# Computing Dense and Sparse Subgraphs of Weakly Closed Graphs

**Tomohiro Koana**
Technische Universität Berlin, Algorithmics and Computational Complexity, Germany
tomohiro.koana@tu-berlin.de

**Christian Komusiewicz**
Philipps-Universität Marburg, Fachbereich Mathematik und Informatik, Germany
komusiewicz@informatik.uni-marburg.de

**Frank Sommer**
Philipps-Universität Marburg, Fachbereich Mathematik und Informatik, Germany
fsommer@informatik.uni-marburg.de

## Abstract

A graph $G$ is weakly $\gamma$-closed if every induced subgraph of $G$ contains one vertex $v$ such that for each non-neighbor $u$ of $v$ it holds that $|N(u) \cap N(v)| < \gamma$. The weak closure $\gamma(G)$ of a graph, recently introduced by Fox et al. [SIAM J. Comp. 2020], is the smallest number such that $G$ is weakly $\gamma$-closed. This graph parameter is never larger than the degeneracy (plus one) and can be significantly smaller. Extending the work of Fox et al. [SIAM J. Comp. 2020] on clique enumeration, we show that several problems related to finding dense subgraphs, such as the enumeration of bicliques and $s$-plexes, are fixed-parameter tractable with respect to $\gamma(G)$. Moreover, we show that the problem of determining whether a weakly $\gamma$-closed graph $G$ has a subgraph on at least $k$ vertices that belongs to a graph class $\mathcal{G}$ which is closed under taking subgraphs admits a kernel with at most $\gamma k^2$ vertices. Finally, we provide fixed-parameter algorithms for Independent Dominating Set and Dominating Clique when parameterized by $\gamma + k$ where $k$ is the solution size.

**2012 ACM Subject Classification** Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Graph algorithms analysis

**Keywords and phrases** Fixed-parameter tractability, $c$-closure, degeneracy, clique relaxations, bicliques, dominating set

**Digital Object Identifier** 10.4230/LIPIcs.ISAAC.2020.20

**Related Version** A continously updated version of the paper is available at `https://arxiv.org/abs/2007.05630`.

**Funding** *Tomohiro Koana*: Supported by the Deutsche Forschungsgemeinschaft (DFG), project FPTinP, NI 369/19.
*Frank Sommer*: Supported by the Deutsche Forschungsgemeinschaft (DFG), project MAGZ, KO 3669/4-1.

## 1 Introduction

In the quest to design efficient algorithms for NP-hard graph problems, a very successful approach is to exploit the sparsity of input graphs: many problems that are assumed to be hard in general graphs turn out to be efficiently solvable in sparse graphs [1, 9, 15, 22, 24, 26, 30]. One popular sparseness measure that has been used for a variety of graph problems is the *degeneracy* of the input graph $G$.

▶ **Definition 1.1.** *For a vertex $v \in V(G)$, let $\deg_G(v) := |N(v)|$ denote the degree of $v$. Then, $G$ is* $d$-degenerate *if there exists a vertex $v$ with $\deg_{G'}(v) \leq d$ in every induced subgraph $G'$ of $G$.*


31st International Symposium on Algorithms and Computation (ISAAC 2020).
Editors: Yixin Cao, Siu-Wing Cheng, and Minming Li; Article No. 20; pp. 20:1–20:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Many graph algorithms which exploit the fact that the input graph has bounded degeneracy have been proposed. For example, there is an algorithm that enumerates all maximal cliques of a graph in $\mathcal{O}(3^{d/3} \cdot dn)$ time and performs very efficiently on real-world input instances [9]. Further applications of degeneracy include FPT algorithms for Dominating Set and related problems [1, 14, 28], for clique relaxations [22, 24], and for biclique enumeration algorithms [8, 16].

In recent work, Fox et al. [11] proposed exploiting a different property of real-world graphs that is motivated by the triadic closure principle. This principle postulates that people in a social network which have many common friends are likely to be friends themselves. This principle is an explanation for the empirical observation that in real-world social networks, we often find no pairs of non-adjacent vertices with many common neighbors. This property is expressed in the *closure number of G*, defined as follows.

▶ **Definition 1.2** ([11]). *Let* $\mathrm{cl}_G(v) = \max_{v' \in V \setminus N[v]} |N(v) \cap N(v')|$ *denote the* closure number *of a vertex v in a graph G. A graph G is c*-closed *if* $\mathrm{cl}_G(v) < c$ *for all* $v \in V(G)$.

Fox et al. [11] showed that a $c$-closed graph has $\mathcal{O}(3^{c/3} \cdot n^2)$ maximal cliques. Given that all maximal cliques can be enumerated in $\mathcal{O}(\alpha \cdot nm)$ time, where $\alpha$ is the number of maximal cliques, this bound implies that all maximal cliques of a $c$-closed graph can be enumerated in $\mathcal{O}^*(3^{c/3})$ time[1]. This means that the clique enumeration problem has an FPT algorithm with respect to the $c$-closure of the input graph. In companion work, we used the $c$-closure of graphs to show that several hard graph problems such as Independent Set, Dominating Set, and Induced Matching admit polynomial kernels when parameterized by the $c$-closure of the graph plus the respective solution size [20]. Very recently, Koana and Nichterlein [21] studied the time complexity of finding and enumerating small induced subgraphs in $c$-closed graphs.

Fox et al. [11] suggested a further graph parameter which combines sparseness and triadic closure properties, the *weak closure* of a graph.

▶ **Definition 1.3** ([11]). *A graph G is weakly $\gamma$-closed*[2] *if one of the following holds:*

- *There exists a* closure ordering $\sigma := v_1, \ldots, v_n$ *of the vertices such that* $\mathrm{cl}_{G_i}(v_i) < \gamma$ *for all* $i \in [n]$ *where* $G_i := G[\{v_i, \ldots, v_n\}]$.
- *Every induced subgraph $G'$ of G has a vertex* $v \in V(G')$ *such that* $\mathrm{cl}_{G'}(v) < \gamma$.

We call $\sigma$ a *closure ordering* of $G$. The weak closure number $\gamma$ of a graph $G$ is never larger than $d + 1$ where $d$ is the degeneracy of $G$ and also never larger than the closure number $c$ of $G$. Consequently, fixed-parameter algorithms for $\gamma$ are, in principle, preferable to those for the closure number $c$ or the degeneracy $d$. From an application point of view, the weak closure number is also an excellent parameter in such graphs since it tends to take on very small values in real-world social networks [11] (see also Table 2 in the arXiv-version of this article [19]). Fox et al. [11] showed that a graph has $\mathcal{O}(3^{\gamma/3} \cdot n^2)$ many maximal cliques which, again using known clique enumeration algorithms, gives an algorithm that enumerates all maximal cliques in $\mathcal{O}^*(3^{\gamma/3})$ time.

**Our Results.** In a nutshell, we show that weak closure can be applied to a variety of graph problems that are related to searching for sparse or dense subgraphs; our main results are listed in Table 1. Our results improve over the state of the art in the following sense: the best

[1] The $\mathcal{O}^*$ notation hides polynomial factors in the input size.
[2] To avoid confusion with the closure number $c$, we denote the weak closure by $\gamma$ instead of $c$.

**Table 1** An overview of our results.

| Problem | Result | Reference |
|---|---|---|
| Independent Set | $\mathcal{O}(\gamma k^2)$-vertex kernel | Corollary 2.4 |
| $s$-Plex | W[1]-hard for $k$ even if $c = 2$ | Theorem 3.3 |
| | $\mathcal{O}(2^\gamma n^{2s+1})$-time algorithm for $s \geq 2$ | Corollary 3.2 |
| $s$-Defective Clique | W[1]-hard for $k$ even if $c = 2$ | [29] |
| | $\mathcal{O}(2^\gamma n^{s+3})$-time algorithm | Corollary 3.5 |
| | $2^{\mathcal{O}(\gamma\sqrt{s}+s\log k)} n^{\mathcal{O}(\sqrt{s})}$-time algorithm | Theorem 3.8 |
| Non-Induced $(k_1, k_2)$-Biclique | $\mathcal{O}^*(2^\gamma)$-time algorithm | Theorem 4.2 |
| Induced $(k, k)$-Biclique | $\mathcal{O}^*(\gamma^{\mathcal{O}(\gamma)})$-time algorithm | Theorem 4.4 |
| Induced $(k_1, k_2)$-Biclique | $\mathcal{O}^*(1.6107^c)$-time algorithm if $k_1 \geq 2$ | Theorem 4.5 |
| | NP-hard for $k_1 = 1$ even if $c = 3$ and $\gamma = 2$ | Theorem 4.6 |
| Independent Dominating Set | $\mathcal{O}^*((\frac{\gamma-1}{2})^k k^{2k})$-time algorithm | Theorem 5.1 |
| Dominating Clique | $\mathcal{O}^*((\gamma - 1)^{k-1})$-time algorithm | Theorem 5.2 |

known tractability results for these problems employ the degeneracy of the input graph as a parameter and, as discussed above, the weak closure is essentially a smaller parameter. For some problems, we also provide results for the $c$-closure parameter. There are two reasons for this. First, for some problems we obtain better running time bounds for the parameter $c$. Second, we provide some lower bounds for the problems under consideration and, whenever possible, we provide them for the larger closure parameter $c$. From a practical point of view, the most important results are, in our opinion, the enumeration algorithms for maximal non-induced bicliques and maximal $s$-plexes whose running time grows moderately with $\gamma$. Our algorithms to enumerate all maximal $s$-plexes and non-induced bicliques are based on the algorithm to enumerate all maximal cliques in weakly $\gamma$-closed graphs [11]. Independently, Husić and Roughgarden [17] obtained similar results for the enumeration of maximal $s$-plexes and further dense subgraphs parameterized by the $c$-closure; it seems that their algorithms for $s$-plex enumeration can be adapted to parameterization by weak closure as well [17].

