The parameterized complexity of $k$-edge induced subgraphs

Bingkai Lin and Yijia Chen

Department of Computer Science and Engineering
Shanghai Jiaotong 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 [16]. 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.

1. Introduction
Induced subgraphs are one of the most natural substructures in graphs. They capture many different combinatorial objects, e.g., clique, independent set, chordless path. 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, 18, 19, 21, 22, 24]. Induced subgraphs with distinct number of edges have also been studied in graph theory [1, 2].

In this paper, we are mainly interested in the problem of finding an induced subgraph which contains exactly $k$ edges, i.e., a $k$-edge induced subgraph. This problem is equivalent to solving a special 0-1 quadratic Diophantine equation $x^TAx = k$, where $A$ is the adjacent matrix of $G$, $x \in \{0, 1\}^n$, $n = |V(G)|$.

It is not difficult to prove that the $k$-edge induced subgraph problem is NP-hard by a reduction from the clique problem. So we approach the problem via parameterized complexity [12, 15, 23] and treat $k$ as the parameter:

$p$-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 $p$-EDGE-INDUCED-SUBGRAPH is fixed-parameter tractable. In fact, there are special cases of $p$-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 $p$-EDGE-INDUCED-SUBGRAPH is fixed-parameter tractable if the graph $G$ has bounded tree-width [11], bounded local tree-width [16], etc., or most generally locally bounded expansion [13]. 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 that solves $p$-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 \( p\text{-Edge-Induced-Subgraph} \) by its own. So our \textsc{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 (\text{FO}) which depends on \( k \) only. It is a well-known result of Frick and Grohe \cite{frick2003first} that the model-checking problem for \text{FO} on graphs of bounded local tree-width is fixed-parameter tractable. The local tree-width for 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 have bounded local tree-width. These give immediately the fixed-parameter tractability of \( p\text{-Edge-Induced-Subgraph} \) on graphs with bounded degree.

With some more efforts, 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 \textsc{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 \( p\text{-Edge-Induced-Subgraph} \) in \textsc{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 \( p\text{-Edge-Induced-Subgraph} \) 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 \( p\text{-Edge-Induced-Subgraph} \) 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 \( V_{11} \cup V_{12} \) 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 \( V_{12} \) 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 in terms of the parameter \( k \) is astronomical, **triple exponential** at least. 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.

\footnote{This is also a direct consequence of Seese’s result that the model-checking problem for \text{FO} on bounded-degree graphs is fixed-parameter tractable \cite{seese1993complexity}. But we find it more natural to work with bounded local tree-width in the following generalization.}
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 $p$-EDGE-INDUCED-SUBGRAPH are both hard. To the best of our knowledge, there are only very few natural problems which exhibit such a phenomenon \cite{14, 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 exist, guarantee the existence of a $k$-edge induced subgraph. Then in Section 4 we establish the fixed-parameter tractability of $p$-Edge-Induced-Subgraph on degree-extreme graphs and bridges using model-checking problems for FO. We present our fpt-algorithm for $p$-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 \cite{16}, we provide a proof of the easy generalization of Frick and Grohe’s algorithm in an appendix.

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 \([n] := \{1, \ldots , n\}\).

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

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., \(\{\{a, b\} \mid a, b \in S\text{ and } a \neq b\}\). A triangular number is \(\binom{k}{2} := \frac{k(k-1)}{2}\) for some \(k \in \mathbb{N}\). In particular, \(\binom{0}{2} = \binom{1}{2} = 0\).

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

An algorithm \(A\) is an fpt-algorithm with respect to a parameterization \(\kappa\) if for every \(x \in \Sigma^*\) the running time of \(A\) on \(x\) is bounded by \(f(\kappa(x)) \cdot |x|^\Theta(1)\) for a computable function \(f : \mathbb{N} \to \mathbb{N}\). Or equivalently, we say that the algorithm \(A\) runs in fpt time. A parameterized problem \((Q, \kappa)\) is fixed-parameter tractable if there is an fpt-algorithm with respect to \(\kappa\) that decides \(Q\).

Let \((Q, \kappa)\) and \((Q', \kappa')\) be two parameterized problems. An fpt-reduction from \((Q, \kappa)\) to \((Q', \kappa')\) 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 \((Q, \kappa)\) to \((Q', \kappa')\), and if \((Q', \kappa')\) is fixed-parameter tractable, then so is \((Q, \kappa)\).

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 \(E \subseteq \binom{V}{2}\). 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 \( E(G[S]) := \left(\frac{S}{2}\right) \cap E(G) \). 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, the 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 \( R_k \), known as the Ramsey number, such that every graph \( G \) with \(|V(G)| \geq R_k \) has either a clique of size \( k \) or an independent set of size \( k \).

It is well-known that \( R_k < 2^{2^k} \) for every \( k \in \mathbb{N} \).

