Hitting forbidden induced subgraphs on bounded treewidth graphs

Ignasi Sau, Uéverton dos Santos Souza

Hitting forbidden induced subgraphs on bounded treewidth graphs

Ignasi Sau
LIRMM, Université de Montpellier, CNRS, Montpellier, France
ignasi.sau@lirmm.fr

Uéverton dos Santos Souza
Instituto de Computação, Universidade Federal Fluminense, Niterói, Brazil
ueverton@ic.uff.br

Abstract

For a fixed graph H, the H-IS-Deletion problem asks, given a graph G, for the minimum size of a set $S \subseteq V(G)$ such that $G \setminus S$ does not contain H as an induced subgraph. Motivated by previous work about hitting (topological) minors and subgraphs on bounded treewidth graphs, we are interested in determining, for a fixed graph H, the smallest function $f_H(t)$ such that H-IS-Deletion can be solved in time $f_H(t) \cdot n^{\mathcal{O}(1)}$ assuming the Exponential Time Hypothesis (ETH), where t and n denote the treewidth and the number of vertices of the input graph, respectively.

We show that $f_H(t) = 2^{\mathcal{O}(t^{h-2})}$ for every graph H on $h \geq 3$ vertices, and that $f_H(t) = 2^{\mathcal{O}(t)}$ if H is a clique or an independent set. We present a number of lower bounds by generalizing a reduction of Cygan et al. [Inf. Comput. 2017] for the subgraph version. In particular, we show that when H deviates slightly from a clique, the function $f_H(t)$ suffers a sharp jump: if H is obtained from a clique of size h by removing one edge, then $f_H(t) = 2^{\Theta(t^{h-2})}$. We also show that $f_H(t) = 2^{\Omega(t^h)}$ when $H = K_{h,h}$, and this reduction answers an open question of Mi. Pilipczuk [MFCS 2011] about the function $f_{C_4}(t)$ for the subgraph version.

Motivated by Cygan et al. [Inf. Comput. 2017], we also consider the colorful variant of the problem, where each vertex of G is colored with some color from $V(H)$ and we require to hit only induced copies of H with matching colors. In this case, we determine, under the ETH, the function $f_H(t)$ for every connected graph H on h vertices: if $h \leq 2$ the problem can be solved in polynomial time; if $h \geq 3$, $f_H(t) = 2^{\Theta(t)}$ if H is a clique, and $f_H(t) = 2^{\Theta(t^{h-2})}$ otherwise.

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

Keywords and phrases parameterized complexity, induced subgraphs, treewidth, hitting subgraphs, dynamic programming, lower bound, Exponential Time Hypothesis.

Related Version A conference version of this article appeared in the *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS), volume 170 of LIPIcs, pages 82:1–82:15, **2020***. A full version of the paper is permanently available at `https://arxiv.org/abs/2004.08324`.

Funding *Ignasi Sau*: CAPES-PRINT Institutional Internationalization Program, process 88887.371209/2019-00, and projects DEMOGRAPH (ANR-16-CE40-0028), ESIGMA (ANR-17-CE23-0010), and ELIT (ANR-20-CE48-0008-01).
Uéverton dos Santos Souza: Grant E-26/203.272/2017 Rio de Janeiro Research Foundation (FAPERJ) and Grant 303726/2017-2 National Council for Scientific and Technological Development (CNPq).

1 Introduction

Graph modification problems play a central role in modern algorithmic graph theory. In general, such a problem is determined by a target graph class $\mathcal{G}$ and some prespecified set $\mathcal{M}$ of allowed *local* modifications, and the question is, given an input graph G and an integer k, whether it is possible to transform G to a graph in $\mathcal{G}$ by applying k modification operations from $\mathcal{M}$. A wealth of graph problems can be formulated for different instantiations of $\mathcal{G}$ and $\mathcal{M}$, and applications span diverse topics such as computational biology, computer vision, machine learning, networking, and sociology [8, 11, 19].

The most studied local modification operation in the literature is vertex deletion and, among the target graph classes, of particular relevance are the ones defined by *excluding* the graphs in a family $\mathcal{F}$ according to some natural graph containment relation, such as minor, topological minor, subgraph, or induced subgraph. By the well-known classification result of Lewis and Yannakakis [25], all interesting cases of these problems are NP-hard.

One of the common strategies to cope with NP-hard problems is that of parameterized complexity [12, 17], where the core idea is to identify a parameter k associated with an input of size n that allows for an algorithm in time $f(k) \cdot n^{\mathcal{O}(1)}$, called *fixed-parameter tractable* (or FPT for short). A natural goal within parameterized algorithms is the quest for the "best possible" function $f(k)$ in an FPT algorithm. Usually, the working hypothesis to prove lower bounds is the *Exponential Time Hypothesis* (ETH) that states, in a simplified version, that the 3-Sat problem on n variables cannot be solved in time $2^{o(n)}$; see [21, 22] for more details.

Among graph parameters, one of the most successful ones is *treewidth*, which –informally speaking– quantifies the topological resemblance of a graph to a tree. The celebrated theorem due to Courcelle [10] states that every graph problem that can be expressed in Monadic Second Order logic is solvable in time $f(t) \cdot n$ on n-vertex graphs given along with a tree decomposition of width at most t. In particular, it applies to most vertex deletion problems discussed above. A very active area in parameterized complexity during the last years consists in optimizing, under the ETH, the function $f(t)$ for several classes of vertex deletion problems. As a byproduct, several cutting-edge techniques for obtaining both lower bounds [26] and algorithms [6, 14, 18] have been obtained, which have become part of the standard toolbox of parameterized complexity. Obtaining tight bounds under the ETH for this kind of vertex deletion problems is, in general, a challenging task, as we proceed to discuss.

Let H be a fixed graph and let $\prec$ be a fixed graph containment relation. In the H-$\prec$-Deletion (meta)problem, given an n-vertex graph G, the objective is to find a set $S \subseteq V(G)$ of minimum size such that $G \setminus S$ does not contain H according to containment relation $\prec$. We parameterize the problem by the treewidth of G, denoted by t, and the objective is to find the smallest function $f_H(t)$ such that H-$\prec$-Deletion can be solved in time $f_H(t) \cdot n^{\mathcal{O}(1)}$.

The case $\prec=$ 'minor' has been object of intense study during the last years [6, 14, 16, 23, 27, 29], culminating in a tight dichotomy about the function $f_H(t)$ when H is connected [2–5].

The case $\prec=$ 'topological minor' has been also studied recently [3–5], but we are still far from obtaining a complete characterization of the function $f_H(t)$. For both minors and topological minors, so far there is no graph H such that[1] $f_H(t) = 2^{\Omega(t^c)}$ for some $c > 1$.

Recently, Cygan et al. [13] started a systematic study of the case $\prec=$ 'subgraph', which turns out to exhibit a quite different behavior from the above cases: for every integer $c \geq 1$ there is a graph H such that $f_H(t) = 2^{\Theta(t^c)}$. Cygan et al. [13] provided a general upper bound

[1] For conciseness, we use (in a non-standard way) the asymptotic notations Ω and Θ to denote conditional lower bounds under the ETH.

and some particular lower bounds on the function $f_H(t)$, but a complete characterization seems to be currently out of reach. Previously, Mi. Pilipczuk [28] had studied the cases where H is a cycle, finding the function $f_{C_i}(t)$ for every $i \geq 3$ except for $i = 4$.

In this article we focus on the case $\prec=$ 'induced subgraph' that, to the best of our knowledge, had not been studied before in the literature, except for the case $K_{1,3}$, for which Bonomo-Braberman et al. [9] showed very recently that $f_{K_{1,3}}(t) = 2^{\mathcal{O}(t^2)}$.