**Preliminaries.** For $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \ldots, n\}$. For a graph $G$, we denote by $V(G)$ and $E(G)$ its vertex set and edge set, respectively. We let $n := |V(G)|$ denote the number of vertices. Let $X \subseteq V(G)$ be a vertex set. We let $G[X]$ denote the subgraph induced by $X$ and $G - X := G[V(G) \setminus X]$ the graph obtained by removing the vertices of $X$. We denote by $N_G(X) := \{y \in V(G) \setminus X \mid xy \in E(G), x \in X\}$ and $N_G[X] := N_G(X) \cup X$, the open and closed neighborhood of $X$, respectively. For all these notations, when $X$ is a singleton $\{x\}$ we may write $x$ instead of $\{x\}$. The maximum degree of $G$ is $\Delta := \max_{v \in V(G)} \deg_G(v)$. We may drop the subscript $\cdot_G$ when it is clear from context. A parameterized problem is *fixed-parameter tractable* if every instance $(I, k)$ can be solved in $f(k) \cdot |I|^{\mathcal{O}(1)}$ time for some computable function $f$. An algorithm with such a running time is an *FPT algorithm*. A *kernelization* is a polynomial-time algorithm which transforms every instance $(I, k)$ into an equivalent instance $(I', k')$ such that $|I'| + k' \leq g(k)$ for some computable function $g$. It is widely believed that W[$t$]-hard problems ($t \in \mathbb{N}$) do not admit an FPT algorithm. For more details on parameterized complexity, we refer to the standard monographs [5, 7].

## 2 Variants of Independent Set

In this section we study Independent Set and related problems.

Independent Set
**Input:** A graph $G$ and $k \in \mathbb{N}$.
**Question:** Is there a vertex set $S \subseteq V(G)$ such that $|S| \geq k$ and the vertices in $S$ are pairwise non-adjacent?

In companion work [20], we provided an $\mathcal{O}(ck^2)$-vertex kernel for Independent Set. Here, we strengthen this result by showing that Independent Set (in fact, a generalization) admits a polynomial kernel with respect to the parameter $k + \gamma$.

Let $\mathcal{G}$ be a graph class. We say that $\mathcal{G}$ is *monotone* if $\mathcal{G}$ is closed under vertex deletions and edge deletions. That is, if $G \in \mathcal{G}$, then for each subgraph $H$ of $G$ we have $H \in \mathcal{G}$. Let us define the problem of finding an induced subgraph belonging to $\mathcal{G}$ as follows:

$\mathcal{G}$-Subgraph
**Input:** A graph $G$ and $k \in \mathbb{N}$.
**Question:** Is there a vertex set $S \subseteq V(G)$ with $|S| \geq k$ such that $G[S] \in \mathcal{G}$?

Note that when $\mathcal{G}$ is the class of edgeless graphs, then $\mathcal{G}$-Subgraph corresponds to Independent Set.

Let $v_1, v_2, \dots, v_n$ be a closure ordering of $G$ and let $G_i = G[V_i]$ for $V_i = \{v_i, v_{i+1}, \dots, v_n\}$. Our kernelization algorithm applies the following reduction rule:

▶ **Reduction Rule 2.1.** *If $\deg_{G_i}(v_i) \geq \gamma k$, then remove $v_i$.*

▶ **Lemma 2.2.** *Reduction Rule 2.1 is correct for monotone graph classes.*

**Proof.** Let $G' := G - v_i$ for $v_i \in V$ with $\deg_{G_i}(v_i) \geq \gamma k$ be the graph obtained by applying Reduction Rule 2.1. Clearly, if $G'[S] \in \mathcal{G}$ for some vertex set $S \subseteq V(G')$, then also $G[S] \in \mathcal{G}$.

Hence, it remains to show that if there exists a vertex set $S \subseteq V(G)$ of size at least $k$ such that $G[S] \in \mathcal{G}$ then there exists a vertex set $S' \subseteq V(G')$ of size at least $k$ such that $G'[S'] \in \mathcal{G}$. If $v_i \notin S$, we observe that $G'[S] \in \mathcal{G}$. Thus, in the following we assume that $v_i \in S$. We prove that $v_i$ and any other vertex $v_j \in V(G)$ have less than $\gamma$ common neighbors in $V_i$, given that they are non-adjacent.

▷ Claim. Let $j \in [n] \setminus \{i\}$. If $v_i v_j \notin E(G)$, then $|N_{G_i}(v_i) \cap N_G(v_j)| < \gamma$.

Proof. First, assume that $j < i$. Then we have $|N_{G_j}(v_i) \cap N_{G_j}(v_j)| < \gamma$ by the definition of closure orderings. Since $V_i \subseteq V_j$ this implies that $|N_{G_i}(v_i) \cap N_G(v_j)| < \gamma$. Second, assume that $j > i$. By the definition of closure orderings we have $|N_{G_i}(v_i) \cap N_{G_i}(v_j)| < \gamma$. ◁

Let $S_i := S \setminus N_G(v_i)$ be the set of vertices in $S$ that are not adjacent to $v_i$. Since $\deg_{G_i}(v_i) \geq \gamma k$, it follows from the claim above that there exists at least one vertex $u$ in the neighborhood of $v_i$ in $G_i$ that is not adjacent to any vertex from $S_i$ (in other words, $u \in N_{G_i}(v_i) \setminus \bigcup_{v_j \in S_i} N_G(v_j)$). Since $\mathcal{G}$ is monotone and $N_G(u) \cap S \subseteq N_G(v_i) \cap S$, for the vertex set $S' := (S \setminus \{v_i\}) \cup \{u\}$ we have $G'[S'] \in \mathcal{G}$. ◀

▶ **Theorem 2.3.** *Let $\mathcal{G}$ be a monotone graph class. Then, $\mathcal{G}$-Subgraph has a kernel with at most $\gamma k^2$ vertices.*

**Proof.** One can exhaustively apply Reduction Rule 2.1 in polynomial time. The resulting graph has degeneracy $d < \gamma k$ with the same vertex ordering. Note that any graph $G$ on at least $(d+1)k$ vertices contains an independent set $S$ of size $k$. Due to the monotonicity of $\mathcal{G}$, $G[S] \in \mathcal{G}$ for an independent set $S$. Thus, returning Yes is correct whenever $|V(G)| \geq \gamma k^2$ and we obtain an equivalent instance with at most $\gamma k^2$ vertices. ◀

Since the class of edgeless graphs is monotone, we obtain the following corollary:

▶ **Corollary 2.4.** *Independent Set has a kernel with at most $\gamma k^2$ vertices.*

Theorem 2.3 also implies kernels for many other problems, including Acyclic Subgraph, Bipartite Subgraph, Planar Subgraph, and Bounded Degree Subgraph. These problems ask whether the input graph $G$ contains a vertex set $S \subseteq V(G)$ such that $|S| \geq k$ and $G[S]$ is acyclic, bipartite, planar, or has bounded maximum degree, respectively. All of these problems are W[1]-hard in general graphs [18].

▶ **Corollary 2.5.** *Each of Acyclic Subgraph, Bipartite Subgraph, Bounded Degree Subgraph, and Planar Subgraph has a kernel with $\gamma k^2$ vertices.*

Moreover, it follows from Theorem 2.3 that the problem of finding a subgraph with exactly $k$ vertices and at most $t$ edges also admits a polynomial kernel in weakly $\gamma$-closed graphs. We call the problem Sparsest-$k$-Subgraph.