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

The disjoint union of two \( \tau \)-structures \( A_1 \) and \( A_2 \) is again a \( \tau \)-structure, denoted by \( A_1 \cup A_2 \), whose universe is \( A_1 \cup A_2 \), and where for each relation symbol \( R \in \tau \) we let \( R^{A_1 \cup A_2} := R^{A_1} \cup R^{A_2} \).

Let \( A \) be a structure of a vocabulary \( \tau \). Then the Gaifman graph of \( A \) is \( G(A) := (V, E) \) with \( V := A \) and

\[
E := \{\{a, b\} \mid a, b \in A \text{ with } a \neq b, \text{ and there exists an } R \in \tau \text{ and a tuple } (a_1, \ldots, a_r) \in R^A \text{ with } \{a, b\} \subseteq \{a_1, \ldots, a_r\}\}.
\]

Note any unary relation in \( A \) has no influence on \( E \).

Let \( r \in \mathbb{N} \) and \( a \in A \). Then the \( r \)-neighborhood of \( a \) is \( N^A_r(a) := \{b \in A \mid d^A(a, b) \leq r\} \).

Moreover, the structure \( N^A_r(a) \) induced by the \( r \)-neighborhood of \( a \) has universe \( N^A_r(a) \), and for each \( r \)-ary relation symbol \( R \in \tau \) the interpretation \( \{ (a_1, \ldots, a_r) \in R^A \mid a_1, \ldots, a_r \in N^A_r(a) \} \).

Formulas of first-order logic of vocabulary \( \tau \) are built up from atomic formulas \( x = y \) and \( Rx_1 \ldots x_r \), where \( x, y, x_1, \ldots, x_r \) 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

\[
is_k := \exists x_1 \ldots \exists x_k \left( \bigwedge_{1 \leq i < j \leq k} (\neg x_i = x_j \land \neg Ex_ix_j) \right).
\]

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

**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 \( \text{tw}(A) \) of a structure \( A \) is simply \( \text{tw}(G(A)) \), that is, the tree-width of the Gaifman graph of \( A \). In fact, to understand most parts of our proofs and algorithms, it is sufficient to know that

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

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

\[
\text{ltw}(A, r) := \max \{ \text{tw}(N^A_r(a)) \mid a \in A \}.
\]

Let \( g : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \) be a function and \( p \in \mathbb{N} \). We say a structure \( A \) has local tree-width bounded by \( g \) with respect to \( p \) if \( \text{ltw}(A, r) \leq g(r, p) \) for every \( r \in \mathbb{N} \). This slightly generalizes the usual notion of local tree-width bounded by a unary function [16].
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 $G$ 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

$$\ell := \left| E(G[u_1, \ldots, v_{k'}]) \right| \leq k.$$

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, $\left| E(G[v_1, \ldots, v_{k'+1}]) \right| > k$. 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 $G$. \hfill \Box

**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 $G$ 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 $\left(\frac{k_0}{2}\right) \leq k$ and set $r := k - \left(\frac{k_0}{2}\right)$. 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

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

If $r' = r$, then $G[V']$ with $V' := \{v_1, \ldots, v_r\} \cup \{u_1, \ldots, u_{r-\ell}, u_{r+1}, \ldots, u_{k_0+\ell}\}$ is a $k$-edge induced subgraph of $G$. Otherwise, $r' < r$ and by the maximality of $r'$ we have $\left| E(G[v_1, \ldots, v_{r'}, v_{r'+1}]) \right| > r$. 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 $V' := \{v_1, \ldots, v_{r'}\} \cup \{u_1, \ldots, u_{r-\ell}, u_{r'+1}, \ldots, u_{r'+k_0-r+\ell}\}$ has exactly $k$ edges. \hfill \Box

**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 $|B| \geq \mathcal{R}_k$.

(A2) $A$ is a clique in $G$. 

\hfill \Box
Let $s$ if we can easily achieve (i). So assume now that $s > 1$ and possibly $v_0 \notin A$. We prove by induction on $s$, noting that Lemma 3.6 is a direct consequence of Gauss’ famous Eureka Theorem \[3\].