Our results and techniques. We first show (Theorem 2) that, for every graph H on $h \geq 3$ vertices, $f_H(t) = 2^{\mathcal{O}(t^{h-2})}$. The algorithm uses standard dynamic programming over a nice tree decomposition of the input graph. However, in order to achieve the claimed running time, we need to use a slightly non-trivial encoding in the tables that generalizes an idea of Bonomo-Braberman et al. [9], by introducing an object that we call *rooted H-folio*, inspired by similar encodings in the context of graph minors [1, 3].

It turns out that when H is a clique or an independent set (in particular, when $|V(H)| \leq 2$), the problem can be solved in *single-exponential* time, that is, $f_H(t) = 2^{\mathcal{O}(t)}$. The case of cliques (Theorem 5), which coincides with the subgraph version, had been already observed by Cygan et al. [13], using essentially the folklore fact that every clique is contained in some bag of a tree decomposition. The case of independent sets (Theorem 8) is more interesting, as we exploit tree decompositions in a novel way, by showing (Lemma 6) that a chordal completion of the complement of a solution can be covered by a constant number of cliques, which implies (Lemma 7) that the complement of a solution is contained in a constant number of bags of the given tree decomposition.

Our main technical contribution consists of lower bounds. Somehow surprisingly, we show (Theorem 10) that when H deviates slightly from a clique, the function $f_H(t)$ suffers a sharp jump: if H is obtained from a clique of size h by removing one edge, then $f_H(t) = 2^{\Omega(t^{h-2})}$, and this bound is tight by Theorem 2. We also provide lower bounds for other graphs H that are "close" to cliques (Theorems 11, 13, and 14), some of them being (almost) tight. In particular, we show (Theorem 14) that when $H = K_{h,h}$, we have that $f_H(t) = 2^{\Omega(t^h)}$. By observing that the proof of the latter lower bound also applies to occurrences of $K_{h,h}$ as a *subgraph*, the particular case $h = 2$ (Corollary 15) answers the open question of Mi. Pilipczuk [28] about the function $f_{C_4}(t)$. All these reductions are inspired by a reduction of Cygan et al. [13] for the subgraph version. We first present the general frame of the reduction together with some properties that the eventual instances constructed for each of the graphs H have to satisfy, yielding in a unified way (Lemma 9) lower bounds for the corresponding problems.

Motivated by the work of Cygan et al. [13], we also consider the *colorful* variant of the problem, where the input graph G comes equipped with a coloring $\sigma : V(G) \to V(H)$ and we are only interested in hitting induced subgraphs of G isomorphic to H such that their colors match. In this case, we first observe that essentially the same dynamic programming algorithm of the non-colored version (Theorem 3) yields the upper bound $f_H(t) = 2^{\mathcal{O}(t^{h-2})}$ for every graph H on $h \geq 3$ vertices. Again, our main contribution concerns lower bounds: we show (Theorem 16), by modifying appropriately the frame introduced for the non-colored version, that $f_H(t) = 2^{\Omega(t^{h-2})}$ for every graph H having a connected component on h vertices that is not a clique. Since the case where H is a clique can also be easily solved in single-exponential time (Theorem 5), which can be shown (Theorem 18) to be optimal, it follows that if H is a connected graph on $h \geq 3$ vertices, $f_H(t) = 2^{\Theta(t)}$ if H is a clique, and $f_H(t) = 2^{\Theta(t^{h-2})}$ otherwise. It is easy to see that the cases where $|V(H)| \leq 2$ can be solved in polynomial time by computing a minimum vertex cover in a bipartite graph.

Organization. In Section 2 we provide some basic preliminaries and formally define the problems. In Section 3 we present the algorithms for both problems, and in Section 4 (resp. Section 5) we provide the lower bounds for the non-colored (resp. colored) version. Finally, we conclude the article in Section 6 with some open questions.

2 Preliminaries

Graphs and functions. We use standard graph-theoretic notation, and we refer the reader to [15] for any undefined notation. We will only consider undirected graphs without loops nor multiple edges, and we denote an edge between two vertices u and v by $\{u, v\}$. A subgraph H of a graph G is *induced* if H can be obtained from G by deleting vertices. A graph G is *H-free* if it does not contain any induced subgraph isomorphic to H. For a graph G and a set $S \subseteq V(G)$, we use the notation $G \setminus S := G[V(G) \setminus S]$. A vertex v is *complete* (resp. *anticomplete*) to a set $S \subseteq V(G)$ if v is adjacent (resp. not adjacent) to every vertex in S. A vertex set S of a connected graph G is a *separator* if $G \setminus S$ is disconnected.

We denote by $\Delta(G)$ (resp. $\omega(G)$) the maximum vertex degree (resp. clique size) of a graph G. For an integer $h \geq 1$, we denote by P_h (resp. I_h, K_h) the path (resp. independent set, clique) on h vertices, and by $K_h - e$ the graph obtained from K_h by deleting one edge. For two integers $a, b \geq 1$, we denote by $K_{a,b}$ the bipartite graph with parts of sizes a and b. We denote the disjoint union of two graphs G_1 and G_2 by $G_1 + G_2$.

A graph property is *hereditary* if whenever it holds for a graph G, it holds for all its induced subgraphs as well. The *open* (resp. *closed*) *neighborhood* of a vertex v is denoted by $N(v)$ (resp. $N[v]$). A vertex is *simplicial* if its (open or closed) neighborhood induces a clique. A graph G is *chordal* if it does not contain induced cycles of length at least four or, equivalently, if $V(G)$ can be ordered $v_1, \ldots, v_n$ such that, for every $2 \leq i \leq n$, vertex v_i is simplicial in the subgraph of G induced by $\{v_1, \ldots, v_{i-1}\}$. Note that being chordal is a hereditary property.

Given a function $f : A \to B$ between two sets A and B and a subset $A' \subseteq A$, we denote by $f|_{A'}$ the restriction of f to A' and by $\mathsf{im}(f)$ the image of f, that is, $\mathsf{im}(f) = \{b \in B \mid \exists a \in A : f(a) = b\}$. For an integer $k \geq 1$, we let $[k]$ be the set containing all integers i with $1 \leq i \leq k$.

Definition of the problems. Before formally defining the problems considered in this article, we introduce some terminology, mostly taken from [13]. Given a graph H, an *H-coloring* of a graph G is a function $\sigma : V(G) \to V(H)$. A *homomorphism* (resp. *induced homomorphism*) from a graph H to a graph G is a function $\pi : V(H) \to V(G)$ such that $\{u, v\} \in E(H)$ implies (resp. if and only if) $\{\pi(u), \pi(v)\} \in E(G)$. When G is H-colored by a function σ, an *(induced) σ-homomorphism* from H to G is an (induced) homomorphism π from H to G with the additional property that every vertex is mapped to the appropriate color, that is, $\sigma(\pi(a)) = a$ for every vertex $a \in V(H)$. An *(induced) H-subgraph* of G is an (induced) injective homomorphism from H to G and, if G is H-colored by a function σ, an *(induced) σ-H-subgraph* of G is an (induced) injective σ-homomorphism from H to G. We say that a vertex set $X \subseteq V(G)$ *hits* an (induced) σ-H-subgraph π if $X \cap \pi(V(H)) \neq \emptyset$.

For a fixed graph H, the problems we consider in this article are defined as follows.

H-IS-DELETION
Input: A graph G.
Output: The minimum size of a set $X \subseteq V(G)$ that hits all induced H-subgraphs of G.

COLORFUL H-IS-DELETION **Input:** A graph G and an H-coloring σ of G. **Output:** The minimum size of a set $X \subseteq V(G)$ that hits all induced σ-H-subgraphs of G.

The H-S-DELETION and COLORFUL H-S-DELETION problems are defined similarly, just by removing the word 'induced' from the above definitions. In the decision version of these problems, we are given a target budget k, and the objective is to decide whether there exists a hitting set of size at most k. Unless stated otherwise, we let n denote the number of vertices of input graph of the problem under consideration. When expressing the running time of an algorithm, we will sometimes use the $\mathcal{O}^*(\cdot)$ notation, which suppresses polynomial factors in the input size.

