The parameterized complexity of k-edge induced subgraphs

Bingkai Lin[a,*], Yijia Chen[b]

[a] *JST, ERATO, Kawarabayashi Large Graph Project, NII, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan*
[b] *School of Computer Science, Fudan University*

Abstract

We prove that finding a k-edge induced subgraph is fixed-parameter tractable, thereby answering an open problem of Leizhen Cai [4]. Our algorithm is based on several combinatorial observations, Gauss' famous *Eureka* theorem [3], and a generalization of the well-known fpt-algorithm for the model-checking problem for first-order logic on graphs with locally bounded tree-width due to Frick and Grohe [17]. On the other hand, we show that two natural counting versions of the problem are hard. Hence, the k-edge induced subgraph problem is one of the very few known examples in parameterized complexity that are easy for decision while hard for counting.

Keywords: k-edge induced subgraph, fixed-parameter tractable, #W[1]-hard

1. Introduction

Imagine that a metropolitan area is to build up a large subway system, which by oversimplification is represented by a graph G in which vertices are stations and edges are direct links between two stations. Due to the high cost of the construction and the limited budget, say we can only build 20 direct links this year. This clearly amounts to find a subgraph graph H of G which contains exactly 20 edges. Note it might happen that there are two stations in H whose planned link in G is not included in H. To save future expense, we would like to ensure that this does not happen. Or equivalently, we demand H to be an *induced subgraph* of G.

Induced subgraphs are one of the most natural substructures in graphs. They capture many different combinatorial objects, e.g., cliques, independent sets, chordless paths. Thus, a great number of algorithmic problems are about finding certain induced subgraphs, and their complexity is among the mostly extensively studied in algorithmic graph theory [5, 9, 10, 24, 25, 27, 28, 30]. As mentioned, the above construction problem can be modelled as finding an induced subgraph which contains exactly k edges, i.e., a k-edge induced subgraph. And in fact, this problem is equivalent to solving a special 0-1 quadratic Diophantine equation $x^T Ax = 2k$, where A is the adjacency matrix of G and $x \in \{0,1\}^n$ with $n = |V(G)|$. Induced subgraphs with distinct number

*Corresponding author
Email addresses: `lin@nii.ac.jp` (Bingkai Lin), `yijiachen@fudan.edu.cn` (Yijia Chen)

of edges have also been studied in graph theory [1, 2].

It is not difficult to prove that the k-edge induced subgraph problem is NP-hard by a reduction from the clique problem. As the number k can be expected to be small, 20 in the above example, in typical applications, we approach the problem via parameterized complexity [13, 16, 29] and treat k as the parameter:

k-EDGE-INDUCED-SUBGRAPH	
Instance:	A graph G and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Decide whether G contains a k-edge induced subgraph.

As the main result of our paper, we show that k-EDGE-INDUCED-SUBGRAPH is fixed-parameter tractable. In fact, there are special cases of k-EDGE-INDUCED-SUBGRAPH whose fixed-parameter tractability has been known for a while. Since we can define a k-edge induced subgraph by a first-order sentence, using logic machinery, it can be shown that k-EDGE-INDUCED-SUBGRAPH is fixed-parameter tractable if the graph G has bounded tree-width [11], bounded local tree-width [17], locally bounded expansion [14], etc., or most generally nowhere dense [22]. Unfortunately, the class of all graphs containing a k-edge induced subgraph does not possess any of these bounded measures. As another previously known case, using his *Random Separation* method [7] and Ramsey's Theorem, Cai [6] gave a very nice combinatorial algorithm[1] that solves k-EDGE-INDUCED-SUBGRAPH when the parameter k is a *triangular number*, i.e., $k = \binom{m}{2}$ for some $m \in \mathbb{N}$. However, it looks very difficult to adapt Cai's algorithm to handle arbitrary k. Therefore neither logic nor combinatorial approach so far seems to be sufficient to settle the complexity of k-EDGE-INDUCED-SUBGRAPH on its own. So our fpt-algorithm is a rather tricky combination of these two methods.

Our approach. As just mentioned, our starting point is that the existence of a k-edge induced subgraph can be characterized by a sentence of first-order logic (FO) which depends on k only. It is a well-known result of Frick and Grohe [17] that the model-checking problem for FO on graphs of bounded *local tree-width* is fixed-parameter tractable. The local tree-width of a graph is a function bounding the tree-width of the induced subgraphs on the neighborhoods within a certain radius of every vertex. For instance, bounded-degree graphs and planar graphs have bounded local tree-width. These give immediately the fixed-parameter tractability of k-EDGE-INDUCED-SUBGRAPH on graphs with bounded degree[2].

---

[1] For the interested reader, we include a proof in Appendix.B.

[2] This is also a direct consequence of Seese's result that the model-checking problem for FO on bounded-degree graphs is fixed-parameter tractable [32]. But we find it more natural to work with bounded local tree-with in the following generalization.

With some more effort, the above result can be extended to graphs G with degree bounded by a function of the parameter k. In that case, we can say the degree $\deg(v)$ of each vertex v is sufficiently small. The corresponding fpt-algorithm generalizes Frick and Grohe's Theorem to graphs with local tree-width bounded by a function of both the radius of the neighborhoods and an additional parameter. As a dual, if $\deg(v)$ of each vertex v in G is sufficiently large, or more precisely, the complement of G has degree bounded by a function of k, then we can decide k-EDGE-INDUCED-SUBGRAPH in fpt time, too.

Moving one step further, we consider graphs in which each $\deg(v)$ is either sufficiently small or sufficiently large, e.g., an n-star. We call such graphs *degree-extreme*. Using the same logic machinery as above, we then are able to show the fixed-parameter tractability of k-EDGE-INDUCED-SUBGRAPH on degree-extreme graphs.

Assume that the graph G is not degree-extreme, i.e., there exists a vertex v_0 whose degree is neither sufficiently small nor sufficiently large. We partition the vertex set of G into two sets V_1 and V_2, where V_1 contains all vertices adjacent to v_0 and V_2 the remaining vertices. Then both V_1 and V_2 are relatively large. Note possibly there are many edges between V_1 and V_2. Nevertheless, we can compute a vertex set B in G such that every edge between V_1 and V_2 has one vertex in B; and if B is large enough, we can show that G contains a k-edge induced subgraph. Otherwise, the graph G consists of two induced subgraphs $G[V_1]$ and $G[V_2]$, plus the edges between V_1 and V_2 adjacent to the set B of bounded size. In case $G[V_1]$ and $G[V_2]$ are both degree-extreme, we call such a graph G a *bridge* (of two degree-extreme graphs). By the logic method again, we prove that k-EDGE-INDUCED-SUBGRAPH is fixed-parameter tractable on bridges.

Now we are left with the case that at least one of $G[V_1]$ and $G[V_2]$ is not degree-extreme, say $G[V_1]$. Then we repeat the above procedure on $G[V_1]$ to get a partition $V_{11} \mid V_{12}$ of V_1. And again, both V_{11} and V_{12} are sufficiently large. Arguing as before, either we already know $G[V_1]$, and hence G, contains a k-edge induced subgraph, or there is a set B_1 of bounded size such that every edge between V_{11} and V_{22} intersects B_1.

Finally we remove the vertex set $B_0 := B \cup B_1$ from G. Then $G[V \setminus B_0]$ is the disjoint union of $G[V_{11} \setminus B_0]$, $G[V_{12} \setminus B_0]$ and $G[V_2 \setminus B_0]$. Moreover, all three induced subgraphs are so large that, by Ramsey's Theorem, either one of them contains a large independent set, or we have three large disjoint cliques which are not adjacent to each other. For both cases, we show that $G[V \setminus B_0]$, and hence G, contains a k-edge induced subgraph. As a matter of fact, the second case is an easy consequence of a famous number-theoretic result of Gauss which states that *every natural number is the sum of three triangular numbers.*

We should mention that the running time of our algorithm is

$$2^{2^{O(k)}} \cdot |G|,$$

i.e., linear in terms of the graph, yet *double exponential* in terms of the parameter. This makes our algorithm impractical. But we hope that similar as it happened in many other cases the knowledge that the k-edge problem is fixed-parameter tractable will encourage to look for faster algorithms or at least for algorithms useful in practice for concrete classes of instances of the problem (For example, see the conclusion section of [20]).

Counting k-edge induced subgraphs. We also study the parameterized complexity of computing the number of k-edge induced subgraphs. For most natural problems, if the decision version is easy, then so is the counting problem. However, it turns out that two natural counting versions of k-EDGE-INDUCED-SUBGRAPH are both hard. To the best of our knowledge, there are only very few natural problems which exhibit such a phenomenon [15, 8].

Organization of our paper. In Section 2 we introduce necessary background and fix our notations. We prove all required combinatorial results in Section 3. In particular, we present several simple structures in a graph which, if it exists, guarantee the existence of a k-edge induced subgraph. Then in Section 4 we establish the fixed-parameter tractability of k-EDGE-INDUCED-SUBGRAPH on degree-extreme graphs and bridges using model-checking problems for FO. In order to have a better analysis of the running time, in Section 4.1 we give combinatorial algorithms for those cases, which can be understood as instantiations of the logic-based algorithm. We present our fpt-algorithm for k-EDGE-INDUCED-SUBGRAPH by putting all the pieces together in Section 5. Finally in Section 6 we prove the hardness of the counting problems. For readers not familiar with [17], we provide a proof of the easy generalization of Frick and Grohe's algorithm in Appendix. A.

2. Preliminaries

$\mathbb{N}$ and $\mathbb{N}^+$ denote the sets of natural numbers (that is, nonnegative integers) and positive integers, respectively. For a natural number n let $[n] := \{1, \dots, n\}$.

We denote the alphabet $\{0, 1\}$ by Σ and identify problems with subsets Q of Σ^*. Clearly, as done mostly, we present concrete problems in a verbal, hence uncodified form over Σ.

For every set S we use $|S|$ to denote its size. Moreover we let $\binom{S}{2}$ be the set of all two-element subsets of S, i.e.,$\big\{\{a, b\} \bigm| a, b \in S \text{ and } a \neq b\big\}$. A triangular number is $\binom{k}{2} := \left|\binom{[k]}{2}\right|$ for some $k \in \mathbb{N}$. In particular, $\binom{0}{2} = \binom{1}{2} = 0$.

Parameterized complexity. A *parameterized problem* is a pair (Q, κ) consisting of a classical problem $Q \subseteq \Sigma^*$ and a polynomial time computable *parameterization* $\kappa : \Sigma^* \to \mathbb{N}$.

An algorithm $\mathbb{A}$ is an fpt-*algorithm with respect to a parameterization* κ if for every $x \in \Sigma^*$ the running time of $\mathbb{A}$ on x is bounded by $f(\kappa(x)) \cdot |x|^{O(1)}$ for a computable function $f : \mathbb{N} \to \mathbb{N}$.

Or equivalently, we say that the algorithm $\mathbb{A}$ runs in fpt time. A parameterized problem (Q, κ) is *fixed-parameter tractable* if there is an fpt-algorithm with respect to κ that decides Q.

Let (Q, κ) and (Q', κ') be two parameterized problems. An fpt-*reduction* from (Q, κ) to (Q', κ') is a mapping $R : \Sigma^* \to \Sigma^*$ such that:

- For every $x \in \Sigma^*$ we have $x \in Q$ if and only if $R(x) \in Q'$.
- R is computable by an fpt-algorithm.
- There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that $\kappa'(R(x)) \leq g(\kappa(x))$ for all $x \in \Sigma^*$.

It is easy to see that if there is an fpt-reduction from (Q, κ) to (Q', κ'), and if (Q', κ') is fixed-parameter tractable, then so is (Q, κ).

We also need some notions from parameterized counting complexity. As they are only required in Section 6, we will introduce them there.

Graphs. We only consider *simple* graphs, that is, finite nonempty undirected graphs without loops and parallel edges. Every graph $G = (V, E)$ is thus determined by a nonempty vertex set V and an edge set $E \subseteq \binom{V}{2}$. The *size* of G is defined by $|G| := |V| + |E|$. For an edge $\{u, v\} \in E$ we say that u is *adjacent* to v, and vice versa. Often we also use $V(G)$ and $E(G)$ to denote the vertex set and the edge set of G, respectively.

Let $G = (V, E)$ be a graph. For every vertex $v \in V$ the set $N^G(v)$ contains all vertices in G that are adjacent to v, i.e., $N^G(v) := \{u \mid \{u, v\} \in E\}$. Moreover, for every $S \subseteq V$ we let $N^G(S) := \bigcup_{v \in S} N^G(v)$. Note the degree of v, written $\deg^G(v)$, is $|N^G(v)|$. If $\deg^G(v) = 0$, then v is an *isolated* vertex. The distance $d^G(u, v)$ between two vertices $u, v \in V$ is the length of a shortest path from u to v in the graph G. If it is clear from the context, we omit the superscript G in the above notations and write $N(v)$, $\deg(v)$, etc., instead.