**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 $k = 1$ is trivially true. So let $k \geq 2$. Moreover, let $v_0, A, B$ be as stated in Definition \[3\]. Since $|B| \geq \mathcal{R}_k$, $G[B]$ 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 $G[B' \cup \{u\}]$ 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 $G[A' \cup B'' \cup \{v_0\}]$ 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 $|N(u) \cap B| \geq 1$ 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 $|N(v) \cap A| \geq m$, then we can easily achieve (i). So assume now that $\forall v \in B \: |N(v) \cap A| \leq m - 1.$ \[1\]
Choose an arbitrary vertex \( v \in B \) and let \( B' := B \setminus \{v\} \). If for every \( u \in A \) we have \( |N(u) \cap B'| \geq 1 \), 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 \( \text{Lemma 3.9} \) it holds that \( |A'| > (m - 1)(n - 2) \). Then by induction hypothesis on \( A' \), \( B \), \( m \), and \( n \), together with \( \text{Lemma 3.9} \), 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 \( u := u \) and \( v := v \), we have \( \{u_i, v_j\} \in E \) if and only if \( i = j \), for every \( i, j \in [n] \). \( \square \)

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

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. \( \square \)

**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 \( |V^G_{[1,d]}| > (d + 1) \cdot (k - 1)^2 \), 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 \( V^G_{[1,d]} \subseteq V' \) and removing their neighbors. \( \square \)

**Remark 3.13.** An immediate consequence of Lemma 3.12 is that \( p \)-EDGE-INDUCED-SUBGRAPH \( G \) is solvable in time \( 2^{O(d^k \cdot k^2)} \) on graphs of degree \( d \leq d \).

### 3.1. A further combinatorial lemma.

For later purpose, we need a generalization of Lemma 3.10.

**Lemma 3.14.** Let \( m, n, p \in \mathbb{N}^+ \) and \( G = (V, E) \) be a graph. Furthermore, let \( A, B \subseteq V \) be disjoint such that \( |N(u) \cap B| \geq p \) 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 \( m \leftarrow (m - 1)(n - 1)^{p-1} + 1 \) and \( n \leftarrow n \).

Thus

(a) either there are \( (m - 1)(n - 1)^{p-1} + 1 \) vertices \( u_1, \ldots, u_{(m-1)(n-1)^{p-1}+1} \) in \( A \) and a vertex \( v \) in \( B \) with \( \{u_i, v\} \in E \) for every \( i \in [(m - 1)(n - 1)^{p-1} + 1] \),
Theorem 4.1. restricted classes of graphs via the model-checking problem for first-order logic.

In this section we show the fixed-parameter tractability of

and

and the edges between a small degree vertex and a large degree one, and takes the complement of

\( A \)

Basically, \( V \)

**Definition 4.3**

\[ A' := \{u_1, \ldots, u_{(m-1)(n-1)}\}, \quad B' := B \setminus \{v\}, \quad m' := m, \quad n' := n, \quad \text{and} \quad p' := p - 1. \]

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

\[ A' \leftarrow A', B \leftarrow B', m \leftarrow m', n \leftarrow n' , \quad \text{and} \quad p \leftarrow p'. \]

If (ii) holds for \( A', B', n' \), then it holds for \( A, B, n, \) too. Otherwise there are \( m \) vertices

\( u_1', \ldots, u_m' \) in \( A' \subseteq A \) and \( p - 1 \) vertices \( v_1', \ldots, v_p' \) in \( B' \subseteq B \) with \( \{u_i', v_j'\} \in E \) for every \( i \in [m] \) and \( j \in [p - 1] \).

Recall now (i) 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 \( u_1', \ldots, u_m', v_1', \ldots, v_p' \in A, v_1', \ldots, v_p' \in B, m, \) and \( p \) by \( v \in B \setminus B' \). \( \square \)

4. Easy instances by model-checking

In this section we show the fixed-parameter tractability of \( p \)-\textsc{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 [16].

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

\[
\begin{array}{|c|}
\hline
p\text{-}\textsc{Mc-Ltw}_g\text{-}\textsc{FO} \\
\hline
\text{Instance:} & A \text{ structure } A, p \in \mathbb{N} \text{ and an FO-sentence } \varphi \text{ such that } A \\
& \text{has local tree-width bounded by } g \text{ with respect to } p. \\
\text{Parameter:} & p + |\varphi|. \\
\text{Problem:} & \text{Decide whether } A \models \varphi. \\
\hline
\end{array}
\]

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 \( \deg(v) \leq d \) or \( \deg(v) \geq |V| - 1 - d \) 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 \( d \geq 1 \).

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 \( A := A(G, d) \) is a \( \tau_{\text{des}} \)-structure defined by \( A := V, P^A := V^G_{\leq d} \), and

\[
R^A := \{ (u, v) \mid \{u, v\} \in E \text{ and } u \in V^G_{\leq d} \text{ or } v \in V^G_{\leq d} \} \\
\cup \{ (u, v) \mid \{u, v\} \notin E, u, v \in V \setminus V^G_{\leq d} \text{ and } u \neq v \}.
\]

Basically, \( A(G, d) \) 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.
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 \( |V_{[1,d]}^G| > (d+1) \cdot (k-1)^2 \), (hence, by Lemma 4.2, \( G \) has a \( k \)-edge induced subgraph),

