48	48	66	0.93	0.99	0.99	0.85	0.84	0.98	0.98	0.64
ppi-Human	81	**130**	**130**	81	-	-	361	0.93	-	-	0.42	0.89	-	-	0.14
email-Enron	10	**16**	**16**	14	12	22	388	0.93	0.98	0.96	0.19	0.82	0.95	0.89	0.02
router-Caida	9	**15**	6	12	15	-	75	0.92	0.97	-	0.55	0.94	0.99	0.95	0.20
Amazon	**7**	**7**	5	7	8	7	50	0.95	0.96	0.90	0.19	0.86	0.90	0.72	0.02

[3] Albert Angel, Nikos Sarkas, Nick Koudas, and Divesh Srivastava. 2012. Dense subgraph maintenance under streaming edge weight updates for real-time story identification. In *Proceedings of the 38th International Conference on Very Large Data Bases*. VLDB Endowment, 574–585.

[4] Gary D Bader and Christopher WV Hogue. 2003. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* 4, 1 (2003), 2.

[5] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *Science* 286, 5439 (1999), 509–512.

[6] Vladimir Batagelj and Matjaz Zaversnik. 2003. An O (m) algorithm for cores decomposition of networks. *arXiv preprint cs/0310049* (2003).

[7] Coen Bron and Joep Kerbosch. 1973. Algorithm 457: Finding all cliques of an undirected graph. *Commun. ACM* 16, 9 (1973), 575–577.

[8] Gregory Buehrer and Kumar Chellapilla. 2008. A scalable pattern mining approach to web graph compression with communities. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 95–106.

[9] Jose Cadena, Anil Kumar Vullikanti, and Charu C Aggarwal. 2016. On dense subgraphs in signed network streams. In *Proceedings of the 16th IEEE International Conference on Data Mining (ICDM)*. IEEE, 51–60.

[10] Moses Charikar. 2000. Greedy approximation algorithms for finding dense components in a graph. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*. Springer, 84–95.

[11] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to algorithms*. MIT press.

[12] Timothy A Davis and Yifan Hu. 2011. The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software (TOMS)* 38, 1 (2011), 1.

[13] David Eppstein, Maarten Löffler, and Darren Strash. 2010. Listing all maximal cliques in sparse graphs in near-optimal time. In *International Symposium on Algorithms and Computation*. Springer, 403–414.

[14] Pooya Esfandiar, Francesco Bonchi, David F Gleich, Chen Greif, Laks VS Lakshmanan, and Byung-Won On. 2010. Fast Katz and commuters: Efficient estimation of social relatedness in large networks. In *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 132–145.

[15] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. 1999. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, Vol. 29. ACM, 251–262.

[16] Alessandro Ferrante, Gopal Pandurangan, and Kihong Park. 2008. On the hardness of optimization in power-law graphs. *Theoretical Computer Science* 393, 1-3 (2008), 220–230.

[17] Michael R Garey and David S Johnson. 2002. *Computers and intractability*.

[18] David Gibson, Ravi Kumar, and Andrew Tomkins. 2005. Discovering large dense subgraphs in massive graphs. In *Proceedings of the 31st International Conference on Very Large Data Bases*. VLDB Endowment, 721–732.

[19] David F Gleich and C Seshadhri. 2012. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 597–605.

[20] Andrew V Goldberg. 1984. *Finding a maximum density subgraph*. Technical report, University of California Berkeley, CA.

[21] Rishi Gupta, Tim Roughgarden, and Comandur Seshadhri. 2014. Decompositions of triangle-dense graphs. In *Proceedings of the 5th conference on Innovations in Theoretical Computer Science*. ACM, 471–482.

[22] Bryan Hooi, Hyun Ah Song, Alex Beutel, Neil Shah, Kijung Shin, and Christos Faloutsos. 2016. Fraudar: Bounding graph fraud in the face of camouflage. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 895–904.

[23] Matthieu Latapy. 2008. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theoretical Computer Science* 407, 1-3 (2008), 458–473.

[24] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[25] Zhi-Quan Luo, Wing-Kin Ma, Anthony Man-Cho So, Yinyu Ye, and Shuzhong Zhang. 2010. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine* 3, 27 (2010), 20–34.

[26] Michael Mitzenmacher, Jakub Pachocki, Richard Peng, Charalampos Tsourakakis, and Shen Chen Xu. 2015. Scalable large near-clique detection in large-scale networks via sampling. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 815–824.

[27] Mark Newman. 2018. *Networks*. Oxford university press.

[28] N Pržulj, Dennis A Wigle, and Igor Jurisica. 2004. Functional topology in a network of protein interactions. *Bioinformatics* 20, 3 (2004), 340–348.