Every nonempty subset $S \subseteq V(G)$ induces a subgraph $G[S]$ with the vertex set S and the edge set $E(G[S]) := \binom{S}{2} \cap E(G)$. Consequently, a graph H is an *induced subgraph of* G if $H = G[V(H)]$. Recall that H is a k-*edge* induced subgraph of G for $k := |E(H)|$.

Again, let S be a set of vertices in G. Then S is a *clique*, if for every $u, v \in S$ we have either $u = v$ or $\{u, v\} \in E(G)$. On the other hand, a set S is an *independent set* in G, if $\{u, v\} \notin E(G)$ for all $u, v \in S$. For every $k \in \mathbb{N}$, there exists a constant $\mathscr{R}_k$, known as the *Ramsey number*, such that every graph G with $|V(G)| \geq \mathscr{R}_k$ has either a clique of size k or an independent set of size k. It is well known that $\mathscr{R}_k < 2^{2 \cdot k}$ for every $k \in \mathbb{N}$.

Relational structures and first-order logic. A *vocabulary* τ is a finite set of relation symbols. Each relation symbol has an *arity*. A *structure* $\mathcal{A}$ of vocabulary τ, or simply structure, consists of a nonempty set A called the *universe*, and an interpretation $R^{\mathcal{A}} \subseteq A^r$ of each r-ary relation symbol $R \in \tau$. For example, a graph G can be identified with a structure $\mathcal{A}(G)$ of vocabulary $\tau_{\text{graph}} := \{E\}$ with the binary relation symbol E such that $A(G) := V(G)$ and $E^{\mathcal{A}(G)} := \big\{(u,v) \bigm| \{u,v\} \in E(G)\big\}$.

The *disjoint union* of two τ-structures $\mathcal{A}_1$ and $\mathcal{A}_2$ is again a τ-structure, denoted by $\mathcal{A}_1 \dot\cup \mathcal{A}_2$, whose universe is $A_1 \dot\cup A_2$, and where for each relation symbol $R \in \tau$ we let $R^{\mathcal{A}_1 \dot\cup \mathcal{A}_2} := R^{\mathcal{A}_1} \dot\cup R^{\mathcal{A}_2}$.

Let $\mathcal{A}$ be a structure of a vocabulary τ. Then the *Gaifman graph* of $\mathcal{A}$ is $G(\mathcal{A}) := (V,E)$ with $V := A$ and

$$E := \big\{\{a,b\} \bigm| a,b \in A \text{ with } a \neq b, \text{ and there exists an } R \in \tau \text{ and a tuple } (a_1,\dots,a_r) \in R^{\mathcal{A}} \text{ with } \{a,b\} \subseteq \{a_1,\dots,a_r\}\big\}.$$

Note any unary relation in $\mathcal{A}$ has no influence on E.

Let $r \in \mathbb{N}$ and $a \in A$. Then the *r-neighborhood* of a is $N_r^{\mathcal{A}}(a) := \big\{b \in A \bigm| d^{G(\mathcal{A})}(a,b) \leq r\big\}$. Moreover, the structure $\mathcal{N}_r^{\mathcal{A}}(a)$ induced by the r-neighborhood of a has universe $N_r^{\mathcal{A}}(a)$, and for each r-ary relation symbol $R \in \tau$ the interpretation $\big\{(a_1,\dots,a_r) \in R^{\mathcal{A}} \bigm| a_1,\dots,a_r \in N_r^{\mathcal{A}}(a)\big\}$.

Formulas of first-order logic of vocabulary τ are built up from atomic formulas $x = y$ and $Rx_1 \dots x_r$ where $x, y, x_1, \dots, x_r$ are variables and $R \in \tau$ is of arity r, using the boolean connectives and existential and universal quantification. To give an example, for every $k \in \mathbb{N}^+$ let

$$is_k := \exists x_1 \dots \exists x_k \left(\bigwedge_{1 \leq i < j \leq k} (\neg x_i = x_j \wedge \neg E x_i x_j) \right).$$

Then a graph G has an independent set of size k if and only if $\mathcal{A}(G) \models is_k$.

Tree-width and local tree-width. We assume that the reader is familiar with the notion of *tree-width* $\text{tw}(G)$ of a graph G. Recall that the tree-width $\text{tw}(\mathcal{A})$ of a structure $\mathcal{A}$ is simply $\text{tw}(G(\mathcal{A}))$, that is, the tree-width of the Gaifman graph of $\mathcal{A}$. In fact, to understand most parts of our proofs and algorithms, it is sufficient to know that

(T) for every structure $\mathcal{A}$ we have $\text{tw}(\mathcal{A}) < |A|$.

Now we are ready to define the *local tree-width* of a structure $\mathcal{A}$. For every $r \in \mathbb{N}$ let

$$\text{ltw}(\mathcal{A}, r) := \max\big\{\text{tw}\left(\mathcal{N}_r^{\mathcal{A}}(a)\right) \bigm| a \in A\big\}.$$

Let $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a function and $p \in \mathbb{N}$. We say a structure $\mathcal{A}$ has *local tree-width bounded by g with respect to p* if $\text{ltw}(\mathcal{A}, r) \leq g(r,p)$ for every $r \in \mathbb{N}$. This slightly generalizes the usual notion of local tree-width bounded by a *unary* function [17].

Example 2.1. Let $d \geq 2$. A structure $\mathcal{A}$ has degree bounded by d if every vertex v in its Gaifman graph $G(\mathcal{A})$ satisfies $\deg(v) \leq d$. It is well known that structures of bounded degree might have arbitrarily large tree-width (e.g., grids). Nevertheless, by (T) it is easy to see

$$\text{ltw}(\mathcal{A}, r) < \sum_{i=0}^{r} d^i = \frac{d^{r+1} - 1}{d - 1}.$$

Thus $\mathcal{A}$ has local tree-width bounded by g with respect to d, where $g : (r, d) \mapsto (d^{r+1} - 1)/(d - 1)$.

3. Some easy positive instances

Definition 3.1 (independent set matching structure). Let $k \in \mathbb{N}$ and $G = (V, E)$ be a graph. Moreover, let $u_1, \ldots, u_k, v_1, \ldots, v_k$ be $2 \cdot k$ vertices in G such that:

(IM1) For every $i, j \in [k]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

(IM2) $\{u_1, \ldots, u_k\}$ is an independent set in G.

Then we say that G contains a *k-independent-set-matching structure* on $u_1, \ldots, u_k$, $v_1, \ldots, v_k$.

Lemma 3.2. *Let $k \in \mathbb{N}$. Every graph containing a k-independent-set-matching structure has a k-edge induced subgraph.*

Proof: The case for $k = 0$ is trivially true. So assume $k \geq 1$ and G contains a k-independent-set-matching structure on the vertices $u_1, \ldots, u_k$, $v_1, \ldots, v_k$.

We choose the maximum $k' \leq k$ such that

$$\ell := \left| E\big(G[\{v_1, \ldots, v_{k'}\}]\big) \right| \leq k.$$

If $k' = k$, then $G[V']$ with $V' := \{u_1, \ldots, u_{k-\ell}\} \cup \{v_1, \ldots, v_k\}$ is a k-edge induced subgraph of G.

Otherwise, $k' < k$. In particular, $\left| E\big(G[\{v_1, \ldots, v_{k'}, v_{k'+1}\}]\big) \right| > k$. As $v_{k'+1}$ can contribute at most k' many new edges, we have $\ell + k' > k$, i.e., $k - \ell < k'$. Then $G[V']$ with $V' := \{u_1, \ldots, u_{k-\ell}\} \cup \{v_1, \ldots, v_{k'}\}$ is a k-edge induced subgraph of G. □

Definition 3.3 (clique matching structure). Let $k \in \mathbb{N}$, $G = (V, E)$ be a graph and $u_1, \ldots, u_k$, $v_1, \ldots, v_k$ pairwise distinct vertices in G such that:

(CM1) For every $i, j \in [k]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

(CM2) $\{u_1, \ldots, u_k\}$ is a clique in G.

Then we say that G contains a *k-clique-matching structure* on $u_1, \ldots, u_k$, $v_1, \ldots, v_k$.

Lemma 3.4. *Let $k \in \mathbb{N}$ and G be a graph containing a k-clique-matching structure. Then there is a k-edge induced subgraph in G.*

Proof: The cases for $k \leq 2$ are trivial. So we consider $k \geq 3$. Let k_0 be maximum with $\binom{k_0}{2} \leq k$ and set $r := k - \binom{k_0}{2}$. It is easy to verify that $k \geq k_0 + r$ by $k \geq 3$ and $k_0 > r$. Now assume G contains a k-clique-matching-structure on the vertices $u_1, \ldots, u_k$, $v_1, \ldots, v_k$. Then, we choose the maximum $r' \leq r$ such that

$$\ell := \Big|E\big(G[\{v_1, \ldots, v_{r'}\}]\big)\Big| \leq r.$$

If $r' = r$, then $G[V']$ with $V' := \big\{v_1, \ldots, v_r\big\} \cup \big\{u_1, \ldots, u_{r-\ell}, u_{r+1}, \ldots, u_{k_0+\ell}\big\}$ is a k-edge induced subgraph of G.

Otherwise, $r' < r$ and by the maximality of r' we have $\Big|E\big(G[\{v_1, \ldots, v_{r'}, v_{r'+1}\}]\big)\Big| > r$. As $v_{r'+1}$ can add at most r' many new edges, we have $\ell + r' > r$, or equivalently $r - \ell < r'$. It follows that $G[V']$ with $V' := \big\{v_1, \ldots, v_{r'}\big\} \cup \big\{u_1, \ldots, u_{r-\ell}, u_{r'+1}, \ldots, u_{r'+k_0-r+\ell}\big\}$ has exactly k edges. □

Definition 3.5 (apex structure). Let $k \in \mathbb{N}$, $G = (V, E)$ be a graph, $A, B \subseteq V$, and a vertex $v_0 \in V$ which satisfy the following conditions:

(A1) A, B are disjoint with $|A| \geq k$ and $|B| \geq \mathscr{R}_k$.

(A2) A is a clique in G.

(A3) $\{u, v_0\} \in E$ for every $u \in A$ and $\{v, v_0\} \notin E$ for every $v \in B$. (Note this implies that $v_0 \notin A$ but possibly $v_0 \in B$.)

(A4) $\{u, v\} \in E$ for every $u \in A$ and $v \in B$.

Then we say that G contains a *k-apex structure* on v_0, A and B.

Lemma 3.6. *Let $k \in \mathbb{N}$ and G be a graph. If G contains a k-apex structure, then it has a k-edge induced subgraph.*

Proof: The case for $k \leq 1$ is trivially true. So let $k \geq 2$. Moreover, let v_0, A, B be as stated in Definition 3.5. Since $|B| \geq \mathscr{R}_k$, $G[B]$ contains either a clique of size k or an independent set of size k.

If $G[B]$ contains an independent set $B' \subseteq B$ with $|B'| = k$. Then for every $u \in A$ the induced subgraph $G\big[B' \cup \{u\}\big]$ has exactly k edges by (A4).

Now assume that there is a clique B' in $G[B]$ of size k. Observe by (A3) and $k \geq 2$, we have $v_0 \notin (A \cup B')$. Furthermore, it is easy to see that we can write $k = \binom{k_0}{2} + r$ for some appropriate $k \geq k_0 \geq r$.

We select arbitrary subsets $A' \subseteq A$ and $B'' \subseteq B'$ with $|A'| = r$ and $|B''| = k_0 - r$. Then it is straightforward to check that $G\big[A' \cup B'' \cup \{v_0\}\big]$ has exactly k edges. □

Lemma 3.7 (three cliques). *Let $k \in \mathbb{N}$ and $G = (V, E)$ be a graph. Assume there exists three subsets S_1, S_2, S_3 such that:*

– *S_1, S_2, S_3 are three disjoint cliques in G, all of size k.*

– *There are no edges between any distinct S_i and S_j.*

Then G has a k-edge induced subgraph.

It is easy to see that Lemma 3.7 is a direct consequence of Gauss' famous Eureka Theorem [3].

Theorem 3.8. *For every $k \in \mathbb{N}$ there exist $k_0, k_1, k_2 \in \mathbb{N}$ such that*

$$k = \binom{k_0}{2} + \binom{k_1}{2} + \binom{k_2}{2}.$$

Lemma 3.9 (large independent set). *Let $k \in \mathbb{N}^+$ and $G = (V, E)$ be a graph without isolated vertices. If G contains an independent set of size $(k-1)^2+1$, then it has a k-edge induced subgraph.*

To prove the above lemma, we need some further preparation.