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

Proof: We assume that (i) is not true, i.e., \( |V_{[1,d]}^G| \leq (d+1) \cdot (k-1)^2 \). For every \( v \in A = V(G) \) it is easy to verify that \( \deg^{G(A)}(v) \leq d + (d+1) \cdot (k-1)^2 \). Together with (T) (see page 4) we conclude

\[
\text{tw}(N_r^A(v)) < |N_r^A(v)| \leq \sum_{i=0}^r (d + (d+1) \cdot (k-1)^2)^i.
\]

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

\[
\text{edge}(x, y) := (Rxy \land (P_x \lor P_y)) \lor (\lnot Rxy \land \lnot P_x \land \lnot P_y).
\]

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

\[
\text{induced}_H := \exists x_1 \ldots \exists x_\ell \left( \bigwedge_{1 \leq i < j \leq \ell} \neg x_i = x_j \land \bigwedge_{(i,j) \in E} \text{edge}(x_i, x_j) \land \bigwedge_{\{i,j\} \in E} \neg \text{edge}(x_i, x_j) \right).
\]

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

\( G \) contains an induced subgraph isomorphic to \( H \) \iff \( A(G, d) \models \text{induced}_H \).

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

| Instance: | \( \text{A graph } G \) 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 \( A := A(G, D(k)) \). By Lemma 4.4 we can assume that

\[
\text{ltw}(A, r) \leq h_0(r, D(k), k).
\]

That is, the structure \( A \) 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

\[
\text{induced}_k := \bigvee_{H \text{ has no isolated vertex}}\text{induced}_H.
\]

It follows that \( G \) has an induced subgraph of exactly \( k \) edges if and only if \( A \models \text{induced}_k \). Note the structure \( A \) can be computed in fpt time, and the sentence \( \text{induced}_k \) can be computed from \( k \). Hence, \( (G, k) \mapsto (A, k, \text{induced}_k) \) gives an fpt-reduction to \( p\text{-MC-LTW}_d\text{-FO} \). The result then follows from Theorem 4.1.  \( \square \)
Remark 4.8. A careful analysis of the above algorithm shows that its running time in terms of the parameter $k$ is at least of the order of $2^{o(D(k))}$.

Definition 4.9 (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 the two degree-extreme graphs).

Similarly to degree-extreme graphs, we translate every bridge to a finite structure. To that end, for every $b \in \mathbb{N}$ let

$$\tau_{\text{bridge}, b} := \{U_1, U_2, P, R, F_1, \ldots, F_b, C_1, \ldots, C_b\},$$

where all symbols are unary except the binary $R$.

Definition 4.10 (bridge structure). Let $d,b \in \mathbb{N}$, $G = (V,E)$ be a graph and $V_1, V_2, B \subseteq V$ with $B = \{v_1, \ldots, v_b\}$ such that $(G,V_1,V_2,B)$ is a $(d,b)$-bridge of two $d$-degree-extreme graphs $G[V_1]$ and $G[V_2]$. Then we define the corresponding $\tau_{\text{bridge},b}$-structure

$$\mathcal{D} := \mathcal{D}(G,V_1,V_2,B,d) := \left( A(G[V_1],d) \cup A(G[V_2],d), U_1^D, U_2^D, F_1^D, \ldots, F_b^D, C_1^D, \ldots, C_b^D \right), \quad (2)$$

where $U_1^D := V_1$, $U_2^D := V_2$ and for every $i \in [b]$

$$F_i^D := \{v_i\}, \quad C_i^D := \{u \in V \mid \{u,v_i\} \in E\}.$$ 

That is, the bridge structure consists of two degree-extreme structures, plus all the edges between them encoded by $2 \cdot b$ unary relations.

Lemma 4.11. Let $d \in \mathbb{N}$, $k \in \mathbb{N}^+$, $G = (V,E)$ be a graph and $V_1, V_2, B \subseteq V$ such that $(G,V_1,V_2,B)$ is a $(d,|B|)$-bridge. Moreover, let $\mathcal{D} := \mathcal{D}(G,V_1,V_2,B,d)$. Then one of the following conditions is satisfied.

(i) $|V^{G[V_1]}_{[1,d]}| > (d+1) \cdot (k-1)^2$.

(ii) $|V^{G[V_2]}_{[1,d]}| > (d+1) \cdot (k-1)^2$.

(iii) $\text{ltw}(\mathcal{D},r) \leq h_0(r,d,k)$ for every $r \in \mathbb{N}$, where the function $h_0$ is defined in Lemma 4.4.

Observe in cases (i) and (ii), by Lemma 2.12, $G[V_1]$ or $G[V_2]$ and hence $G$ has a $k$-edge induced subgraph.