▶ **Corollary 2.6.** *Sparsest-$k$-Subgraph has a kernel with at most $\gamma k^2$ vertices.*

This is in sharp contrast to Densest-$k$-Subgraph, which is W[1]-hard even in 2-closed graphs [29]. In Densest-$k$-Subgraph one asks for a vertex subset $S \subseteq V(G)$ of exactly $k$ vertices such that $G[S]$ has at least $t$ edges.

## 3 Clique Relaxations

In this section we present algorithms for problems which contain Clique as special case.

### 3.1 $s$-Plex

A clique is a vertex set $C$ such that each vertex $v \in S$ is adjacent to each other vertex in $S$. One way to relax this definition is to allow each vertex $v \in S$ to have at most $s$ non-neighbors in $S$. A set fulfilling this condition is said to be an $s$-plex and cliques are exactly the 1-plexes. This relaxation leads to the following problem.

$s$-Plex
**Input:** A graph $G$ and $s, k \in \mathbb{N}$.
**Question:** Is there a set $S \subseteq V(G)$ of at least $k$ vertices such that the minimum degree in $G[S]$ is at least $|S| - s$?

It is known that $s$-Plex is W[1]-hard when parameterized by $k$ for all $s \in \mathbb{N}$ [18, 23] and a simple algorithm can enumerate all maximal $s$-plexes in $2^d n^{s+\mathcal{O}(1)}$ time [22]. The task of enumerating $s$-plexes is also studied in practice [2, 3]. Here, we present an algorithm to list all maximal $s$-plexes in weakly $\gamma$-closed graphs.

▶ **Theorem 3.1.** *For $s \geq 2$, there are $\mathcal{O}(2^\gamma n^{2s-1})$ maximal $s$-plexes. Moreover, all maximal $s$-plexes can be enumerated in $\mathcal{O}(2^\gamma n^{2s+1})$ time.*

**Proof.** Let $v \in V(G)$ be a vertex such that $\mathrm{cl}_G(v) < \gamma$ and let $G' := G - v$ be the graph obtained by deleting $v$. Let $\mathcal{S}$ and $\mathcal{S}'$ be the collections of all maximal $s$-plexes (without duplicates) in $G$ and $G'$, respectively. We show that $|\mathcal{S}| \leq |\mathcal{S}'| + 2^\gamma n^{2s-2}$ and that $\mathcal{S}$ can be constructed from $\mathcal{S}'$ in $\mathcal{O}(|\mathcal{S}'| \cdot n + 2^\gamma n^{2s+1})$ time. Each maximal $s$-plex $S$ in $G$ is of one of the following four types:

1. $S$ does not contain $v$. Then, $S$ is also maximal in $G'$.
2. $S$ contains $v$ and $S \setminus \{v\}$ is maximal in $G'$.
3. $S$ contains $v$, $S \setminus \{v\}$ is not maximal in $G'$, and $S$ contains a non-neighbor of $v$ (that is, $S \setminus N_G(v) \neq \emptyset$).
4. $S$ contains $v$, $S \setminus \{v\}$ is not maximal in $G'$, and $S$ is in the neighborhood of $v$ (that is, $S \subseteq N_G[v]$).

It is easy to see that there are $|\mathcal{S}'|$ maximal $s$-plexes of type 1 and type 2. Moreover, these $s$-plexes can be enumerated in $\mathcal{O}(|\mathcal{S}'| \cdot n)$ time.

Now, we enumerate maximal $s$-plexes of type 3. Consider such an $s$-plex $S$. We partition $S$ into three parts as follows: We first divide $S$ into $S_v := S \cap N_G[v]$ and $\overline{S_v} := S \setminus N_G[v]$. We divide $S_v$ further into $S_{uv} := S_v \cap N_G(u)$ and $\overline{S_{uv}} := S_v \setminus N_G(u)$ for some vertex $u \in \overline{S_v}$. By the definition of $s$-plexes, $|\overline{S_v}| < s$ and $|\overline{S_{uv}}| < s$. Hence, there are at most $n^{2s-2}$ choices for $\overline{S_v}$ and $\overline{S_{uv}}$. For $S_{uv}$, there are at most $2^{\gamma-1}$ choices because $S_{uv} \subseteq N_G(v) \cap N_G(u)$ and $|N_G(v) \cap N_G(u)| < \mathrm{cl}_G(v) < \gamma$. Overall, there are at most $2^{\gamma-1}n^{2s-2}$ maximal $s$-plexes of type 3.

We then enumerate maximal $s$-plexes of type 4. Let $S$ be one of these $s$-plexes. Since $S' := S \setminus \{v\}$ is not maximal in $G'$, there exists a vertex $u \in V(G) \setminus S$ such that $S' \cup \{u\}$ is an $s$-plex in $G'$. If $u \in N_G(v)$, then $S \cup \{u\}$ is also an $s$-plex in $G$, which contradicts the fact that $S$ is maximal in $G$. Hence, we can assume that $u \notin N_G(v)$. Then, $S \setminus N_G(u)$ contains at most $s-1$ vertices, which in turn implies that there are at most $n^{s-1}$ choices for $S \setminus N_G(u)$. Since $S \subseteq N(v)$ we observe that $S \cap N_G(u) \subseteq N_G(v) \cap N_G(u)$ and $|N_G(v) \cap N_G(u)| \leq \mathrm{cl}_G(v) < \gamma$. Thus, we have $2^{\gamma-1}$ choices for $S \cap N_G(v)$. All in all, there are at most $2^{\gamma-1}n^s$ maximal $s$-plexes of type 4.

By the above analysis, we obtain $|\mathcal{S}| \leq |\mathcal{S}'| + 2^{\gamma-1}n^{2s-2} + 2^{\gamma-1}n^s \leq |\mathcal{S}'| + 2^\gamma n^{2s-2}$. For the time complexity, recall that all maximal $s$-plexes of type 1 and 2 can be found in $\mathcal{O}(|\mathcal{S}'| \cdot n)$ time. Furthermore, maximal $s$-plexes of type 3 and 4 can be enumerated in $\mathcal{O}((2^{\gamma-1}n^{2s-2} + 2^{\gamma-1}n^s) \cdot n^2)$ time, because it takes $\mathcal{O}(n^2)$ time to verify whether a vertex set is a maximal $s$-plex or not. Finally, we remove duplicates in $\mathcal{O}((|\mathcal{S}'| + 2^{\gamma-1}n^{2s-2} + 2^{\gamma-1}n^s) \cdot n) = \mathcal{O}(|\mathcal{S}'| \cdot n + 2^\gamma n^{2s-1})$ time, using radix sort. Altogether, the algorithm needs $\mathcal{O}(|\mathcal{S}'| \cdot n + 2^\gamma n^{2s})$ time to enumerate all maximal $s$-plexes in $G$.

Let $a_n$ be the number of maximal $s$-plexes in weakly $\gamma$-closed graphs on $n$ vertices. Clearly, $a_1 = 1$. Furthermore, the above analysis showed that $a_n - a_{n-1} = |\mathcal{S}| - |\mathcal{S}'| \leq 2^\gamma n^{2s-2}$. Hence, by induction we obtain $a_n = a_1 + \sum_{i=2}^n (a_i - a_{i-1}) \leq 2^\gamma n^{2s-1} + 1$. Thus, all maximal $s$-plexes can be enumerated in $\mathcal{O}((a_n \cdot n + 2^\gamma n^{2s}) \cdot n) = \mathcal{O}(2^\gamma n^{2s+1})$ time. ◀

A factor of $n^{2s-2}$ in Theorem 3.1 is unavoidable: Consider a graph $G$ consisting of two cliques $C_1$ and $C_2$ of equal size. Clearly, $G$ is 1-closed. Each subset of $C_1$ of size exactly $s-1$ and each subset of $C_2$ of size exactly $s-1$ together form a maximal $s$-plex. Hence, there exist 1-closed graphs with $\Omega((n/2)^{2s-2})$ maximal $s$-plexes.

From Theorem 3.1, we obtain the following.

▶ **Corollary 3.2.** *For $s \geq 2$, $s$-Plex can be solved in $\mathcal{O}(2^\gamma n^{2s+1})$ time.*

We show that there is presumably no $f(k) \cdot n^{\mathcal{O}(1)}$-time algorithm for $s$-Plex in 2-closed graphs. Our reduction also shows that $s$-Plex is W[1]-hard for the parameter $k+d$, answering an open question from the literature [22].

