FPT Algorithms for Finding Near-Cliques in c-Closed Graphs

Balaram Behera
Georgia Institute of Technology, Atlanta, GA, USA

Edin Husić
London School of Economics and Political Science, UK

Shweta Jain
University of Illinois, Urbana-Champaign, IL, USA

Tim Roughgarden
Columbia University, New York, NY, USA

C. Seshadhri
University of California, Santa Cruz, CA, USA

Abstract

Finding large cliques or cliques missing a few edges is a fundamental algorithmic task in the study of real-world graphs, with applications in community detection, pattern recognition, and clustering. A number of effective backtracking-based heuristics for these problems have emerged from recent empirical work in social network analysis. Given the $\mathbb{NP}$-hardness of variants of clique counting, these results raise a challenge for *beyond worst-case analysis* of these problems. Inspired by the triadic closure of real-world graphs, Fox et al. (SICOMP 2020) introduced the notion of c-closed graphs and proved that maximal clique enumeration is fixed-parameter tractable with respect to c.

In practice, due to noise in data, one wishes to actually discover "near-cliques", which can be characterized as cliques with a sparse subgraph removed. In this work, we prove that many different kinds of maximal near-cliques can be enumerated in polynomial time (and FPT in c) for c-closed graphs. We study various established notions of such substructures, including k-plexes, complements of bounded-degeneracy and bounded-treewidth graphs. Interestingly, our algorithms follow relatively simple backtracking procedures, analogous to what is done in practice. Our results underscore the significance of the c-closed graph class for theoretical understanding of social network analysis.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Social networks

Keywords and phrases c-closed graph, dense subgraphs, FPT algorithm, enumeration algorithm, k-plex, Moon-Moser theorem

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.17

Related Version *Previous Version*: `https://arxiv.org/abs/2007.09768v3` [42]
Full Version: `https://arxiv.org/abs/2007.09768`

Funding *Tim Roughgarden*: Supported in part by NSF Award CCF-1813188, CCF-2006737 and ARO grant W911NF1910294.
C. Seshadhri: Supported by NSF DMS-2023495, CCF-1740850, 1839317, 1813165, 1908384, 1909790, and ARO Award W911NF1910294.

Acknowledgements We would like to thank anonymous referees for their comments and suggestions.

1 Introduction

The discovery of cliques and clique-like subgraphs is a fundamental tool in modern graph analysis, especially for social networks. Such substructures have been used in many different applications including community detection in social networks [54, 71], identification of

13th Innovations in Theoretical Computer Science Conference (ITCS 2022).
Editor: Mark Braverman; Article No. 17; pp. 17:1–17:24
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

real-time stories in the news [3] and even detection of regulatory motifs in DNA [34]. They have been used for graph visualization [79, 80] and for creating index structures for answering reachability and distance queries in databases [23, 46].

In practice, due to noise in data, one is also interested in large "near-cliques". While this is an ill-defined term, applications require cliques that are missing a small sparse subgraph. For example, incomplete cliques have been used to predict missing pairwise interactions [78] and for identifying functional groups [38] in a protein interaction network. They have been used for community detection [82] and for detecting test collusion [8]. Recent works have used the fraction of near-cliques to k-cliques to define higher order variants of clustering coefficients [77]. A common notion is that of k-plexes (a clique minus a subgraph with degree bound k). They have been used in community detection [74, 4], for partitioning of sparse biological networks [35], and for determining molecular similarity [40].

From a worst-case standpoint, even the simpler problem of maximum clique is a notoriously difficult computational problem. Even getting $O(n^{1-\delta})$-approximations is $\mathbb{NP}$-hard [39, 83], and it is hard to non-trivially approximate even with algorithm parameterized by solution size [17]. On the other hand, there have been many recent successes in clique enumeration/approximation in the data mining community [44, 45, 57, 28, 18, 29, 43]. Many of these results employ backtracking heuristics [44, 43, 28, 29]. These algorithms can even get the exact maximum clique for graphs with millions of edges. Moreover, the basic backtracking techniques work for approximating counts of cliques missing a few edges [45, 75, 9, 70].

This gap between theory and practice is the main focus of our work. *Can we prove the existence of efficient (hopefully, backtracking) algorithms for near-clique discovery, assuming the input has "reasonable" properties of social networks?*

The starting point for our work is the recent notion of *c-closed graphs*, defined by Fox et al. [32, 33]. Triadic closure – the property that friends of friends are often friends – is a well-observed property of social networks. A c-closed graph has the property that two vertices sharing at least c common neighbors are connected by an edge. Fox et al. empirically show that real-world social network are often (or approximately) c-closed for small values of c. Theoretically, they proved that maximal clique enumeration can be done in time $2^{O(c)}n^2$, and is hence fixed parameter tractable (FPT) in c. (The basic brute force algorithm can be shown to run in $O(n^c)$ time.)

1.1 Main results

Our focus is on counting the number of maximal near-clique structures, which we can roughly define as "a clique minus a sparse subgraph", or alternately, the complement of a sparse subgraph. The input graph G has n vertices, m edges, and is assumed to be c-closed.

We define the various pattern subgraphs that will be counted. We begin with the classic notion of a $(d+1)$-plex.

▶ **Definition 1** ($(d+1)$-plex, [69]). *A subset of vertices S is called a* $(d+1)$-plex *if each $v \in S$ is adjacent to all but at most d vertices of S (excluding itself).*

Observe that a $(d+1)$-plex is precisely the complement of a graph with maximum degree at most d. Our first result is that enumerating maximal $(d+1)$-plexes (for constant d) in an input c-closed graph is FPT in c.

▶ **Theorem 2.** *For c-closed graphs and a fixed $d \geq 0$, there is an algorithm running in time $O(n^{2d} \cdot \kappa_d^c \cdot p(c))$ for enumerating $(d+1)$-plexes, where $\kappa_d < 2$ is the root of $x^{d+4} - 2x^{d+3} + 1 = 0$; and for a polynomial p. For 2-plexes, a stronger bound $O(n^2 \cdot 10^{c/5} \cdot p(c))$ applies.*

We go further and show analogous results for other patterns that can be expressed as complements of sparse graphs. A pattern has *bounded co-degeneracy* if the degeneracy of the complement is bounded. The degeneracy can be thought of as a more robust notion of maximum degree, and has a significant role in social network analysis. Bounded co-degenerate graphs are a natural generalization of $(d+1)$-plexes. Analogously, we also consider counting maximum *bounded co-treewidth* graphs.

▶ **Theorem 3.** *For c-closed graphs and a fixed $d \geq 0$, there is an algorithm running in time $O(n^{2d+4}4^c)$ that outputs all maximal induced subgraphs with co-degeneracy d in an input c-closed graph.*

▶ **Theorem 4.** *For c-closed graphs and a fixed $t \geq 0$, there is an algorithm running in time $O(n^{t+4}2^{2c})$ that outputs all maximal induced subgraphs with co-treewidth $\leq t$.*

The exponential dependence n^d in Theorems 2 and 3 is necessary, as is the dependence n^t in Theorem 4 as we show with examples.

We note that not all natural notions of "co-sparse" subgraphs lead to FPT bounds. For example, the maximal subgraphs with bounded *average* co-degree cannot be listed by an FPT algorithm, even for average co-degree of at most 2.

▶ **Example 5.** Let $\ell \in \mathbb{N}$ and let $c = \frac{\ell}{2}(\ell+1)+1$. By the hand-shaking lemma, a subgraph $G[S]$ has average co-degree at most 2 if and only if $G[S]$ contains at most $|S|$ non-edges. Consider a graph G consisting of a clique K on $c-1$ vertices and an independent set I on n vertices, where any vertex in I is adjacent to every vertex in K. G is c-closed since any two non-adjacent vertices are adjacent only to K. Note that G contains exactly $n+c-1$ vertices.

Let us show that the number of maximal subgraphs $G[S]$ with at most $|S|$ non-edges is at least n^ℓ and hence not FPT with respect to c. In particular, consider a set of the form $S \cup K$ where $S \subseteq I$. If $|S| = s$, then the number of non-edges in $G[S \cup K]$ is exactly $s(s-1)/2$. By the choice of c, any set S of size ℓ is a maximal subgraph with at most $|S|$ non-edges. Thus, there are at least $O(n^\ell) \approx O(n^{\sqrt{2c}})$ maximal subgraphs $G[S]$ with at most $|S|$ non-edges.

The backtracking connection: One of the first steps in proving the above theorems is a different, simpler proof that maximal clique enumeration is FPT in c. (This is the main result of Fox et al. [33].) Typical backtracking algorithms exhaustively and incrementally build candidates for solutions until they have discovered all candidates. We analyze a simple backtracking procedure that finds cliques (Section 3), and give a bound on its running time. Moreover, we use this result to show that maximal bounded co-degenerate subgraphs can be enumerated efficiently. We consider these proofs as mathematical justification for the empirical success of backtracking algorithms, and see our results as "beyond the worst-case analysis" results [68].

Organization: In Section 1.2, we describe our results in more detail. Section 1.3 covers related work. Section 2 describes the definitions and terms required for the proofs. Sections 3, 4, 5 and 6 respectively gives proofs for FPT bounds for cliques, $(d+1)$-plexes, bounded co-degeneracy and bounded co-treewidth graphs.

1.2 Discussion of results

Cliques. We first provide a simple proof that uses a backtracking tree to show that the number of maximal cliques is bounded by $O(cm2^c)$ where m represents the number of edges in the complement graph. (Fox et al. prove a bound of $\min\{3^{(c-1)/3}n^2, 4^{(c+4)(c-1)/2}n^{2-2^{1-c}}\}$). We convert this result into a simple backtracking algorithm for enumerating maximal cliques that runs in time $O(cmn^2 2^c)$. Although the running time bound we obtain is slightly worse than that of Fox et al., the algorithm and proof are simpler, in particular, as Fox et al. black-box clique enumeration. We also believe that our proof provides theoretical understanding for the practical efficiency of common backtracking methods, such as the Bron-Kerbosch algorithm [14] and a recent work of Jain-Seshadhri [44].

Two approaches. For the other dense subgraph types, we do the following: for each type, we provide structural results bounding the maximum possible number of maximal subgraphs of that type. Our results come in two flavors. In one flavor, the *backtracking approach*, we show that any subgraph of that type can be split into parts which are either bounded in size or are cliques. For parts that are cliques, we use the simple backtracking algorithm for counting cliques mentioned above. For parts that are not cliques (and are thus bounded in size), we simply find candidate vertices for each part, enumerate all subsets of these candidate sets and combine them to give a set of subgraphs that is a superset of the set of all maximal subgraphs of that type (for cliques, k-plexes and co-degenerate subgraphs, there exist simple tests for checking if a subgraph is a maximal subgraph of that type). Because the parts are of bounded size, we get FPT bounds for the size of this superset. In some cases (cliques and $d+1$-plexes), this approach leads to slightly worse exponential factors than bounds obtained using the second approach, but leads to simple algorithms that are easy to describe. Indeed, the enumeration algorithms follow from the structural results; obtaining the structural results is the main challenge. Interestingly, the algorithms obtained using this approach have significant portions that use backtracking, reflecting the fact that backtracking has proven to be effective in practice.