Proof: Assume that neither (i) nor (ii) holds. Let $v \in D = V$, $r \in \mathbb{N}$ and consider the structure $N_r^D(v)$. Observe that all unary relations $U_1^D, \ldots, C_b^D$ have no impact on the tree-width of $N_r^D(v)$, i.e.,

$$\text{tw} \left( N_r^D(v) \right) = \text{tw} \left( N_r^{A(G[V_1],d)} \cup A(G[V_2],d)(v) \right),$$

by (2). Hence

$$\text{tw} \left( N_r^D(v) \right) = \begin{cases} \text{tw} \left( N_r^{A(G[V_1],d)}(v) \right), & \text{if } v \in V_1 \\ \text{tw} \left( N_r^{A(G[V_2],d)}(v) \right), & \text{if } v \in V_2. \end{cases}$$

Then (iii) follows from Lemma 4.4.
\textbf{Definition 4.12.} For every \(b \in \mathbb{N}\) let
\[
edged(x, y) := (U_1x \wedge U_1y \wedge \text{edge}(x, y)) \lor (U_2x \wedge U_2y \wedge \text{edge}(x, y)) \lor \bigvee_{i \in [b]} \left( (F_i x \wedge C_i y) \lor (F_i y \wedge C_i x) \right).
\]
Recall the formula \(\text{edge}(x, y)\) is defined in Definition 4.9.

Then for every graph \(H = (V, E)\), where \(V = [\ell]\) for some \(\ell \in \mathbb{N}\), we define
\[
\text{induced}^2_{b,H} := \exists x_1 \ldots \exists x_\ell \left( \bigwedge_{1 \leq i < j \leq \ell} \neg x_i = x_j \bigwedge_{\{i,j\} \in E} \text{edge}^2_b(x_i, x_j) \bigwedge_{\{i,j\} \notin E} \neg \text{edge}^2_b(x_i, x_j) \right).
\]

\textbf{Lemma 4.13.} Let \(d, b \in \mathbb{N}\), \(G = (V, E)\) a graph and \(V_1, V_2, B \subseteq V\) such that \((G, V_1, V_2, B)\) is a \((d, b)\)-bridge. Then for every graph \(H\) we have
\[
G \text{ contains an induced subgraph isomorphic to } H \iff \mathcal{D}(G, V_1, V_2, B, d) \models \text{induced}^2_{b,H}.
\]

We omit the trivial proof.

\textbf{Proposition 4.14.} Let \(D : \mathbb{N} \rightarrow \mathbb{N}\) be a computable function. Then the problem

\begin{center}
\begin{tabular}{|l|}
\hline
\textbf{Instance:} & A graph \(G = (V, E)\), \(V_1, V_2, B \subseteq V\) and \(k \in \mathbb{N}\) such that \\
\textit{(S1)} \(G, V_1, V_2, B)\) is a \((D(k), |B|)\)-bridge. \\
\textbf{Parameter:} & \(k + |B|\). \\
\textbf{Problem:} & Decide whether \(G\) contains a \(k\)-edge induced subgraph. \\
\hline
\end{tabular}
\end{center}

is fixed-parameter tractable.

\textit{Proof:} This is similar to Proposition 4.7. \qed

\section{5. The algorithm}

The main component of our fpt-algorithm for \textsc{p-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 4.9).

For every \(k \in \mathbb{N}\) we let
\[
p_k := 2^{2^k} (> \mathcal{R}_k).
\]

\textbf{Lemma 5.1.} For every computable function \(D : \mathbb{N} \rightarrow \mathbb{N}\) there is an fpt-algorithm \(\mathcal{A}_D\) such that for every graph \(G = (V, E)\) and every \(k \in \mathbb{N}\) exactly one of following conditions is satisfied.
\begin{enumerate}
\item[(S1)] \(G\) is \(D(k)\)-degree-extreme and \(\mathcal{A}_D\) correctly decides whether \(G\) contains a \(k\)-edge induced subgraph.
\item[(S2)] \(G\) is not \(D(k)\)-degree-extreme and \(\mathcal{A}_D\) correctly outputs that \(G\) contains a \(k\)-edge induced subgraph.
\item[(S3)] \(G\) is not \(D(k)\)-degree-extreme and \(\mathcal{A}_D\) outputs three subsets \(V_1, V_2, B \subseteq V\) such that
\begin{enumerate}
\item[(S3.1)] \(V = V_1 \dot{\cup} V_2\) with \(|V_1| > D(k)\) and \(|V_2| > D(k) + 1\);
\item[(S3.2)] every edge between \(V_1\) and \(V_2\) in \(G\) has one vertex in \(B\) and \(|B| \leq (p_k - 1)^{p_k+1} + (p_k - 1)^2\).
\end{enumerate}
\end{enumerate}
Claim 2. If from Lemma 3.4. \( G \) graph, Lemma 3.6 implies the claim. Otherwise (ii) holds. And say

Proof of the claim. So,

Claim 1. Proposition 4.7 to achieve (S1). Otherwise let \( V \) and \( S \) set or a clique. If

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

\[
D(k) < \text{deg}(v_0) < |V| - 1 - D(k).
\]

(3)

Then we set \( V_1 := N(v_0) \) and \( V_2 := V \setminus V_1 \). By (3) it holds that \(|V_1| > D(k)\) and \(|V_2| = |V| - |V_1| = |V| - \text{deg}(v_0) > D(k) + 1\), i.e., (S3.1). Let

\[
W_1 := \left\{ u \in V_1 \mid |N(u) \cap V_2| \geq p_k \right\} \quad \text{and} \quad W_2 := V_1 \setminus W_1.
\]

Figure 1 illustrates our construction.