Tree decompositions. A *tree decomposition* of a graph G is a pair $\mathcal{D} = (T, \mathcal{X})$, where T is a tree and $\mathcal{X} = \{X_w \mid w \in V(T)\}$ is a collection of subsets of $V(G)$, called *bags*, such that:

- $\bigcup_{w \in V(T)} X_w = V(G)$,
- for every edge $\{u, v\} \in E$, there is a $w \in V(T)$ such that $\{u, v\} \subseteq X_w$, and
- for each $\{x, y, z\} \subseteq V(T)$ such that z lies on the unique path between x and y in T, $X_x \cap X_y \subseteq X_z$.

We call the vertices of T *nodes* of $\mathcal{D}$ and the sets in $\mathcal{X}$ *bags* of $\mathcal{D}$. The *width* of a tree decomposition $\mathcal{D} = (T, \mathcal{X})$ is $\max_{w \in V(T)} |X_w| - 1$. The *treewidth* of a graph G, denoted by $\mathsf{tw}(G)$, is the smallest integer t such that there exists a tree decomposition of G of width at most t. We need to introduce nice tree decompositions, which will make the presentation of the algorithms much simpler.

Nice tree decompositions. Let $\mathcal{D} = (T, \mathcal{X})$ be a tree decomposition of G, r be a vertex of T, and $\mathcal{G} = \{G_w \mid w \in V(T)\}$ be a collection of subgraphs of G, indexed by the vertices of T. A triple $(\mathcal{D}, r, \mathcal{G})$ is a *nice tree decomposition* of G if the following conditions hold:

- $X_r = \emptyset$ and $G_r = G$,
- each node of $\mathcal{D}$ has at most two children in T,
- for each leaf $\ell \in V(T)$, $X_\ell = \emptyset$ and $G_\ell = (\emptyset, \emptyset)$. Such an ℓ is called a *leaf node*,
- if $w \in V(T)$ has exactly one child w', then either
  - $X_w = X_{w'} \cup \{v_{\text{in}}\}$ for some $v_{\text{in}} \notin X_{w'}$ and $G_w = G[V(G_{w'}) \cup \{v_{\text{in}}\}]$. The node w is called an *introduce node* and the vertex v_{in} is the *introduced vertex* of X_w,
  - $X_w = X_{w'} \setminus \{v_{\text{out}}\}$ for some $v_{\text{out}} \in X_{w'}$ and $G_w = G_{w'}$. The node w is called a *forget node* node and v_{out} is the *forget vertex* of X_w.
- if $w \in V(T)$ has exactly two children w_1 and w_2, then $X_w = X_{w_1} = X_{w_2}$, $E(G_{w_1}) \cap E(G_{w_2}) = E(G[X_w])$, and $G_w = (V(G_{w_1}) \cup V(G_{w_2}), E(G_{w_1}) \cup E(G_{w_2}))$. The node w is called a *join node*.

For each $w \in V(T)$, we denote by V_w the set $V(G_w)$. Given a tree decomposition, it is possible to transform it in polynomial time to a *nice* one of the same width [24]. Moreover, by Bodlaender et al. [7] we can find in time $2^{\mathcal{O}(\mathsf{tw})} \cdot n$ a tree decomposition of width $\mathcal{O}(\mathsf{tw})$ of any graph G with treewidth tw. Since the running time of our algorithms dominates this function, we may assume that a nice tree decomposition of width $t = \mathcal{O}(\mathsf{tw})$ is given along with the input.

Exponential Time Hypothesis. The *Exponential Time Hypothesis* (ETH) of Impagliazzo and Paturi [21] implies that the 3-SAT problem on n variables cannot be solved in time $2^{o(n)}$. The Sparsification Lemma of Impagliazzo et al. [22] implies that if the ETH holds, then there is no algorithm solving a 3-SAT formula with n variables and m clauses in time $2^{o(n+m)}$.

Using the terminology from Cygan et al. [13], a 3-Sat formula φ in conjunctive normal form is said to be *clean* if each variable of φ appears exactly three times, at least once positively and at least once negatively, and each clause of φ contains two or three literals and does not contain twice the same variable. Cygan et al. [13] observed the following useful lemma.

▶ **Lemma 1** (Cygan et al. [13]). *The existence of an algorithm in time $2^{o(n)}$ deciding whether a clean 3-Sat formula with n variables is satisfiable would violate the* ETH.

3 Algorithms

In this section we present algorithms for H-IS-Deletion and Colorful H-IS-Deletion. We start in Subsection 3.1 with a general dynamic programming algorithm that solves H-IS-Deletion and Colorful H-IS-Deletion in time $\mathcal{O}^*(2^{\mathcal{O}(t^{h-2})})$ for any graph H on at least $h \geq 3$ vertices. In Subsection 3.2 we focus on hitting cliques and independent sets.

3.1 A general dynamic programming algorithm

We present the algorithm for H-IS-Deletion, and then we discuss that essentially the same algorithm applies to Colorful H-IS-Deletion as well. Our algorithm to solve H-IS-Deletion in time $\mathcal{O}^*(2^{\mathcal{O}(t^{h-2})})$ uses standard dynamic programming over a nice tree decomposition of the input graph; we refer the reader to [12] for a nice exposition. However, in order to achieve the claimed running time, we need to use a slightly non-trivial encoding in the tables, which we proceed to explain.

Let $|V(H)| = h$ and assume that we are given a nice tree decomposition of the input graph G such that its bags contain at most t vertices (in a tree decomposition of width t, the bags have size at most $t+1$, but to simplify the exposition we assume that they have size at most t, which does not change the asymptotic complexity of the algorithm).

Intuitively, our algorithm proceeds as follows. At each bag X_w of the nice tree decomposition of G, a state is indexed by the intersection of the desired hitting set constructed so far with the bag, and the collection of *proper* subgraphs of H that occur as induced subgraphs in the graph obtained from G_w after removing the current solution. In order to be able to proceed with the dynamic programming routine while keeping the complement of the hitting set H-free, we need to remember how these proper subgraphs of H intersect with X_w, and this is the most expensive part of the algorithm in terms of running time. We encode this collection of rooted subgraphs (where the "roots" correspond to the vertices in X_w) of H with an object $\mathcal{H}_w$ that we call a *rooted H-folio*, inspired by similar encodings in the context of graph minors [1,3]. Since we need to remember proper subgraphs of H on at most $h-1$ vertices, and we have up to t choices to root each of their vertices in the bag X_w, the number of rooted proper subgraphs of H is at most t^{h-1}. Therefore, the number of rooted H-folios, each corresponding to a collection of rooted proper subgraphs of H, is bounded from above by $2^{t^{h-1}}$. This encoding naturally leads to a dynamic programming algorithm to solve H-IS-Deletion in time $\mathcal{O}^*(2^{\mathcal{O}(t^{h-1})})$, where the hidden constants (but not the degree of the polynomial in n) may depend on H.

In order to further reduce the exponent to $h-2$, we use the following trick inspired by the dynamic programming algorithm of Bonomo-Braberman et al. [9] to solve $K_{1,3}$-IS-Deletion in time $\mathcal{O}^*(2^{\mathcal{O}(t^2)})$. The crucial observation is the following: the existence of proper induced subgraphs of H that are *fully* contained in the current bag X_w can be checked *locally* within that bag, without needing to root their vertices. That is, we distinguish these *local occurrences* of proper induced subgraphs of H, and we encode them separately in $\mathcal{H}_w$, without rooting their vertices in X_w. Note that the number of choices for those local occurrences depends only on H. In particular, since the proper subgraphs of H have at most $h-1$ vertices, the previous observation implies that we never need to root exactly $h-1$ vertices of an induced subgraph of H, since such occurrences would be fully contained in X_w. This permits to improve the running time to $\mathcal{O}^*(2^{\mathcal{O}(t^{h-2})})$. The details follow.