In the other flavor, the *three-step approach*, we use the approach taken by Fox et al. for proving their result for maximal cliques. We view their proof as being composed of three parts. The first part uses a combinatorial bound on the number of maximal cliques, the classic Moon-Moser theorem [63, 62]. This theorem states that the number of maximal cliques in an arbitrary N-vertex graph is bounded above by $3^{N/3}$ (with a matching lower bound furnished by a complete $(N/3)$-partite graph).

The second and most interesting part of the proof exploits the c-closed condition to translate the Moon-Moser theorem into an FPT bound of at most $n^2 3^{(c-1)/3}$ maximal cliques in a c-closed graph with n vertices. Roughly, this step of the proof works as follows. For (almost) every maximal clique, one can identify two non-adjacent vertices such that the clique is contained in the common neighborhood of the two vertices. Such a maximal clique in the original graph is also maximal in an induced subgraph on at most $c-1$ vertices, by the c-closure property. The upper bound follows by applying the Moon-Moser theorem to these subgraphs (of which there is a polynomial number), each of size at most $c-1$.

The third step is to translate the FPT combinatorial bound on the number of maximal cliques into an FPT algorithm for enumerating them. For the case of cliques, there is a well known algorithm [73] that can be used to list all maximal cliques in $O(mn)$ time per clique.[1]

Thus, for proofs using the three-step approach, we use the same three-part framework outlined above for the special case of cliques:

1. **Combinatorial bound:** Find an upper bound on the number of maximal dense subgraphs in an arbitrary N-vertex graphs, in the spirit of the Moon-Moser theorem. (Either relying on an existing bound or proving a new one from scratch.)

[1] Replacing the Moon-Moser bound with the trivial bound of 2^N would also lead to an FPT result, albeit one that is exponentially worse. Fox et al. [32, 33] also prove an incomparable bound with better dependence on n ($n^{2-2^{1-c}}$) but worse dependence on c ($4^{(c+4)(c-1)/2}$).

2. **FPT bound:** Exploit the c-closed condition to translate the combinatorial bound into an FPT-type upper bound (with parameter c) on the number of maximal dense subgraphs in a c-closed graph on n vertices.
3. **Enumeration:** Give an FPT enumeration algorithm for listing all maximal dense subgraphs in a c-closed graph. (Either relying on an existing enumeration algorithm or devising a new one.)

We describe our contributions in more detail below:

$(d+1)$-plexes.[2] A subset $S \subseteq V(G)$ is called a $(d+1)$*-plex* if every vertex $v \in S$ is non-adjacent to at most d other vertices in S. Equivalently, a subset S is a $(d+1)$-plex if $G[S]$ has co-degree at most d. Thus, a clique is 1-plex. This is a common relaxation of cliques used in practice [31, 69]. For each fixed d, we give an FPT algorithm for enumerating $(d+1)$-plexes. In general graphs, an FPT algorithm for finding a largest $(d+1)$-plex is impossible (assuming P $\neq$ NP) [56].

For the backtracking approach, we show that every maximal $(d+1)$-plex is either a maximal clique, or contains a pair of non-adjacent vertices (u,v) such that the $(d+1)$-plex can be split into two parts – one part of size at most $2d-2$ consisting of vertices that are non-adjacent to either u or v, and the other of size at most c consisting of (a subset of) common neighbors of u and v. Since the number of pairs of non-adjacent vertices in the given c-closed graph is equal to the number of edges in its complement graph, m, this gives the maximum number of maximal $(d+1)$-plexes as $O(mn^{2d-2}2^c)$ and the enumeration algorithm follows.

For the three-step approach, we use $\mathcal{M}_d(N)$ – the maximum number of maximal $(d+1)$-plexes in an N vertex graph. (Equivalently, $\mathcal{M}_d(N)$ is the number of maximal subgraphs of degree at most d in an N vertex graph.) For the combinatorial bound we need an upper bound on $\mathcal{M}_d(N)$. A recent result shows that for every fixed d there is a constant $\kappa_d < 2$ such that $\mathcal{M}_d(N) \leq \kappa_d^N$ [81].

Determining a tight bound for $\mathcal{M}_d(N)$ appears to be challenging. To the best of our knowledge, the only tight bound is the Moon-Moser theorem stating that $\mathcal{M}_0(N) \leq 3^{N/3} \approx 1.442^N$. One of our contributions is to give a tight bound for $\mathcal{M}_1(N)$: $\mathcal{M}_1(N) \leq 10^{N/5} \approx 1.585^N$ which requires a much more involved proof than the Moon-Moser theorem (see the full version of the paper [7] for the proof).[3]

In the second step of the three-step approach, we give an FPT bound with a smaller (than in the case of backtracking) exponential factor $O(n^{2d} \cdot \kappa_d^c)$ using a more careful analysis of the structure of a $(d+1)$-plex. (Example 21 shows that the exponential dependence n^d is necessary.) Moreover, using the tight bound for $\mathcal{M}_1(N)$ we give a stronger bound $O(n^2 \cdot 10^{c/5})$ for the number of maximal 2-plexes in a c-closed graph on n vertices.

To convert the tighter bound into an enumeration algorithm and complete the third step, the simplest approach is to apply black-box one of the recent polynomial delay algorithms for efficiently listing $(d+1)$-plexes [10, 15]. E.g., Berlowitz et al. [10] give an algorithm which enumerates all maximal $(d+1)$-plexes in time $O((d+1)^{2d+2}p(n))$ per maximal $(d+1)$-plex,

[2] Similar result for $(\boldsymbol{d+1})$-plexes was proved independently and concurrently with the previous version of this paper by Koana, Komusiewicz, and Sommer [47]. The results in [47, 48] apply more generally to the class of weakly $\boldsymbol{c}$-closed graphs defined in [32, 33] (The paper [47] also includes several results showing polynomial-size kernels for various problems in weakly $\boldsymbol{c}$-closed graphs, an important direction that is not pursued here.)

[3] The induced subgraphs with maximum degree at most one are also called *dissociation sets* [76]. Thus, we show that the number of maximal dissociation sets in an N-vertex graph is at most $10^{N/5}$.

where $p(n)$ is a polynomial in n. By the FPT bound, the enumeration algorithm runs in FPT time. However, we can obtain a better running time by translating our proof of the FPT bound into a bespoke enumeration algorithm.

Bounded co-degeneracy. We say that a graph has *co-degeneracy* at most d if its complement is d-degenerate. (Recall that a graph is d-degenerate if every induced subgraph has at least one vertex with degree at most d.) In Section 5 we give, for each fixed d, FPT algorithms for enumerating maximal subgraphs with co-degeneracy at most d.

For the backtracking approach, we first show that every subgraph with bounded co-degeneracy is either a clique, or the degeneracy ordering of the complement of the subgraph contains an edge that splits the subgraph into three parts; two of whose sizes are bounded ($2d-2$ and c, respectively) and the third is a maximal independent set (in the complement graph) which can be discovered using the algorithm for enumerating cliques. This gives a bound of $O(cm^2n^{2d}4^c)$ on the number of maximal subgraphs with co-degeneracy d and an enumeration algorithm follows.

For the three-step approach, for the combinatorial bound, we define $\mathcal{D}_d(N)$ to be the maximum number of maximal subgraphs with co-degeneracy at most d in an arbitrary N-vertex graph. For every fixed d there is a constant $\gamma_d < 2$ such that $\mathcal{D}_d(N) \leq \gamma_d^N$, see [65].

For the FPT bound, we show that the number of maximal subgraphs with co-degeneracy at most d is at most $O(n^{8d} \cdot \mathcal{D}_d(2dc)) \leq O(n^{8d} \cdot \gamma_d^{2dc})$. The idea is to show that there are two types of maximal subgraphs with co-degeneracy at most d: either they have the structure of a generalized co-star, or we can find $2d$ pairs of non-adjacent edges such that the maximal subgraph is contained in the common neighborhoods of these non-adjacent pairs and an additional $4d$ vertices. Counting generalized stars reduces to counting cliques, and we control the other case using the c-closed condition.

An FPT algorithm is obtained by applying the recent enumeration algorithm [26] that lists all maximal subgraphs with bounded degeneracy in time $O(mn^{d+2})$ per maximal subgraph.

Bounded co-treewidth. A graph is said to have *co-treewidth* at most t if its complement has treewidth at most t. The class of graphs with co-treewidth at most t is denoted by $\mathcal{T}_t$. In Section 6, we give, for each fixed t, FPT algorithms for enumerating $\mathcal{T}_t$-graphs using (only) the three-step approach.

Obtaining non-trivial combinatorial bounds on the number of maximal subgraphs with (co-)treewidth at most t in an arbitrary N-vertex graph is an open question in graph theory, so we use the trivial upper bound of 2^N. (In any case, there are no known polynomial-delay algorithms for listing subgraphs of bounded (co-)treewidth that would allow us to algorithmically exploit (black-box) the savings that a better bound would give us.)

For our FPT bound, we show that for almost every maximal subgraph of bounded co-treewidth we can either find two pairs of non-adjacent vertices and show that the subgraph is contained in the common neighborhoods of these two pairs (plus t additional vertices), or else that the subgraph is a generalized co-star. In the former case we use the c-closure condition and reduce the latter case to counting maximal cliques in smaller graphs. We show that there are $O(n^{t+4}2^{2c})$ maximal subgraphs with co-treewidth at most t. Exponential dependence n^t is necessary, even when $c = 1$ (Example 32).

While there are no known polynomial-delay enumeration algorithms for listing maximal subgraphs of bounded (co-)treewidth, we show how to turn our FPT bound into an FPT algorithm for enumerating $\mathcal{T}_t$-graphs.

In the full version of the paper, we also extend these results to the subgraphs of bounded *local* co-treewidth.

1.3 Further related work

Polynomial-time solvable special cases of the Maximum Clique **problem and its generalizations in hereditary graph classes.** The problems we consider generalize the fundamental Maximum Independent Set and Maximum Clique problems. It is well known that polynomial-time and fixed-parameter tractability results for these problems require significant restrictions on the allowable input graphs. For example, it is known that Maximum Independent Set is NP-hard already for subcubic graphs, and for H-free graphs (for H connected) whenever H is not a path nor a subdivision of the claw ($K_{1,3}$) [2]. Similarly, the problem is $W[1]$-hard when parameterized by the solution size for H-free graphs whenever H is not a suitable generalization of a path or a subdivision of the claw [11] (obtained by replacing each vertex by a clique); in fact, the problem does not even admit an FPT constant-factor approximation for these graph classes (assuming Gap ETH) [30]. Known polynomial-time solvable special cases of the Maximum Independent Set problem include input graphs that are perfect (including (co-)chordal and (co-)bipartite graphs), P_6-free graphs [58], fork-free graphs [59], and other highly restricted classes [1, 19, 20, 41].