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

Proof of the claim. We apply Lemma 3.4 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 \( p_k > \mathcal{R}_k \), 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 \( G[S \cup \{v_0\}] \) 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 3.4 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.

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

Proof of the claim. It is easy to verify that we can apply Lemma 3.4 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 $|N(u) \cap V_2| < p_k$, therefore (ii) must hold. Recall $p_k > \mathcal{R}_k$, hence $G[N[u_1, \ldots, u_{p_k}]]$ 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 $G[N(V_0, v_1, \ldots, v_k)]$ 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 5.1.

Let $$B := W_1 \cup (N(W_2) \cap V_2),$$ i.e., the grey area in Figure 1. 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 $(V_1, V_2, B)$.

Finally we are ready to present our fpt-algorithm for $p$-EDGE-INDUCED-SUBGRAPH.

**Theorem 5.2.** $p$-EDGE-INDUCED-SUBGRAPH is fixed-parameter tractable.

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

$$D_0(k) := 2 \cdot ((p_k - 1)^{p_k+1} + (p_k - 1)^2) + 2^{2^{((k-1)^2+1)}}. \quad (4)$$

Note $2^{2^{((k-1)^2+1)}} > \mathcal{R}_{(k-1)^2+1}$. Then let $\mathcal{A}_{D_0}$ 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 $p$-EDGE-INDUCED-SUBGRAPH. First, we remove all the isolated vertices in $G$. For simplicity, the resulting graph is denoted by $G$ again. Then, we simulate the algorithm $\mathcal{A}_{D_0}$ on $(G, k)$. If the result is either (S1) or (S2) in Lemma 5.1, we already get the correct answer. Otherwise, $\mathcal{A}_{D_0}$ 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 an appropriate computable function of $k$. The fixed-parameter tractability of whether $G$ contains a $k$-edge induced subgraph follows from Proposition 4.14. 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 simulate the algorithm $\mathcal{A}_{D_0}$ on $(G[V_1], k)$. 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

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

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 $i \in [3]$

$$|U_i| > D_0(k) - 2 \cdot ((p_k - 1)^{p_k+1} + (p_k - 1)^2) = 2^{2^{((k-1)^2+1)}} > \mathcal{R}_{(k-1)^2+1},$$
where the equality is by \([4]\).

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. \(\square\)

**Remark 5.3.** We mentioned in the Introduction that the running time of our fpt-algorithm in terms of \(k\) is triple exponential at least. To see this, recall the function \(D_0\) as defined in \([4]\) is of the order \(2^{2^{\Theta(k)}}\). This gives the quadruple exponential lower bound for the algorithm \(k_{D_0}\) by Remark \([4.8]\). So the same lower bound applies to our algorithm for \(p\)-\(\text{EDGE-INDUCED-SUBGRAPH}\).

### 6. Counting \(k\)-edge induced subgraphs

In this section we study two counting versions of \(p\)-\(\text{EDGE-INDUCED-SUBGRAPH}\). Of course, the most natural version is:

\[
\text{\#\text{\(k\)-edge induced subgraphs}}
\]

**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 \((F, \kappa)\), where \(F : \Sigma^* \rightarrow \mathbb{N}\) and \(\kappa\) is a parameterization. \((F, \kappa)\) is fixed-parameter tractable if \(F\) can be computed by an fpt-algorithm with respect to \(\kappa\). For more background of parameterized counting complexity, the reader is referred to \([14, 20]\).

In fact, the hardness of \(p\)-\#\(\text{EDGE-INDUCED-SUBGRAPH}\) 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 \(p\)-\#\(\text{EDGE-INDUCED-SUBGRAPH}\), i.e., counting the number of 0-edge induced subgraphs is exactly the classical problem:

\[
\text{\#\text{INDEPENDENT-SET}}
\]

**Instance:** A graph \(G\).

**Problem:** Compute the number of independent sets in \(G\).

Recall that \#\(\text{INDEPENDENT-SET}\) is \#P-hard \([27, 25]\). Hence:

**Theorem 6.1.** Assume \#P \(\neq\) P. Then \(p\)-\#\(\text{EDGE-INDUCED-SUBGRAPH}\) 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 \(p\)-\(\text{EDGE-INDUCED-SUBGRAPH}\). Therefore, it also makes sense to consider:

\[
\text{\(p\)-\#\text{\(k\)-edge induced subgraphs}}
\]