Lemma 3.10. *Let $m, n \in \mathbb{N}^+$ and $G = (V, E)$ be a graph. Furthermore, let $A, B \subseteq V$ be disjoint such that $\big|N(u) \cap B\big| \geq 1$ for every $u \in A$. If $|A| > (m-1)(n-1)$, then*

(i) either there are m vertices $u_1, \ldots, u_m$ in A and a vertex v in B with $\{u_i, v\} \in E$ for every $i \in [m]$,

(ii) or there are n vertices $u_1, \ldots, u_n$ in A and n vertices $v_1, \ldots, v_n$ in B such that for all $i, j \in [n]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

Proof: Let $s := |B|$. We prove by induction on s and n. If $n = 1$, then (ii) is trivially true. And if $s = 1$ and $n > 1$, then clearly (i) holds.

Now assume both $s > 1$ and $n > 1$. If there exists a vertex $v \in B$ with $\big|N(v) \cap A\big| \geq m$, then we can easily achieve (i). So assume now that

$$\text{for every } v \in B \text{ we have } \big|N(v) \cap A\big| \leq m - 1. \tag{1}$$

Choose an arbitrary vertex $v \in B$ and let $B' := B \setminus \{v\}$. If for every $u \in A$ we have $\big|N(u) \cap B'\big| \geq 1$, then the result follows from the induction hypothesis on A and B' with $|B'| = s - 1$. Otherwise, there exists a vertex $u \in A$ such that $N(u) \cap B' = \emptyset$, i.e., $N(u) \cap B = \{v\}$. Let $A' := A \setminus N(v)$. By (1) it holds that $|A'| > (m-1)(n-2)$. Then by induction hypothesis on

$$A \leftarrow A', B \leftarrow B', m \leftarrow m, \text{ and } n \leftarrow n - 1,$$

together with (1), the property (ii) holds for A', B', and $n-1$. That is, there are $n-1$ vertices $u_1, \ldots, u_{n-1}$ in A' and $n-1$ vertices $v_1, \ldots, v_{n-1}$ in B' such that for all $i, j \in [n-1]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$. As $N(u) \cap B' = N(v) \cap A' = \emptyset$, by taking $u_n := u$ and $v_n := v$, we have $\{u_i, v_j\} \in E$ if and only if $i = j$, for every $i, j \in [n]$. □

Proof of Lemma 3.9: Let $S \subseteq V$ be an independent set in G with $|S| > (k-1)^2$. Since G has no isolated vertex, $|N(u) \cap N(S)| \geq 1$ for every $u \in S$. So we can apply Lemma 3.10 on

$$A \leftarrow S, B \leftarrow N(S), m \leftarrow k, \quad \text{and } m \leftarrow k.$$

If (i) holds, then we have an induced k-star of exactly k edges. Otherwise, we have (ii). Hence, there exist vertices $u_1, \ldots, u_k \in S$ and $v_1, \ldots, v_k \in N(S)$ such that G contains k-independent-set-matching structure on those vertices. The result follows from Lemma 3.2. □

Definition 3.11. Let $G = (V, E)$ be a graph and $d \in \mathbb{N}$. We define

$$V^G_{[1,d]} := \{v \in V \mid 1 \leq \deg(v) \leq d\}.$$

Lemma 3.12 (sufficiently many small degree vertices). *Let $d, k \in \mathbb{N}^+$ and $G = (V, E)$ be a graph. If $\left|V^G_{[1,d]}\right| > (d+1) \cdot (k-1)^2$, then G contains a k-edge induced subgraph.*

Proof: Let $G' = (V', E')$ be the graph resulting by removing all isolated vertices from G. Then, by Lemma 3.9 it suffices to show that G' contains an independent set S of size $(k-1)^2 + 1$. In fact, such a set S can be constructed by repeatedly picking vertices from $V_{[1,d]} \subseteq V'$ and removing their neighbors. □

3.1. A further combinatorial lemma. For later purpose, we need a generalization of Lemma 3.10.

Lemma 3.13. *Let $m, n, p \in \mathbb{N}^+$ and $G = (V, E)$ be a graph. Furthermore, let $A, B \subseteq V$ be disjoint such that $\left|N(u) \cap B\right| \geq p$ for every $u \in A$. If $|A| > (m-1)(n-1)^p$, then*

(i) either there are m vertices $u_1, \ldots, u_m$ in A and p vertices $v_1, \ldots, v_p$ in B with $\{u_i, v_j\} \in E$ for every $i \in [m]$ and $j \in [p]$,

(ii) or there are n vertices $u_1, \ldots, u_n$ in A and n vertices $v_1, \ldots, v_n$ in B such that for all $i, j \in [n]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

Proof: We proceed by induction on p. The case $p = 1$ is precisely Lemma 3.10. So let $p > 1$. We apply Lemma 3.10 on

$$m \leftarrow (m-1)(n-1)^{p-1} + 1 \quad \text{and} \quad n \leftarrow n.$$

Thus

(a) either there are $(m-1)(n-1)^{p-1}+1$ vertices $u_1, \dots, u_{(m-1)(n-1)^{p-1}+1}$ in A and a vertex v in B with $\{u_i, v\} \in E$ for every $i \in \big[(m-1)(n-1)^{p-1}+1\big]$,

(b) or there are n vertices $u_1, \dots, u_n$ in A and n vertices $v_1, \dots, v_n$ in B such that for all $i, j \in [n]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

Clearly (b) is exactly (ii). So we assume that (a) holds. Let

$$A' := \big\{u_1, \dots, u_{(m-1)(n-1)^{p-1}+1}\big\}, \quad B' := B \setminus \{v\}, \quad m' := m, \quad n' := n, \quad \text{and} \quad p' := p-1.$$

It is easy to verify that we can apply the induction hypothesis on

$$A \leftarrow A', B \leftarrow B', m \leftarrow m', n \leftarrow n', \text{ and } p \leftarrow p'.$$

If (ii) holds for A', B', and n', then it holds for A, B, n, too. Otherwise, there are m vertices $u'_1, \dots, u'_m$ in $A' \subseteq A$ and $p-1$ vertices $v'_1, \dots, v'_{p-1}$ in $B' \subseteq B$ with $\{u'_i, v'_j\} \in E$ for every $i \in [m]$ and $j \in [p-1]$.

Recall now (a) is true for the vertices in A and the vertex v in B. Therefore, $\{u'_i, v\} \in E$ for every $i \in [m]$. Then (i) holds for $u'_1, \dots, u'_m \in A$, $v'_1, \dots, v'_{p-1}, v \in B$, m, and p by $v \in B \setminus B'$. □

4. Tractable instances by model-checking

In this section we show the fixed-parameter tractability of k-Edge-Induced-Subgraph on some restricted classes of graphs via the model-checking problem for first-order logic.

As mentioned in the Introduction, the following is a generalization of a well-known result due to Frick and Grohe [17].

Theorem 4.1. *For every computable function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ the problem*

p-Mc-Ltw$_g$-FO	
Instance:	A structure $\mathcal{A}$, $p \in \mathbb{N}$ and an FO-sentence φ such that $\mathcal{A}$ has local tree-width bounded by g with respect to p.
Parameter:	$p + \lvert\varphi\rvert$.
Problem:	Decide whether $\mathcal{A} \models \varphi$.

is fixed-parameter tractable.

For the sake of completeness we include a proof in the appendix.

Definition 4.2 (degree-extreme graph). Let $d \in \mathbb{N}$ and $G = (V, E)$ be a graph. If $\deg(v) \le d$ or $\deg(v) \ge |V| - 1 - d$ for every $v \in V$, then the graph G is *d-degree-extreme.*

For example, let $n \in \mathbb{N}$, then an n-star is d-degree-extreme for every $d \ge 1$.

Now we translate every degree-extreme graph to a finite structure over the vocabulary $\tau_{\text{des}} := \{P, R\}$ where P is a unary relation symbol and R a binary relation symbol.

Definition 4.3 (degree-extreme structure). Let $d \in \mathbb{N}$ and $G = (V, E)$ be a d-degree-extreme graph. We set $V^G_{\leq d} := \{v \in V \mid \deg(v) \leq d\}$. Then $\mathcal{A} := \mathcal{A}(G, d)$ is a τ_{des}-structure defined by

$$\begin{aligned} A &:= V, \\ P^{\mathcal{A}} &:= V^G_{\leq d}, \\ R^{\mathcal{A}} &:= \{(u,v) \mid \{u,v\} \in E \text{ and } (u \in V^G_{\leq d} \text{ or } v \in V^G_{\leq d})\} \\ &\quad \cup \{(u,v) \mid \{u,v\} \notin E,\ u,v \in V \setminus V^G_{\leq d} \text{ and } u \neq v\}. \end{aligned}$$

Basically, $\mathcal{A}(G, d)$ has the same vertex set as G, keeps the edges between two small degree vertices and the edges between a small degree vertex and a large degree one, and takes the complement of remaining edges between large degree vertices.

It should be clear that $\mathcal{A}(G, d)$ is polynomial time computable from G and d.

Lemma 4.4. *There is a computable function $h_0 : \mathbb{N} \times \mathbb{N} \times \mathbb{N}^+ \to \mathbb{N}^+$ such that for every $d \in \mathbb{N}$, $k \in \mathbb{N}^+$ and every d-degree-extreme graph G we have*

(i) either $\left|V^G_{[1,d]}\right| > (d+1) \cdot (k-1)^2$, (hence, by Lemma 3.12, G has a k-edge induced subgraph),

(ii) or for the structure $\mathcal{A} := \mathcal{A}(G, d)$ as defined in Definition 4.3 we have $\operatorname{ltw}(\mathcal{A}, r) \leq h_0(r, d, k)$ *for every $r \in \mathbb{N}$.*

Proof: We assume that (i) is not true, i.e., $\left|V^G_{[1,d]}\right| \leq (d+1) \cdot (k-1)^2$. For every $v \in A = V(G)$ it is easy to verify that $\deg^{G(\mathcal{A})}(v) \leq d + (d+1) \cdot (k-1)^2$. Together with the property (T)(see page 6) we conclude

$$\operatorname{tw}\left(\mathcal{N}^{\mathcal{A}}_r(v)\right) < \left|N^{\mathcal{A}}_r(v)\right| \leq \sum_{i=0}^{r} \left(d + (d+1) \cdot (k-1)^2\right)^i.$$

Thus we can define the desired function h_0 accordingly. □

Definition 4.5. Recall the vocabulary of degree-extreme structures is $\tau_{\text{des}} = \{P, R\}$. We let

$$\mathit{edge}(x, y) := \big(Rxy \wedge (Px \vee Py)\big) \vee (\neg Rxy \wedge \neg Px \wedge \neg Py).$$

Moreover, let $H = (V, E)$ be a graph. We assume that $V = [\ell]$ for some $\ell \in \mathbb{N}$. We define

$$\mathit{induced}_H := \exists x_1 \ldots \exists x_\ell \left(\bigwedge_{1 \leq i < j \leq \ell} \neg x_i = x_j \wedge \bigwedge_{\{i,j\} \in E} \mathit{edge}(x_i, x_j) \wedge \bigwedge_{\{i,j\} \in \binom{V}{2} \setminus E} \neg \mathit{edge}(x_i, x_j) \right).$$

Then the following lemma is straightforward.

Lemma 4.6. *Let $d \in \mathbb{N}$ and G be a d-degree-extreme-graph. For every graph H we have*

$$G \text{ contains an induced subgraph isomorphic to } H \quad \Longleftrightarrow \quad \mathcal{A}(G,d) \models \mathit{induced}_H.$$

Proposition 4.7. *Let $D : \mathbb{N} \to \mathbb{N}$ be a computable function. Then the problem*

Instance:	A graph G and $k \in \mathbb{N}$ such that G is $D(k)$-degree-extreme.
Parameter:	k.
Problem:	Decide whether G contains a k-edge induced subgraph.

is fixed-parameter tractable.

Proof: We only consider $k \in \mathbb{N}^+$ and let $G = (V,E)$ be a $D(k)$-degree-extreme graph. Moreover, let $\mathcal{A} := \mathcal{A}(G, D(k))$. By Lemma 4.4 we can assume that

$$\mathrm{ltw}(\mathcal{A}, r) \leq h_0(r, D(k), k).$$

That is, the structure $\mathcal{A}$ has local tree-width bounded by the function $g(r,k) := h_0(r, D(k), k)$ with respect to k.

Then we define the following FO-sentence

$$\mathit{induced}_k := \bigvee_{\substack{H \text{ has no isolated vertex} \\ \text{and } |E(H)| = k}} \mathit{induced}_H.$$

It follows that G has an induced subgraph of exactly k edges if and only if $\mathcal{A} \models \mathit{induced}_k$. Note the structure $\mathcal{A}$ can be computed in fpt time, and the sentence $\mathit{induced}_k$ can be computed from k. Hence, $(G,k) \mapsto \big(\mathcal{A}, k, \mathit{induced}_k\big)$ gives an fpt-reduction to p-MC-LTW$_g$-FO. The result then follows from Theorem 4.1. □

Definition 4.8 (bridge). Let $d, b \in \mathbb{N}$. Moreover let $G = (V,E)$ be a graph such that:

(B1) $V = V_1 \cup V_2$ for some disjoint V_1 and V_2.