Real worlds graphs. It is widely accepted that the real-world graphs possess several nice properties that differentiate them from arbitrary graphs. The established ones include heavy-tailed degree distributions, a high density of triangles and communities, the small world property (low diameter), and triadic closure. Over the years there has been a lot of significant and influential work trying to capture the special structure of real-world graphs. The literature is almost entirely focused on the generative (i.e., probabilistic) models. A few most popular ones include preferential attachment [6], the copying model [53], Kronecker graphs [55], the Chung-Lu random graph model [21, 22], with many new models introduced every year. For example, already in 2006, the survey by Chakrabarti and Faloutsos [16] examines 23 different models. Generative approaches are very enticing as they, by definition, give an easy way of producing synthetic data, and are a good proxy for studying random processes on graphs. On the other hand, if one is to design an algorithm for real-world graphs with good worst-case guarantees, a hard choice of the exact model arises as there is a little consensus about which of the many models is the "right" one, if any.

An idea is to find algorithms that are not suited to any specific generative model, but only assume a deterministic condition. In other words, isolate a parameter of the real-world graphs that differentiates them from arbitrary graphs and use it give stronger guarantees for particular algorithms/problems. Fox, Roughgarden, Seshadhri, Wei, and Wein [32, 33] took this approach and introduced the class of c-closed graphs, where they showed that the maximum clique problem is FPT when parameterized by c.

There are only a few other algorithmic results in the same spirit. Notably, several problems can be solved faster for graphs with a power-law degree distribution: Barch, Cygan, Łacki, and Sankowski [13] gave faster algorithms for transitive closure, maximum matching, determinant, PageRank and matrix inverse; and Borassi, Crescenzi, and Trevisan [12] gave faster algorithms for diameter, radius, distance oracles, and computing the most "central" vertices by assuming additional axioms satisfied by real-world graphs.

Motivated by triadic closure, Gupta, Roughgarden, and Seshadhri [36] define triangle-dense graphs and proved relevant structural results. Informally, they proved that if a constant fraction of two-hop paths are closed into triangles, then (most of) the graph can be decomposed into clusters with diameter at most 2.

c-closed graphs. The c-closed graph model was introduced by Fox et al. [33] (see book chapter in [68] by some of the authors). After Fox et al. introduced c-closed graphs, Koana, Komusiewicz, and Sommer wrote several papers further exploting c-closure to design FPT algorithm for hard problems. In [50] they showed that the dominating set problem, the induced matching problem, and the irredundant set problem admit kernels of size $k^{O(c)}$, $O(c^7k^8)$, $O(c^{5/2}k^3)$ respectively; where k is the size of the solution. In [49], they show that enumerating maximal bicliques and $(d+1)$-plexes, is FPT with respect to c and study fixed parameter tractability of related hard problems with respect to the parameter c and size of the solution. In [51], they give the kernels for Capacitated Vertex Cover, Connected Vertex Cover, and Induced Matching of sizes $k^{O(c)}$, and $(ck)^{O(c)}$, respectively. Moreover, Koana and Nichterlein [52] explore the fixed parameter tractability of enumerating small induced subgraphs in a c-closed graph.

We note that the densest subgraph problem is trivially solvable in polynomial time for c-closed graph when $c = 1$, and NP-hard already for $c = 2$, see [66].

$(d + 1)$-plexes. The maximal cliques often fail to detect cohesive subgraphs. To address the issue, Seidman and Foster [69] in 1978 introduced the notion of $(d + 1)$-plex. We refer the reader to [75, 61, 64, 10, 24, 9] and references therein for an overview of the literature. The literature is mostly focused on heuristic algorithms for finding large $(d + 1)$-plexes or enumerating (several) maximal $(d + 1)$-plexes without providing any worst-case guarantees. For example, recently Conte, Firmani, Patrignani, and Torlone [25] gave a novel approach for the detection of 2-plexes. We point out that Lewis and Yannakakis [56] proved that the problem of finding a maximum $(d + 1)$-plex is NP-hard for any fixed d. Alternate proof is given in [5].

Counting and enumerating maximal subgraphs. Counting (maximal) induced subgraphs in an arbitrary N-vertex graph is a crucial part when it comes to design of faster exact algorithms. We mention a few related results. Moon and Moser [63] and also Miller and Muller [62] prove that the number of maximal cliques (equivalently maximal independent sets) in a graph on N vertices is at most $3^{N/3}$. Tomita, Tanaka and Takahashi [72] gave an algorithm for finding a maximum clique by enumerating all maximal cliques in time $O(3^{N/3})$.

Gupta, Raman and Saurabh [37, Theorem 4] show that the number of maximal 1-regular induced graphs in an N-vertex graph is at most $10^{N/5}$ and gave an algorithm for finding a maximum such subgraph with similar running time. Note that in any graph, the number of maximal induced matchings is not larger than the number of maximal induced subgraphs with degree at most 1. Therefore, it is somewhat surprising that the number of maximal induced subgraphs with degree at most 1 is also bounded by $10^{N/5}$, as we show in the full version of the paper [7]). The same paper [37] shows that for each integer r there is a constant $\rho_r < 2$, such that the number of maximal r-regular graphs in an N vertex graph is at most ρ_r^N.

Zhou, Xu, Guo, Xiao, and Jin [81] show that for each d there is a constant $\kappa_d < 2$ such that all maximal $(d + 1)$-plexes can be enumerated in time $O(\kappa_d^N N^2)$. Implicitly, they also show that the number of maximal $(d + 1)$-plexes is at most κ_d^N, i.e., $\mathcal{M}_d(N) \leq \kappa_d^N$.

Pilipczuk and Pilipczuk [65] show that for every fixed d there is a constant $\gamma_d < 2$ such that the number of maximal induced d-degenerate subgraphs in a graph on N vertices is at most γ_d^N, i.e., $\mathcal{D}_d(N) \leq \gamma_d^N$.

2 Preliminaries and complementary terminology

We consider finite, simple, undirected graphs. Let $G = (V, E)$ be a graph. We write $uv \in E(G)$ for an edge $\{u, v\} \in E(G)$ and we say that the vertices u and v are *adjacent* or that u is a *neighbor* of v and vice versa. If $w \in N_G(u) \cap N_G(v)$ we say that w is a *common* neighbor of u and v. For a vertex $v \in V(G)$ we denote by $N_G(v) = \{u \in V(G) : uv \in E(G)\}$ the *neighborhood* of v in G and $N_G[v] = N_G(v) \cup \{v\}$ the *closed neighborhood* of v in G. For $U \subseteq V(G)$, we define $N_G(U) = \cup_{u \in U} N_G(u) \setminus U$ and $N_G[U] = N_G(U) \cup U$. For simplicity, if the set U is given implicitly as a collection of vertices $u_1, \dots, u_\ell$ we write $N_G(u_1, \dots, u_\ell)$ instead of $N_G(\{u_1, \dots, u_\ell\})$, and similarly for $N_G[u_1, \dots, u_\ell]$. We drop the subscript G when the graph is clear from the context.

Let $W \subseteq V(G)$. The induced subgraph $G[W]$ is defined as the graph $H = (W, E(G) \cap \binom{W}{2})$, where $\binom{W}{2}$ is the set of all unordered pairs with elements in W. The graph $G[V(G) \setminus W]$ is also denoted as $G \setminus W$. Set W is *separator* in G if $G \setminus W$ has strictly more connected components than graph G. A connected component is *non-trivial* if it contains at least two vertices (equivalently at least one edge). The diameter of G, denoted $\mathrm{diam}(G)$, is the length of a longest shortest path among two vertices in G. If G is disconnected, then $\mathrm{diam}(G) = \infty$.

The *complement* of a graph $G = (V, E)$ is the graph $\overline{G} := (V, \binom{V}{2} \setminus E)$. We say that W is a clique (in G) if for any two vertices $u, v \in W$ we have $uv \in E(G)$. A set I is an *independent set* in G if I is a clique in $\overline{G}$. A set $U \subseteq V(G)$ is a vertex cover in G if $V(G) \setminus U$ is an independent set in G.

The *degree* of v in G is $\deg_G(v) = |N_G(v)|$, and the *maximum degree* of G is $\Delta(G) = \max_{v \in V(G)} \deg_G(v)$. Graph is *$d$-degenerate* (has degeneracy at most d) if every induced subgraph of $G[S]$ contains a vertex v such that $\deg_{G[S]}(v) \leq d$.

▶ **Definition 6** (Treewidth, [67]). *Let G be a graph. A* tree decomposition *of G is a pair $(T, \mathcal{W})$, where T is a tree and $\mathcal{W} = \{W_t \subseteq V(G) : t \in V(T)\}$ is a set of* bags *satisfying*

- $\cup_{t \in V(T)} W_t = V(G)$ *and for every edge uv in G there is bag W_t containing u and v; and*
- *if $t, t', t'' \in V(T)$ and t' lies on the path between t and t'' in T, then $W_t \cap W_{t''} \subseteq W_{t'}$.*

The width *of $(T, \mathcal{W})$ is $\max_{t \in V(T)}(|W_t| - 1)$. The* treewidth *of G, denoted* $\mathrm{tw}(G)$, *is the smallest number t such that there is a tree decomposition $(T, \mathcal{W})$ of G with width t.*

Co-degree, co-treewidth, and co-degeneracy refer to the degree, treewidth and degeneracy in the complement graph, respectively.

▶ **Definition 7** (c-closed, [32]). *A graph G is c-*closed *if any two non-adjacent vertices have at most $c - 1$ common neighbors.*

Finding the smallest c for which a given graph G is c-closed can be done by squaring the adjacency matrix in $O(n^\omega)$ time, where $\omega < 2.373$ is the matrix multiplication exponent.

A problem is said to be *fixed-parameter tractable* with respect to a parameter k if there is an algorithm that solves it in time $O(f(k)n^\alpha)$ where f can be an arbitrary function and α is a constant, for more details on parameterized algorithms and complexity we refer to [27]. Throughout the paper, unless otherwise stated the parameter is c, the number of vertices (resp. edges) in a c-closed graph (or its complement) is denoted by n (resp. m), and the number of vertices in a generic graph is denoted by N.

We state the main theorem of Fox et al. proving that maximal clique enumeration is FPT in c.

▶ **Theorem 8** (Fox et al.[32, 33]). *In any c-closed graph, a set of cliques containing all maximal cliques can be generated in time $O(p(n, c) + 3^{c/3}n^2)$, where $p(n, c) = O(n^{2+o(1)}c + c^{2-\omega-\alpha/(1-\alpha)}n^\omega + n^\omega \log(n))$ for the matrix multiplication exponent ω and $\alpha > 0.29$.*

Complementary terminology. We are interested in finding the dense subgraphs in c-closed graphs, but it is more convenient to present the rest of the paper in the complementary terminology. This means that we will be working with the complements of c-closed graphs. We will use $\mathbf{m}$ to denote the number of edges in the **co-graph** (short for complement graph) of a c-closed graph.