**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.** \(p\)-\#\(\text{EDGE-INDUCED-SUBGRAPH}^*\) 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 $(F, \kappa)$ and $(F', \kappa')$ be two parameterized counting problems. An \textit{fpt Turing reduction} from $(F, \kappa)$ to $(F', \kappa')$ is an algorithm $A$ with an oracle to $F'$ which satisfies the following conditions:

- $A$ computes the function $F$ in fpt-time (with respect to $\kappa$).
- There is a computable function $g : \mathbb{N} \to \mathbb{N}$ such that for all oracle queries “$F'(y) = ?” posed by $A$ on input $x$ we have $\kappa'(y) \leq g(\kappa(x))$.

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

**Proof of Theorem 6.2** We give an \textit{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 \textit{nice}.

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

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

For each $i \in [2k-1]$ we call $V_i$ a \textit{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

$$G[\{(v, i) \mid i \in [k]\} \cup \{e_i \mid i \in [k-1]\}]$$

is a $(2k-2)$-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 $2k-2$ edges. If $V(H') \cap V_i \neq \emptyset$ for every $i \in [2k-1]$, i.e., $H'$ intersects all blocks $V_i$’s, then

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

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 $i \in [2k-1]$. 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 $V_{2i-1}$
and $V_{2i+1}$, 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 $V(H') \cap V_{2i-1} = \{(v_i, 2i-1)\}$. At this point, we already know that $H'$ contains the following $2 \cdot k - 2$ edges

$$\{v_1, e_1\}, \{e_1, v_2\}, \ldots, \{v_{k-1}, e_{k}\}, \{e_k, v_k\}. \quad (5)$$

We prove that $\{v_1, \ldots, v_k\}$ 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 $\{(v_i, 2i-1), (v_j, 2j-1)\}$ and hence have more than $2 \cdot k - 2$ edges by (5).

It follows that

\[
\text{(the number of independent sets of size } k \text{ in } G) \cdot k! = \text{the number of } (2 \cdot k - 2)\text{-edge nice induced subgraphs in } H \text{ which intersect every } V_i. \quad (6)
\]

Thus our goal is to compute the right hand side of (6) using $p$-$\#$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

\[
s_X := \text{the number of } (2 \cdot k - 2)\text{-edge nice induced subgraphs in } H_X,
\]

\[
t_X := \text{the number of } (2 \cdot k - 2)\text{-edge nice induced subgraphs in } H_X
\]

which intersect $V_i$ for every $i \in X$.

Therefore, the right hand side of (6) is exactly $t_{[2 \cdot k - 1]}$.

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

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

Hence, by simple dynamic programming using $p$-$\#$EDGE-INDUCED-SUBGRAPH as an oracle, we can compute every $t_X$ in fpt time. \qed

**Acknowledgement.** We thank Leizhen Cai for bringing the problem $p$-EDGE-INDUCED-SUBGRAPH to our attention, and Jörg Flum for comments on earlier versions of this paper.

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

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

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

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

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

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

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

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

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

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

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

[23] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.
[24] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V-C dimension. *Journal of Computer and System Science*, 53(2):161–170, 1996.

[25] 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.

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

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

For the reader not familiar with [16] we give a detailed proof of Theorem 4.1. Our presentation closely follows that of [15, Section 12.2]. Overall we will reduce \( p\)-Mc-Ltw\(_g\)-FO to a generalization of the parameterized independent set problem.

**Definition 6.3.** Let \( G = (V, E) \) be a graph and \( \ell, r \in \mathbb{N} \). A set \( S \subseteq V \) is \((\ell, r)\)-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 6.4.** Let \( g : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \) be a computable function. Then the following parameterized problem is fixed-parameter tractable.

| Input          | Parameter | Problem             |
|----------------|-----------|---------------------|
| \( p\)-Scattered-Set-Ltw\(_g\) | \( p + \ell + r \) | Decide whether \( S \) is \((\ell, r)\)-scattered. |

To prove this proposition we need another simple combinatorial result (for a proof see, e.g., [15, Lemma 12.12]).

**Lemma 6.5.** Let \( G = (V, E) \) be a connected graph and \( S \subseteq V \) a dominating set\(^2\) 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 6.4. By Courcelle’s Theorem [11] it is easy to see that the problem

| Input          | Parameter | Problem             |
|----------------|-----------|---------------------|
| \( p\)-Scattered-Set-Tw | \( \text{tw}(G) + \ell + r \) | Decide whether \( S \) is \((\ell, r)\)-scattered. |

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

\[ |T| < \ell. \tag{7} \]

**Claim 1.** \( S \subseteq N^G_T \left( := \{ v \in V \mid d^G(u, v) \leq r \text{ for some } u \in T \} \right) \).

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

**Claim 2.** \( S \) is \((\ell, r)\)-scattered in \( G \) if and only if \( S \) is \((\ell, r)\)-scattered in \( N^G_{2r}(T) \left( := G[N^G_{2r}(T)] \right) \).