(B2) $G[V_1]$ and $G[V_2]$ are both d-degree-extreme.

(B3) There exists a subset $B \subseteq V$ with $|B| = b$ such that for every edge $\{u,v\}$ with $u \in V_1$ and $v \in V_2$ we have either $u \in B$ or $v \in B$.

Then (G, V_1, V_2, B) is a (d,b)-*bridge* (of two degree-extreme graphs).

Similarly to degree-extreme graphs, it is now routine to show:

Proposition 4.9. *Let $D : \mathbb{N} \to \mathbb{N}$ be a computable function. Then the problem*

Instance:	A graph $G = (V, E)$, $V_1, V_2, B \subseteq V$ and $k \in \mathbb{N}$ such that (G, V_1, V_2, B) is a $(D(k), \|B\|)$-bridge.
Parameter:	$k + \|B\|$.
Problem:	Decide whether G contains a k-edge induced subgraph.

is fixed-parameter tractable.

4.1. Combinatorial algorithms for degree-extreme graphs and bridges. We have seen that k-EDGE-INDUCED-SUBGRAPH is fixed-parameter tractable on degree-extreme graphs and bridges using a powerful model-checking algorithm for first-order logic. This logic-based model-checking algorithm is treated as a black-box, and its precise running time is very hard to analyze. In fact, the worse-case running time is astronomical on the parameter, and cannot be improved under widely-accepted complexity assumptions [18].

Of course, as all the involved first-order sentences are given explicitly, we can in principle analyze the algorithm $\mathbb{A}$ in Theorem 4.1 on those specific formulas. There are two reasons for the huge running time of $\mathbb{A}$.

(i) Using Gaifman's Theorem (see Theorem 7.5) $\mathbb{A}$ first translates a given first-order sentence φ to an equivalent Boolean combination ψ of so-called *local sentences*. In general there is no elementary bound on $|\psi|$ in terms of $|\varphi|$ [12]. And even if we restrict ourself to structures of bounded degree, the known reduction produces a ψ with $|\psi|$ being 4-fold exponential in $|\varphi|$; and there is a double exponential lower bound [12]. Recall that the sentence $\mathit{induced}_k$ in the proof of Proposition 4.7 already has size exponential in k.

(ii) To evaluate ψ on a given a structure, $\mathbb{A}$ further translates ψ to tree-automata. The size of these automata grows exponentially with each quantifier alternation in ψ. Although the first-order sentences we need (e.g., $\mathit{induced}_k$) are all existential, none of the known translations for Gaifman's theorem in step (i) preserves the quantifier structure of φ. That is, ψ might not be existential.

Nevertheless, using an *existential locality theorem* due to Grohe and Wöhrle [23], we could have an algorithm with much better running time than the general one in Theorem 4.1. But a careful inspection shows that this could only give a 4-fold exponential fpt algorithm. We will discuss this more technical issue in Remark 4.11.

Therefore, we now present some purely combinatorial algorithms whose running time is routine to analyze. Let us emphasize that, although these algorithms seem to rend the logic machinery useless, without thinking from a logic perspective in the first place, it would have been very difficult, if not impossible, to even come up with those cases that the algorithms are designed to solve.

First, we have a more refined version of Proposition 4.7.

Lemma 4.10. *There is an algorithm which solves the problem*

Instance:	$d \in \mathbb{N}$, a d-degree-extreme graph G, and $k \in \mathbb{N}$.
Problem:	Decide whether G contains a k-edge induced subgraph.

in time $2^{O(k^3 + k \cdot \log d)} \cdot |G|$.

Remark 4.11. Let us directly analyze the running time of an algorithm for Proposition 4.7 with $d := D(k)$. Thereby, we construct a structure $\mathcal{A} = \mathcal{A}(G, d)$ with

$$\mathrm{ltw}(\mathcal{A}, r) \leq \lambda(r) = (d \cdot k)^{O(r)},$$

where the estimation $(d \cdot k)^{O(r)}$ is due to the function h_0 defined in the proof of Lemma 4.4. Moreover, we observe that the existential sentence $\varphi = \mathit{induced}_k$ has length

$$|\varphi| = 2^{O(k^2)}.$$

Thus the best algorithm [23, Theorem 9] for checking whether $\mathcal{A} \models \varphi$ (which we are aware of) has running time bounded by

$$2^{2^{\left(\lambda\left(|\varphi|^{O(1)}\right)^{O(1)}\right)}} \cdot |G|^{O(1)} = 2^{2^{\left((d \cdot k)^{2^{O(k^2)}}\right)}} \cdot |G|^{O(1)},$$

This is clearly far worse than the bound stated in Lemma 4.10. In our final fpt-algorithm for k-Edge-Induced-Subgraph we will solve a subproblem on d-degree-extreme graph with

$$d = 2^{2^{O(k)}}.$$

Hence, using Proposition 4.7 instead of Lemma 4.10 will result in running time

$$2^{2^{2^{2^{O(k^2)}}}} |G|^{O(1)}.$$

Before giving the formal proof, let us briefly explain the underlying strategy of our algorithm.

The intuition of the algorithm. Clearly the input graph G can be assumed to have no isolated vertices. First, by Lemma 3.12, we can handle the case where G has sufficient many small degree vertices. On the other hand, when G has many vertices with large degree, we can safely assume that its complement $\bar{G}$ has a k-edge induced matching M_k, and there are plenty isolated vertices in $\bar{G}$ outside of M_k. In such a case, we can obtain induced subgraphs whose edges number can be

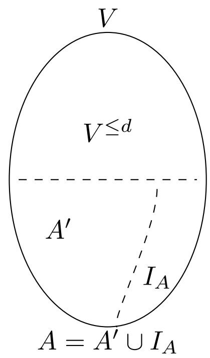

Figure 1.

written as $\binom{p}{2} - q$, where $p, q \in \mathbb{N}$, $p \leq k$, and $q \leq p/2$. If k can be expressed in such a form, we are done. Otherwise, suppose $k = \binom{p}{2} - q$ and $q > p/2$. We need to check whether there is a graph C' in $\bar{G}$ with p' vertices and q' edges, where $q - q' = \Delta$ and $p - p' \geq 2\Delta$. If so, we can return C' with a Δ-induced matching and $(p - p' - 2\Delta)$ isolated vertices from $\bar{G}$. Fortunately, we can show that if $\bar{G}$ contains a connected component with more than k edges, then it also contains such a graph C'. On the other hand, if every connected component of $\bar{G}$ has small edges number, then we can conduct brute-force search. More details can be found in the proof below.

In the following we will tacitly use the fact that every k-edge induced subgraph *without isolated vertices* has at most $2k$ vertices.

Proof of Lemma 4.10: Without loss of generality, let $G = (V, E)$ contain no isolated vertices. Therefore, for $V^{\leq d} := \{v \in V \mid \deg(v) \leq d\}$, if $|V^{\leq d}| > (d+1)(k-1)^2$, then G has an induced subgraph with exactly k edges by Lemma 3.12. Hence we can assume

$$|V^{\leq d}| \leq (d+1)(k-1)^2.$$

Next, since G is d-degree-extreme, every vertex in $A := V \setminus V^{\leq d}$ has degree at least $|V| - 1 - d$. Similarly to the degree-extreme structures in Definition 4.3, we consider the *complement* graph $\bar{G}_A$ of $G[A]$ defined by

$$V(\bar{G}_A) := A,$$
$$E(\bar{G}_A) := \binom{A}{2} \setminus E(G[A]) = \big\{\{u, v\} \mid u, v \in A \text{ with } u \neq v \text{ and } \{u, v\} \notin E(G(A))\big\}.$$

Furthermore, as shown in Figure 1, we partition A into two parts $A' \dot\cup I_A$, where

$$I_A := \{v \in A \mid v \text{ is an isolated vertex in } \bar{G}_A\} \quad \text{and} \quad A' := A \setminus I_A.$$

Equivalently, I_A contains all vertices in $G[A]$ that are adjacent to every other vertex in $G[A]$.

We distinguish the following cases.

Case 1. We first assume $|A'| \leq (6d^2+2)k$, and then the number of subsets

$$U \subseteq V^{\leq d} \cup A'$$

of size at most $2k$ is bounded by

$$\Big((d+1)\cdot(k-1)^2 + (6d^2+2)k\Big)^{2k} = (d\cdot k)^{O(k)}.$$

Fix such a U, and the set I_A can be partitioned into $I_0, \ldots, I_{|U|}$, where

$$I_i := \big\{v \in I_A \;\big|\; v \text{ has exactly } i \text{ neighbors in } U \text{ in the graph } G\big\}.$$

The reason for defining those I_i's is the following simple fact. Let $I \subseteq I_A$ and $v, v' \in I_i$ for some $0 \leq i \leq |U|$ with $v \in I$ and $v' \notin I$. Then

$$\Big|E\big(G[U \cup I]\big)\Big| = \Big|E\big(G[U \cup I \cup \{v'\} \setminus \{v\}]\big)\Big|.$$

Thus, G has a k-edge induced subgraph H with

$$V(H) \cap \Big(V^{\leq d} \cup A'\Big) = U$$

if and only if there are j_i's for all $0 \leq i \leq k+1$ and $0 \leq j_i \leq \min\Big\{\big|I_i\big|, k+1\Big\}$ with

$$\Big|E\big(G[U]\big)\Big| + \sum_{i=0}^{k+1} i \cdot j_i + \binom{\sum_{i=0}^{k+1} j_i}{2} = k. \tag{2}$$

Thereby, we enumerate all possible U's and j_i's to check whether (2) holds, whose running time is bounded by

$$O\Big((d\cdot k)^{O(k)} \cdot (k+2)^{k+2} \cdot |G|\Big) = (d\cdot k)^{O(k)} \cdot |G|.$$

Case 2. Every connected component of $\bar{G}_A$ contains less than k edges.

Let $\mathbf{C}_k$ be the set of all connected graphs C with $V(C) = \big[|V(C)|\big]$ and $|E(C)| < k$. Observe that $\big|\mathbf{C}_k\big| \leq 2^{k^2}$, since every $C \in \mathbf{C}_k$ is a subgraph of $\Big([k], \binom{[k]}{2}\Big)$ (i.e., a k-clique).

We fix an ordering on A. Let $H \subseteq \bar{G}_A$ be a connected *induced* subgraph, thus by assumption $|E(H)| < k$. The ordering on A induces a unique enumeration of $V(H) = \big\{v_1, \ldots, v_\ell\big\}$ with $\ell := |V(H)| \leq k$, which then determines a unique $C_H \in \mathbf{C}_k$ such that

$$\big(v_i \mapsto i\big)_{i \in [\ell]}$$

is an isomorphism from H to C_H.

In order to find a k-edge induced subgraph in G, we first guess a subset X of $V^{\leq d}$ with at most $2k$ vertices. There are $|V^{\leq d}|^{O(k)} \leq (dk)^{O(k)}$ possibilities. For each subset X, we remove all the vertices $V^{\leq d} \setminus X$ from G. Furthermore, we define

$$P_H := \Big\{ \big(i, N^X(v_i)\big) \;\Big|\; i \in [\ell] \Big\},$$

where $N^X(v_i)$ is the set of vertices in X that are adjacent to v_i in the original graph G. Then, the *type* of H (with respect to X) is

$$\big(C_H, P_H\big).$$

It is routine to verify that if two induced subgraphs H_1 and H_2 of $\bar{G}_A$ have the same type, then for any $U \subseteq X$, the induced subgraphs $G\big[U \dot\cup V(H_1)\big]$ and $G\big[U \dot\cup V(H_2)\big]$ are isomorphic and hence have the same number of edges.

By $\big|\mathbf{C}_k\big| \leq 2^{k^2}$, $\ell \leq k$, and $\big|X\big| \leq 2k$, there are at most

$$2^{k^2} \cdot (2^{2k})^k = 2^{O(k^2)} \tag{3}$$

types of connected induced subgraphs in $\bar{G}[A]$. In particular, (3) also bounds the number of different types of connected components in $\bar{G}_A$. Then for each possible type, if there are more than $2k$ many connected components of this type, then we only keep $2k$ of them in G. We use G_X to denote the resulting graph. By the discussion above, G_X has a k-edge induced subgraph if and only if the original graph G has a k-edge induced subgraph H with $V(H) \cap V^{\leq d} = X$. We now observe that

$$\big|V(G_X)\big| \leq 2k + 2^{O(k^2)} \cdot 2k \cdot k = 2^{O(k^2)}.$$

Thus, to find a k-edge induced subgraph of G, we enumerate all X with at most $2k$ vertices from $V^{\leq d}$. For each X, we compute a graph G_X from G, we enumerate all subsets $V' \subseteq V(G_X)$ of size at most $2k$, and check if

$$\big|E(G_X[V'])\big| = \big|E(G[V'])\big| = k.$$

This can be done in time

$$O\Big((dk)^{O(k)} \cdot |V(G_X)|^{O(k)} \cdot |G|\Big) = 2^{O(k^3 + k \cdot \log d)} \cdot |G|.$$

Case 3. We have both $\big|A'\big| > (6d^2 + 2)k$ and that $\bar{G}_A$ contains a connected component with at least k edges. We claim that $G[A]$ has a k-edge induced subgraph.