▶ **Proposition 9.** *A graph G is the complement of a c-closed graph if and only if for any two adjacent vertices u, v in G it holds $|V(G) \setminus N_G[u, v]| \leq c - 1$.*

As the notions of co-treewidth and co-degeneracy are already introduced in the complementary notions, it is clear that we are interested in the subgraphs of bounded treewidth and bounded degeneracy in the complement of a c-closed graph.

We provide an alternate definition of degenerate graphs, that follows by results of Matula-Beck [60].

Given an ordering of vertices $(v_1, \ldots, v_n)$, we will let $V^+(v)$ denote the set of vertices following v in the ordering, and $N^+(v)$ denote the neighbors of v that are after v in the ordering. Thus, $N^+(v) \subseteq V^+(v)$. Note that $N^+(v)$ and $V^+(v)$ depend on the ordering, but for brevity we do not it include in the notation as the ordering will always be clear from the context.

▶ **Definition 10** (Degeneracy Ordering). *An ordering of vertices $(v_1, \ldots, v_n)$ is a degeneracy ordering if for all $1 \leq i \leq n$, v_i is the minimum degree vertex in $G[\{v_i, \ldots, v_n\}]$, breaking ties lexicographically.*

▶ **Definition 11** (d-Degenerate Graph). *A graph $G = (V, E)$ is d-degenerate if there exists an ordering $(v_1, \ldots, v_n)$ such that for all $1 \leq i \leq n$, we have $|N^+(v_i)| \leq d$. The degeneracy ordering of a d-degenerate satisfies this property.*

We recall that whenever we say maximal subgraph this is referred to a maximal vertex induced subgraph.

3 Cliques

For enumerating cliques, we only consider the backtracking approach, as the three-step approach is already given by Fox et al. [33].

▶ **Definition 12** (Independent Set Backtracking Tree). *Let $G = (V, E)$ denote the co-graph of a c-closed graph and fix an ordering of the vertices. The backtracking tree of G is denoted as $T = (X, F)$ where X is a node-set and F a link-set (we will use nodes and links for the backtracking tree and vertices and edges for G). A node in X is labeled by a $U \subseteq V$, and a link is labeled by a $v \in V$. The tree has the following properties.*

- *The root node is labeled by V.*
- *All nodes that are labeled by an independent set are leaves.*
- *For all internal nodes labeled by U, there is a child node for each $v \in U$ labeled by $U' = V^+(v) \setminus N(v)$ with the corresponding link (U, U') labeled by v.*

The root node is at level 0 and the children of any vertex are at exactly one level lower than the vertex. We call every $P \cup Q$ an independent set path where P is a root-to-leaf path in T and Q is the last node label of P.

Consider any root-to-leaf path $P = (v_1, \ldots, v_k)$. By definition of T, we have $v_i \in V^+(v_{i-1}) \setminus N(v_1, \ldots, v_{i-1})$ for all $1 \leq i \leq k$. Hence, P is an induced independent set since $P \subseteq V \setminus N(P)$. Let the last node label of P be Q which is an independent set since it is a leaf label. Then, $P \cup Q$ is also an independent set since $Q \subseteq V \setminus N[P]$. Compiling the above conclusions, it follows that every independent set path in T indeed is an induced independent set in G. Moreover, by the fixed ordering, no two independent set paths correspond to the same independent set. Now the following converse theorem is fairly straightforward and it does not use the c-closure property.

▶ **Theorem 13.** *Every maximal independent set of G is an independent set path in the backtracking tree T.*

Proof. Consider a maximal independent set S of size k, and let $(v_1, \ldots, v_k)$ be the ordered form of S according to our fixed ordering (in Definition 12). Choose the minimum j such that $V^+(v_j) \setminus N(v_1, \ldots, v_j)$ is an independent set. We now show that $P = (v_1, \ldots, v_j)$ is a root-to-leaf path in T and that $Q = \{v_{j+1}, \ldots, v_k\}$ is the last node label of P; hence, S is an independent set path of T. Further observe that if P is a path starting at the root (a root-originating path), its last node must be a leaf by our choice of j.

We prove that P is a root-originating path by induction on j. For $j = 0$, this is vacuously true, and for $j = 1$, the claim holds since $v_1 \in V$. Now, consider some $j \geq 2$ and assume the inductive hypothesis for $j - 1$, so $(v_1, \ldots, v_{j-1})$ is a root-originating path. Since $v_j \in V \setminus N(v_1, \ldots, v_{j-1})$, since S is an independent set, and since v_j is of higher order than the vertices $v_1, \ldots, v_{j-1}$, we have $v_j \in V^+(v_{j-1}) \setminus N(v_1, \ldots, v_{j-1})$. Thus, by definition of T, the path P exists and is a root-originating path.

Next, since P is a root-to-leaf path, the last node label of P is $U = V^+(v_j) \setminus N(P)$. Since S is an independent set, for all $j < i \leq k$, we have $v_i \in U$ since v_i has higher order than any vertex in P. Further, if there exists a $v \in U \setminus \{v_{j+1}, \ldots, v_k\}$, we have an independent set $P \cup U$ whose subset is S, contradicting the maximality of S. Hence, $Q = U$ as required. ◀

The key argument that bounds the size of the backtracking tree follows. It shows a surprising connection with the c-closure parameter.

▶ **Lemma 14.** *The backtracking tree T has at most c levels.*

Proof. We show the lemma by showing that for every independent set S of size c, the set of its non-neighbours is also an independent set.

Let $U = V \setminus N[S]$ be the set of non-neighbours of S. We claim U is an independent set. If $G[U]$ were to contain an edge $\{u, v\}$, then $S \subseteq V \setminus N[u, v]$ since $S \cup \{u\}$ and $S \cup \{v\}$ are independent sets. Since $|S| = c$, we breach the c-closed condition; thus, U must be an independent set. Hence T has at most c levels, since every node at level c is a leaf node. ◀

▶ **Theorem 15.** *The size of the backtracking tree T is $O(cm2^c)$.*

Proof. For any non-leaf node label U, the induced subgraph $G[U]$ contains an edge. For any edge $e = \{u, v\}$, let us count the number of such tree nodes such that $G[U]$ contains e. Let P be the path in T from the root to U. Then we have $P \subseteq V \setminus N[u, v]$ since $P \cup \{u\}$ and $P \cup \{v\}$ are independent sets. Since $|V \setminus N[u, v]| < c$ and all paths are unique, the edge e can appear in at most $\binom{c}{i}$ non-leaf nodes at level i. In other words, the number of occurrences of edge e at level i can be at most $\binom{c}{i}$. Thus the total number of occurrences of all edges at level i is at most $\sum\limits_{e \in E(G)} \binom{c}{i} = m\binom{c}{i}$. In other words, if we let $\boldsymbol{U}_i$ be the set of all non-leaf nodes at level i, then $\sum\limits_{U \in \boldsymbol{U}_i} |E(U)| \leq m\binom{c}{i}$. Note that this means that $|\boldsymbol{U}_i| \leq m\binom{c}{i}$.

The number of isolated vertices in $G[U]$ is less than c since $G[U]$ contains an edge, and the number of non-isolated vertices in $G[U]$ is at most $2|E(U)|$. Hence, the node labeled by

U can have at most $2|E(U)| + c$ children. Thus the number of all children produced at level i (i.e. the total number of nodes in the tree T at level $i + 1$) is at most

$$\sum_{U \in \boldsymbol{U}_i} (2|E(U)| + c) \leq 2 \sum_{U \in \boldsymbol{U}_i} (|E(U)|) + c|\boldsymbol{U}_i| \leq 2m\binom{c}{i} + cm\binom{c}{i} = (2 + c)m\binom{c}{i},$$

Thus, the total number of nodes in T is given by

$$(2 + c)m \sum_{i=0}^{c} \binom{c}{i} = O(cm2^c)$$

as desired. ◀

To construct the children for every internal node of this tree will take $O(n^2)$ time, so to build T and enumerate a superset of maximal independent sets in G (equivalently, a superset of maximal cliques in the c-closed graph whose complement is G) will take $O(cmn^2 2^c)$ time. Thus, the backtracking algorithm runs in FPT time with parameter c. Interestingly, the backtracking algorithm does not need to know the value of the parameter c.

▶ **Corollary 16.** *The backtracking algorithm enumerates a superset of all maximal independent sets in the co-graph of a c-closed graph in time $O(cmn^2 2^c)$, where m is the number of edges in the co-graph and n is the number of vertices.*

4 $(d + 1)$-plexes

For any fixed d, we show that the number of maximal subgraphs with degree at most d in the complement of a c-closed graph admits an FPT bound. This implies that the number of maximal $(d + 1)$-plexes in a c-closed graph admits an FPT bound and an FPT enumeration algorithm.

We give proofs using both approaches, starting with the approach that uses backtracking as a subroutine.

▶ **Theorem 17.** *Let G be the complement of a c-closed graph. The number of maximal subgraphs with degree at most d in G is bounded by $O(mn^{2d-2}2^c)$.*

Proof. We count two types of maximal subsets S that induce a subgraph with degree at most d:

- subsets S for which $G[S]$ is edgeless, and
- subsets S for which $G[S]$ contains at least one edge.

If $G[S]$ is a maximal subgraph with degree at most d and $G[S]$ is edgeless, then S is also a maximal independent set in G. By Corollary 16, a superset of all maximal independent sets in G can be enumerate in time $O(cmn^2 2^c)$.

Suppose $G[S]$ has an edge, say (u, v). Let $Y = S \cap N(u, v)$ and $Z = S \setminus N[u, v]$, then by the c-closed condition, $|Z| \leq c$. Moreover, since Y consists of neighbors of (u, v) and u and v can have at most $d - 1$ neighbors, $|Y| \leq 2d - 2$. For any edge, there are 2^c possible choices for Z and $O(n^{2d-2})$ choices for Y. Hence, the number of maximal $(d + 1)$-plexes containing at least one edge is $O(mn^{2d-2}2^c)$. By simply enumerating all possible choices for Y and Z for every edge and combining them, in total time $O(mn^{2d-2}2^c)$, we will have enumerated a superset of all $(d + 1)$-plexes containing an edge. ◀

▶ **Corollary 18.** *Let G be the complement of a c-closed graph. A superset of all maximal subgraphs with degree at most d in G can be enumerated in time $O(mn^{2d-2}2^c + cmn^2 2^c)$.*

4.1 Enumerating $(d+1)$-plexes via the three-step approach

Next, we give an alternate bound with exponential improvement in c is using the three step approach. The running time bound we obtain is $O(n^{2d} \cdot \kappa_d^c \cdot p(c))$ where $\kappa_d < 2$ is the root of $x^{d+4} - 2x^{d+3} + 1 = 0$; and for a polynomial p.