[29] Stephen B Seidman. 1983. Network structure and minimum degree. *Social networks* 5, 3 (1983), 269–287.

[30] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 817–826.

[31] Charalampos Tsourakakis. 2015. The k-clique densest subgraph problem. In *Proceedings of the 24th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1122–1132.

[32] Charalampos Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria Tsiarli. 2013. Denser than the densest subgraph: extracting optimal quasi-cliques with quality guarantees. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 104–112.

[33] Takeaki Uno. 2005. Maximal Clique Enumerator (MACE). http://research.nii.ac.jp/~uno/codes.htm.

[34] Bimal Viswanath, Alan Mislove, Meeyoung Cha, and Krishna P Gummadi. 2009. On the evolution of user interaction in Facebook. In *Proceedings of the 2nd ACM Workshop on Online social networks*. ACM, 37–42.

[35] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature* 393, 6684 (1998), 440.

SUPPLEMENTARY MATERIAL

In order to facilitate reproducibility, this section contains a detailed description of the mechanisms used to generate the neighborhood seed sets for initializing localSearchOQC, guidelines for choosing the tuning parameter α in the OQC objective function (26) for both the localSearchOQC and greedyOQC algorithms, and additional experiments showcasing how the choice of these parameters influences the obtained results. Additionally, we provide an example to illustrate the quality of the lower bound on the neighborhood quasi-clique value derived in Theorem 3.5 on a real-world graph.

We begin by discussing the choice of α for localSearchOQC. While the recommendation of [32] is to set $\alpha = 1/3$, the algorithm performs much better in practice with a larger value. Such a beneficial effect can be partially explained via the following intuitive argument: consider the case where $\bar{C} > 1/3$ for a given graph $\mathcal{G}$. Note that the term $\alpha\binom{|\mathcal{S}|}{2}$ in $f_\alpha(\mathcal{S})$ can be interpreted as the expected number of edges in a subgraph $\mathcal{G}_\mathcal{S}$ of a random Erdos-Renyi graph with edge-density α. This random graph model serves as a null model which is used to compare and contrast the number of edges of a subgraph $\mathcal{G}_\mathcal{S}$ in the given graph $\mathcal{G}$. We now point out that α can also be equivalently viewed as the expected local clustering coefficient of a random Erdos-Renyi graph. This observation suggests that given a graph, we can set the value of α to be equal to the average clustering coefficient $\bar{C}$ of $\mathcal{G}$, as the random Erdos-Renyi graph model will exhibit the same clustering coefficient as $\mathcal{G}$ on average, and hence, may constitute a more appropriate parameter setting when $\bar{C} > 1/3$. In practice though, we observed that irrespective of the actual value of $\bar{C}$, it never hurts to increase α to a value larger than $\max\{1/3, \bar{C}\}$. This effect is illustrated via the following two strategies for generating seed sets for the localSearchOQC algorithm.

(**S1**): In this strategy, from the NDP of a graph, we select all vertices whose neighborhood density lies in the interval $[0.70, 0.95]$. On average, this yields a small number of $20-30$ vertices, with the worst-case extremes arising in the case of the facebook-A graph, where 3.5% of the $4,039$ vertices (a total of 153) where returned and the Amazon graph, where only 4 vertices were returned. Every such vertex v is then combined with its neighborhood $\mathcal{N}(v)$ to generate a seed set $\{v\} \cup \mathcal{N}(v)$, which is used as initialization for the localSearchOQC algorithm with $\alpha = 1$. For this choice of α, the edge-surplus objective function $f_\alpha(\mathcal{S})$ attaches a high penalty to any subset of vertices which do not form a clique, i.e., we "encourage" the algorithm to discover cliques.

(**S2**): In an alternative strategy, we partition the interval of neighborhood density values $[0.70, 0.95)$ into 5 sets of disjoint, equi-spaced sub-intervals $[0.7, 0.75)$, $[0.75, 0.8)$, $[0.8, 0.85)$, $[0.85, 0.9)$, and $[0.9, 0.95)$. Next, we list the vertices of the graph whose neighborhood edge-densities lie in one of these 5 sub-intervals. For graphs with small C_g the size of the list was always $< 1\%$ of the total number of vertices, whereas it was up to 5% for larger C_g. From each sub-interval, we select the vertex whose neighborhood subgraph attains the highest edge-surplus value according to (26), where the parameter α in $f_\alpha(\mathcal{S})$ is set to the lower bound of the sub-interval; e.g., for the sub-interval $[0.9 - 0.95)$, $\alpha = 0.9$. This vertex v is then combined with its neighborhood to form the seed set $\{v\} \cup \mathcal{N}(v)$, which is then used to initialize localSearchOQC, with the same value of α as the sub-interval lower bound. A total of 5 such seed sets are generated (one for each sub-interval). In this case, our objective is to induce the algorithm to unearth large quasi-cliques.