▶ **Theorem 3.3.** *$s$-Plex is W[1]-hard in 2-closed graphs when parameterized by $k + d$.*

**Proof.** We reduce from Clique. Let $(G, k)$ be an instance of Clique for $k \geq 4$. First, we subdivide each edge $uv$ of $G$ twice. That is, we remove the edge $uv$ and add edges $ux_u^v$, $x_u^v x_v^u$, and $x_v^u v$, where $x_u^v$ and $x_v^u$ are two new vertices. For each edge $uv \in E(G)$, we introduce $k-3$ vertices $x_{uv}^1, \dots, x_{uv}^{k-3}$. Let $X_{uv} = \{x_u^v, x_v^u, x_{uv}^1, \dots, x_{uv}^{k-3}\}$ and let $X = \bigcup_{uv \in E(G)} X_{uv}$. We then add edges so that $X_{uv}$ forms a clique. Lastly, we introduce a set $T = \{t^1, \dots, t^{k-3}\}$ of $k-3$ vertices and add edges between $x_{uv}^i$ and $t^i$ for each $uv \in E(G)$ and each $i \in [k-3]$. Let $G'$ be the resulting graph.

It is easy to verify that $G'$ is 2-closed. Moreover, $G'$ is $(k-1)$-degenerate: Each vertex $x \in X_{uv}$ is of degree $k-1$ and there is no edge in $G' - X$. We show that $G$ has a clique of size $k$ if and only if $G'$ has an $s$-plex of size $k'$, where $k' = 2k-3+(k-1)\binom{k}{2}$ and $s = k' - (k-1)$.

Suppose that $G$ has a clique $S$ of size exactly $k$. Let $S' = S \cup T \cup \bigcup_{uv \in E(G[S])} X_{uv}$. Observe that $|S'| \geq k'$. We verify that each vertex in $G'[S']$ has degree at least $k' - s = k-1$.

- Let $v \in S$. By construction, we have $x_v^u \in N_{G'}(v)$ for each $u \in S \setminus \{v\}$. Since $x_v^u$ is contained in $S'$, $v$ has at least $k-1$ neighbors in $G'[S']$.
- We have $\deg_{G'[S']}(t^i) \geq \binom{k}{2} \geq k-1$ for each $i \in [k-3]$, because $t^i$ is adjacent to $x_{uv}^i$ for all $uv \in E(G[S])$.
- Consider $x_u^v$ for $uv \in E(G[S])$. We have $u \in N_{G'}(x_u^v)$ by construction. Moreover, $x_u^v$ is adjacent to all $k-2$ vertices in $X_{uv} \setminus \{x_u^v\}$. Thus, we have $\deg_{G'[S']}(x_u^v) \geq k-1$.
- Consider $x_{uv}^i$ for $uv \in E(G[S])$ and $i \in [k-3]$. We have $t^i \in N_{G'}(x_{uv}^i)$ by construction. Moreover, $x_{uv}^i$ is adjacent to all $k-2$ vertices in $X_{uv} \setminus \{x_{uv}^i\}$. Thus, we have $\deg_{G'[S']}(x_{uv}^i) \geq k-1$.

Thus, every vertex has at least $k-1 = k'-s$ neighbors in $G'[S']$.

Conversely, suppose that $S'$ is an $s$-plex of size exactly $k'$. We start with the following claim.

▷ Claim. If $S'$ contains a vertex $x$ of $X_{uv}$ for some $uv \in E(G)$, then $S'$ also contains all vertices in $N_{G'}[X_{uv}]$ (that is, $u, v \in S'$, $X_{uv} \subseteq S'$, and $T \subseteq S'$).

Proof. By construction, $\deg_{G'}(x) = k-1$. Since each vertex in $G'[S']$ has degree $|S'|-s \geq k-1$ by the definition of $s$-plexes, we have $N_{G'}[X_{uv}] \subseteq S'$. The claim follows because $X_{uv}$ is a clique. ◁

Let $\ell = |S' \cap V(G)|$. By the claim above, there are at most $\binom{\ell}{2}$ edges $uv \in E(G)$ with $X_{uv} \cap S' \neq \emptyset$. By construction, we have $|X_{uv}| = k-1$ for each $uv \in E(G)$. Thus, we have

$$|S'| = |S' \cap V(G)| + |T| + |S' \cap X| \leq \ell + k - 3 + (k-1)\binom{\ell}{2}.$$

Since $|S'| = k' = 2k - 3 + \binom{k}{2}$, we obtain $\ell \geq k$.

By definition, each vertex $v \in S' \cap V(G)$ has at least $|S'| - s \geq k-1$ neighbors in $G'[S']$. So there are at least $\ell(k-1)/2$ edges $uv \in E(G)$ such that $S' \cap X_{uv} \neq \emptyset$. Hence, we see from the above claim that

$$|S'| \geq |S' \cap V(G)| + |T| + |S' \cap X| \geq \ell + k - 3 + (k-1) \cdot \ell(k-1)/2.$$

Since $|S'| = k' = 2k-3+(k-1)\binom{k}{2}$, we obtain $\ell = k$ and $|S' \cap X| = (k-1)\binom{k}{2}$. Thus, $S' \cap V(G)$ is a clique of $k$ vertices in $G$ by construction. ◀

## 3.2 $s$-Defective Clique

Another way to relax the clique model is to allow at most $s$ non-edges, which leads us to the following problem:

$s$-Defective Clique
**Input:** A graph $G$ and $k \in \mathbb{N}$.
**Question:** Is there a set $S \subseteq V(G)$ of at least $k$ vertices such that $G[S]$ has at least $\binom{|S|}{2} - s$ edges?

A clique is a 0-defective clique. One can show that $s$-Defective Clique is W[1]-hard with respect to $k$ even if $c = 2$ by adapting a previous hardness proof for Densest-$k$-Subgraph [29, Theorem 20]. Next, we adapt the algorithm of Theorem 3.1 to obtain an algorithm to enumerate all maximal $s$-defective cliques.

The only difference to the proof of Theorem 3.1 is the following: For bounding the number of $s$-plexes of type 3 the sets $\overline{S_v}$ and $\overline{S_{uv}}$ were bounded by $s-1$ each. Since a maximal $s$-defective clique contains at most $s$ non-edges and $uv \notin E(G)$ we observe that $|\overline{S_v} \cup \overline{S_{uv}}| < s$. Hence, there are at most $2^{\gamma-1}n^{s-1}$ maximal $s$-defective cliques of type 3. Thus, we can bound the overall number of maximal $s$-defective cliques by $2^{\gamma}n^{s+1} + 1$. Since the rest of the proof is completely analogous, we omit it.

▶ **Theorem 3.4.** *For $s \geq 2$, there are $\mathcal{O}(2^{\gamma}n^{s+1})$ maximal $s$-defective cliques in weakly $\gamma$-closed graphs and they can be enumerated in $O(2^{\gamma}n^{s+3})$ time.*

A factor of $n^{s+1}$ in Theorem 3.4 is inevitable due to the following lower bound: Again we consider the graph $G$ consisting of two disjoint cliques $C_1$ and $C_2$, each of size $n/2$. Observe that for each clique $C \subseteq C_1$ of size $s$ and $v \in C_2$, the vertex set $C \cup \{v\}$ is a maximal $s$-defective clique. Thus, $G$ has $\Omega((n/2)^{s+1})$ maximal $s$-defective cliques.

From Theorem 3.4, we obtain the following.

▶ **Corollary 3.5.** *$s$-Defective Clique can be solved in $O(2^{\gamma}n^{s+3})$ time.*

Next, we present faster algorithms in terms of the dependence on $s$. First, we show that each $s$-defective clique can be covered by $\mathcal{O}(\sqrt{s})$ maximal cliques.

▶ **Lemma 3.6.** *Let $S$ be an $s$-defective clique for $s \geq 1$. Then, there is a collection $\mathcal{C}$ of at most $\mathcal{O}(\sqrt{s})$ cliques such that $S \subseteq \bigcup_{C \in \mathcal{C}} C$.*

**Proof.** Consider the graph $H$ obtained by taking the complement of $G[S]$. By definition, $H$ has at most $s$ edges. It suffices to show that there is an $\mathcal{O}(\sqrt{s})$-coloring of $H$ (that is, $\chi(H) = \mathcal{O}(\sqrt{s})$). Although this is known folklore, we describe its proof for the sake of completeness. Consider an optimal coloring. Then, for each pair of colors, say red and blue, there is at least one edge with one endpoint red and the other blue (otherwise we find a coloring with fewer colors). Since $H$ has at most $s$ edges, we obtain $s \leq \binom{\chi(H)}{2}$, or equivalently, $\chi(H) \leq \sqrt{2s + \frac{1}{4}} + \frac{1}{2}$. ◀

Since all cliques (that are not necessarily maximal) can be enumerated in $\mathcal{O}(2^d dn)$ time, we obtain the following:

▶ **Theorem 3.7.** *$s$-Defective Clique can be solved in $2^{\mathcal{O}(d\sqrt{s})}n^{\mathcal{O}(\sqrt{s})}$ time.*

We can also use Lemma 3.6 to obtain an algorithm where the exponent on $n$ is $\mathcal{O}(\sqrt{s})$ in its running time.