Note that k can be written as

$$k = \binom{p}{2} - q \tag{4}$$

for some $p, q \in \mathbb{N}$ with $p < \sqrt{2k} + 1$ and $0 \leq q < p - 1$. Our goal is to find an induced subgraph H of $\bar{G}_A$ with p vertices and q edges. This suffices as the complement of H has k edges by (4).

Let C be a connected component of $\bar{G}_A$ with $\geq k$ edges. Choose a vertex $v \in V(C)$ of minimum degree such that $C \setminus \{v\}$ is still connected, and then remove v from C. We repeat this procedure until the number of edges is $\leq q$. Suppose we obtain a connected induced subgraph C' of $\bar{G}_A$ with p' vertices and q' edges for some $q' \leq q$. Let

$$\Delta := q - q'. \tag{5}$$

Assume without loss of generality $k \geq 2$, thus $\Delta \leq k$.

Claim 1. If $q' = 0$, then $p' = 1$, $q = 0$, and $\Delta = 0$.

Proof of the claim. $q' = 0$ means that C' has one single vertex, i.e., $p' = 1$. This further implies $q = 0$, as otherwise we would have kept the last two vertices together with the edge between them. $\dashv$

Claim 2. Assume $q' > 0$. Then there is a vertex $v \in V(C')$ so that $C' \setminus \{v\}$ is still connected, and the degree of v in C' is at least Δ.

Proof of the claim. This is trivial for $\Delta = 0$. So assume $\Delta \geq 1$. Let v^* be the last vertex being deleted from C, and C^* the graph before its deletion. In particular $C' = C^* \setminus \{v^*\}$. Thus v^* has degree $> \Delta$ in C^*, otherwise we would have not deleted it by (5). Note this implies C' has more than $\Delta \geq 1$ vertices. Choose an arbitrary vertex v in C' such that $C' \setminus \{v\}$ is still connected. v has degree at least Δ in C'. Otherwise, the degree of v in C^* is at most Δ. On the other hand, v^* has degree $> \Delta \geq 1$ in C^*, so

$$C^* \setminus \{v\}$$

is connected, and we should have deleted v instead of v^*. $\dashv$

Since C' has $q' < \sqrt{2k}$ edges and is connected, we conclude

$$|V(C')| \leq k.$$

Then, let

$$N[C'] := V(C') \cup \big\{u \in A \bigm| \{u, v\} \in E(\bar{G}_A) \text{ for some } v \in C'\big\}\Big(\subseteq V(C)\Big).$$

$N[C']$ has at most $(d+1)k$ vertices, since $\bar{G}_A$ has degree bounded by d. We consider the graph

$$H := \bar{G}_A \setminus N[C']$$

Recall that the set A' of non-isolated vertices in $\bar{G}_A$ has size $> (6d^2 + 2)k$. It follows that $\bar{G}_A$ contains more than

$$\frac{(6d^2 + 2)k}{2} = (3d^2 + 1)k$$

edges, and in turn H has more than

$$(3d^2+1)k - d\cdot(d+1)k \geq (2d^2-2d+1)\Delta$$

edges.

Claim 3. Every graph H with vertex degree at most d and $(2d^2-2d+1)\Delta$ many edges contains a set

$$V_M := \big\{u_1, \cdots, u_\Delta, v_1, \cdots, v_\Delta\big\}, \tag{6}$$

such that

$$E(H) \cap \binom{V_M}{2} = \big\{\{u_i, v_i\} \bigm| i \in [\Delta]\big\},$$

i.e., $H[V_M]$ is an induced matching of size Δ.

Proof of the claim. We prove by induction on Δ. It is clear for $\Delta = 1$. Otherwise, we choose an arbitrary edge $e = \{u_\Delta, v_\Delta\}$ and let

$$E_e := \Big\{\{u,v\} \Bigm| \big\{u, u_\Delta\big\} \in E(H) \text{ or } \big\{u, v_\Delta\big\} \in H\Big\}.$$

Thus, every edge in $E(H)\setminus E_e$ is *independent*[3] to $\{u_\Delta, v_\Delta\}$. It is also easy to see

$$|E_e| \leq 1 + 2(d-1) + 2(d-1)^2 = 2d^2 - 2d + 1.$$

Then we apply the induction hypothesis on the graph $\big(V(H), E(H)\setminus E_e\big)$ and $\Delta - 1$. ⊣

Let V_M be as in the above claim. Since $V_M \cap [N(C')] = \emptyset$, there is no edge between V_M and C' in $\bar{G}_A$. So $\bar{G}_A\big[V(C') \cup V_M\big]$ has $p' + 2\Delta$ vertices and $q' + \Delta = q$ edges.

Of course, if $p = p' + 2\Delta$, then we are done. Assume $p > p' + 2\Delta$. Thus $p < \sqrt{2k} + 1$ and $k \geq 2$ implies

$$\big|V(C') \cap V_M\big| \leq k. \tag{7}$$

Let

$$N\big[V(C') \cap V_M\big] = \big(V(C') \cap V_M\big) \cap \big\{u \in A \bigm| \{u,v\} \in E(\bar{G}_A) \text{ for some } v \in \big(V(C') \cap V_M\big)\big\}.$$

Observe that by (7)

$$\Big|N\big[V(C') \cap V_M\big]\Big| \leq (d+1)k,$$

which, together with $|A'| > (6d^2+2)k$, implies that

$$\bar{G}_A \setminus N\big[V(C') \cap V_M\big]$$

[3] Two edges are independent if none of their vertices are adjacent.

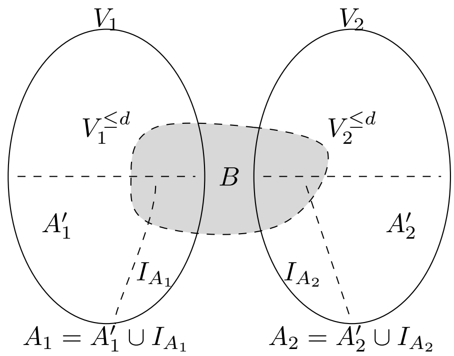

Figure 2.

has more than $d \cdot k$ vertices. Again using the fact that each vertex in $\bar{G}_A$ has degree at most d, we can choose an independent set I in $\bar{G}_A \setminus N\big[V(C') \cap V_M\big]$ with

$$|I| = p - p' - 2\Delta \leq k.$$

Then $\bar{G}_A[C' \cup V_M \cup I]$ has p vertices and q edges, as desired.

As the last step, we prove that it is always the case $p \geq p' + 2\Delta$. Recall that $p - 2 \geq q$ and $q = q' + \Delta$. So it is sufficient to show

$$q' + 2 \geq p' + \Delta. \tag{8}$$

If $q' = 0$, then Claim 1 immediately applies. Otherwise, $q' > 0$. By Claim 2, we choose a vertex v in C' such that $C' \setminus \{v\}$ is still connected, and v has degree $\geq \Delta$ in C'. So the number q' of edges in C' is at least $\Delta + p' - 2$, which is exactly what we want for (8). □

Lemma 4.12. *The problem*

Instance:	Given $d, k \in \mathbb{N}$ and $G = (V_1, V_2, B)$ be a $(d, \|B\|)$-bridge.
Problem:	Decides whether G contains an induced subgraph of exactly k edges.

can be solved in time $2^{O(k^3)}(d^2k^2 + |B|)^{O(k)}|G|$.

Proof: Our algorithm follows the same strategy as that of Lemma 4.10. Firstly, $G[V_1]$ and $G[V_2]$ are decomposed into several pieces, as shown in Figure 2. Let $V_1^{\leq d}$ be the set of vertices with degree $\leq d$ in $G[V_1]$, $A_1 := V_1 \setminus V_1^{\leq d}$, and $\bar{G}_{A_1}$ the complement of $G[A_1]$. Moreover, I_{A_1} is the set of isolated vertices in $\bar{G}_{A_1}$ and $A'_1 := A_1 \setminus I_{A_1}$. We define $V_2^{\leq d}$, A_2, $\bar{G}_{A_2}$, I_{A_2}, and A'_2 similarly with respect to $G[V_2]$.

Without loss of generality, we can again assume that G has no isolated vertices. Thus the numbers of vertices in $V_1^{\le d}$ and $V_2^{\le d}$ are both at most $(d+1)(k-1)^2$, otherwise we already know that G has a k-edge induced subgraph by Lemma 3.12.

As argued in Case 3 in the proof of Lemma 4.10, if for some $i \in [2]$ we have $|A_i'| > (6d^2+2)k$ and $\bar{G}_{A_i}$ contains a connected component with $\ge k$ edges, then $G[V_i]$ has a k-edge induced subgraph, and so does G.

So we are left with the the following cases.

Case 1. $|A_1'| \le (6d^2+2)k$ and $|A_2'| \le (6d^2+2)k$. We enumerate all subsets U of

$$V_1^{\le d} \cup A_1' \cup V_2^{\le d} \cup A_2' \cup B$$

of size bounded by $2k$. For each U we partition $I_{A_1} \setminus B$ into $I_0^1, \cdots, I_{|U|}^1$ in such a way that each vertex in $I_i^1 \subseteq \big(I_{A_1} \setminus B\big)$ has exactly i neighbors in U. Similarly, $I_{A_2} \setminus B$ is partitioned into $I_0^2, \cdots, I_{|U|}^2$, where each vertex in $V_i^2 \subseteq \big(I_{A_2} \setminus B\big)$ has exactly i neighbors in U.

Then, we check whether there are

$$\big(j_{t,i}\big)_{t\in[2], 0\le i\le k+1}$$

with $0 \le j_{t,i} \le \max\big\{|I_j^t|, k+1\big\}$ such that

$$\big|E(G[U])\big| + \sum_{i=0}^{k+1} i \cdot j_{1,i} + \binom{\sum_{i=0}^{k+1} j_{1,i}}{2} + \sum_{i=0}^{k+1} i \cdot j_{2,i} + \binom{\sum_{i=0}^{k+1} j_{2,i}}{2} = k.$$

The left hand side of the above equation is the number of the edges of an induced subgraph where for each t and i we choose an (arbitrary) vertex subset of I_i^t of size $j_{t,i}$. Observe that there are at most

$$\Big(2(6d^2+2)k + 2(d+1)(k-1)^2 + |B|\Big)^{2k} \cdot (k+2)^{2(k+2)} = \big(d \cdot k + |B|\big)^{O(k)}$$

possibilities.

Case 2. $|A_1'| \le (6d^2+2)k$, and each connected component in $\bar{G}_{A_2}$ has at most k edges.

To find a k-edge induced subgraph of G, we first enumerate all subsets X of

$$V_1^{\le d} \cup A_1' \cup V_2^{\le d} \cup B$$

with $|X| \le 2k$ and remove all the vertices of $(V_1^{\le d} \cup A_1' \cup V_2^{\le d} \cup B) \setminus X$ from G. Note that

$$\Big|V_1^{\le d} \cup A_1' \cup V_2^{\le d} \cup B\Big| \le 2(d+1)(k-1)^2 + (6d^2+2)k + |B| = O(d^2 \cdot k^2 + |B|).$$

The choices of X is upper bounded by

$$(d^2 \cdot k^2 + |B|)^{O(k)}.$$

Let H be a connected induced subgraph of $\bar{G}_{A_2} \setminus B$. Then, the type (C_H, P_H) of H (with respect to X) is determined by:

– Assume that $V(H) = \{v_1, \ldots, v_\ell\}$. Then $C_H \in \mathbf{C}_k$ is the graph with vertex set $[\ell]$ that is isomorphic to H. Recall that $\mathbf{C}_k$ is the set of all connected graphs C with $V(C) = \big[|V(C)|\big]$ and $|E(C)| < k$.

– Let $P_H := \Big\{ \big(i, N^X(v_i)\big) \;\Big|\; i \in [\ell] \Big\}$, where $N^X(v_i)$ is the set of vertices in X which are adjacent to v_i in the original graph G.