Combinatorial bound. Our bound depends on an extension of $\mathcal{M}_d(N)$. For a (not necessarily c-closed) graph G and $P \subseteq V(G)$, the number of maximal subgraphs containing P and with degree at most d is denoted by $\mathcal{M}_d(G;P)$. Analogously, $\mathcal{M}_d(N+p;p)$ is the maximum value $\mathcal{M}_d(G;P)$ takes over all graphs on $N+p$ vertices and all sets $P \subseteq V(G)$ with size p. In particular, $\mathcal{M}_d(N) = \mathcal{M}_d(N;0)$. By adding isolated vertices, it is easy to see that $\mathcal{M}_d(N+p;p) \leq \mathcal{M}_d(N+p';p')$ for all $p \leq p'$.

By closely examining the result by Zhou et al. [81, Theorem 1], we note that they implicitly show that for each d and every p there is a constant $\kappa_d < 2$ such that $\mathcal{M}_d(N+p;p) \leq \kappa_d^N$. More precisely, they show that the bound holds if κ_d is the positive solution of $x^{d+3} - 2x^{d+2} + 1 = 0$. For $d = 0, \ldots, 4$ we have $\kappa_d = 1.618, 1.839, 1.928, 1.966$ and 1.984. To the best of our knowledge, next to the Moon-Moser theorem, these are the best (and only) existing bounds for $\mathcal{M}_d(N)$ and $\mathcal{M}_d(N+p;p)$.

The Moon-Moser theorem states that $\kappa_0 = 3^{1/3}$ suffices. We prove a tight upper bound on $\mathcal{M}_1(N)$. In other words we show that we can set $\kappa_1 = 10^{1/5} \leq 1.585$. The proof uses similar recursive bound(s) as in the Moon-Moser theorem, and in the proof for 1-regular graphs given by Gupta et al. [37, Theorem 4], but our proof requires a significantly more extensive case analysis. See [7].

▶ **Theorem 19.** $\mathcal{M}_1(N) \leq 10^{N/5} \leq 1.585^N$.

To see that the bound is tight consider any N a multiple of 5. The graph consisting of $\frac{N}{5}$ copies of K_5 contains $10^{N/5}$ maximal subgraphs with degree at most 1. The same number of subgraphs is attained if we remove a matching from each of the K_5s.

FPT bound. Our next goal is to give an upper bound on the number of subgraphs with degree at most d in the complement of a c-closed graph using $\mathcal{M}_d(N+p;p)$ for $d > 1$, and $\mathcal{M}_1(N)$. For the case when $d = 0$, we already have Theorem 8 which we use in the proof.

▶ **Theorem 20.** *Let G be the complement of a c-closed graph. The number of maximal induced subgraphs with degree at most d in G, is bounded by $2n^{2d} \cdot \mathcal{M}_d(c-1+2d;2d)$. Moreover, for $d=1$ the bound simplifies to $2n^2 \cdot \mathcal{M}_1(c-1)$.*

Proof. Similar to the proof for the first bound for counting $(d+1)$-plexes, we count two types of maximal subsets S that induce a subgraph with degree at most d:

- subsets S for which $G[S]$ is edgeless, and
- subsets S for which $G[S]$ contains at least one edge.

If $G[S]$ is a maximal subgraph with degree at most d and $G[S]$ is edgeless, then S is also a maximal independent set in G. By Theorem 8, the number of maximal independent sets in G is bounded by $n^2 \cdot \mathcal{M}_0(c-1)$. By definition, it is not hard to see that $\mathcal{M}_0(c-1) \leq \mathcal{M}_0(c-1+d;d) \leq \mathcal{M}_d(c-1+2d;2d)$ holds. Therefore, in order to prove the theorem, it suffices to show that the number of maximal subgraphs that contain an edge and with degree at most d is bounded by $n^{2d} \cdot \mathcal{M}_d(c-1+2d;2d)$.

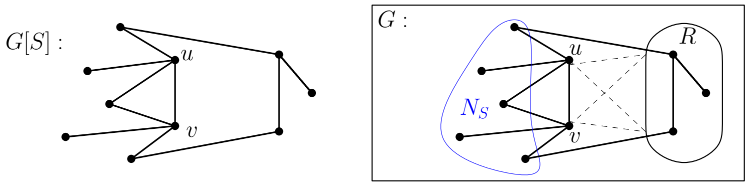

Figure 1 Proof of Theorem 20. Left: $G[S]$ represents an induced subgraph with maximum degree 4. Right: depiction of $G[S]$ within G. Recall that $R = V(G) \setminus N_G[u, v]$ and that $|R| \leq c - 1$ as G is the complement of a c-closed graph. The dashed lines represent non-edges.

We refer to Figure 1. Let uv be an edge in G. Suppose that S is a maximal set such that $\Delta(G[S]) \leq d$ and $u, v \in S$. Let $N_S = S \cap N_G(u, v)$. By the maximum degree assumption and since u and v are adjacent to each other, there are at most $2d - 2$ vertices in N_S. To prove the theorem, we show that the number of maximal sets S satisfying the following two

- degree of $G[S]$ is at most d, and
- S contains $\{u, v\}$ and $S \cap N(u, v) = N_S$ (S contains $\leq 2d$ fixed vertices);

is bounded by $\mathcal{M}_d(c - 1 + 2d; 2d)$.

We claim that any such maximal set S also induces a maximal subgraph (with the same properties) in graph $G[\{u, v\} \cup N_S \cup R]$ where $R = V \setminus N_G[u, v]$. Namely, we can obtain $G[\{u, v\} \cup N_S \cup R]$ from G by removing some vertices that are not in S. As removal of such vertices does not influence the maximality of S, it follows that S induces a maximal subgraph (with the above stated properties) in $G[\{u, v\} \cup N_S \cup R]$.

Since G is the complement of a c-closed graph and by definition of R, we have $|R| \leq c-1$. Let $k = |\{u, v\} \cup N_S|$. Then, by definition of $\mathcal{M}_d(c - 1 + k; k)$ it follows that the number of maximal sets S that induce a subgraph with degree at most d and contain $\{u, v\} \cup N_S$ is bounded by $\mathcal{M}_d(c - 1 + k; k)$. As $|\{u, v\} \cup N_S| = k \leq 2d$ we have $\mathcal{M}_d(c - 1 + k; k) \leq \mathcal{M}_d(c - 1 + 2d; 2d)$ and the proof follows.

Next, we deal with the case $d = 1$. The proof is largely the same and we make a small change in the way we count the subsets S that contain u, v. As the maximum degree of $G[S]$ is at most 1 and since u and v are adjacent to each other we have that $N_S = \emptyset$. We claim that if S is maximal set with degree at most 1 in G containing uv, then $S \setminus \{u, v\}$ is a maximal set with degree at most 1 in $G[R]$.

For a contradiction, suppose that $S \setminus \{u, v\}$ is not a maximal such set, and let $S' \subseteq R$ such that $S \setminus \{u, v\} \subset S'$ and $\Delta(G[S']) \leq 1$. Since $S' \subseteq R = V(G) \setminus N[u, v]$ it follows that u and v are non-adjacent to S'. Thus, $\Delta(G[S' \cup \{u, v\}]) \leq 1$ contradicting maximality of S.

It follows that the number of maximal subsets S with $\Delta(G[S]) \leq 1$ and that contain edge uv is at most $\mathcal{M}_1(c - 1)$. Thus, the number of maximal subsets S with $\Delta(G[S]) \leq 1$ is bounded by $n^2\mathcal{M}_0(c - 1) + n^2\mathcal{M}_1(c - 1) \leq 2n^2\mathcal{M}_1(c - 1)$. ◀

We give an example showing that the dependency on n and d cannot be improved.

▶ **Example 21.** Any complete bipartite graph is the complement of a 1-closed graph as any two adjacent vertices have no common non-neighbors. Let $K_{i,j}$ be the complete bipartite graph with parts of size i and j. It is easy to see that the number of maximal subgraphs with degree at most d in $K_{\ell,\ell}$ for $\ell > d$, is at least $\Omega(\ell^{2d}) = \Omega\left(\frac{|V(K_{\ell,\ell})|^{2d}}{2^{2d}}\right)$ for any fixed d.

Enumeration. Equipped with Theorem 20 it is straightforward to obtain an algorithm, with running time similar to the FPT bound, for enumeration of all maximal $(d + 1)$-plexes in c-closed graph. A simple way is to run a polynomial delay algorithm for listing all maximal subgraphs with degree at most d on the complement graph [10]. The FPT bound then implies that the enumeration algorithm indeed runs in FPT time. A better running time can be obtained if the enumeration algorithm is incorporated directly into the proof of the FPT bound. We sketch it below.

▶ **Corollary 22** (Restatement of Theorem 2). *For c-closed graphs and a fixed $d \geq 0$, there is an algorithm running in time $O(n^{2d} \cdot \kappa_d^c \cdot p(c))$ for enumerating $(d+1)$-plexes, where $\kappa_d < 2$ is the root of $x^{d+4} - 2x^{d+3} + 1 = 0$; and for a polynomial p. For 2-plexes, a stronger bound $O(n^2 \cdot 10^{c/5} \cdot p(c))$ applies.*

Proof of Corollary 22. We enumerate all maximal subgraphs with degree at most d in the complement graph. If a maximal subgraph with degree at most d is edgeless, then it is also a maximal independent set and we use the algorithm by Fox et al. [33] stated in Theorem 8.

Hence, we only need to enumerate the maximal subgraphs with degree at most d and that contain at least one edge. Similarly, as in the proof of Theorem 20 once we fix an edge uv, and the neighbors of u and v the rest of maximal induced subgraph is contained in a subset of at most $c-1$ vertices. By applying the polynomial delay algorithm [10] to these vertices, we can obtain all maximal subgraphs of degree at most d that contain the fixed vertices in time $O(\mathcal{M}_d(c-1+2d; 2d) \cdot p(c)) \leq O(\kappa_d^c \cdot p(c))$ for a polynomial p. ◀

5 Bounded co-degeneracy

As with $(d+1)$-plexes, we first give the result with the backtracking approach.

Any d-degenerate graph (with possible isolated vertices) can either be an independent set or it can be separated into 3 components, characterized by an edge in the graph. This decomposition is unrelated to the c-closed property, but we exploit this structure for faster enumeration in a c-closed co-graph.

▶ **Lemma 23.** *Consider a d-degenerate graph H with the degeneracy ordering of $(u_1, \dots, u_n)$. If H is not an independent set, there exists an edge (u_s, u_t) such that for*

$$X = \{u_1, \dots, u_{s-1}\}, \qquad Y = \{u_{s+1}, \dots, u_{t-1}\}, \qquad Z = \{u_{t+1}, \dots, u_n\},$$

X is an independent set, Y is a subset of $V \setminus N[u_s, u_t]$, and Z is a subset $V \setminus N[u_s, u_t]$ with at most $2d-2$ additional vertices.