▶ **Theorem 3.8.** *s*-Defective Clique *can be solved in* $2^{\mathcal{O}(\gamma\sqrt{s}+s\log k)}n^{\mathcal{O}(\sqrt{s})}$ *time.*

**Proof.** We first enumerate all maximal cliques in $\mathcal{O}^*(3^{\gamma/3})$ time [11]. If there is a clique of size at least $k$, then return Yes. Now we assume that there is no clique of size $k$. By Lemma 3.6, it suffices to check whether there is an $s$-defective clique of size $k$ in $\bigcup_{C\in\mathcal{C}} C$ for each collection $\mathcal{C}$ of $\mathcal{O}(\sqrt{s})$ maximal cliques. Since for each fixed collection in $\mathcal{C}$ there are $\mathcal{O}(k\sqrt{s})$ vertices in $\bigcup_{C\in\mathcal{C}} C$, this can be done in $\mathcal{O}(2^\gamma(\sqrt{s}k)^{\mathcal{O}(s+3)})$ time by applying the algorithm of Corollary 3.5. Since there are $\mathcal{O}^*(3^{\gamma/3})$ maximal cliques, this procedure requires $(3^{\gamma/3}\cdot n^{\mathcal{O}(1)})^{\mathcal{O}(\sqrt{s})}\cdot\mathcal{O}(2^\gamma(\sqrt{s}k)^{\mathcal{O}(s+3)}) = 2^{\mathcal{O}(\gamma\sqrt{s}+s\log k)}n^{\mathcal{O}(\sqrt{s})}$ time. ◀

For $c$-closed graphs, we can obtain an algorithm whose running time does not depend on $k$. This is due to the following lemma.

▶ **Lemma 3.9.** *Let $S\subseteq V(G)$ be an $s$-defective clique in $G$, in which at least one pair of vertices are non-adjacent. Then, $|S|\le c+s$.*

**Proof.** Let $u,v\in S$ be vertices such that $uv\notin E(G)$. We show that $|S'|\le c+s-2$ for $S' := S\setminus\{u,v\}$. Since $G$ is $c$-closed, there are at most $c-1$ vertices in $S'$ adjacent to both $u$ and $v$. Moreover, there are at most $s-1$ vertices in $S'$ which are non-adjacent to either $u$ or $v$ in $S'$, by the definition of $s$-defective cliques. Thus, we obtain $|S'|\le(c-1)+(s-1)=c+s-2$. ◀

▶ **Corollary 3.10.** *s*-Defective Clique *can be solved in* $2^{\mathcal{O}(c\sqrt{s}+s\log(c+s))}n^{\mathcal{O}(\sqrt{s})}$ *time.*

# 4 Bicliques

## 4.1 Non-Induced Biclique

In this subsection, we study problems of finding non-induced maximal bicliques fulfilling certain cardinality constraints. The main problem under consideration is defined as follows.

Non-Induced $(k_1,k_2)$-Biclique
**Input:** A graph $G$ and $k_1,k_2\in\mathbb{N}$.
**Question:** Are there two disjoint sets $S,T$ such that $|S|\ge k_1$, $|T|\ge k_2$, and $st\in E(G)$ for each $s\in S$ and $t\in T$?

Non-Induced $(k_1,k_2)$-Biclique is W[1]-hard even if $k_1=k_2$ [25]. We also consider Non-Induced Max-Edge Biclique in which we demand that $|S|\cdot|T|\ge k$ instead of putting constraints on the partition sizes. Non-Induced Max-Edge Biclique can be solved by solving $\sqrt{k}$ instances of Non-Induced $(k_1,k_2)$-Biclique and thus the latter problem can be considered to be more difficult in our setting. Non-Induced Max-Edge Biclique can be solved in $\mathcal{O}(k^{2.5}k^{\sqrt{k}}n)$ time by applying the algorithm for Induced Max-Edge Biclique on bipartite graphs [10].

Our algorithm for Non-Induced $(k_1,k_2)$-Biclique is based on an FPT algorithm for enumerating all maximal bicliques of the graph and we use this algorithm to solve the aforementioned biclique problems. We need to be careful, however, about what we mean by enumerating bicliques: There is an algorithm that enumerates in $\mathcal{O}^*(2^d)$ time all maximal pairs of sets $S$ and $T$ such that each vertex of $S$ is adjacent to each vertex of $T$ [8].[3] For this enumeration problem, an FPT algorithm for the weak closure is impossible since any

[3] Eppstein [8] describes an algorithm with running time $\mathcal{O}^*(2^{2a})$ for the graph parameter arboricity $a$ which is linearly bounded in $d$ by the inequality $a\le d\le 2a-1$. It can be shown that this algorithm also has running time $\mathcal{O}^*(2^d)$.

clique of size $n$ is 1-closed and admits $\Theta(2^n)$ bipartitions that need to be enumerated. To circumvent this issue, we view a biclique as a vertex set that can be partitioned into such sets $S$ and $T$. Thus, in order to improve the parameterization from $d$ to $\gamma$, we go from an explicit listing of bicliques with bipartitions to a compact representation of bicliques as vertex sets and this is indeed necessary. We say that a vertex set $U \subseteq V(G)$ is a non-induced biclique if $G[U]$ contains a biclique as a (not necessarily induced) subgraph. Note that it can be decided in $\mathcal{O}(n^2)$ time whether a vertex set $U \subseteq V(G)$ is a non-induced biclique or not, because $U$ is a non-induced biclique if and only if the complement of $G[U]$ has multiple connected components. We adapt the algorithm of Theorem 3.1 to obtain an $\mathcal{O}^*(2^\gamma)$-time algorithm to enumerate all maximal non-induced bicliques.

As in the proof of Theorem 3.1, we aim to enumerate all maximal non-induced bicliques in $G$, provided with the collection $\mathcal{S}'$ of all non-induced maximal bicliques in $G' := G - v$. Again we define the same four types of non-induced bicliques. First and foremost, all maximal non-induced bicliques of type 1 and type 2 can be enumerated from $\mathcal{S}'$ in $|\mathcal{S}'| \cdot n^2$ time. We claim that there are at most $2^{\gamma-1}n$ maximal non-induced bicliques of type 3: Let $S$ be such a non-induced biclique. There are at most $n$ choices for $u \in S \setminus N_G[v]$ and there are at most $2^{\gamma-1}$ choices for $N_G(v) \cap S \subseteq N_G(v) \cap N_G(u)$. Note that $S \setminus N_G(v) = \bigcap_{w \in N_G(v) \cap S} N_G(w)$. Finally, there is only one maximal non-induced biclique of type 4, namely $N_G[v]$. Thus, we obtain the following theorem.

▶ **Theorem 4.1.** *All maximal non-induced bicliques can be enumerated in $\mathcal{O}^*(2^\gamma)$ time.*

We show that Non-Induced $(k_1, k_2)$-Biclique can be solved in $\mathcal{O}^*(2^\gamma)$ time, using this enumeration algorithm.

▶ **Theorem 4.2.** *Non-Induced $(k_1, k_2)$-Biclique can be solved in $\mathcal{O}^*(2^\gamma)$ time.*

**Proof.** For each non-induced biclique $U$ of size at least $k_1 + k_2$, we construct an instance of Subset Sum defined as follows:

Subset Sum
**Input:** A set $A = \{a_1, \ldots, a_n\}$ of $n$ positive integers and $k_1 \le k_2 \in \mathbb{N}$.
**Question:** Is there a set $B \subseteq A$ such that $k_1 \le \sum_{b \in B} \le k_2$?

A standard dynamic programming algorithm can solve Subset Sum in $\mathcal{O}(n \cdot \sum_{a \in A} a)$ time. To solve Non-Induced $(k_1, k_2)$-Biclique, we construct an instance $(A', k_1', k_2')$ of Subset Sum, where $k_1' := k_1$, $k_2' := |U| - k_2$, and $A' := \{|C_i| \mid i \in [\ell]\}$ for the connected components $C_1, \ldots, C_\ell \subseteq V(G)$ of the complement of $G[U]$. It is easy to see that $(G, k_1, k_2)$ is a Yes-instance if and only if the constructed instance of Subset Sum is a Yes-instance for some maximal non-induced biclique $U$. ◀

We obtain the following result by the abovementioned reduction.

▶ **Corollary 4.3.** *Non-Induced Max-Edge Biclique can be solved in $\mathcal{O}^*(2^\gamma)$ time.*

## 4.2 Induced Biclique

In this subsection, we study problems of finding *induced* maximal bicliques fulfilling certain cardinality constraints. Formally, these problems are defined as follows.