Again, for two connected induced subgraphs $H_1, H_2 \subseteq \bar{G}_{A_2} \setminus B$ of the same type, and for any $U \subseteq X$, $G\big[U \cup V(H_1)\big]$ and $G\big[U \cup V(H_2)\big]$ have the same number of edges. Observe that there are at most

$$2^{k^2} \cdot (2^{2k})^k = 2^{O(k^2)}$$

types of connected induced subgraphs in $\bar{G}_{A_2} \setminus B$ with respect to X.

Then for every possible type (C_H, P_H) we make sure that $G[A_2] \setminus B$ contains at most $2k$ connected components with this type by removing those additional ones. Thus we can assume that the resulting graph (which we still denote by $G[A_2]$) has at most

$$2^{O(k^2)} \cdot 2k \cdot k = 2^{O(k^2)}$$

vertices.

To find a k-edge induced subgraph of G, we then enumerate all subsets X' of A_2 with $|X' \cup X| \leq 2k$ and let $U = X \cup X'$. For each fixed U, the set $I_{A_1} \setminus B$ can be partitioned into $I_0, \cdots, I_{|U|}$ such that each vertex in I_i has i neighbors in U in the original graph G. Then, G has a k-edge induced subgraph H with

$$V(H) \cap \Big(V_1^{\leq d} \cup A_1' \cup V_2^{\leq d} \cup A_2 \cup B\Big) = U$$

if and only if there are j_i's for all $0 \leq i \leq k+1$ and $0 \leq j_i \leq \min\Big\{\big|I_i\big|, k+1\Big\}$ such that

$$\Big|E\big(G[U]\big)\Big| + \sum_{i=0}^{k+1} i \cdot j_i + \binom{\sum_{i=0}^{k+1} j_i}{2} = k.$$

holds. The number of possible U's and j_i's is bounded by

$$(d^2 \cdot k^2 + |B|)^{O(k)} \cdot 2^{O(k^3)} \cdot (k+2)^{2(k+2)} = 2^{O(k^3)}(d^2k^2 + |B|)^{O(k)}.$$

Case 3. $|A_2'| \leq (6d^2+2)k$ and each connected component in $\bar{G}_{A_1}$ has at most k edges.

This is symmetric to Case 2.

Case 4. $|A_1'| > (6d^2+2)k$, $|A_2'| > (6d^2+2)k$, and each connected component in $\bar{G}_{A_1}$ or in $\bar{G}_{A_2}$ has at most k edges.

As in Case 2, we first guess a subset X of $V_1^{\leq d} \cup V_2^{\leq d} \cup B$ with $|X| \leq 2k$. Then we define for every connected induced subgraph of $\bar{G}_{A_1}$ and of $\bar{G}_{A_2}$ its type with respect to X. Then we remove those redundant connected components to ensure that there are at most $2k$ connected components for each type. It can be verified as in Case 2 the resulting graph, which we still denote by G, contains at most

$$2k + 2 \cdot 2k \cdot k \cdot 2^{O(k^2)} = 2^{O(k^2)}$$

vertices. Then, by brute-force we enumerate all vertex subset of size at most $2k$ in G, and check whether it induces a k-edge induced subgraph. The running time can be bounded by

$$2^{O(k^3)}(d^2k^2 + |B|)^{O(k)}|G|. \qquad \square$$

5. The algorithm

The main component of our fpt-algorithm for k-Edge-Induced-Subgraph is the following procedure that either already solves the problem or decomposes the given graph into potentially a bridge of two large degree-extreme graphs (cf. Definition 4.8).

For every $k \in \mathbb{N}$ we let

$$p_k := 2^{2 \cdot k}(> \mathscr{R}_k).$$

Lemma 5.1. *For every computable function $D : \mathbb{N} \to \mathbb{N}$ there is an* fpt-*algorithm $\mathbb{A}_D$ such that for every graph $G = (V, E)$ and every $k \in \mathbb{N}$ exactly one of following conditions is satisfied.*

(S1) G is $D(k)$-degree-extreme and $\mathbb{A}_D$ correctly decides whether G contains a k-edge induced subgraph.

(S2) G is not $D(k)$-degree-extreme and $\mathbb{A}_D$ correctly outputs that G contains a k-edge induced subgraph.

(S3) G is not $D(k)$-degree-extreme and $\mathbb{A}_D$ outputs three subsets $V_1, V_2, B \subseteq V$ such that

(S3.1) $V = V_1 \dot\cup V_2$ with $|V_1| > D(k)$ and $|V_2| > D(k) + 1$;

(S3.2) every edge between V_1 and V_2 in G has one vertex in B and $|B| \le (p_k - 1)^{p_k+1} + (p_k - 1)^2$.

The running time of $\mathbb{A}_D$ can be bounded by

$$2^{O(k^3 + k \cdot \log D(k))} \cdot |G|.$$

Proof: Let $G = (V, E)$ be a graph and $k \in \mathbb{N}$. If G is $D(k)$-degree-extreme, then we apply Proposition 4.7 to achieve (S1). Otherwise let $v_0 \in V$ be a vertex with

$$D(k) < \deg(v_0) < |V| - 1 - D(k). \qquad (9)$$

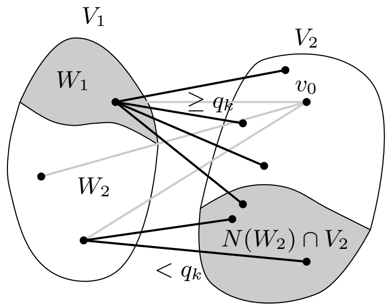

Figure 3.

Then we set $V_1 := N(v_0)$ and $V_2 := V \setminus V_1$. By (9) it holds that $|V_1| > D(k)$ and $|V_2| = |V| - |V_1| = |V| - \deg(v_0) > D(k) + 1$, i.e., (S3.1). Let

$$W_1 := \Big\{ u \in V_1 \ \Big| \ \big|N(u) \cap V_2\big| \geq p_k \Big\} \quad \text{and} \quad W_2 := V_1 \setminus W_1.$$

Figure 3 illustrates our construction.

Claim 1. If $|W_1| > (p_k - 1)^{p_k+1}$, then G contains a k-edge induced subgraph.

Proof of the claim. We apply Lemma 3.13 on

$$A \leftarrow W_1, B \leftarrow V_2, m \leftarrow p_k, n \leftarrow p_k, \text{ and } p \leftarrow p_k.$$

So there are p_k vertices $u_1, \ldots, u_{q_k}$ in W_1 and p_k vertices $v_1, \ldots, v_{p_k}$ in V_2 such that

(i) either $\{u_i, v_j\} \in E$ for every $i, j \in [p_k]$,

(ii) or for all $i, j \in [p_k]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

Recall $p_k > \mathscr{R}_k$, so there is a subset $S \subseteq \{u_1, \ldots, u_{p_k}\}$ such that S is either an independent set or a clique. If S is an independent set, then $G\big[S \cup \{v_0\}\big]$ has exactly k edges. So suppose S is a clique.

Assume that (i) is true, then G contains a k-apex structure on $v_0, S, \{v_1, \ldots, v_{p_k}\}$. Hence, Lemma 3.6 implies the claim. Otherwise (ii) holds. And say $S = \{u_{i_1}, \ldots, u_{i_k}\}$. Then the graph G contains an k-clique-matching structure on $u_{i_1}, \ldots, u_{i_k}, v_1, \ldots, v_k$. The result follows from Lemma 3.4. $\dashv$

Claim 2. If $\big|N(W_2) \cap V_2\big| > (p_k - 1)^2$, then G contains a k-edge induced subgraph.

Proof of the claim. It is easy to verify that we can apply Lemma 3.10 on

$$A \leftarrow N(W_2) \cap V_2, B \leftarrow W_2, m \leftarrow p_k, \text{ and } n \leftarrow p_k.$$

So,

(i) either there are p_k vertices $u_1, \ldots, u_{q_k}$ in $N(W_2) \cap V_2$ and a vertex v in W_2 such that $\{u_i, v\} \in E$ for every $i \in [p_k]$,

(ii) or there are p_k vertices $u_1, \ldots, u_{p_k}$ in $N(W_2) \cap V_2$ and p_k vertices $v_1, \ldots, v_{p_k}$ in W_2 such that for all $i, j \in [p_k]$ we have $\{u_i, v_j\} \in E$ if and only if $i = j$.

But (i) contradicts our definition of W_2, i.e., for every $u \in W_2$ we have $\big|N(u) \cap V_2\big| < p_k$, therefore (ii) must hold. Recall $p_k > \mathscr{R}_k$, hence $G\big[\{v_1, \ldots, v_{p_k}\}\big]$ contains either a clique of size of k or an independent set of size k. Without loss of generality, let $\{v_1, \ldots, v_k\} \subseteq W_2 \subseteq V_1$ be a clique or an independent set.

For the independent set case, as $v_0 \notin V_1$, then $G\big[\{v_0, v_1, \ldots, v_k\}\big]$ is a k-induced subgraph. For the clique case, G contains a k-clique-matching structure on $u_1, \ldots, u_k, v_1, \ldots, v_k$. We are done by Lemma 3.4. ⊣

Let

$$B := W_1 \cup \big(N(W_2) \cap V_2\big),$$

i.e., the grey area in Figure 3. If $|B| > (p_k - 1)^{p_k+1} + (p_k - 1)^2$, then, by Claim 1 and Claim 2, the graph G contains a k-edge induced subgraph, and (S2) follows. Otherwise

$$|B| \leq (p_k - 1)^{p_k+1} + (p_k - 1)^2.$$

Observe that every edge between V_1 and V_2 has at least one vertex in B. Thus, we achieve (S3) by outputting (V_1, V_2, B).

Observe that in the case of (S2) or (S3) the running time of $\mathbb{A}$ can be easily bounded by $O(|G|)$, since it checks whether some simple size constraints are satisfied. In the case of (S1) we can invoke Lemma 4.10 instead of Proposition 4.7, which gives us the claimed upper bound. □

Finally we are ready to present our fpt-algorithm for k-Edge-Induced-Subgraph.

Theorem 5.2. k-Edge-Induced-Subgraph *is fixed-parameter tractable. More precisely, there is an algorithm deciding whether a given graph G contains a k-edge induced subgraph in time*

$$2^{2^{O(k)}} \cdot |G|.$$

Proof: We define a computable function $D_0 : \mathbb{N} \to \mathbb{N}$ by

$$D_0(k) := 2 \cdot \big((p_k - 1)^{p_k+1} + (p_k - 1)^2\big) + 2^{2\cdot((k-1)^2+1)} \leq 2^{2^{O(k)}}. \tag{10}$$

Note $2^{2\cdot((k-1)^2+1)} > \mathscr{R}_{(k-1)^2+1}$. Then let $\mathbb{A}_{D_0}$ be the algorithm as stated in Lemma 5.1 for the function D_0.

Let (G, k) with $G = (V, E)$ be an instance of k-Edge-Induced-Subgraph. First, we remove all the isolated vertices in G. For simplicity, the resulting graph is denoted by G again. Then, we run the algorithm $\mathbb{A}_{D_0}$ on (G, k), which takes time

$$2^{O(k^3+k\log D_0(k))} \cdot |G| \leq 2^{2^{O(k)}} \cdot |G|.$$

If the result is either (S1) or (S2) in Lemma 5.1, we already get the correct answer. Otherwise, $\mathbb{A}_{D_0}$ outputs three subsets $V_1, V_2, B \subseteq V$ satisfying (S3.1) and (S3.2).

If $G[V_1]$ and $G[V_2]$ are both $D_0(k)$-degree-extreme, then (G, V_1, V_2, B) is a $(D_0(k), |B|)$-bridge with $|B|$ bounded by

$$(p_k - 1)^{p_k+1} + (p_k - 1)^2 \leq 2^{2^{O(k)}}.$$

The fixed-parameter tractability of whether G contains a k-edge induced subgraph follows from Proposition 4.9. And in fact, this can be done in time

$$2^{O(k^3)}(D_0(k)^2k^2 + |B|)^{O(k)}|G| \leq 2^{2^{O(k)}} \cdot |G|$$

by Lemma 4.12. Otherwise, either $G[V_1]$ or $G[V_2]$ is not $D_0(k)$-degree-extreme.

We assume that $G[V_1]$ is not $D_0(k)$-degree-extreme. (The case for $G[V_2]$ is symmetric.) Then we run the algorithm $\mathbb{A}_{D_0}$ on $(G[V_1], k)$ in time $2^{2^{O(k)}} \cdot |G|$. Observe that the result cannot be (S1). If the output is (S2), since $G[V_1]$ is an induced subgraph of G, we conclude that G has an induced subgraph of exactly k edges.