Proof. Choose minimum t such that u_t is a terminal vertex of an edge in H. Then choose maximum s such that (u_s, u_t) is an edge in H (this must exist since H is not an independent set). By the minimality of t, X is an independent set.

By the minimality of t, u_s is not adjacent to any vertex in Y. By the maximality of s, u_t is not adjacent to any vertex in Y. Hence, $Y \subseteq V \setminus N[u_s, u_t]$.

Furthermore, since u_t and u_s are connected, each can be adjacent to at most $d-1$ vertices in Z to ensure the d-degeneracy condition. Thus, the rest of the vertices in Z are non-adjacent from both u_t and u_s. ◀

Notice that since $|V \setminus N[u_s, u_t]| < c$ by the c-closed condition, we have $|Y| < c$ and $|Z| < 2d - 2 + c$. Furthermore, note that if H is maximal, then so is the independent set X.

▶ **Theorem 24.** *[Restatement of Theorem 3] For c-closed graphs and a fixed $d \geq 0$, there is an algorithm running in time $O(cm^2 n^{2d} 4^c + cmn^2 2^c)$ that outputs a set containing all maximal induced subgraphs with co-degeneracy d in the c-closed graph, where m is the number of edges in the complement graph of the c-closed graph.*

Proof. We describe an algorithm that generates supersets of all maximal induced d-degenerate subgraphs in a c-closed co-graph G. (We can check in linear time whether each such subgraph is truly d-degenerate.)

Start with any edge $\{u, v\}$ and pick an orientation (say) (u, v). Then, we construct all possible choices of Y and Z, which takes $O(n^{2d-2}2^c)$ time. Next, we choose Y and Z such that $G[Y \cup Z \cup \{u, v\}]$ is a d-degenerate subgraph whose degeneracy ordering is (u, Y, v, Z). Then, we can build a set S of vertices s where s is the first vertex in the degeneracy ordering of $G[\{s, u, v\} \cup Y \cup Z]$. Lastly, we enumerate all maximal independent sets X in $G[S]$ which takes $O(cmn^2 2^c)$ time by Corollary 16. Then, any maximal d-degenerate subgraph H of G is $X \cup Y \cup Z \cup \{u, v\}$ for some chosen X, Y, Z, and $\{u, v\}$ according to the above algorithm. The total run-time for this algorithm is $O(cm^2n^{2d}4^c + cmn^2 2^c)$. ◀

5.1 Enumerating subgraphs of bounded co-degeneracy with the three-step approach

We give another FPT algorithm for enumerating all maximal subgraphs with degeneracy at most d in the complement of a c-closed graph using the three-step approach. For this (as well as for bounded-treewidth) we use the notion of a generalized star and of an (ℓ, k)-partition. We define these below. The bound obtained using this approach is worse than the algorithm described above but we include it for the sake of completeness and since the same notions are used in the case of bounded treewidth. The proof uses an alternate characterization of the structure of a bounded-degeneracy graph in the co-graph of a c-closed graph. For details and missing proofs see the full version.

Generalized stars. We say that that a graph H is a k-star if there is a partition $\{A, B\}$ of $V(H)$ such that $|A| \leq k$ and B is an independent set. Equivalently, graph is a k-star if and only if it has a vertex cover of size at most k. We say that A is the *head* of the k-star H, and B is the set of *tails*. A k-star is *proper* if every tail is adjacent to at most $k - 1$ vertices (in the head). In particular, any $(k - 1)$-star is a proper k-star. We note that an edgeless graph is a proper 1-star and a (vertex disjoint) union of an edgeless graph and a star is a proper 2-star.

▶ **Lemma 25.** *Let G be the complement of a c-closed graph. The number of subsets $S \subseteq V(G)$ that induce a proper k-star with a maximal set of tails is at most $2 \cdot n^{k+2} \cdot \mathcal{M}_0(c - 1)$.*

Note that we only require that the set of tails is maximal: there is no proper k-star with the same head and a strictly larger (inclusion-wise) set of tails.

Proof of Lemma 25. Let $A \subseteq V(G)$ be a set of at most k vertices. For a proper k-star with head A and the set of tails B it holds that B is an independent set in $G \setminus A$. Suppose that the B is the maximal set of tails for the k-star $G[A \cup B]$.

Let X be the set of vertices $v \in V(G) \setminus A$ that are adjacent to every vertex in A. If $|A| = k$, then since $G[A \cup B]$ is proper and by maximality of the tail, it follows that B is a maximal independent set in $G \setminus (A \cup X)$. If $|A| < k$ then by the maximality of tail, B is a maximal independent set in $G \setminus A$.

By Theorem 8 there are at most $n^2\mathcal{M}_0(c - 1)$ maximal independent sets in $G \setminus A$ and similarly at most $n^2\mathcal{M}_0(c - 1)$ maximal independent sets in $G \setminus (A \cup X)$. The lemma follows. ◀

Good (ℓ, k)-partitions. Next, we introduce a definition that captures the property of graphs we can count by fixing several edges. Informally, we say that a graph H admits a good (ℓ, k)-partition if there are k edges and a set A_0 on at most ℓ vertices such that the rest of the graph can be partitioned into non-neighborhoods of the edges. We show that the subgraphs admitting a good (ℓ, k)-partition are easy to count.

▶ **Definition 26.** *We say that a graph H admits a* good (ℓ, k)-partition *if there exist k edges $e_1, \dots, e_k$ and a $(k+1)$-partition $\{A_0, A_1, \dots, A_k\}$ of the set $V(H) \setminus \left(\cup_{i=1}^{k} e_i\right)$ such that $N_H(e_i) \cap A_i = \emptyset$ for every $i \in [k]$ and $|A_0| \leq \ell$.*

▶ **Lemma 27.** *Let G be the complement of a c-closed graph. The number of subsets $S \subseteq V$ for which graph $G[S]$ admits a good (ℓ, k)-partition, is bounded by $n^{\ell+2k} \cdot 2^{k(c-1)}$.*

Proof of Lemma 27. Let H be induced subgraph of G that let $e_1, \dots, e_k$ and $A_0, \dots, A_k$ be the edges and sets defining a good (ℓ, k)-partition of H. To prove the lemma, it suffices to show that the number of induced subgraphs that admit a good (ℓ, k)-partition with the same edges $e_1, \dots, e_k$ and the same set A_0 is bounded by $2^{k(c-1)}$.

Denote with U the vertices of G that are neither incident to the edges $e_1, \dots, e_k$ nor in the set A_0, i.e., $U = V(G) \setminus (A_0 \cup_{i=1}^{k} e_i)$. By definition of a good (ℓ, k)-partition, for any induced subgraph with a good (ℓ, k)-partition $e_1, \dots, e_k$ and $A_0, A'_1, \dots A'_k$ it holds $A'_i \subseteq U \setminus N_G(e_i)$ for each $i \in [k]$. Since G is complement of a c-closed graph, it follows that $|U \setminus N(e_i)| \leq c-1$ for each $i \in [k]$. Hence, there are at most $2^{k(c-1)}$ induced subgraphs $G[S]$ that admit a good (ℓ, k)-partition with A_0 and the edges $e_1, \dots, e_k$. The lemma follows. ◀

.

We obtain an FPT algorithm for bounded-degeneracy graphs in the following way.

Combinatorial bound. Recall that the maximum number of maximal d-degenerate subgraph with in an arbitrary N-vertex graph is denoted by $\mathcal{D}_d(N)$. Pilipczuk and Pilipczuk [65] show that for every d there is a constant $\gamma_d < 2$ such that $\mathcal{D}_d(N) \leq \gamma_d^N$.

FPT bound. It can be shown that a d-degenerate graph is either a $4d$-star or admits a good $(4d, 2d)$-partition. Then, by Lemmas 25 and 27 we obtain an FPT upper bound.

▶ **Theorem 28.** *Let G be the complement of a c-closed graph. The number of maximal d-degenerate subgraphs in G is bounded by $O(n^{8d}\mathcal{D}_d(2dc))$.*

Enumeration. Maximal d-degenerate subgraphs can be listed in time $O(mn^{d+2})$ per maximal subgraph [26]. We obtain the following corollary.

▶ **Corollary 29.** *For each fixed integer d, there is a constant $\gamma_d < 2$ and an FPT algorithm running in time $O(n^{9d+4} \cdot \gamma_d^{2dc})$ for enumerating all maximal subgraphs with co-degeneracy at most d in a c-closed graph G.*

6 Bounded co-treewidth

We give FPT algorithms for enumerating all maximal subgraphs of bounded treewidth in the complement of a c-closed graph using (only) the three-step approach. For the combinatorial bound, we use the trivial upper bound 2^N for the number of maximal subgraphs of bounded treewidth in an N-vertex graph. For the enumeration, we are unaware of any polynomial delay algorithms for enumerating maximal subgraphs of bounded treewidth. Nevertheless, the proof of the FPT bound is easily turned into an FPT enumeration algorithm. Therefore, we are only concerned with proving the FPT bound. We refer to the full version version of this paper [7] for an extension of the upper bound (and consequently the algorithm) to the subgraphs of bounded local treewidth.

FPT bound. To count star-like maximal subgraphs with treewidth at most t in the complement of a c-closed graph, we use Lemma 25. The counting reduces to counting maximal independent sets in smaller graphs.

To count the non-star-like graphs with treewidth at most t, we use Lemma 27. The lemma shows how to count all subgraphs that contain several edges and show that any other vertex is non-adjacent to at least one of the fixed edges.

The upper bound is proved by combining the two mentioned cases. More precisely, we show that any subgraph of bounded treewidth is counted by either Lemma 25 or Lemma 27.

We present the main theorem of this section.

▶ **Theorem 30.** *Let G be the complement of a c-closed graph and let $t \in \mathbb{N}$. The number of maximal subsets $S \subseteq V(G)$ for which $\mathrm{tw}(G[S]) \leq t$ is at most $3n^{t+4}2^{2(c-1)}$.*

Before we prove the theorem, we mention that the class of all graphs with treewidth at most t contains all "proper" $(t+1)$-stars but not all $(t+1)$-stars. Simply, K_{t+2} is a $(t+1)$-star but has treewidth $t+1$. The proof relies on the following claim.

▷ Claim 31. Let $S \subseteq V(G)$ such that $\mathrm{tw}(G[S]) \leq t$. Then, $G[S]$ is either a proper $(t+1)$-star or admits a good $(t,2)$-partition.