The performance of localSearchOQC using the seeding strategy **(S1)**, is depicted in Figure 3 on 2 representative datasets. By setting $\alpha = 1$, localSearchOQC is indeed capable of discovering cliques when initialized from appropriate vertex neighborhoods. While the size of the discovered cliques is smaller than the largest ego-clique for a small number of seeds, the majority of trials produced cliques of larger sizes. We empirically verified that these cliques are maximal, which concurs with our intuition regarding the algorithm, i.e., if the current solution set is a non-maximal clique, by design, the algorithm will seek to add vertices which will produce a larger, maximal clique (note that the extreme setting $\alpha = 1$ discourages any other vertices from being added in this case). A list of these maximal cliques of size larger than the largest ego-clique for the datasets considered are depicted in the right-hand column of Figure 3. We point out that on the web-Google dataset, a few seeds produced subgraphs of small size and low density. This illustrates a potential drawback of setting $\alpha = 1$: if the initial seed set is not in the local vicinity of a denser subgraph, then localSearchOQC compensates by seeking out a small subgraph with low density. To appreciate this behavior, we focus on one such seed set of size 60 and density 0.77. For this subgraph, the edge-surplus objective function has a value of -407. When used as initialization for localSearchOQC, the algorithm yields a subgraph of size 11 and edge-density 0.18. However, the objective function $f_\alpha(\mathcal{S})$ has a value of -45, which marks a near 10-fold improvement over the initial set. While this is a worst-case scenario for such a "all-or-nothing" approach, we observed that it seldom occurs in practice (only $6/32$ trials on the web-Google graph and no such occurrences on the Facebook-B graph). Overall, our experiments indicate that these vertex neighborhoods can indeed serve as favorable initialization points for discovering maximal cliques using localSearchOQC.

As a performance benchmark, we also added the greedyOQC algorithm of [32], with α also set equal to 1. Interestingly, the algorithm *always* produced a clique with this setting on all the datasets we tried. With regard to detecting cliques, Figure 3 reveals that the performance of greedyOQC is competitive with localSearchOQC. On the Facebook-B graph, localSearchOQC detects 3 distinct cliques of size 25, while greedyOQC also discovers a different clique of the same size. Finally, on the web-Google graph, the size of the largest clique discovered by localSearchOQC is 43, which is comparable in size to the largest clique on 46 vertices produced by greedyOQC. We also empirically observed that the clique returned by greedyOQC does not subsume any of the smaller cliques produced by localSearchOQC, thereby highlighting the contrasting nature of the two approaches.

We now focus on the effectiveness of localSearchOQC in discovering large quasi-cliques when using the seeding strategy **(S2)**. We used greedyOQC again as a benchmark, with the range of parameter settings varying from $\alpha \in \{1/3, 0.7, 0.75, 0.8, 0.85, 0.9, 1\}$, i.e., from the recommended setting $1/3$ to the highest possible value

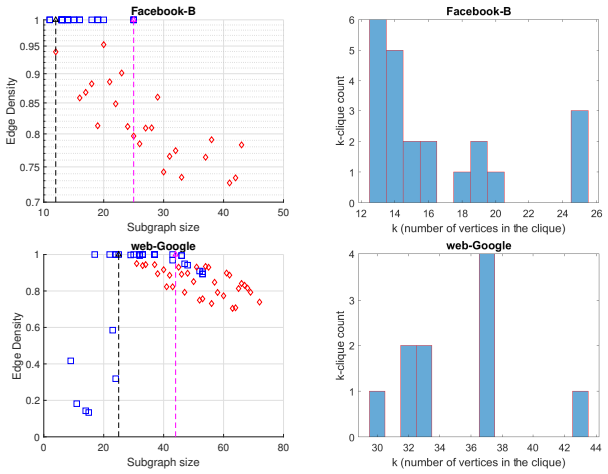

Figure 3: **Results of using LocalSearchOQC with seeds (S1) on three real-world graphs. Left column: Edge Density versus subgraph size. The red diamonds denote the neighborhood subgraphs selected using the seeding strategy (S1), the black vertical line highlights the size of the largest ego-clique, the blue squares denote the subgraphs obtained using LocalSearchOQC with seeds (S1) and $\alpha = 1$, and the magenta vertical line marks the size of the largest clique returned by GreedyOQC. Right column: list of k-cliques obtained by LocalSearchOQC of size larger than the largest ego-clique.**