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

\[ d^{N^G_{2r}(T)}(v_i, v_j) > r \quad \tag{8} \]

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 \( v_j \)

\(^2\)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 \( \{u, v\} \in E(G) \).
satisfies \( d^G(u, v_i) \leq r \), and hence, \( u \in N^G_r(S) := \{ v \in V \mid d^G(u, v) \leq r \text{ for some vertex } u \in S \} \). Then by Claim 1, \( u \in N^G_r(T) \). As a consequence \( d^{N^G_r}(v_i, v_j) \leq r \), which contradicts \( (\text{8}) \). 

Claim 2 shows that the mapping

\[
R(G, S, p, \ell, r) := (N^G_r(T), S, \ell, r)
\]

is a correct reduction from \( p\text{-SCATTERED-SET-LTW}_g \) to \( p\text{-SCATTERED-SET-Tw} \). It remains to show \( R \) is an fpt-reduction. To that end, we need to bound \( \text{tw} (N^G_r(T)) + \ell + r \) in terms of \( p + \ell + r \).

**Claim 3.** \( \text{tw} (N^G_r(T)) \leq g(2 \cdot r \cdot (3 \cdot \ell - 4), p) \).

**Proof of the claim.** Let \( H \) be a graph with

\[
V(H) := N^G_r(T) \quad \text{and} \quad E(H) := \{ \{u, v\} \mid u, v \in V(H), u \neq v \text{ and } d^G(u, v) \leq 2 \cdot r \}.
\]

It is then easy to verify that \( T \) is a dominating set in \( H \). Hence by Lemma 6.5, every connected component of \( H \) has diameter at most \( 3 \cdot |T| - 1 \leq 3 \cdot \ell - 4 \) by \( (\text{9}) \). It follows that every connected component \( C \) of \( N^G_r(T) \) has diameter at most \( 2 \cdot r \cdot (3 \cdot \ell - 4) \). This implies that \( C = N^G_{r,(3 \cdot \ell - 4)}(v) \) for every \( v \in C \). Recall that \( G \) has local tree-width bounded by \( g \) with respect to \( p \). Hence,

\[
\text{tw} (N^G_r(T)) \leq g(2 \cdot r \cdot (3 \cdot \ell - 4), p)
\]

This finishes the proof.

Now we recall Gaifman’s Theorem \( [17] \).

**Lemma 6.6.** Let \( \tau \) be a vocabulary and \( r \in \mathbb{N} \). Then there is an FO-formula \( \delta_r(x, y) \) such that for all \( \tau \)-structure \( A \) and all elements \( a, b \in A \) we have \( d^G(A)(a, b) \leq r \) if and only if \( A \models \delta_r(a, b) \).

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

An FO \( \tau \)-formula \( \psi(x) \) is \( r \)-local if for all \( \tau \)-structure \( A \) and \( a \in A \):

\[
A \models \psi(a) \iff \mathcal{N}^A_r(a) \models \psi(a).
\]

**Theorem 6.7 (Gaifman’s Theorem).** Every FO-sentence \( \varphi \) is equivalent to a Boolean combination of sentences of the form

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

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

Now we have all the tools for proving Theorem 6.8 which for the reader’s convenience we repeat as below:

**Theorem 6.8.** For every computable function \( g : \mathbb{N} \times \mathbb{N} \to \mathbb{N} \) the problem \( p\text{-Mc-LTW}_g\text{-FO} \) is fixed-parameter tractable.

**Proof:** Let \( (A, p, \varphi) \) be an instance of \( p\text{-Mc-LTW}_g\text{-FO} \). 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 \( \psi \)

\[
\varphi = \exists x_1 \ldots \exists x_{\ell} \left( \bigwedge_{1 \leq i < j \leq \ell} d(x_i, x_j) > 2 \cdot r \land \bigwedge_{i \in [\ell]} \psi(x_i) \right).
\]
Let $G = (V, E)$ be a graph with $V := A$ and $E := \{\{a, b\} \mid a, b \in A$ and $d^{G(A)}(a, b) = 1\}$. That is, $G$ is Gaifman’s graph of $A$. Moreover, let $S := \{a \in A \mid A \models \psi(a)\}$. By the $r$-locality of $\psi$ we have $S = \{a \in A \mid \mathcal{N}^{A}(a) \models \psi(a)\}$. Since $tw(\mathcal{N}^{A}(a)) \leq g(r, p)$, we can compute the set $S$ in fpt time, again by Courcelle’s Theorem.

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

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

Now the result follows from Proposition 6.4. \qed