Proof of Claim 31. Let $(T, \mathcal{W})$ be a tree decomposition of $G[S]$ of width at most t; W_a is the bag corresponding to vertex $a \in V(T)$ and $\mathcal{W}$ is the set of bags, i.e., $\mathcal{W} = \{W_a : a \in V(T)\}$. Without loss of generality, we may assume that for any edge $ab \in E(T)$ the bags W_a and W_b are crossing, i.e., it holds $W_a \setminus W_b \neq \emptyset \neq W_b \setminus W_a$. On the contrary, if $W_a \subseteq W_b$ we can simply remove the vertex a and the bag W_a and reconnect the tree in the natural way to obtain a tree decomposition with the same width and a smaller tree.

Let $ab \in E(T)$ and let T_a, T_b be the trees in $T \setminus ab$. Tree T_a (resp. T_b) is the tree in $T \setminus ab$ containing the vertex a (resp. b). It is easy to check that there is no edge between $U_a := \cup_{t \in V(T_a)} W_t \setminus (W_a \cap W_b)$ and $U_b := \cup_{t \in V(T_b)} W_t \setminus (W_a \cap W_b)$. In other words, $W_a \cap W_b$ is a separator of $G[S]$ whenever $U_a \neq \emptyset \neq U_b$. Since the adjacent bags in T are crossing we do have $U_a \neq \emptyset \neq U_b$. Moreover, since $|W_a|, |W_b| \leq t+1$ and $W_a \setminus W_b \neq \emptyset$ it follows that $|W_a \cap W_b| \leq t$. Thus, $W_a \cap W_b$ is a separator of size at most t in $G[S]$ for every $ab \in E(T)$. If U_a and U_b both contain an edge, say e_1 and e_2 respectively, then $G[S]$ admits a good $(t,2)$-partition. Namely, we can set $A_0 = W_a \cap W_b$, $A_1 = U_b$, and $A_2 = U_a$. Therefore, we assume that for each edge $ab \in E(T)$ at least one of U_a or U_b is an independent set. We show, that this implies that $G[S]$ is a proper $(t+1)$-star.

If for some $ab \in E(T)$ both U_a and U_b are independent sets, then so is $U_a \cup U_b$. As $U_a \cup U_b = S \setminus (W_a \cap W_b)$ and $|W_a \cap W_b| \leq t$, it follows that $G[S]$ is a t-star. Hence, for the rest of the proof we assume that for each edge $ab \in E(T)$ either U_a or U_b is not an independent set. Combining with the previous paragraph, we have that for each $ab \in E(T)$ exactly one of U_a, U_b is an independent set and the other one is not.

Such a property gives a natural orientation of the edges in T. In particular, if U_a is an independent set we orient the edge ab as (a,b) and say that edge ab is oriented *towards* b. Otherwise we orient ab as (b,a) as say that ab is oriented *towards* a. Since T is a tree, there is a vertex $s \in V(T)$ such that all incident edges are oriented towards s. (Start with an arbitrary vertex $x \in V(T)$ and move to any vertex $y \in N_T(x)$ such that xy is oriented towards y. We keep iterating until we encounter a vertex s such that all incident edges are oriented towards s. The process terminates as T is a tree.) We show that $S \setminus W_s$ is an independent set.

Suppose on the contrary that there is an edge $uv \in G[S] \setminus W_s$. By the definition of tree decomposition $(T, \mathcal{W})$, the vertices u and v are both contained in some bag W_p for $p \in V(T)$. Moreover, it holds that $p \neq s$. Let q be the neighbor of s on the undirected s-p path in T (possibly $q = p$). Then, U_q is not an independent set: we have $uv \in U_q$ since $W_p \setminus W_s \subseteq U_q$. It follows that the edge sq is oriented from s to q. A contradiction with the choice of s. As $|W_s| \leq t + 1$ we conclude that S is a $(t + 1)$-star.

It remains to show that the $(t + 1)$-star is proper, i.e., that every vertex $v \in S \setminus W_s$ is adjacent to at most t vertices in W_s. If $|W_s| \leq t$, then there is nothing to prove, so assume $|W_s| = t + 1$. For the sake of contradiction, let $v \in S \setminus W_s$ be a vertex adjacent to all $t + 1$ vertices of W_s. Let W_r be the bag containing v that is closest to the bag W_s in the tree T. Let T_v be the tree in $T \setminus s$ that contains r. Since $v \notin W_s$, for any bag W_x that contains v it holds $x \in T_v$. Moreover, the unique $s - x$ path in T contains the vertex r. By the properties of tree decomposition, and since v is adjacent to every vertex in W_s it follows that $W_s \subset W_r$. Thus, $|W_r| \geq |W_s \cup \{v\}| = t + 2$. A contradiction with the width of $(T, \mathcal{W})$. ◁

Proof of Theorem 30. Let S be a maximal subset of vertices of G such that $\mathrm{tw}(G[S]) \leq t$. By Claim 31, either $G[S]$ admits a good $(t, 2)$-partition or S induces a proper $(t + 1)$-star. The number of sets S that admit a good $(t, 2)$-partition is at most $n^{t+4}2^{2c-2}$ by Lemma 27.

Let us consider the case when $G[S]$ is a proper $(t+1)$-star. Since S is a maximal set with property that $G[S] \in \mathcal{C}$ it follows that S is also a set that induces a proper $(t + 1)$-star with maximal tail. It is not hard to see that the class of graph with bounded treewidth contains all proper $(t+1)$-stars. The number of sets S that induce a proper $(t+1)$-star with maximal tail is at most $2n^{t+3}2^{c-1}$ by Lemma 25. The theorem follows. ◀

▶ **Example 32.** Recall that $K_{a,b}$ is the complement of a 1-closed graph, and that $\mathrm{tw}(K_{a,b}) = \min\{a, b\}$ for any $a, b \in \mathbb{N}$. Trivially, $K_{\ell,t+1}$ contains at least $\Omega(\ell^t) = \Omega\left((|V(K_{\ell,t+1})| - t - 1)^t\right)$ maximal induced subgraphs with treewidth at most t. Hence, the dependence on n^t in Theorem 30 is necessary.

Enumeration. Let us explain how to turn the above proof in an enumeration algorithm. In the proof of Theorem 30 we showed that any maximal induced subgraph of treewidth at most t is either a proper $(t + 1)$-star or admits a good $(t, 2)$-partition.

Enumeration of all proper $(t + 1)$-stars reduces to the enumeration of all maximal independnet sets in the complement of smaller c-closed graphs by the same reduction as in the proof of Lemma 25. Thus, listing all proper $(t + 1)$-stars takes $O(n^{t+3}\mathcal{M}_0(c - 1))$ time.

To enumerate all subgraphs admitting a good $(t, 2)$-partition we use the defintion of the good $(t, 2)$-partition and the c-closure condition. For two edges e, f there are at most $2c$ vertices that are non-adjacent to either e or f by the complementary c-closure property. After fixing a set A of size at most t and particular two edges e, f, by brute-force we can find all subgraphs with treewidth at most t that admit a good $(t, 2)$-partition with the set A and the edges e and f. Since there are at most $2c$ vertices over which we have to apply the bruce-force this takes $O(2^{2c})$ time. In total, going over all sets of size at most t and every two edges e, f takes $O(n^{t+4}2^{2c})$ time.

▶ **Corollary 33** (Restatement of Theorem 4). *For c-closed graphs and a fixed $t \geq 0$, there is an algorihtm running in time $O(n^{t+4}2^{2c})$ that outputs a set containing all maximal induced subgraphs with co-treewidth $\leq t$.*

References

1 Graph classes (problem: Problem: Independent set). URL: `https://www.graphclasses.org/classes/problem_Independent_set.html`.

2 Vladimir E Alekseev. The effect of local constraints on the complexity of determination of the graph independence number. *Combinatorial-algebraic methods in applied mathematics*, pages 3–13, 1982.

3 Albert Angel, Nick Koudas, Nikos Sarkas, and Divesh Srivastava. Dense subgraph maintenance under streaming edge weight updates for real-time story identification. *arXiv preprint arXiv:1203.0060*, 2012.

4 Davide Bacciu, Alessio Conte, Roberto Grossi, Francesco Landolfi, and Andrea Marino. K-plex cover pooling for graph neural networks. *Data Mining and Knowledge Discovery*, pages 1–21, 2021.

5 Balabhaskar Balasundaram, Sergiy Butenko, and Illya V Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59(1):133–142, 2011.

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

7 Balaram Behera, Edin Husić, Shweta Jain, Tim Roughgarden, and C. Seshadhri. FPT algorithms for finding near-cliques in c-closed graphs, 2021. `arXiv:2007.09768`.

8 Dmitry I Belov and James A Wollack. Graph theory approach to detect examinees involved in test collusion. *Applied Psychological Measurement*, page 01466216211013902, 2021.

9 Matthias Bentert, Anne-Sophie Himmel, Hendrik Molter, Marco Morik, Rolf Niedermeier, and René Saitenmacher. Listing all maximal k-plexes in temporal graphs. *Journal of Experimental Algorithmics (JEA)*, 24(1):1–27, 2019.

10 Devora Berlowitz, Sara Cohen, and Benny Kimelfeld. Efficient enumeration of maximal k-plexes. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 431–444, 2015.

11 Édouard Bonnet, Nicolas Bousquet, Pierre Charbit, Stéphan Thomassé, and Rémi Watrigant. Parameterized complexity of independent set in H-free graphs. *Algorithmica*, 82(8):2360–2394, 2020.

12 Michele Borassi, Pierluigi Crescenzi, and Luca Trevisan. An axiomatic and an average-case analysis of algorithms and heuristics for metric properties of graphs. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 920–939. SIAM, 2017.

13 Paweł Brach, Marek Cygan, Jakub Łacki, and Piotr Sankowski. Algorithmic complexity of power law networks. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1306–1325. SIAM, 2016.

14 Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

15 Yixin Cao. Enumerating maximal induced subgraphs. *arXiv preprint arXiv:1912.13446*, 2020.

16 Deepayan Chakrabarti and Christos Faloutsos. Graph mining: Laws, generators, and algorithms. *ACM computing surveys (CSUR)*, 38(1):2–es, 2006.

17 Parinya Chalermsook, Marek Cygan, Guy Kortsarz, Bundit Laekhanukit, Pasin Manurangsi, Danupon Nanongkai, and Luca Trevisan. From gap-ETH to FPT-inapproximability: Clique, dominating set, and more. In *Proceedings of the 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 743–754. IEEE, 2017.

18 Xiaowei Chen and John CS Lui. Mining graphlet counts in online social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(4):1–38, 2018.

19 Maria Chudnovsky, Marcin Pilipczuk, Michał Pilipczuk, and Stéphan Thomassé. On the maximum weight independent set problem in graphs without induced cycles of length at least five. *SIAM Journal on Discrete Mathematics*, 34(2):1472–1483, 2020.

20 Maria Chudnovsky, Stéphan Thomassé, Nicolas Trotignon, and Kristina Vušković. Maximum independent sets in (pyramid, even hole)-free graphs. *arXiv preprint arXiv:1912.11246*, 2019.