Note that we may assume that H has at least three vertices, as otherwise it is a clique or an independent set, and then H-IS-DELETION can be solved in single-exponential time by the algorithms in Subsection 3.2.

▶ **Theorem 2.** *For every graph H on $h \geq 3$ vertices, the H-IS-DELETION problem can be solved in time $2^{\mathcal{O}(t^{h-2})} \cdot n$, where n and t are the number of vertices and the treewidth of the input graph, respectively.*

Proof. As discussed in Section 2, we may assume that we are given are *nice* tree decomposition $(\mathcal{D}, r, \mathcal{G})$ of G of width $\mathcal{O}(\mathsf{tw}(G))$. To simplify the exposition, suppose that the bags of $\mathcal{D}$ contain at most t vertices, and recall that this assumption does not change the asymptotic complexity of the algorithm.

At each bag X_w of the given nice tree decomposition of G, a valid state of our dynamic programming table is indexed by the following two objects:

- A subset $\hat{S}_w \subseteq X_w$ that corresponds to the intersection of the desired H-hitting set with the current bag. With this in mind, we say that a set $S_w \subseteq V_w$ is *feasible for* $\hat{S}_w$ if $G_w \setminus S_w$ is H-free and $S_w \cap X_w = \hat{S}_w$.
- A rooted H-folio $\mathcal{H}_w$, containing the following two collections of (rooted) proper induced subgraphs of H:
  - The set $\mathcal{L}_w$ of *local* occurrences of proper induced subgraphs of H, consisting of the set of proper induced subgraphs of H that occur in $G[X_w \setminus \hat{S}_w]$.
  - The set $\mathcal{R}_w$ of *rooted* occurrences of proper induced subgraphs of H, consisting of a set of triples $(\tilde{H}, R, \rho)$ where $\tilde{H}$ is a proper induced subgraph of H, $R \subseteq V(\tilde{H})$ is a set with $0 \leq |R| \leq h-2$ corresponding to the *roots* of $\tilde{H}$ in $X_w \setminus \hat{S}_w$, and $\rho : R \to X_w \setminus \hat{S}_w$ is an injective function that maps each vertex in R to its corresponding vertex in $X_w \setminus \hat{S}_w$.

Note that the number of local occurrences of proper induced subgraphs of H depends only on H, and that the number of tuples $(\tilde{H}, R, \rho)$ of rooted occurrences is at most $2^h \cdot 2^{h-1} \cdot t^{h-2}$, and therefore the number of rooted H-folios is at most $2^{\mathcal{O}(t^{h-2})}$, as desired.

We say that the rooted H-folio of a subgraph $G'_w \subseteq G_w$ is $\mathcal{H}_w$ if the local and rooted occurrences of induced subgraphs of H in G'_w correspond exactly to the collections $\mathcal{L}_w$ and $\mathcal{R}_w$ of $\mathcal{H}_w$, respectively. Our algorithm stores, for each state $(\hat{S}_w, \mathcal{H}_w)$ of a node w of a nice tree decomposition of G, the minimum size of a set $S_w \subseteq V_w$ feasible for $\hat{S}_w$ such that the rooted H-folio of $G_w \setminus S_w$ is $\mathcal{H}_w$, or $+\infty$ if such a set does not exist. We denote this value by $\mathsf{opt}(\hat{S}_w, \mathcal{H}_w)$.

When r is the root of the nice tree decomposition, note that the solution of the H-IS-DELETION problem in G equals $\min_{\mathcal{H}_r}\{\mathsf{opt}(\emptyset, \mathcal{H}_r)\}$, where $\mathcal{H}_r = (\mathcal{L}_r, \mathcal{R}_r)$ runs over all rooted H-folios such that $\mathcal{L}_r = \emptyset$ and $\mathcal{R}_r$ contains the triples $(\tilde{H}, \emptyset, \emptyset)$ for all proper induced subgraphs $\tilde{H}$ of H.

We now show how these valid states and associated values can be computed recursively in a typical bottom-up fashion starting from the leaves, by distinguishing the distinct types of nodes in a nice tree decomposition. We let $V(H) = \{z_1, \ldots, z_h\}$.

Leaf node. The unique valid state is $(\emptyset, \emptyset)$ and $\mathsf{opt}(\emptyset, \emptyset) = 0$.

Introduce node. Let w be an introduce node with child w' such that $X_w \setminus X_{w'} = \{v\}$. For each valid state $(\hat{S}_{w'}, \mathcal{H}_{w'})$ for w', with $\mathcal{H}_{w'} = (\mathcal{L}_{w'}, \mathcal{R}_{w'})$, we generate the following valid states for w, depending on whether v is included in the current partial hitting set in X_w or not:

- $(\hat{S}_{w'} \cup \{v\}, \mathcal{H}_{w'})$. In this case, we just include v into the partial hitting set, hence the rooted H-folio remains the same. Therefore, $\mathsf{opt}(\hat{S}_{w'} \cup \{v\}, \mathcal{H}_{w'}) = \mathsf{opt}(\hat{S}_{w'}, \mathcal{H}_{w'}) + 1$.
- $(\hat{S}_{w'}, \mathcal{H}_w)$ only if $G[X_w \setminus \hat{S}_{w'}]$ is H-free, where $\mathcal{H}_w = (\mathcal{L}_w, \mathcal{R}_w)$ is defined as follows:
  - $\mathcal{L}_w$ contains all the proper induced subgraphs of H that occur in $G[X_w \setminus \hat{S}_{w'}]$. Note that this set can be computed in time $\mathcal{O}(t^h)$, where the hidden constant depends on h.
  - In order to define the set of triples contained in $\mathcal{R}_w$, we first check that H does not occur when introducing v: if for some triple $(\tilde{H}', R', \rho') \in \mathcal{R}_{w'}$, the graph obtained from $\tilde{H}'$ by adding a new vertex u and an edge $\{u, z\}$ for every root vertex $z \in R'$ such that $\rho'(z)$ is adjacent to v in G, is isomorphic to H, we discard the state $(\hat{S}_{w'}, \mathcal{H}_w)$ from the table of X_w, and we move on to the next state for w'. If there is no such triple, we add the whole collection $\mathcal{R}_{w'}$ to $\mathcal{R}_w$. Moreover, we add to $\mathcal{R}_w$ every triple $(\tilde{H}, R, \rho)$ that can be obtained from a triple $(\tilde{H}', R', \rho') \in \mathcal{R}_{w'}$ with $|V(\tilde{H}')| \leq h - 2$, $|R'| \leq h - 3$, $v \in \mathsf{im}(\rho)$, $|R| = |R'| + 1$, $\rho|_{R'} = \rho'$, $\tilde{H}$ is a proper induced subgraph of H, and $\tilde{H}'$ is isomorphic to $\tilde{H} \setminus \{\rho^{-1}(v)\}$. That is, since in this case v does not belong to the partial hitting set, we also add to $\mathcal{R}_w$ any rooted occurrence that can be obtained from a rooted occurrence in $\mathcal{R}_{w'}$ by adding vertex v to the set of roots R to form a larger $\tilde{H}$.

  In this case we set $\mathsf{opt}(\hat{S}_{w'}, \mathcal{H}_w) = \mathsf{opt}(\hat{S}_{w'}, \mathcal{H}_{w'})$.