Induced $(k_1, k_2)$-Biclique
**Input:** A graph $G$ and $k_1, k_2 \in \mathbb{N}$ such that $k_1 \le k_2$.
**Question:** Are there two disjoint vertex sets $S, T$ such that (1) $|S| = k_1$ and $|T| = k_2$, (2) $ss' \notin E(G)$ for each $s, s' \in S$, (3) $tt' \notin E(G)$ for each $t, t' \in T$, and (4) $st \in E(G)$ for each $s \in S$ and $t \in T$?

When $k_1 = k_2$, we will refer to the problem as INDUCED $(k,k)$-BICLIQUE. Moreover, we refer to the problem variant where we aim to maximize the number of edges in the biclique as INDUCED MAX-EDGE BICLIQUE. INDUCED $(k_1,k_2)$-BICLIQUE is W[1]-hard even if $k_1 = k_2$ [5]. INDUCED MAX-EDGE BICLIQUE is NP-hard [27] and W[1]-hardness can be shown by a simple reduction from INDEPENDENT SET parameterized by the solution size $k$. As in the non-induced case, INDUCED MAX-EDGE BICLIQUE can be solved by solving $\sqrt{k}$ instances of INDUCED $(k_1,k_2)$-BICLIQUE. Thus, positive results for INDUCED $(k_1,k_2)$-BICLIQUE transfer to INDUCED MAX-EDGE BICLIQUE. All maximal induced bicliques can be enumerated in $\mathcal{O}^*(3^{(\Delta+d)/3})$ time [16] and it is impossible to obtain an FPT algorithm for the enumeration of maximal induced bicliques for the parameter $d$ because a graph may have too many maximal induced bicliques [16].

First, we present an FPT algorithm for INDUCED $(k,k)$-BICLIQUE parameterized by $\gamma$.

▶ **Theorem 4.4.** *INDUCED $(k,k)$-BICLIQUE can be solved in $\mathcal{O}^*(\gamma^{\mathcal{O}(\gamma)})$ time.*

**Proof.** Since a biclique $K_{\gamma,\gamma}$ is not weakly $\gamma$-closed, $(G,k)$ is a No-instance if $k \geq \gamma$. Moreover, INDUCED $(k,k)$-BICLIQUE is trivially solvable in polynomial time when $k \leq 1$. Hence, we may assume that $2 \leq k < \gamma$. Let $\sigma$ be a fixed closure ordering of $G$. Suppose that $(S,T)$ is a solution of $(G,k)$. We consider each choice of the vertex $v \in S \cup T$ that appears in $\sigma$ before all other vertices of $S \cup T$. We assume without loss of generality that $v$ lies in $S$. Let $G'$ be the graph obtained by removing all vertices preceding $v$ in $\sigma$. Then, we additionally consider each choice of a vertex $v' \in V(G') \setminus \{v\}$ which is contained in $S$ and an independent set $T \subseteq N_{G'}(v) \cap N_{G'}(v')$ of at least $k$ vertices. Since $|N_{G'}(v) \cap N_{G'}(v')| < \gamma$, there are at most $2^\gamma$ possibilities for $T$. Now, it remains to find an independent set $S \subseteq \bigcap_{u \in T} N_{G'}(u)$ of size at least $k$ in $G'$. By Corollary 2.4, this can be achieved in $\mathcal{O}^*((\gamma k^2)^k)$ time. Since $k < \gamma$, the overall running time is $\mathcal{O}^*(2^\gamma \gamma^{3\gamma}) = \mathcal{O}^*(\gamma^{\mathcal{O}(\gamma)})$. ◀

For $c$-closed graphs, we show that there is a single-exponential time algorithm when $k_1 \geq 2$. Our algorithm is based on a reduction to a variant of INDEPENDENT SET called BICOLORED INDEPENDENT SET [4].

▶ **Theorem 4.5.** *INDUCED $(k_1,k_2)$-BICLIQUE can be solved in $O^*(1.6107^c)$ time if $k_1 \geq 2$.*

**Proof.** Our algorithm is based on reductions to the following variant of INDEPENDENT SET:

BICOLORED INDEPENDENT SET
**Input:** A graph $G$, a partition $(V_1,V_2)$ of $V(G)$, and $k_1,k_2 \in \mathbb{N}$.
**Question:** Is there an independent set $I \subseteq V(G)$ with $|I \cap V_1| = k_1$ and $|I \cap V_2| = k_2$?

For each induced cycle $(u_S,u_T,v_S,v_T)$ on four vertices in $G$, we construct an instance $(G',V_1',V_2',k_1,k_2)$ of BICOLORED INDEPENDENT SET, where $V_1' := N_G(u_S) \cap N_G(v_S)$, $V_2' := N_G(u_T) \cap N_G(v_T)$, and $G' = (V_1' \cup V_2', E(G[V_1']) \cup E(G[V_2']) \cup \{v_1'v_2' \mid v_1' \in V_1', v_2' \in V_2', v_1'v_2' \notin E(G)\})$. By the $c$-closure of $G$, there are at most $2c-2$ vertices on $G'$. It is easy to verify that there is a $(k_1,k_2)$-biclique containing $u_S,u_T,v_S,v_T$ if and only if $(G',V_1',V_2',k_1,k_2)$ is a Yes-instance. Since BICOLORED INDEPENDENT SET is $O^*(1.2691^n)$-time solvable on $n$-vertex graphs [4], we obtain an $O^*(1.6107^c)$-time algorithm for INDUCED $(k_1,k_2)$-BICLIQUE. ◀

Gaspers et al. [13] provided an $\mathcal{O}^*(3^{n/3})$-time algorithm to enumerate all maximal bicliques. By using a reduction similar to the one in the proof Theorem 4.5, we can thus enumerate all maximal bicliques in which each part has at least two vertices in $\mathcal{O}^*(3^{2c/3})$ time. However, even 2-closed graphs may have $\Omega(3^{n/3})$ maximal bicliques: Consider the graph with a single universal vertex $u$ and $(n-1)/3$ disjoint triangles. Observe that there are $3^{(n-1)/3}$ maximal bicliques where one part consists of $u$.

In contrast to this positive result for $k_1 \geq 2$, we prove that Induced $(1, k)$-Biclique is NP-hard even on graphs with constant $h$-index, $c$-closure, and weak $\gamma$-closure.

▶ **Theorem 4.6.** *Induced Max-Edge Biclique remains NP-hard even on graphs with $h$-index four, $c$-closure three and weak $\gamma$-closure two.*

**Proof.** We will reduce from Independent Set, which is NP-hard even on cubic graphs (graphs in which each vertex has degree at most three) [12]. We assume that $k \geq 10$, since otherwise the instance $(G, k)$ can be solved in polynomial time. We construct an instance $(G', k)$ of Induced $k$-Edge Biclique as follows: We begin with a copy of $G$. Then, each edge $uv \in E(G)$ is replaced by a path on four vertices $u, u_v, v_u$, and $v$. Finally, we introduce a new universal vertex $w$. It is easy to see that $G'$ has $h$-index four, is 3-closed, and weakly 2-closed. Next, we prove that $G$ contains an independent set of size $k$ if and only if $G'$ contains an induced biclique with at least $k + |E(G)|$ edges.

Suppose that $G$ contains an independent set $I$ of size at least $k$. Then, there is an independent set $I'$ of size $k + |E(G)|$ in $G' - w$. Thus, the set $I' \cup \{w\}$ is an induced biclique with at least $k + |E(G)|$ edges in $G'$.

Conversely, suppose that $G'$ contains a biclique $(S, T)$ with at least $k$ edges. Since each vertex in $G' - w$ has degree at most three and $k \geq 10$, we see that vertex $w$ is contained in $(S, T)$. Without loss of generality, assume that $w \in S$. Since $w$ is a universal vertex, we obtain $S = \{w\}$. It follows that $T$ is an independent set of size at least $k + |E(G)|$ in $G'$. Hence, $G$ contains an independent set of size at least $k$. ◀

The reduction in the proof of Theorem 4.6 also implies the following.

▶ **Theorem 4.7.** *Induced $(k_1, k_2)$-Biclique is NP-hard on graphs with $h$-index four, $c$-closure three and weak $\gamma$-closure two even if $k_1 = 1$.*

To complete the dichotomy with respect to $c$, we prove that Induced Max-Edge Biclique can be solved in polynomial time if $c = 2$. To do so, we first show that Induced $(1, k_2)$-Biclique can be solved in polynomial time if the input graph is diamond-free.

▶ **Proposition 4.8.** *Induced $(k_1, k_2)$-Biclique can be solved in polynomial time on diamond-free graphs if $k_1 = 1$.*

**Proof.** Suppose that the input graph $G$ is diamond-free. Then, $G[N(v)]$ is a disjoint union of cliques for each vertex $v \in V(G)$. Thus, $(G, k_1, k_2)$ is a Yes-instance if and only if there is a vertex $v \in V(G)$ such that $G[N(v)]$ has at least $k_2$ connected components. ◀

Since 2-closed graphs are diamond-free, we obtain the following from Proposition 4.8 and Theorem 4.5.