21 Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 99(25):15879–15882, 2002.

22 Fan Chung and Linyuan Lu. Connected components in random graphs with given expected degree sequences. *Annals of Combinatorics*, 6(2):125–145, 2002.

23 Edith Cohen, Eran Halperin, Haim Kaplan, and Uri Zwick. Reachability and distance queries via 2-hop labels. *SIAM Journal on Computing*, 32(5):1338–1355, 2003.

24 Alessio Conte, Donatella Firmani, Caterina Mordente, Maurizio Patrignani, and Riccardo Torlone. Fast enumeration of large k-plexes. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 115–124, 2017.

25 Alessio Conte, Donatella Firmani, Maurizio Patrignani, and Riccardo Torlone. Shared-nothing distributed enumeration of 2-plexes. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2469–2472, 2019.

26 Alessio Conte, Andrea Marino, Roberto Grossi, Takeaki Uno, and Luca Versari. Proximity search for maximal subgraph enumeration. *arXiv preprint arXiv:1912.13446*, 2019.

27 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 4(8). Springer, 2015.

28 Maximilien Danisch, Oana Balalau, and Mauro Sozio. Listing k-cliques in sparse real-world graphs. In *Proceedings of the 2018 World Wide Web Conference*, pages 589–598, 2018.

29 Laxman Dhulipala, Quanquan C Liu, Julian Shun, and Shangdi Yu. Parallel batch-dynamic k-clique counting. In *Symposium on Algorithmic Principles of Computer Systems (APOCS)*, pages 129–143. SIAM, 2021.

30 Pavel Dvořák, Andreas Emil Feldmann, Ashutosh Rai, and Paweł Rzążewski. Parameterized inapproximability of independent set in H-free graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 40–53. Springer, 2020.

31 Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

32 Jacob Fox, Tim Roughgarden, C. Seshadhri, Fan Wei, and Nicole Wein. Finding cliques in social networks: A new distribution-free model. In *45th International Colloquium on Automata, Languages, and Programming, (ICALP)*, volume 107 of *LIPIcs*, pages 55:1–55:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

33 Jacob Fox, Tim Roughgarden, C Seshadhri, Fan Wei, and Nicole Wein. Finding cliques in social networks: A new distribution-free model. *SIAM Journal on Computing*, 49(2):448–464, 2020.

34 Eugene Fratkin, Brian T Naughton, Douglas L Brutlag, and Serafim Batzoglou. Motifcut: regulatory motifs finding with maximum density subgraphs. *Bioinformatics*, 22(14):e150–e157, 2006.

35 Milana Grbić, Aleksandar Kartelj, Savka Janković, Dragan Matić, and Vladimir Filipović. Variable neighborhood search for partitioning sparse biological networks into the maximum edge-weighted k k-plexes. *IEEE/ACM transactions on computational biology and bioinformatics*, 17(5):1822–1831, 2019.

36 Rishi Gupta, Tim Roughgarden, and Comandur Seshadhri. Decompositions of triangle-dense graphs. *SIAM Journal on Computing*, 45(2):197–215, 2016.

37 Sushmita Gupta, Venkatesh Raman, and Saket Saurabh. Maximum r-regular induced subgraph problem: Fast exponential algorithms and combinatorial bounds. *SIAM Journal on Discrete Mathematics*, 26(4):1758–1780, 2012.

38 Kyungsook Han, Guangyu Cui, and Yu Chen. Identifying functional groups by finding cliques and near-cliques in protein interaction networks. In *2007 Frontiers in the Convergence of Bioscience and Information Technologies*, pages 159–164. IEEE, 2007.

39 Johan Håstad. Clique is hard to approximate withinn 1- ε. *Acta Mathematica*, 182(1):105–142, 1999.

40 Maritza Hernandez, Arman Zaribafiyan, Maliheh Aramon, and Mohammad Naghibi. A novel graph-based approach for determining molecular similarity. *arXiv preprint arXiv:1601.06693*, 2016.

41 Edin Husić and Martin Milanič. A polynomial-time algorithm for the independent set problem in $\{P_{10}, C_4, C_6\}$-free graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 271–284. Springer, 2019.

42 Edin Husić and Tim Roughgarden. FPT algorithms for finding dense subgraphs in c-closed graphs, 2021. `arXiv:2007.09768v3`.

43 Shweta Jain and C Seshadhri. A fast and provable method for estimating clique counts using turán's theorem. In *Proceedings of the 26th international conference on world wide web*, pages 441–449, 2017.

44 Shweta Jain and C Seshadhri. The power of pivoting for exact clique counting. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 268–276, 2020.

45 Shweta Jain and C Seshadhri. Provably and efficiently approximating near-cliques using the Turán shadow: PEANUTS. In *Proceedings of The Web Conference 2020 (WWW)*, pages 1966–1976, 2020.

46 Ruoming Jin, Yang Xiang, Ning Ruan, and David Fuhry. 3-hop: a high-compression indexing scheme for reachability query. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, pages 813–826, 2009.

47 Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. Computing dense and sparse subgraphs of weakly closed graphs. *arXiv preprint arXiv:2007.05630*, 2020.

48 Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. Computing Dense and Sparse Subgraphs of Weakly Closed Graphs. In *31st International Symposium on Algorithms and Computation (ISAAC 2020)*, volume 181, pages 20:1–20:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

49 Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. Computing dense and sparse subgraphs of weakly closed graphs. In Yixin Cao, Siu-Wing Cheng, and Minming Li, editors, *31st International Symposium on Algorithms and Computation, ISAAC 2020, December 14-18, 2020, Hong Kong, China (Virtual Conference)*, volume 181 of *LIPIcs*, pages 20:1–20:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.ISAAC.2020.20`.

50 Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. Exploiting c-Closure in Kernelization Algorithms for Graph Problems. *arXiv preprint arXiv:2005.03986*, 2020.

51 Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. Essentially tight kernels for (weakly) closed graphs. *CoRR*, abs/2103.03914, 2021. `arXiv:2103.03914`.

52 Tomohiro Koana and André Nichterlein. Detecting and enumerating small induced subgraphs in c-closed graphs. *Discret. Appl. Math.*, 302:198–207, 2021. `doi:10.1016/j.dam.2021.06.019`.

53 Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D Sivakumar, Andrew Tomkins, and Eli Upfal. Stochastic models for the web graph. In *Proceedings 41st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 57–65. IEEE, 2000.

54 Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. *Computer networks*, 31(11-16):1481–1493, 1999.

55 Jure Leskovec, Deepayan Chakrabarti, Jon Kleinberg, Christos Faloutsos, and Zoubin Ghahramani. Kronecker graphs: an approach to modeling networks. *Journal of Machine Learning Research*, 11(2), 2010.

56 John M Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

57 Ronghua Li, Sen Gao, Lu Qin, Guoren Wang, Weihua Yang, and Jeffrey Xu Yu. Ordering heuristics for k-clique listing. *Proc. VLDB Endow.*, 2020.

58 Daniel Lokshtanov, Marcin Pilipczuk, and Erik Jan Van Leeuwen. Independence and efficient domination on P_6-free graphs. *ACM Transactions on Algorithms (TALG)*, 14(1):1–30, 2017.

59 Vadim V Lozin and Martin Milanič. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *Journal of Discrete Algorithms*, 6(4):595–604, 2008.

60 David W Matula and Leland L Beck. Smallest-last ordering and clustering and graph coloring algorithms. *Journal of the ACM (JACM)*, 30(3):417–427, 1983.

61 Benjamin McClosky and Illya V Hicks. Combinatorial algorithms for the maximum k-plex problem. *Journal of Combinatorial Optimization*, 23(1):29–49, 2012.

62 Raymond E Miller and David E Muller. A problem of maximum consistent subsets. Technical report, IBM Research Report RC-240, JT Watson Research Center, Yorktown Heights, NY, 1960.

63 John W Moon and Leo Moser. On cliques in graphs. *Israel Journal of Mathematics*, 3(1):23–28, 1965.

64 Jeffrey Pattillo, Nataly Youssef, and Sergiy Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 226(1):9–18, 2013.

65 Marcin Pilipczuk and Michał Pilipczuk. Finding a maximum induced degenerate subgraph faster than 2^n. In *International Symposium on Parameterized and Exact Computation (IPEC)*, pages 3–12. Springer, 2012.

66 Venkatesh Raman and Saket Saurabh. Short cycles make W -hard problems hard: FPT algorithms for W -hard problems in graphs with no short cycles. *Algorithmica*, 52(2):203–225, 2008. doi:10.1007/s00453-007-9148-9.

67 Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.

68 Tim Roughgarden. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021. doi:10.1017/9781108637435.

69 Stephen B Seidman and Brian L Foster. A graph-theoretic generalization of the clique concept. *Journal of Mathematical Sociology*, 6(1):139–154, 1978.

70 Xiaoli Song, Changjun Zhou, Bin Wang, and Qiang Zhang. A method of motif mining based on backtracking and dynamic programming. In *International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pages 317–328. Springer, 2015.

71 Mauro Sozio and Aristides Gionis. The community-search problem and how to plan a successful cocktail party. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 939–948, 2010.

72 Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42, 2006.

73 Shuji Tsukiyama, Mikio Ide, Hiromu Ariyoshi, and Isao Shirakawa. A new algorithm for generating all the maximal independent sets. *SIAM Journal on Computing*, 6(3):505–517, 1977.

74 Yue Wang, Xun Jian, Zhenhua Yang, and Jia Li. Query optimal k-plex based community in graphs. *Data Science and Engineering*, 2(4):257–273, 2017.

75 Bin Wu and Xin Pei. A parallel algorithm for enumerating all the maximal k-plexes. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 476–483. Springer, 2007.

76 Mihalis Yannakakis. Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10(2):310–327, 1981.

77 Hao Yin, Austin R Benson, Jure Leskovec, and David F Gleich. Local higher-order graph clustering. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 555–564, 2017.

78 Haiyuan Yu, Alberto Paccanaro, Valery Trifonov, and Mark Gerstein. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, 22(7):823–829, 2006.

79 Yang Zhang and Srinivasan Parthasarathy. Extracting analyzing and visualizing triangle k-core motifs within networks. In *2012 IEEE 28th international conference on data engineering*, pages 1049–1060. IEEE, 2012.

80 Feng Zhao and Anthony KH Tung. Large scale cohesive subgraphs discovery for social network visual analysis. *Proceedings of the VLDB Endowment*, 6(2):85–96, 2012.

81 Yi Zhou, Jingwei Xu, Zhenyu Guo, Mingyu Xiao, and Yan Jin. Enumerating maximal k-plexes with worst-case time guarantee. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34(03), pages 2442–2449, 2020.

82 Jinrong Zhu, Bilian Chen, and Yifeng Zeng. Community detection based on modularity and k-plexes. *Information Sciences*, 513:127–142, 2020.

83 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing (STOC)*, pages 681–690, 2006.