Forget node. Let w be a forget node with child w' such that $X_{w'} \setminus X_w = \{v\}$. For each valid state $(\hat{S}_{w'}, \mathcal{H}_{w'})$ for w', with $\mathcal{H}_{w'} = (\mathcal{L}_{w'}, \mathcal{R}_{w'})$, we generate the following valid states for w, depending on whether $v \in \hat{S}_{w'}$ or not:

- If $v \in \hat{S}_{w'}$ we add the state $(\hat{S}_{w'} \setminus \{v\}, \mathcal{H}_{w'})$ and we set $\mathsf{opt}(\hat{S}_{w'} \setminus \{v\}, \mathcal{H}_{w'}) = \mathsf{opt}(\hat{S}_{w'}, \mathcal{H}_{w'})$. In this case, we just forget vertex v, which was in the solution, and $\mathcal{H}_{w'}$ remains the same.
- Otherwise, if $v \notin \hat{S}_{w'}$, we add the state $(\hat{S}_{w'}, \mathcal{H}_w)$ where $\mathcal{H}_w = (\mathcal{L}_w, \mathcal{R}_w)$ is defined as follows:
  - $\mathcal{L}_w$ contains all the proper induced subgraphs of H that occur in $G[X_w \setminus \hat{S}_{w'}]$. Again, this set can be computed locally in time $\mathcal{O}(t^h)$.
  - $\mathcal{R}_w$ contains every triple $(\tilde{H}, R, \rho)$ that can be constructed by any of the following two operations:
    * If there is a local occurrence $\tilde{H}' \in \mathcal{L}_{w'}$ such that v belongs to an induced $\tilde{H}'$-subgraph F of $G[X_{w'} \setminus \hat{S}_{w'}]$, we add to $\mathcal{R}_w$ the triple $(\tilde{H}, R, \rho)$ defined as $\tilde{H} = \tilde{H}'$, $R = V(\tilde{H}') \setminus \{z\}$ where z is the vertex of $\tilde{H}'$ mapped to v, and ρ mapping every vertex of R to their image in F. That is, if v was part of a local occurrence for node w', now this occurrence becomes a rooted one for node w, defined in the natural way.
    * Let $(\tilde{H}', R', \rho') \in \mathcal{R}_{w'}$ be a rooted occurrence in $\mathcal{R}_{w'}$. We distinguish two cases:
      - If $v \notin \mathsf{im}(\rho')$, we add $(\tilde{H}', R', \rho')$ to $\mathcal{R}_w$.
      - Otherwise, if $v \in \mathsf{im}(\rho')$, we add to to $\mathcal{R}_w$ the triple $(\tilde{H}, R, \rho)$ defined as $\tilde{H} = \tilde{H}'$, $R = R' \setminus \{\rho'^{-1}(v)\}$, and $\rho = \rho'|_R$. That is, we just remove the forgotten vertex v from the root set of the corresponding rooted occurrence.

In this case we set $\mathsf{opt}(\hat{S}_{w'}, \mathcal{H}_w) = \mathsf{opt}(\hat{S}_{w'}, \mathcal{H}_{w'})$.

Join node. Let w be a join node with children w_1 and w_2. For each pair of valid states $(\hat{S}_{w_1}, \mathcal{H}_{w_1})$ and $(\hat{S}_{w_2}, \mathcal{H}_{w_2})$ for w_1 and w_2, with $\mathcal{H}_{w_1} = (\mathcal{L}_{w_1}, \mathcal{R}_{w_1})$ and $\mathcal{H}_{w_2} = (\mathcal{L}_{w_2}, \mathcal{R}_{w_2})$, respectively, such that $\hat{S}_{w_1} = \hat{S}_{w_2}$ and $\mathcal{L}_{w_1} = \mathcal{L}_{w_2}$, we generate the valid state $(\hat{S}_{w_1}, \mathcal{H}_w)$ for w, where $\mathcal{H}_w = (\mathcal{L}_{w_1}, \mathcal{R}_w)$ and $\mathcal{R}_w$ is defined as follows. For every pair of rooted triples $(\tilde{H}_1, R_1, \rho_1) \in \mathcal{R}_{w_1}$ and $(\tilde{H}_2, R_2, \rho_2) \in \mathcal{R}_{w_2}$ with $R_1 = R_2$, $\rho_1 = \rho_2$, and $(V(\tilde{H}_1) \setminus R_1) \cap (V(\tilde{H}_2) \setminus R_2) = \emptyset$ (recall that the vertices of H are labeled, so this condition is well-defined), we add to $\mathcal{R}_w$ the triple $(\tilde{H}_1 \cup \tilde{H}_2, R_1, \rho_1)$. That is, we just merge the rooted triples that coincide in $X_w = X_{w_1} = X_{w_2}$, by taking the union of the corresponding subgraphs of H.

Finally, for $(\hat{S}_{w_1}, \mathcal{H}_w)$ to be indeed a valid state for w, we have to check that an occurrence of H has not been created in those triples: if for some such a triple $(\tilde{H}_1 \cup \tilde{H}_2, R_1, \rho_1) \in \mathcal{R}_w$, we have that $\tilde{H}_1 \cup \tilde{H}_2 = H$, we discard the state $(\hat{S}_{w_1}, \mathcal{H}_w)$ for w, and we move on to the next pair of valid states $(\hat{S}_{w_1}, \mathcal{H}_{w_1})$ and $(\hat{S}_{w_2}, \mathcal{H}_{w_2})$ for w_1 and w_2, respectively.

If the state $(\hat{S}_{w_1}, \mathcal{H}_w)$ has not been discarded, we set

$$\mathsf{opt}(\hat{S}_{w_1}, \mathcal{H}_w) = \mathsf{opt}(\hat{S}_{w_1}, \mathcal{H}_{w_1}) + \mathsf{opt}(\hat{S}_{w_2}, \mathcal{H}_{w_2}) - |\hat{S}_{w_1}|.$$

In all cases, if distinct valid states of the child(ren) node(s) generate the same valid state $(\hat{S}_w, \mathcal{H}_w)$ at the current node w, we update $\mathsf{opt}(\hat{S}_w, \mathcal{H}_w)$ to be the minimum among all the obtained values, as usual. This concludes the description of the algorithm, whose correctness follows from the definition of the tables and the fact that the solution of the H-IS-Deletion problem on G is computed at the root of the nice tree decomposition. Clearly, all the above operations can be performed at each node in time $2^{\mathcal{O}(t^{h-2})}$, and the proof is complete by taking into account that we may assume that the given nice tree decomposition has $\mathcal{O}(t \cdot n)$ nodes [24]. ◀

A dynamic programming algorithm similar to the one provided in Theorem 2 can also solve the Colorful H-IS-Deletion problem in time $2^{\mathcal{O}(t^{h-2})} \cdot n$ for every graph H on $h \geq 3$ vertices. Indeed, the algorithm remains basically the same, except that we have to keep track only of *colorful* copies of proper subgraphs of H, and to discard only the states in which a *colorful* occurrence of H appears. In order to do that, in the tables of the dynamic programming algorithm we just need to replace rooted H-folios by *rooted σ-H-folios*, defined in the natural way. Since the number of further computations at each node in order to verify that the colors match in the obtained rooted subgraphs of H is a function dominated by $2^{\mathcal{O}(t^{h-2})}$, we obtain the same asymptotic running time. We omit the details.

▶ **Theorem 3.** *For every graph H on $h \geq 3$ vertices, the* Colorful H-IS-Deletion *problem can be solved in time $2^{\mathcal{O}(t^{h-2})} \cdot n$, where n and t are the number of vertices and the treewidth of the input graph, respectively.*

It is easy to check that small adaptations of the algorithms of Theorems 2 and 3 also work for the (not necessarily induced) subgraph version of both problems. Nevertheless, the obtained running times never outperform those obtained by Cygan et al. [13] for those problems.

3.2 Hitting cliques and independent sets

The following folklore lemma follows easily from the definition of tree decomposition.

▶ **Lemma 4.** *Let G be a graph and let $\mathcal{D}$ be a tree decomposition of G. Then every clique of G is contained in some bag of $\mathcal{D}$.*

Note that if H is a clique, then the (Colorful) H-IS-Deletion problem is the same as the (Colorful) H-S-Deletion problem. Cygan et al. [13] observed that, by Lemma 4, in order to solve (Colorful) K_h-IS-Deletion it is enough to do the following: store, for every bag of a (nice) tree decomposition of the input graph, the subset of vertices of the bag that belongs to the partial hitting set, and check locally within the bag that the remaining vertices do not induce a K_h. A typical dynamic programming routine yields the following result[2].