Now we are left with case (S3). In particular, there are subsets $V_{11}, V_{12}, B_1 \subseteq V_1$ such that the corresponding properties of (S3.1) and (S3.2) are satisfied. Let

$$U_1 := V_{11} \setminus (B \cup B_1), \quad U_2 := V_{12} \setminus (B \cup B_1), \quad \text{and} \;\; U_3 := V_2 \setminus (B \cup B_1).$$

Observe that in G if we remove the vertex set B, then there is no edge left between V_1 and V_2. Similarly, if we remove the vertex set B_1, every edge between V_{11} and V_{12} is destroyed. Thus, by (S3.2), in the original graph G, there is no edge between each pair of U_1, U_2 and U_3. Moreover by (S3.1) and (S3.2) for every $i \in [3]$

$$|U_i| > D_0(k) - 2 \cdot \left((p_k - 1)^{p_k+1} + (p_k - 1)^2\right) = 2^{2\cdot((k-1)^2+1)} > \mathscr{R}_{(k-1)^2+1},$$

where the equality is by (10).

We use Ramsey's Theorem again. If there is an independent set of size $(k-1)^2 + 1$ in one of the U_1, U_2 and U_3, as G has no isolated vertex, then G contains a k-edge induced subgraph by Lemma 3.9. Otherwise every U_i contains a clique of size $(k-1)^2 + 1 \geq k$. As we have seen that there is no edge between U_1, U_2 and U_3 in G, Lemma 3.7 implies that G contains an induced subgraph of exactly k edges. □

6. Counting k-edge induced subgraphs

In this section we study two counting versions of k-EDGE-INDUCED-SUBGRAPH. Of course, the most natural version is:

k-#EDGE-INDUCED-SUBGRAPH	
Instance:	A graph G and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Compute the number of k-edge induced subgraphs in G.

In general, a parameterized counting problem is a pair (F, κ), where $F : \Sigma^* \to \mathbb{N}$ and κ is a parameterization. (F, κ) is fixed-parameter tractable if F can be computed by an fpt-algorithm with respect to κ. For more background of parameterized counting complexity, the reader is referred to [15, 26].

In fact, the hardness of k-#EDGE-INDUCED-SUBGRAPH is rather easy to show. We observe that the vertex set of every induced subgraph *without any edge* is an independent set, and vice versa. Hence the *first slice* of k-#EDGE-INDUCED-SUBGRAPH, i.e., counting the number of 0-edge induced subgraphs is exactly the classical problem:

#INDEPENDENT-SET	
Instance:	A graph G.
Problem:	Compute the number of independent sets in G.

Recall that #INDEPENDENT-SET is #P-hard [33, 31]. Hence:

Theorem 6.1. *Assume* #P $\neq$ P. *Then* k-#EDGE-INDUCED-SUBGRAPH *is not fixed-parameter tractable.*

One might attribute the above hardness result to the fact that we allow induced subgraphs to have isolated vertices. Note these isolated vertices play no role in the decision problem k-EDGE-INDUCED-SUBGRAPH. Therefore, it also makes sense to consider:

k-#EDGE-INDUCED-SUBGRAPH*	
Instance:	A graph G and $k \in \mathbb{N}$.
Parameter:	k.
Problem:	Compute the number of k-edge induced subgraphs *without isolated vertices* in G.

Then we show:

Theorem 6.2. k-#EDGE-INDUCED-SUBGRAPH* *is hard for* #W[1].

Here, #W[1] is the counting version of the parameterized class W[1]. One standard complete problem of #W[1] is:

> p-#Independent-Set
>
> *Instance:* A graph G and $k \in \mathbb{N}$.
>
> *Parameter:* k.
>
> *Problem:* Compute the number of independent sets of size k in G.

To prove the #W[1]-hardness, we need an appropriate notion of reduction. Let (F, κ) and (F', κ') be two parameterized counting problems. An fpt *Turing reduction* from (F, κ) to (F', κ') is an algorithm $\mathbb{A}$ with an oracle to F' which satisfies the following conditions:

– $\mathbb{A}$ computes the function F in fpt-time (with respect to κ).

– There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that for all oracle queries "$F'(y) =?$" posed by $\mathbb{A}$ on input x we have $\kappa'(y) \leq g(\kappa(x))$.

It is easy to verify that if (F, κ) is #W[1]-hard and there is an fpt Turing reduction from (F, κ) to (F', κ'), then (F', κ') is #W[1]-hard.

Proof of Theorem 6.2: We give an fpt Turing reduction from p-#Independent-Set to p-#Edge-Induced-Subgraph*. To simplify the presentation, let us call an induced subgraph without isolated vertices *nice*.

Let (G, k) be an instance of p-#Independent-Set. For each $i \in [k]$ we define $V_{2\cdot i-1} := \big\{(v, i) \bigm| v \in V(G)\big\}$. Moreover, for $i \in [k-1]$ let $V_{2\cdot i} := \{e_i\}$, where all e_i's are new vertices not in $V(G)$. Then we define a new graph H with

$$
\begin{aligned}
V(H) &:= \bigcup_{i\in[2\cdot k-1]} V_i \\
E(H) &:= \bigcup_{i\in[k]} \big\{\{(u,i),(v,i)\} \bigm| u, v \in V(G) \text{ with } u \neq v\big\} \\
&\quad \cup \bigcup_{1\leq i<j\leq k} \big\{\{(u,i),(v,j)\} \bigm| u = v \text{ or } \{u,v\} \in E\big\} \\
&\quad \cup \bigcup_{i\in[k-1]} \big\{\{(v,j),e_i\} \bigm| v \in V(G) \text{ and } (j = i \text{ or } j = i+1)\big\}.
\end{aligned}
$$

For each $i \in [2 \cdot k - 1]$ we call V_i a *block* of G. Observe that each odd block is a clique of size $|V(G)|$ and each even block a singleton set.

Let $\{v_1, \ldots, v_k\} \subseteq V(G)$ be an independent set of size k in G. Clearly

$$G\Big[\big\{(v_i, i) \bigm| i \in [k]\big\} \cup \big\{e_i \bigm| i \in [k-1]\big\}\Big]$$

is a $(2 \cdot k - 2)$-edge nice induced subgraph of G. The crucial observation is that the following converse is also true.

Claim. Let H' be a nice induced subgraph of H containing exactly $2 \cdot k - 2$ edges. If $V(H') \cap V_i \neq \emptyset$ for every $i \in [2 \cdot k - 1]$, i.e., H' intersects all blocks V_i's, then

$$\big\{ v \in V \mid \text{for some } i \in [k] \text{ we have } (v, 2 \cdot i - 1) \in V(H') \big\}$$

is an independent set in G of size k.

Proof of the claim. First we show that $|V(H') \cap V_i| = 1$ for all $i \in [2 \cdot k - 1]$. This is obviously true for even i's, i.e., H' contains all e_i's. As e_i is adjacent to every vertex in the blocks $V_{2 \cdot i - 1}$ and $V_{2 \cdot i + 1}$, if H' contains two vertices in one odd block, then H' would have more than $2 \cdot k - 2$ edges, a contradiction.

Next for every $i \in [k]$ let v_i be the vertex in G such that $V(H') \cap V_{2 \cdot i - 1} = \big\{ (v_i, 2 \cdot i - 1) \big\}$. At this point, we already know that H' contains the following $2 \cdot k - 2$ edges

$$\{v_1, e_1\}, \{e_1, v_2\}, \dots, \{v_{k-1}, e_k\}, \{e_k, v_k\}. \tag{11}$$

We prove that $\{v_1, \dots, v_k\}$ is an independent set in G of size k. Otherwise for some $1 \leq i < j \leq k$ we have $v_i = v_j$ or $\{v_i, v_j\} \in E(G)$. Then H' would contain a further edge $\big\{ (v_i, 2 \cdot i - 1), (v_j, 2 \cdot j - 1) \big\}$ and hence have more than $2 \cdot k - 2$ edges by (11). $\dashv$

It follows that

$$\begin{aligned} &\big(\text{the number of independent sets of size } k \text{ in } G\big) \cdot \ k! \\ &\qquad = \text{the number of } (2 \cdot k - 2)\text{-edge nice induced subgraphs in } H \text{ which intersect every } V_i. \end{aligned} \tag{12}$$

Thus our goal is to compute the right hand side of (12) using k-#Edge-Induced-Subgraph as an oracle. To that end for every $X \subseteq [2 \cdot k - 1]$ we let

$$H_X := H \left[\bigcup_{i \in X} V_i \right]$$

and

$$\begin{aligned} s_X &:= \text{the number of } (2 \cdot k - 2)\text{-edge nice induced subgraphs in } H_X, \\ t_X &:= \text{the number of } (2 \cdot k - 2)\text{-edge nice induced subgraphs in } H_X \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{which intersect } V_i \text{ for every } i \in X. \end{aligned}$$

Therefore, the right hand side of (12) is exactly $t_{[2 \cdot k - 1]}$.

Note every s_X can be computed by an oracle query to k-#Edge-Induced-Subgraph on the instance $(H_X, 2 \cdot k - 2)$. Moreover it is easy to see

$$t_X = s_X - \sum_{Y \subsetneq X} t_Y.$$

Hence, by simple dynamic programming using k-#Edge-Induced-Subgraph as an oracle, we can compute every t_X in fpt time. $\square$

7. Conclusion

In this paper, we present an fpt-algorithm for k-Edge-Induced-Subgraph. In particular, this shows that for every fixed $k \in \mathbb{N}$ deciding whether a graph G has a k-edge induced subgraph can be done in linear time. Moreover, we also prove that two natural counting versions of k-Edge-Induced-Subgraph are both hard. Such a phenomenon might have theoretical interest on its own.

When designing our algorithm, we first use an algorithmic meta-theorem to prove that some crucial cases are efficiently solvable. With this fact in mind, we then provide refined combinatorial algorithms for those cases. It results in a double exponential fpt-algorithm, instead of a 4-fold exponential one in case we use the logic-machinery, for k-Edge-Induced-Subgraph. Through this approach, we hope to convey a similar message as in [20, 21] that logic can help designing combinatorial algorithms.

As we have mentioned, k-Edge-Induced-Subgraph can be formulated as a special quadratic Diophantine equation $x^T Ax = 2k$, where A is the adjacency matrix of an n-vertex simple graph G and $x \in \{0,1\}^n$. Our technique might have application on other variances of quadratic Diophantine equations, e.g. where $x \in \{0,1,2,..,d\}^n$ or G is a multi-graph.

The obvious question is whether our algorithm can be improved to single exponential. In case not, can we prove a matching lower bound? At the moment, it seems to be out of reach.

Acknowledgement. We thank Leizhen Cai for bringing the problem k-Edge-Induced-Subgraph to our attention, and Jörg Flum for comments on earlier versions of this paper.

Special thanks also go to an anonymous reviewer for improving the algorithm for Case 2 in the proof of Lemma 4.10, and his idea also applies to Case 2 in the proof of Lemma 4.12. This significantly lowers the running time of our overall algorithm from triple exponential to double exponential.

References

[1] Noga Alon and Béla Bollobás. Graphs with a small number of distinct induced subgraphs. *Discrete Mathematics*, 75(1-3):23–30, 1989.

[2] Noga Alon and Alexandr V. Kostochka. Induced subgraphs with distinct sizes. *Random Struct. Algorithms*, 34(1):45–53, 2009.

[3] G. Andrews. Eureka! num $= \Delta + \Delta + \Delta$. *Journal of Number Theory*, 23(3):285–293, 1986.

[4] H.L. Bodlaender, L. Cai, J. Chen, M.R. Fellows, J.A. Telle, and D. Marx. Open problems

in parameterized and exact computation - IWPEC 2006. Technical Report UU-CS-2006-052, Department of Information and Computing Sciences, Utrecht University, 2006.

[5] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996.

[6] L. Cai. Private communication, 2008.

[7] L. Cai, S.M. Chan, and S.O. Chan. Random separation: A new method for solving fixed-cardinality optimization problems. In *Proceedings of the second International Workshop on Parameterized and Exact Computation (IWPEC'06)*, Lecture Notes in Computer Science 4169, pages 239–250. Springer, 2006.

[8] Y. Chen and J. Flum. On parameterized path and chordless path problems. In *Proceedings of 22nd Annual IEEE Conference on Computational Complexity (CCC 2007)*, pages 250–263. IEEE Computer Society, 2007.

[9] Y. Chen, M. Thurley, and M. Weyer. Understanding the complexity of induced subgraph isomorphisms. In *Proceedings of the 35th International Colloquium on Automata Languages and Programming (ICALP'08)*, pages 587–596. Springer, 2008.

[10] M. Chudnovsky, K. Kawarabayashi, and P. D. Seymour. Detecting even holes. *Journal of Graph Theory*, 48(2):85–111, 2005.

[11] B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. Van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 192–242, Amsterdam, The Netherlands, 1990. Elsevier Science Publishers.

[12] A. Dawar, M. Grohe, S. Kreutzer, and N. Schweikardt. Model theory makes formulas large. In *Proceedings of the 34th International Colloquium on Automata Languages and Programming (ICALP'07)*, pages 913–924. Springer, 2078.