▶ **Corollary 4.9.** *Induced $(k_1, k_2)$-Biclique and Induced Max-Edge Biclique can be solved in polynomial time on 2-closed graphs.*

## 5 Variants of Dominating Set

In companion work [20], we showed that Dominating Set admits a kernel of size $k^{\mathcal{O}(c)}$. We were not able to resolve the parameterized complexity of Dominating Set in weakly $\gamma$-closed graphs. However, we develop FPT algorithms for the related Independent Dominating Set and Dominating Clique problems in weakly $\gamma$-closed graphs.

**Algorithm 1** An FPT algorithm *SolveIDS* to solve INDEPENDENT DOMINATING SET.

```
1: function SolveIDS(G, k)
2:     if k = 0 and V(G) ≠ ∅ then return No
3:     Let I := ∅ and G' := G.    ▷ I will be an independent set of size at most k + 1 in G
4:     while V(G') ≠ ∅ and |I| ≤ k do
5:         Let v be a vertex such that cl_{G'}(v) ≤ γ − 1.
6:         I := I ∪ {v} and G' := G' − N_{G'}[v].
7:     if |I| ≤ k then return Yes
8:     else
9:         P := {v | v is a common neighbor of at least two vertices in I}
10:        for each u ∈ P do
11:            if SolveIDS(G − N_G[u], k − 1) returns Yes then return Yes
12:    return No
```

## 5.1 Independent Dominating Set

We consider the INDEPENDENT DOMINATING SET problem. The task in this problem is to find an independent set $S$ of size at most $k$ dominating all vertices.

INDEPENDENT DOMINATING SET
**Input:** A graph $G$ and $k \in \mathbb{N}$.
**Question:** Is there a vertex set $S \subseteq V(G)$ such that $|S| \leq k$, $S$ induces an independent set in $G$, and for each $v \in V$ it holds that $S \cap N[v] \neq \emptyset$?

INDEPENDENT DOMINATING SET is W[2]-complete [7]. There are several fixed-parameter tractability results in restricted graph classes: INDEPENDENT DOMINATING SET has a kernel of $\mathcal{O}(d^2 k^{d+1})$ vertices computable in $\mathcal{O}^*(2^d)$ time [28]. Moreover, when the graph contains no cycles of length three or four, INDEPENDENT DOMINATING SET can be solved in $\mathcal{O}^*(k^{\mathcal{O}(k)})$ time [29].

We present an FPT algorithm *SolveIDS* (Algorithm 1) with running time $\mathcal{O}^*((\frac{\gamma-1}{2})^k k^{2k})$. Note that our algorithm extends the $\mathcal{O}^*(k^{\mathcal{O}(k)})$ time algorithm of Raman and Saurabh [29], because any graph without cycles of length three or four is 2-closed. Algorithm 1 first greedily computes an independent set $I$ of size at most $k+1$ by iteratively choosing vertices $v$ such that $\mathrm{cl}_{G'}(v) \leq \gamma - 1$ (Line 5). If $I$ is inclusion-maximal and of size at most $k$, then $I$ constitutes a solution. Otherwise, we find a vertex set $P$ to branch on. The choice of $I$ will ensure that $P$ has at most $(\gamma-1)\binom{k+1}{2}$ vertices.

▶ **Theorem 5.1.** *INDEPENDENT DOMINATING SET can be solved in $\mathcal{O}^*((\frac{\gamma-1}{2})^k k^{2k})$ time.*

**Proof.** We show that Algorithm 1 solves an instance $(G, k)$ of INDEPENDENT DOMINATING SET in the claimed time. First, we prove the correctness of Algorithm 1. Let $I$ be the independent set of size at most $k+1$ of $G$ obtained in Lines 3 to 6. Suppose that $|I| \leq k$. Since $I$ is a maximal independent set, each vertex $v \in V(G)$ is either contained in $I$ or a neighbor of a vertex in $I$. Hence, $I$ is an independent dominating set of size at most $k$ of $G$. Thus, $(G, k)$ is a Yes-instance. Now suppose that $|I| = k+1$. Let $P$ be the set of vertices in $G$ which have at least two neighbors in $I$ (Line 9). Since $|I| = k+1$, the sought solution must contain at least one vertex of $P$. Thus, the branching in Line 11 is correct.

Now, we analyze the time complexity of Algorithm 1. To do so, we prove that $|P| \leq (\gamma-1)\binom{k+1}{2}$. Let $v_i$ be the $i$th vertex added to $I$ in Line 6 and let $G_i := G - N_G[\{v_1, \ldots, v_{i-1}\}]$ for each $i \in [k+1]$. Since $P \subseteq \bigcup_{i \in [k+1]} N_{G_i}(v_i)$, we see that $|P| \leq \sum_{i \in [k+1]} |N_{G_i}(v_i) \cap P|$.

**Algorithm 2** An algorithm for finding a dominating clique $S$. Vertex $v_i$ is the first vertex of the dominating clique $S$ in the fixed closure ordering $\sigma$ of $G$. Initially we have $T := \{v_i\}$.

```
function SolveDC(G, k, T)                                   ▷ T ⊆ {v_i, ..., v_n} and v_i ∈ T
    if k = 0 and V(G) ≠ N[T] then return No
    if V(G) = N[T] then return Yes
    Compute a vertex w such that v_i w ∉ E(G)
    for each u ∈ ⋂_{x∈T} N(x) ∩ N(w) ∩ V(G_i) do             ▷ G_i := G[v_i, ..., v_n]
        if SolveDC(G − (N(u) \ N[v_i]), k − 1, T ∪ {u}) returns Yes then return Yes
    return No
```

Moreover, we have $|N_{G_i}(v_i) \cap P| \leq \sum_{j \in [i+1,k+1]} |N_{G_i}(v_i) \cap N_{G_i}(v_j)|$ for each $i \in [k]$. Therefore,

$$|P| \leq \sum_{i<j \in [k+1]} |N_{G_i}(v_i) \cap N_{G_i}(v_j)| \leq (\gamma - 1)\binom{k+1}{2}.$$

Here, the last inequality is due to the fact that $\mathrm{cl}_{G_i}(v_i) \leq \gamma - 1$ for each $i \in [k]$.

It is easy to see that finding an independent set $I$ in Lines 3 to 6 only requires polynomial time. Since each node has at most $(\gamma-1)\binom{k+1}{2}$ children in the search tree and its depth is at most $k$, the overall running time of Algorithm 1 is $\mathcal{O}^*((\gamma - 1 \cdot \binom{k+1}{2})^k) = \mathcal{O}^*((\frac{\gamma-1}{2})^k k^{2k})$. ◀

## 5.2 An FPT algorithm for Dominating Clique

We now consider the Dominating Clique problem. The task in this problem is to find a clique $S$ of size at most $k$ dominating all vertices.

Dominating Clique
**Input:** A graph $G$ and a parameter $k \in \mathbb{N}$.
**Question:** Is there a vertex set $S \subseteq V(G)$ such that $|S| \leq k$, $S$ induces a clique in $G$, and for each $v \in V$ it holds that $S \cap N[v] \neq \emptyset$?

It is known that Dominating Clique is W[2]-hard even on graphs which do not contain a 4-claw (a $K_{1,4}$) as an induced subgraph [6].

Note that there is a straightforward $\mathcal{O}^*(d^k)$-time algorithm: Choose a vertex $v$ with minimum degree and consider the case that the dominating clique contains $v$. Since $G$ is $d$-degenerate, there are $\mathcal{O}(d^{k-1})$ choices for the remaining vertices of the dominating clique. If there is no dominating clique that contains $v$, then continue to search the dominating clique in the graph $G - v$. Eventually, we find either a dominating clique of size at most $k$ or arrive at an empty graph.

In this subsection, we describe an FPT algorithm for weakly $\gamma$-closed graphs, resulting in an $\mathcal{O}^*((\gamma-1)^{k-1})$-time algorithm. Note that a maximal clique of a weakly $\gamma$-closed graph may be arbitrarily large. Thus, a simple brute-force search on maximal cliques may require $\Omega(n^k)$ time. Moreover, we want to avoid enumerating all maximal cliques since this alone incurs a running time of $\Omega(3^{\gamma/3})$. Instead, we will use Algorithm 2 for each vertex $v_i$ in a fixed closure ordering $\sigma$. The key idea is that we assume that $v_i$ is the first vertex in the dominating clique with respect to $\sigma$. As we shall see in the proof of Theorem 5.2 this guarantees that for each vertex $w$ which is not adjacent to $v_i$, we may branch into at most $\gamma - 1$ cases to determine a vertex that dominates $w$.