▶ **Theorem 5** (Cygan et al. [13]). *For every integer $h \geq 1$, K_h-IS-Deletion and Colorful K_h-IS-Deletion can be solved in time $2^{\mathcal{O}(t)} \cdot n$, where n and t are the number of vertices and the treewidth of the input graph, respectively.*

The case where H is an independent set, which is NP-hard by [25], turns out to be more interesting. We proceed to present a single-exponential algorithm for I_h-IS-Deletion, and we remark that this algorithm does *not* apply to the colorful version.

Note that I_2-IS-Deletion is the dual problem of Maximum Clique, since a minimum I_2-hitting set is the complement of a maximum clique. This duality together with Lemma 4 yield the following key insight: in any graph G, after the removal of an optimal solution of I_2-IS-Deletion, all the remaining vertices are contained in a single bag of any tree decomposition of G. Our algorithm is based on a generalization of this property to any $h \geq 1$, stated in Lemma 7, which gives an alternative way to exploit tree decompositions in order to solve the H-IS-Deletion problem.

We first need a technical lemma. A *clique cover* of a graph G is a collection of cliques of G that cover $V(G)$, and its *size* is the number of cliques in the cover.

▶ **Lemma 6.** *Every I_h-free chordal graph G admits a clique cover of size at most $h-1$.*

Proof. We prove the lemma by induction on h. For $h = 2$, G itself is a clique and the claim is trivial. Suppose inductively that any I_{h-1}-free chordal graph admits a clique cover of size at most $h-2$, let G be an I_h-free chordal graph, and let v be a simplicial vertex of G. Since $N[v]$ is a clique and G is I_h-free, it follows that $G \setminus N[v]$ is I_{h-1}-free. Since being chordal is a hereditary property, $G \setminus N[v]$ is an I_{h-1}-free chordal graph, so by induction $G \setminus N[v]$ admits a clique cover of size at most $h-2$. These $h-2$ cliques together with $N[v]$ define a clique cover of G of size at most $h-1$. ◀

▶ **Lemma 7.** *Let $h \geq 2$ be an integer, let G be a graph, let $\mathcal{D}$ be a tree decomposition of G, and let S be any solution for I_h-IS-Deletion on G. Then there are at most $h-1$ bags $X_1, X_2, \ldots, X_{h-1}$ of $\mathcal{D}$ such that $V(G) \setminus S \subseteq \bigcup_{i \in [h-1]} X_i$.*

Proof. Let $\mathcal{D}$ be a tree decomposition of G, let S be a solution for I_h-IS-Deletion on G, and let $G^\star$ be the graph obtained from G by adding an edge between any pair of vertices contained in the same bag of $\mathcal{D}$. Note that $G^\star$ is a chordal graph, and that $\mathcal{D}$ is also a tree decomposition of $G^\star$. Since being a chordal graph is a hereditary property, it follows that $G^\star \setminus S$ is chordal. Since $G \setminus S$ is I_h-free, and the property of being I_h-free is closed under edge addition, we have that $G^\star \setminus S$ is also I_h-free. Thus, $G^\star \setminus S$ is an I_h-free chordal graph,

[2] In fact, Cygan et al. [13] presented an algorithm only for Colorful K_h-S-Deletion, but the algorithm for K_h-S-Deletion is just a simplified version of the colorful version, just by forgetting the colors.

and Lemma 6 implies that $G^\star \setminus S$ admits a clique cover of size at most $h-1$. Since any clique in $G^\star \setminus S$ is also a clique in $G^\star$, and $\mathcal{D}$ is a tree decomposition of $G^\star$, Lemma 4 implies that every clique of $G^\star \setminus S$ is contained in some bag of $\mathcal{D}$, and therefore there are at most $h-1$ bags of $\mathcal{D}$ that cover all vertices in $V(G^\star) \setminus S = V(G) \setminus S$. ◀

Recall that I_h-IS-Deletion is NP-hard even for $h=2$, thus the problem cannot be solved in time $n^{f(h)}$ for any function f, unless $\mathsf{P} = \mathsf{NP}$.

▶ **Theorem 8.** *For every integer $h \geq 1$, I_h-IS-Deletion can be solved in time $2^{\mathcal{O}(t)} \cdot n^h$, where n and t are the number of vertices and the treewidth of the input graph, respectively.*

Proof. For $h=1$ the problem can be trivially solved in linear time, so assume $h \geq 2$. Let $\mathcal{D}$ be a tree decomposition of G with width t, and let S be an (unknown) optimal solution for I_h-IS-Deletion on G. By Lemma 7, there are at most $h-1$ bags $X_1, X_2, \ldots, X_{h-1}$ of $\mathcal{D}$ such that $V(G) \setminus S \subseteq \bigcup_{i \in [h-1]} X_i$. Since we may assume that $\mathcal{D}$ has $\mathcal{O}(n)$ nodes [24], we can enumerate the candidate sets of bags $X_1, X_2, \ldots, X_{h-1}$ in time $\mathcal{O}(n^{h-1})$. For each such fixed set $X_1, X_2, \ldots, X_{h-1}$, we generate all subsets $\bar{S} \subseteq \bigcup_{i \in [h-1]} X_i$, which are at most $2^{(h-1)(t+1)}$ many, and for each $\bar{S}$ we check whether the graph $G[\bar{S}]$ is I_h-free, in time $2^t \cdot t^{\mathcal{O}(1)} \cdot n$, by computing a maximum independent set of $G[\bar{S}]$ using dynamic programming based on treewidth [12] (note that having treewidth at most t is a hereditary property). Note that, by Lemma 7, there exists some of the considered sets $\bar{S}$ such that $V(G) \setminus \bar{S} = S$, and therefore an optimal solution S of I_h-IS-Deletion on G can be found in time $\mathcal{O}(n^{h-1} \cdot 2^{(h-1)(t+1)} \cdot 2^t \cdot t^{\mathcal{O}(1)} \cdot n) = 2^{\mathcal{O}(t)} \cdot n^h$, as claimed. ◀

We would like to mention that the approach used in the algorithm of Theorem 8 does not seem to be easily applicable to the colorful version of the problem. Indeed, the colored version of Lemma 6 fails: removing a clique from a σ-I_h-free chordal graph does not necessarily yield a σ-I_{h-1}-free chordal graph, and the inductive argument does not apply.

To conclude this section, note that, for any graph H and any instance (G, σ) of Colorful H-IS-Deletion, any edge between two vertices u, v with $\sigma(u) = \sigma(v)$ can be safely deleted without affecting the instance. Hence, if $H = K_2$ we can assume that the input graph is bipartite, and therefore the Colorful K_2-IS-Deletion problem (where the goal is to hit all edges) is equivalent to computing a minimum vertex cover in a bipartite graph, which can be done in polynomial time. Similarly, the Colorful I_2-IS-Deletion problem can also be solved in polynomial time, by computing a minimum vertex cover in the bipartite complement of the input graph. This is in sharp contrast to the uncolored version, where both problems are NP-hard [25].

4 Lower bounds for H-IS-Deletion

In this section we present lower bounds for the H-IS-Deletion problem. Our reductions will be from the 3-Sat problem restricted to clean formulas (see Section 2 for the definition), and are strongly inspired by a reduction of Cygan et al. [13, Theorem 4] for the H-S-Deletion problem when H is the graph obtained from $K_{2,h}$ by attaching a triangle to each of the two vertices of degree h. More precisely, the reduction from the 3-Sat problem restricted to clean formulas of Cygan et al. [13, Theorem 4] is based on a frame graph that is a simplified version of the general one that we define below, but that enjoys its essential properties, namely that each occurrence of the forbidden (induced) graph H corresponds to a satisfying variable/clause pair. Our technical contribution is to enhance this basic frame graph in order to deal with different forbidden subgraphs H, in particular (as discussed in detail below) by adding edges inside the central part, redefining the "attached graphs" L, changing the adjacencies given by the functions $f_{C,\ell}$ defined below, or adding a new vertex set T into the central part (cf. Figure 1).