[13] R.G. Downey and M.R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[14] Z. Dvorak, D. Král, and R. Thomas. Deciding first-order properties for sparse graphs. In *Proceedins of the 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS'10)*, pages 133–142. IEEE Computer Society, 2010.

[15] J. Flum and M. Grohe. The parameterized complexity of counting problems. *SIAM Journal on Computing*, 33(4):892–922, 2004.

[16] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[17] M. Frick and M. Grohe. Deciding first-order properties of locally tree-decomposable structures. *Journal of ACM*, 48(6):1184–1206, 2001.

[18] M. Frick and M. Grohe. The complexity of first-order and monadic second-order logic revisited. *Annual of Pure and Applied Logic*, 130(1-3):3–31, 2004.

[19] H. Gaifman. On local and non-local properties. In *Proceedings of the Herbrand Symposium, Logic Colloquium'91*, pages 105–135. North-Holland, 1982.

[20] M. Grohe. Computing crossing numbers in quadratic time. *Journal of Computer and System Sciences*, 68(2):285–302, 2004.

[21] M. Grohe, K. Kawarabayashi, and B. A. Reed. A simple algorithm for the graph minor decomposition - logic meets structural graph theory. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'13)*, pages 414–431, 2013.

[22] M. Grohe, S. Kreutzer, and S. Siebertz. Deciding first-order properties of nowhere dense graphs. In *Proceedins of the 46th Symposium on Theory of Computing, (STOC'14)*, pages 89–98. ACM, 2014.

[23] M. Grohe and S. Wöhrle. An existential locality theorem. *Annals of Pure and Applied Logic*, 129(1-3):131–148, 2004.

[24] R. Haas and M. Hoffmann. Chordless paths through three vertices. *Theoretical Computer Science*, 351(3):360–371, 2006.

[25] S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289(2):997–1008, 2002.

[26] C. McCartin. Parameterized counting problems. *Annals of Pure and Applied Logic*, 138(1-3):147–182, 2006.

[27] H. Moser and S. Sikdar. The parameterized complexity of the induced matching problem. *Discrete Applied Mathematics*, 157(4):715–727, 2009.

[28] H. Moser and D. M. Thilikos. Parameterized complexity of finding regular induced subgraphs. *Journal of Discrete Algorithms*, 7(2):181–190, 2009.

[29] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.

[30] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. *Journal of Computer and System Sciences*, 53(2):161–170, 1996.

[31] J. S. Provan and M. O. Ball. The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM Journal on Computing*, 12(4):777–788, 1983.

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

[33] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

Appendix. A

For the reader not familiar with [17] we give a detailed proof of Theorem 4.1. Our presentation closely follows that of [16, Section 12.2]. Overall we will reduce p-MC-LTW$_g$-FO to a generalization of the parameterized independent set problem.

Definition 7.1. Let $G = (V, E)$ be a graph and $\ell, r \in \mathbb{N}$. A set $S \subseteq V$ is (ℓ, r)-*scattered* if there exist $v_1, \ldots, v_\ell \in S$ such that for every $1 \leq i < j \leq \ell$ we have $d(v_i, v_j) > r$.

Proposition 7.2. *Let $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ be a computable function. Then the following parameterized problem is fixed-parameter tractable.*

p-SCATTERED-SET-LTW$_g$	
Instance:	A graph $G = (V, E)$, $S \subseteq V$ and $p, \ell, r \in \mathbb{N}$ such that G has local tree-width bounded by g with respect to p.
Parameter:	$p + \ell + r$.
Problem:	Decide whether S is (ℓ, r)-scattered.

To prove this proposition we need another simple combinatorial result (for a proof see, e.g., [16, Lemma 12.12]).

Lemma 7.3. *Let $G = (V, E)$ be a connected graph and $S \subseteq V$ a dominating set*[4] *in G. Then $d(u, v) \leq 3 \cdot |S| - 1$ for every $u, v \in V$. That is, the diameter of G is bounded by $3 \cdot |S| - 1$.*

Proof of Proposition 7.2: By Courcelle's Theorem [11] it is easy to see that the problem

p-SCATTERED-SET-TW	
Instance:	A graph $G = (V, E)$, $S \subseteq V$ and $\ell, r \in \mathbb{N}$.
Parameter:	$\mathrm{tw}(G) + \ell + r$.
Problem:	Decide whether S is (ℓ, r)-scattered.

[4] Recall, $S \subseteq V(G)$ is a dominating set if for every $u \in V(G)$ either $u \in S$ or there is a vertex $v \in S$ with $\{u, v\} \in E(G)$.

is fixed-parameter tractable. So our goal is to give an fpt-reduction from p-Scattered-Set-Ltw$_g$ to p-Scattered-Set-Tw.

First, using a simple greedy algorithm, we can compute in linear time a *maximal* set $T \subseteq S$ such that for every distinct $u, v \in T$ we have $d^G(u,v) > r$. If $|T| \geq \ell$, then we are done. Otherwise

$$|T| < \ell. \tag{13}$$

Claim 1. $S \subseteq N_r^G(T) \Big(:= \big\{ v \in V \bigm| d^G(u,v) \leq r \text{ for some vertex } u \in T \big\} \Big)$.

Proof of the claim. Otherwise let $v \in S \setminus N_r^G(T)$. Thus $d^G(v,u) > r$ for every $u \in T$. This contradicts the maximality of T. $\dashv$

Claim 2. S is (ℓ, r)-scattered in G if and only if S is (ℓ, r)-scattered in $\mathcal{N}_{2\cdot r}^G(T)$ $\Big(:= G\big[N_{2\cdot r}^G(T)\big] \Big)$.

Proof of the claim. The direction from left to right is trivial. So let us assume that S is (ℓ, r)-scattered in $\mathcal{N}_{2\cdot r}^G(T)$. In particular, there exist $v_1, \ldots, v_\ell \in S$ such that

$$d^{\mathcal{N}_{2\cdot r}^G(T)}(v_i, v_j) > r \tag{14}$$

for every $1 \leq i < j \leq \ell$. Towards a contradiction assume that there exist some $i, j \in \mathbb{N}$ with $1 \leq i < j \leq \ell$ and $d^G(v_i, v_j) \leq r$. Note every vertex u in a shortest path between v_i and v_j satisfies $d^G(u, v_i) \leq r$, and hence, $u \in N_r^G(S)\big(:= \big\{ v \in V \bigm| d^G(u,v) \leq r \text{ for some vertex } u \in S \big\}\big)$. Then by Claim 1, $u \in N_{2\cdot r}^G(T)$. As a consequence $d^{\mathcal{N}_{2\cdot r}^G(T)}(v_i, v_j) \leq r$, which contradicts (14). $\dashv$

Claim 2 shows that the mapping

$$R(G, S, p, \ell, r) := \big(\mathcal{N}_{2\cdot r}^G(T), S, \ell, r \big)$$

is a correct reduction from p-Scattered-Set-Ltw$_g$ to p-Scattered-Set-Tw. It remains to show R is an fpt-reduction. To that end, we need to bound $\mathrm{tw}\big(\mathcal{N}_{2\cdot r}^G(T)\big) + \ell + r$ in terms of $p + \ell + r$.

Claim 3. $\mathrm{tw}\big(\mathcal{N}_{2\cdot r}^G(T)\big) \leq g(2 \cdot r \cdot (3 \cdot \ell - 4), p)$.

Proof of the claim. Let H be a graph with

$$V(H) := N_{2\cdot r}^G(T) \text{ and } E(H) := \big\{ \{u,v\} \mid u, v \in V(H),\ u \neq v \text{ and } d^G(u,v) \leq 2 \cdot r \big\}.$$

It is then easy to verify that T is a dominating set in H. Hence by Lemma 7.3, every connected component of H has diameter at most $3 \cdot |T| - 1 \leq 3 \cdot \ell - 4$ by (13). It follows that every connected component C of $\mathcal{N}_{2\cdot r}^G(T)$ has diameter at most $2 \cdot r \cdot (3 \cdot \ell - 4)$. This implies that $C = N_{2\cdot r\cdot(3\cdot\ell-4)}^G(v)$ for every $v \in C$. Recall that G has local tree-width bounded by g with respect to p. Hence,

$$\mathrm{tw}\big(\mathcal{N}_{2\cdot r}^G(T)\big) \leq g(2 \cdot r \cdot (3 \cdot \ell - 4), p)$$

$\dashv$

This finishes the proof. □

Now we recall Gaifman's Theorem [19].

Lemma 7.4. *Let τ be a vocabulary and $r \in \mathbb{N}$. Then there is an* FO*-formula $\delta_r(x, y)$ such that for all τ-structure $\mathcal{A}$ and all elements $a, b \in A$ we have $d^{G(\mathcal{A})}(a, b) \le r$ if an only if $\mathcal{A} \models \delta_r(a, b)$.*

For simplicity we will write $d(x, y) \le r$ and $d(x, y) > r$ instead of $\delta_r(x, y)$ and $\neg\delta_r(x, y)$, respectively.

An FO τ-formula $\psi(x)$ is *r-local* if for all τ-structure $\mathcal{A}$ and $a \in A$:

$$\mathcal{A} \models \psi(a) \iff \mathcal{N}_r^{\mathcal{A}}(a) \models \psi(a).$$

Theorem 7.5 (Gaifman's Theorem). *Every* FO*-sentence φ is equivalent to a Boolean combination of sentences of the form*

$$\exists x_1 \ldots \exists x_\ell \left(\bigwedge_{1 \le i < j \le \ell} d(x_i, x_j) > 2 \cdot r \wedge \bigwedge_{i \in [\ell]} \psi(x_i) \right).$$

with $\ell, r \in \mathbb{N}^+$. Moreover, such a Boolean combination can be computed from φ.

Now we have all the tools for proving Theorem 4.1 which for the reader's convenience we repeat as below:

Theorem 7.6. *For every computable function $g : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ the problem p-*Mc-Ltw$_g$-FO *is fixed-parameter tractable.*

Proof: Let $(\mathcal{A}, p, \varphi)$ be an instance of p-Mc-Ltw$_g$-FO. It is easy to see that, by Gaifman's Theorem, we can assume without loss of generality that for some $\ell, r \in \mathbb{N}$ and r-local FO-formula ψ

$$\varphi = \exists x_1 \ldots \exists x_\ell \left(\bigwedge_{1 \le i < j \le \ell} d(x_i, x_j) > 2 \cdot r \wedge \bigwedge_{i \in [\ell]} \psi(x_i) \right).$$

Let $G = (V, E)$ be a graph with $V := A$ and $E := \big\{ \{a, b\} \bigm| a, b \in A \text{ and } d^{G(\mathcal{A})}(a, b) = 1 \big\}$. That is, G is Gaifman's graph of $\mathcal{A}$. Moreover, let $S := \big\{ a \in A \bigm| \mathcal{A} \models \psi(a) \big\}$. By the r-locality of ψ we have $S = \big\{ a \in A \bigm| \mathcal{N}_r^{\mathcal{A}}(a) \models \psi(a) \big\}$. Since $\mathrm{tw}\left(\mathcal{N}_r^{\mathcal{A}}(a)\right) \le g(r, p)$, we can compute the set S in fpt time, again by Courcelle's Theorem.

It is now easy to verify that $\mathcal{A} \models \varphi$ if and only if S is (ℓ, r)-scattered in G, i.e.,

$$(\mathcal{A}, p, \varphi) \in p\text{-Mc-Ltw}_g\text{-FO} \iff (G, S, p, \ell) \in p\text{-Scattered-Set-Ltw}_g.$$

Now the result follows from Proposition 7.2. □

Appendix. B

Theorem 7.7 (Cai [6])**.** *It is is fixed-parameter tractable to decide whether a given graph G contains a $\binom{k}{2}$-edge induced subgraph where $k \in \mathbb{N}$ is the parameter.*

Proof: Let G and k be given. If every vertex of G has degree less than $\mathscr{R}_{\binom{k}{2}}$, then by Proposition 4.7, we can decide in fpt time whether G contains a $\binom{k}{2}$-edge induced subgraph.

Otherwise, assume v is vertex in G with $\deg(v) \geq \mathscr{R}_{\binom{k}{2}}$. Consider the subgraph H of G induced by $N^G(v)$, in particular, $|V(H)| \geq \mathscr{R}_{\binom{k}{2}}$. By Ramsey's Theorem, either H has an independent set S of size of $\binom{k}{2}$ or a clique S of size $\binom{k}{2}$. In the first case $S \cup \{v\}$ induces a $\binom{k}{2}$-edge subgraph in H and hence also in G. In the second case, clearly H (and thus G) contains a smaller clique of size k which has exactly $\binom{k}{2}$ edges. □

Remark 7.8. As mentioned in the Introduction, Cai's original argument uses the Random Separation method [7] instead of Proposition 4.7 as what we have done above.