▶ **Theorem 5.2.** *Dominating Clique can be solved in $\mathcal{O}^*((\gamma-1)^{k-1})$ time.*

**Proof.** To solve an instance $(G, k)$ of Dominating Clique, we first compute a closure ordering $\sigma$. Afterwards, we invoke *SolveDC* on input $(G, k-1, \{v_i\})$ for each vertex $v_i \in V$. In the call *SolveDC*$(G, k-1, \{v_i\})$, we assume that $v_i$ is the first vertex in the dominating clique $S$ with respect to the closure ordering $\sigma$.

We first show that *SolveDC*$(G, k, T)$ is correct in the following sense: it returns Yes if and only if there is a dominating clique $S$ of size at most $k$ which contains all vertices of $T$, and vertex $v_i$ is the first vertex in $S$ with respect to $\sigma$ (where $v_i$ is the minimal vertex of $T$ with respect to $\sigma$). It is easy to see that the terminal conditions in Lines 2 and 3 are correct. Let $w$ be the vertex computed in Line 4. Since we want to compute a dominating clique $S$ which contains $T$, where vertex $v_i$ is the first vertex in $S$ with respect to the closure ordering $\sigma$ and since $v_i w \notin E(G)$, any dominating clique must contain at least one vertex $u$ of $\bigcap_{x \in T} N(x) \cap N(w) \cap V(G_i)$. Thus, the branching in Lines 5 and 6 is correct. Since each vertex $u$ chosen in Line 5 is a common neighbor of all vertices in $T$, we conclude that the set $T$ is a clique and thus Line 3 returns Yes if and only if $G$ contains a dominating clique of size at most $k$. Furthermore, each vertex $u$ chosen in Line 5 is contained in $G_i$. Hence $v_i <_\sigma u$. In other words, vertex $v_i$ is the smallest vertex in $T$ with respect to $\sigma$.

Let us analyze the time complexity of *SolveDC*. It is easy to see that Lines 2 to 4 can be performed in polynomial time. Consider the search tree where each node corresponds to an invocation of *SolveDC*. We show that each node in the search tree has at most $\gamma - 1$ children. To this end, we bound the size of $|N(v_i) \cap N(w) \cap V(G_i)|$ which is an upper bound on the number of branches created in Line 5. If $v_i <_\sigma w$, then $|N(v_i) \cap N(w) \cap V(G_i)| \leq \gamma - 1$ by Definition 1.3. Otherwise, if $w <_\sigma v_i$, then $v_i$ and $w$ have at most $\gamma - 1$ common neighbors in $\{v' \mid w <_\sigma v'\}$ and thus also in $V(G_i)$. Hence, each node has at most $\gamma - 1$ children. Moreover, the depth of the search tree is at most $k - 1$. Thus, we spend $\mathcal{O}^*((\gamma-1)^{k-1})$ time for each vertex $v_i \in V(G)$ and the claimed running time bound follows. ◀

In companion work [20], we provided a reduction from $\lambda$-Hitting Set to Dominating Set that implies that kernels of size $\mathcal{O}(k^{c-1-\epsilon})$ are impossible under some standard complexity-theoretic assumptions. We obtain the following from the same reduction.

▶ **Proposition 5.3.** *For $c \geq 3$, Dominating Clique has no kernel of size $\mathcal{O}(k^{c-1-\epsilon})$ unless coNP ⊆ NP/poly.*

▶ **Proposition 5.4.** *Unless the ETH fails, there is no $n^{o(k)}$-time algorithm for Dominating Clique.*

Hence, it is unlikely that the running time of Theorem 5.2 can be substantially improved.

## 6 Conclusion

We have provided further applications of the weak closure parameter $\gamma$ which was introduced for clique enumeration [11]. Given the usefulness of the class of weakly closed graphs, it seems important to further study their properties. For example, it would be nice to obtain a forbidden subgraph characterization. We note that the weakly-1-closed graphs are exactly the graphs that do not contain a $C_4$ or a $P_4$ as an induced subgraph. These graphs are also known as quasi-threshold graphs. Can we obtain a similar characterization for weakly 2-closed graphs? Further FPT algorithms for the parameter $\gamma$ would also be very interesting from a theoretical and practical point of view. For example, it is open whether Dominating Set has an FPT algorithm for the parameter $\gamma + k$; so far it is known only to have FPT algorithms for $d + k$ [1] and $c + k$ [20].

## References


1. Noga Alon and Shai Gutner. Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. *Algorithmica*, 54(4):544–556, 2009.
2. 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 (KDD '17)*, pages 115–124. ACM, 2017.
3. Alessio Conte, Tiziano De Matteis, Daniele De Sensi, Roberto Grossi, Andrea Marino, and Luca Versari. D2K: scalable community detection in massive networks via small-diameter $k$-plexes. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '18)*, pages 1272–1281. ACM, 2018.
4. Jean-François Couturier and Dieter Kratsch. Bicolored independent sets and bicliques. *Information Processing Letters*, 112(8-9):329–334, 2012.
5. Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
6. Marek Cygan, Geevarghese Philip, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Dominating set is fixed parameter tractable in claw-free graphs. *Theoretical Computer Science*, 412(50):6982–7000, 2011.
7. Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
8. David Eppstein. Arboricity and bipartite subgraph listing algorithms. *Information Processing Letters*, 51(4):207–211, 1994.
9. David Eppstein, Maarten Löffler, and Darren Strash. Listing all maximal cliques in large sparse real-world graphs. *ACM Journal of Experimental Algorithmics*, 18, 2013.
10. Qilong Feng, Shaohua Li, Zeyang Zhou, and Jianxin Wang. Parameterized algorithms for edge biclique and related problems. *Theoretical Computer Science*, 734:105–118, 2018.
11. 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.
12. M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
13. Serge Gaspers, Dieter Kratsch, and Mathieu Liedloff. On independent sets and bicliques in graphs. *Algorithmica*, 62(3-4):637–658, 2012.
14. Petr A. Golovach and Yngve Villanger. Parameterized complexity for domination problems on degenerate graphs. In *Proceedings of the 34th International Workshop Graph-Theoretic Concepts in Computer Science (WG '08)*, volume 5344 of *Lecture Notes in Computer Science*, pages 195–205, 2008.
15. Martin Grohe, Stephan Kreutzer, and Sebastian Siebertz. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM*, 64(3):17:1–17:32, 2017.
16. Danny Hermelin and George Manoussakis. Efficient enumeration of maximal induced bicliques. *Discrete Applied Mathematics*, 2020. To appear. `doi:10.1016/j.dam.2020.04.034`.
17. Edin Husić and Tim Roughgarden. FPT algorithms for finding dense subgraphs in $c$-closed graphs. *CoRR*, abs/2007.09768, 2020. `arXiv:2007.09768`.
18. Subhash Khot and Venkatesh Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289(2):997–1008, 2002.
19. Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. Computing dense and sparse subgraphs of weakly closed graphs. *CoRR*, abs/2007.05630, 2020. `arXiv:2007.05630`.
20. Tomohiro Koana, Christian Komusiewicz, and Frank Sommer. Exploiting c-closure in kernelization algorithms for graph problems. In *Proceedings of the 28th Annual European Symposium on Algorithms (ESA '20)*, volume 173 of *LIPIcs*, pages 65:1–65:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

21 Tomohiro Koana and André Nichterlein. Detecting and enumerating small induced subgraphs in $c$-closed graphs. *CoRR*, abs/2007.12077, 2020. arXiv:2007.12077.

22 Christian Komusiewicz. Multivariate algorithmics for finding cohesive subnetworks. *Algorithms*, 9(1):21, 2016.

23 Christian Komusiewicz, Falk Hüffner, Hannes Moser, and Rolf Niedermeier. Isolation concepts for efficiently enumerating dense subgraphs. *Theoretical Computer Science*, 410(38-40):3640–3654, 2009.

24 Christian Komusiewicz and Manuel Sorge. An algorithmic framework for fixed-cardinality optimization in sparse graphs applied to dense subgraph problems. *Discrete Applied Mathematics*, 193:145–161, 2015.

25 Bingkai Lin. The parameterized complexity of the $k$-biclique problem. *Journal of the ACM*, 65(5):34:1–34:23, 2018.

26 Eugene M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982.

27 René Peeters. The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131(3):651–654, 2003.

28 Geevarghese Philip, Venkatesh Raman, and Somnath Sikdar. Polynomial kernels for dominating set in graphs of bounded degeneracy and beyond. *ACM Transactions on Algorithms*, 9(1):11:1–11:23, 2012.

29 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.

30 Detlef Seese. Linear time computable problems and first-order descriptions. *Mathematical Structures in Computer Science*, 6(6):505–526, 1996.