We start by presenting the general frame of the reductions together with some generic properties that our eventual instances of H-IS-Deletion will satisfy, which allow to prove in a unified way (cf. Lemma 9) the equivalence of the instances. Variations of this general frame will yield the concrete reductions for distinct graphs H (cf. Theorems 10, 11, 13, and 14).

General frame of the reductions. Given a clean 3-Sat formula φ with n variables and m clauses, we proceed to build a so-called *frame graph* $F_{H,\varphi}$. For each graph H considered in the reductions, $F_{H,\varphi}$ will be enhanced with additional vertices and edges, obtaining a graph $G_{H,\varphi}$ that will be the constructed instance of the H-IS-Deletion problem.

Let h be an integer depending on H, to be specified in each particular reduction, let s be the smallest positive integer such that $s^h \geq 3n$, and note that $s = \mathcal{O}(n^{1/h})$. We introduce a set of vertices $M = \{w_{i,j} \mid i \in [s], j \in [h]\}$, which we call the *central part* of the frame. One may think of this set M as a matrix with s rows and h columns. We will sometimes add an extra set T of vertices to the central part, with $|T|$ depending only on H, obtaining an *enhanced central part* $M' = M \cup T$.

Let L be a graph, to be specified according to each particular considered graph H. By *attaching a copy of L between two vertices $u, v \in V(F_{H,\varphi})$* we mean adding a new copy of L, choosing two arbitrary distinct vertices of L, and identifying them with u and v, respectively.

For each variable x of φ and for each clause C containing x in a literal $\ell \in \{x, \bar{x}\}$, we add to F_φ a new vertex $a_{x,C,\ell}$. We also introduce another "dummy" vertex a_x. Since φ is clean, we have introduced four vertices in $F_{H,\varphi}$ for each variable x. Let $a_{x,C_1,\ell}$, $a_{x,C_2,\bar{\ell}}$, $a_{x,C_3,\ell}$, a_x be the four introduced vertices (recall that x appears at least once positively and negatively in φ). We attach a copy of L between the following four pairs of vertices: $(a_{x,C_1,\ell}, a_{x,C_2,\bar{\ell}})$, $(a_{x,C_2,\bar{\ell}}, a_{x,C_3,\ell})$, $(a_{x,C_3,\ell}, a_x)$, and $(a_x, a_{x,C_1,\ell})$. We denote by A the union of all the vertices in these variable gadgets.

For each clause C of φ and for each literal ℓ in C, we add to F_φ a new vertex $b_{C,\ell}$. Since φ is clean, we have introduced two or three vertices in $F_{H,\varphi}$ for each clause C. We attach a copy of L between every pair of these vertices. We denote by B the union of all the vertices in these clause gadgets. This concludes the construction of the frame $F_{H,\varphi}$; cf. Figure 1.

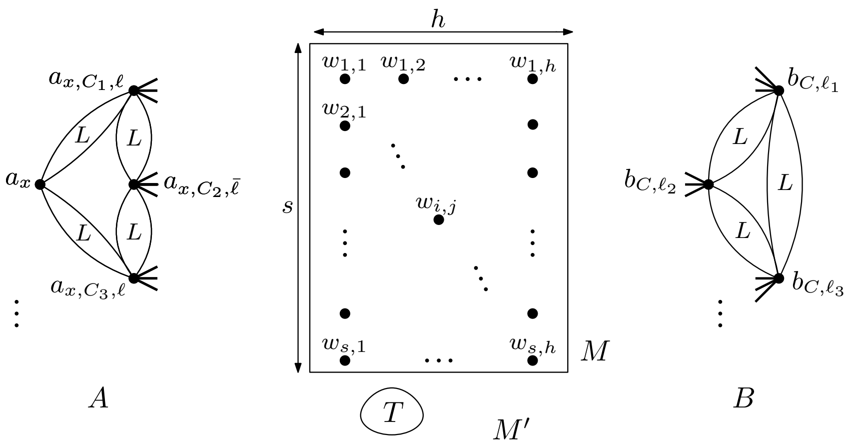

Figure 1 Illustration of the general frame graph $F_{H,\varphi}$.

In all our reductions, the graph $G_{H,\varphi}$ will satisfy the following property:

P1: All the connected components of $G_{H,\varphi} \setminus M'$ are of size bounded by a function of H.

Also, in all our reductions the budget that we set for the solution of H-IS-Deletion on $G_{H,\varphi}$ is $k := 2n + \sum_{C\in\varphi}(|C|-1) = 5n - m$, where $|C|$ denotes the number of literals in clause C. For each fixed graph H, the choice of k, the edges within M', and the edges between M' and the sets A, B will force the following behavior in $G_{H,\varphi}$:

P2: For each gadget corresponding to a variable x, at least one of the pairs $(a_{x,C_1,\ell}, a_{x,C_3,\ell})$ and $(a_x, a_{x,C_2,\bar{\ell}})$ needs to be in the solution and, for each gadget corresponding to a clause C, at least $|C|-1$ vertices in the set $\{b_{C,\ell} \mid \ell \in C\}$ need to be in the solution.

The above property together with the choice of k imply that the budget is *tight*: exactly one of the pairs $(a_{x,C_1,\ell}, a_{x,C_3,\ell})$ and $(a_x, a_{x,C_2,\bar{\ell}})$ is in the solution, thereby defining the true/false assignment of variable x; and exactly one of the vertices in $\{b_{C,\ell} \mid \ell \in C\}$ is *not* in the solution, corresponding to a satisfied literal in C. More precisely, our graph $G_{H,\varphi}$ will satisfy the following key property:

P3: Let $X \subseteq V(G_{H,\varphi})$ contain exactly one of $(a_{x,C_1,\ell}, a_{x,C_3,\ell})$ and $(a_x, a_{x,C_2,\bar{\ell}})$ for each variable x, and exactly $|C|-1$ vertices in $\{b_{C,\ell} \mid \ell \in C\}$ for each clause C. If $G_{H,\varphi} \setminus X$ contains H as an induced subgraph, then it has an occurrence of H as an induced subgraph containing exactly one vertex $a_{x,C,\ell} \in A$ and exactly one vertex $b_{C',\ell'} \in B$, with $(C,\ell) = (C',\ell')$. Moreover, each such a pair of vertices gives rise to an occurrence of H in $G_{H,\varphi} \setminus X$.

Note that property P3 states that the occurrences of H described above are "representative" of all occurrences of H, in the sense that it is enough that the set X hits these particular occurrences in order to guarantee that $G_{H,\varphi} \setminus X$ contains no occurrence of H at all. We now show that the above three properties are enough to construct the desired reductions.

▶ **Lemma 9.** *Let H be a fixed graph and, given a clean 3-Sat formula φ, let $G_{H,\varphi}$ be a graph constructed starting from the frame graph $F_{H,\varphi}$ described above, where the central part M has h columns for some constant $h \geq 1$ depending on H. If $G_{H,\varphi}$ satisfies properties* P1*,* P2*, and* P3*, then the H-IS-Deletion problem cannot be solved in time $\mathcal{O}^*(2^{o(t^h)})$ unless the* ETH *fails, where t is the width of a given tree decomposition of the input graph.*

Proof. Since H is a fixed graph, property P1 implies that we can easily construct in polynomial time a tree decomposition of $G_{H,\varphi}$ of width $\mathcal{O}(|M'|) = \mathcal{O}(|M|) = \mathcal{O}(s) = \mathcal{O}(n^{1/h})$. Let t be the width of the constructed tree decomposition of $G_{H,\varphi}$. We set $k := 5n - m$, where n and m are the number of variables and clauses of φ, respectively. We claim that φ is satisfiable if and only if $G_{H,\varphi}$ has a solution of H-IS-Deletion of size at most k. This will conclude the proof of the lemma, since an algorithm in time $\mathcal{O}^*(2^{o(t^h)})$ to solve H-IS-Deletion on $G_{H,\varphi}$ would imply the existence of an algorithm in time $2^{o(n)}$ to decide whether φ is satisfiable, which would contradict the ETH by Lemma 1.