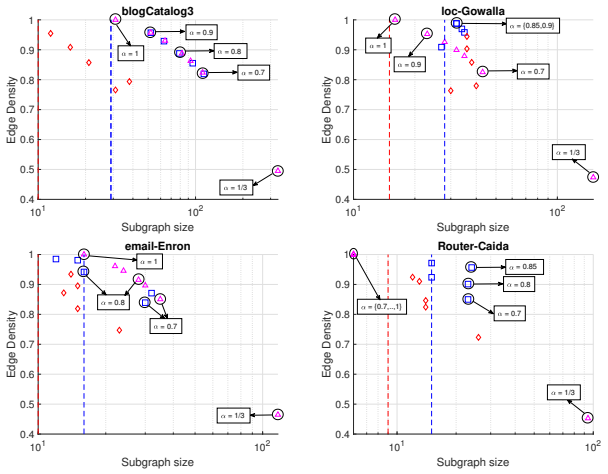

Figure 4: **Edge density versus subgraph size for four real-world graphs as a function of the parameter α used in LocalSearchOQC and GreedyOQC. The red diamonds denote the neighborhood subgraphs selected using the seeding strategy (S2), the red vertical line highlights the size of the largest ego-clique, the blue squares denote the subgraphs obtained using LocalSearchOQC with seeds (S2), the blue vertical line marks the size of the largest clique obtained using LocalSearchOQC with seeds (S1), and the magenta triangles denote the output of GreedyOQC.**

1. Figure 4 displays the results of our experiments on 4 datasets, which are representative of all the possible outcomes that we observed. Regarding the performance of GreedyOQC, we point out that the recommended setting of $\alpha = 1/3$ can be *very* sub-optimal with respect to the neighborhood subgraphs we selected. For example, on the BlogCatalog3 dataset, using $\alpha = 1/3$ outputs a subgraph on 330 vertices with edge-density 0.5, which is 33% less dense and 10 times larger in size than the least-dense neighborhood subgraph obtained. The algorithm demonstrates marked improvement only upon using a more aggressive choice of α, with the subgraph size decreasing and the density increasing progressively as α is increased, and ultimately yielding a clique when $\alpha = 1$.

On the BlogCatalog3 dataset, in terms of size and edge-density, the quasi-cliques computed by LocalSearchOQC are a close match to those computed by GreedyOQC for a given α. On the other hand, on the Loc-Gowalla graph, it can be noted that the initial seed sets themselves are large quasi-cliques. In this case, further refinement using LocalSearchOQC does not result in a significant improvement, although it does identify a near-clique on 32 vertices. In comparison, the largest clique detected by GreedyOQC is only marginally larger than the largest ego-clique, and is much smaller than the largest clique recovered by LocalSearchOQC. On the Email-Enron graph, we observe the opposite trend, i.e., LocalSearchOQC produces dense quasi-cliques of smaller size compared to GreedyOQC overall. On the Router-Caida graph, we made a curious observation regarding GreedyOQC - the subgraph produced is invariant with respect to all choices of $\alpha > 1/3$. In this case, the algorithm completely fails to unveil any dense quasi-cliques, while LocalSearchOQC discovers a 0.95-quasi-clique on 24 vertices. Furthermore, it can be seen that the clique computed by GreedyOQC is of size 6, which is smaller than *both* the largest ego-clique and the largest clique computed by LocalSearchOQC.

Finally, we compare the lower bound on the neighborhood edge-density derived in Theorem 3.5 against its actual value for the Facebook-A graph in Figure 5. We chose this particular dataset as it has a large value of $C_g = 0.52$, and its degree distribution closely conforms with our assumptions (C1)–(C2). Note that for a fixed C_g, the lower bound $(C_g - \beta)/(1 - \beta)$ decreases monotonically with $\beta \in (d_{\min}/d_{\max}, C_g)$. We plot the value of this lower bound for every unique degree in the graph that lies between a fraction $\beta_{\min} = 0.05$ and $\beta_{\max} = C_g$ of the largest degree $d_{\max} = 1,045$, and also plot the largest clustering coefficient C_v (i.e., the actual neighborhood edge-density) for every such degree. The figure reveals that our lower bound is pessimistic in general, although it becomes tighter for larger degrees. A very small number of neighborhoods of large degree also violate the lower bound, which we attribute to the fact that there are missing degrees in practice and that the degree distribution approximately obeys a power-law with exponent 2.

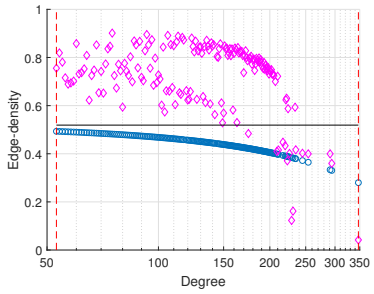

Figure 5: **Lower bound of Theorem 3.5 (blue) vs actual neighborhood edge-density (magenta) as a function of the degree for the Facebook-A graph. Black line – C_g, red lines – admissible range of degrees.**