Suppose first that α is an assignment of the variables that satisfies all the clauses in φ, and we define a set $X \subseteq V(G_{H,\varphi})$ as follows. For each variable x, add to X all vertices $a_{x,C,\ell}$ such that $\alpha(\ell)$ is true. If only one vertex was added in the previous step, add to X vertex a_x as well. For each clause X, choose a literal ℓ that satisfies C, and add to X the set $\{b_{C,\ell'} \mid \ell' \neq \ell\}$. By construction we have that $|X| = k$, and property P3 guarantees that H does not occur in $G_{H,\varphi} \setminus X$ as an induced subgraph.

Conversely, suppose that there exists $X \subseteq V(G_{H,\varphi})$ with $|X| \leq k$ such that $G_{H,\varphi} \setminus X$ does not contain H as an induced subgraph. By property P2 and the choice of k, X contains exactly one of $(a_{x,C_1,\ell}, a_{x,C_3,\ell})$ and $(a_x, a_{x,C_2,\bar{\ell}})$ for each variable x, and exactly $|C|-1$ vertices in $\{b_{C,\ell} \mid \ell \in C\}$ for each clause C. We define the following assignment α of the variables: for each variable x, let $\ell \in \{x, \bar{x}\}$ such that $a_{x,C,\ell} \in X$ for some clause C. Then we set $\alpha(x)$ to true if $\ell = x$, and to false if $\ell = \bar{x}$. By the above discussion, this is a valid assignment. Consider a clause C of φ, and let ℓ be the literal in C such that $b_{C,\ell} \notin X$. Property P3 and the hypothesis that X is a solution imply that there exists a variable $x \in \{\ell, \bar{\ell}\}$ such that $a_{x,C,\ell} \in X$, as otherwise there would be an occurrence of H in $G_{H,\varphi} \setminus X$. By the definition of α, necessarily $\alpha(\ell)$ is true, and therefore α satisfies C. Since this argument holds for every clause, we conclude that φ is satisfiable. ◀

We now proceed to describe concrete reductions for several instantiations of H. In order to add edges between the enhanced central part M' and the sets A, B, we use the following nice trick introduced in [13]. To each pair (C, ℓ), where C is a clause of φ and ℓ is a literal in C, we assign a function $f_{C,\ell} : [h] \to [s]$. Note that there are s^h many such functions, and recall that s has been chosen so that $s^h \geq 3n$. We assign these functions in such a way that $f_{C,\ell} \neq f_{C',\ell'}$ whenever $(C, \ell) \neq (C', \ell')$; note that this is possible by the choice of s and the fact that, since φ is clean, each clause contains at most three literals. We assume henceforth that these functions are fixed.

We start with the following result that provides a tight lower bound for a graph that is very "close" to a clique, namely a clique minus one edge.

▶ **Theorem 10.** *For any integer $h \geq 1$, the $(K_{h+2}-e)$-IS-Deletion problem cannot be solved in time $\mathcal{O}^*(2^{o(t^h)})$ unless the ETH fails, where t is the width of a given tree decomposition of the input graph.*

Proof. We first treat the case $h = 1$ separately, by presenting a polynomial-time reduction from the Vertex Cover problem, which is well-known not to be solvable, assuming the ETH, in time $2^{o(n+m)}$ on graphs with n vertices and m edges [21, 22]. In fact, we will prove the stronger lower bound of $\mathcal{O}^*(2^{o(n)})$, which implies the lower bound $\mathcal{O}^*(2^{o(t)})$ corresponding to the case $h = 1$ claimed in the statement of the theorem. Note that, in the case $h = 1$, $K_3 - e = P_3$. Given an instance G of Vertex Cover, let G' be obtained from G by attaching a private neighbor to every vertex of G, and note that $|V(G')| = 2|V(G)|$. It can be easily verified that the size of a minimum vertex cover of G equals the minimum size of a vertex set of G' intersecting all its induced P_3's. Hence, the $(K_3 - e)$-IS-Deletion problem cannot be solved in time $\mathcal{O}^*(2^{o(n)})$ under the ETH.

Suppose henceforth that $h \geq 2$, and let $H = K_{h'+2} - e$ for some $h' \geq 2$ (we relabel h as h' in H to keep the index h for the number of columns in the frame graph $F_{H,\varphi}$). We will present a reduction from the 3-Sat problem restricted to clean formulas. Given such a formula φ, let $F_{H,\varphi}$ be the frame graph described above the statement of the theorem, with $h = h'$, $L = K_{h+2} - e$, and $T = \emptyset$. In this construction, when attaching copies of L, we choose the attachment vertices to be two distinct vertices of L different from the endvertices of its unique non-edge. Note that this is always possible as $h \geq 2$. We proceed to build, starting from $F_{H,\varphi}$, an instance $G_{H,\varphi}$ of H-IS-Deletion with budget $k = 5n - m$ satisfying properties P1, P2, and P3, and then Lemma 9 will imply the claimed lower bound.

We add an edge between any two vertices $w_{i,j}, w_{i',j'} \in M$ with $j \neq j'$. That is, we turn $G_{H,\varphi}[M]$ into a complete h-partite graph, where each part has size s. For each clause C and each literal ℓ in C, where $\ell \in \{x, \bar{x}\}$ for some variable x, we add the edges $\{a_{C,x,\ell}, w_{f_{C,\ell}(j),j}\}$ and $\{b_{C,\ell}, w_{f_{C,\ell}(j),j}\}$ for every $j \in [h]$. That is, the function $f_{C,\ell}$ indicates the *unique* neighbor of $a_{C,x,\ell}$ and $b_{C,\ell}$ in the j-th column of M, for every $j \in [h]$. This concludes the construction of $G_{H,\varphi}$, which clearly satisfies property P1. By the choice of k and the fact that there is a copy of H between the corresponding vertices of A and B (cf. Figure 1), property P2 holds as well. Let $X \subseteq V(G_{H,\varphi})$ be a set as in property P3, and let $\tilde{H}$ be an induced subgraph of $G_{H,\varphi} \setminus X$ isomorphic to H. Since $\omega(G_{H,\varphi}[M]) = h$, $\omega(G_{H,\varphi}[(A \cup B) \setminus X]) \leq h$ (here we use that $h \geq 2$ and the choice of the attachment vertices of L), and $\omega(H) = h + 1$, $\tilde{H}$ intersects both M and $A \cup B$. Moreover, since no two adjacent vertices in $(A \cup B) \setminus X$ both have neighbors in M, necessarily $|V(\tilde{H}) \cap (A \cup B)| = 2$ and $V(\tilde{H}) \cap M$ induces a clique of size h, which implies that $\tilde{H}$ contains a vertex in each column of M. By the definition of the functions $f_{C,\ell}$ and the construction of $G_{H,\varphi}$, the two vertices in $V(\tilde{H}) \cap (A \cup B)$ must be $a_{x,C,\ell} \in A$ and $b_{C',\ell'} \in B$ with $(C, \ell) = (C', \ell')$, and therefore property P3 follows and we are done by Lemma 9. ◀

In the following result we provide an almost tight lower bound for another graph H that is also "close" to a clique, in this case a clique of size h plus two isolated vertices. Since this graph is somehow symmetric to the one considered in Theorem 10, the natural approach is to reverse the roles of neighbors and non-neighbors given by the functions $f_{C,\ell}$. However, in this way there would be many cliques of size h consisting of a vertex in $A \cup B$ together with $h - 1$ of its neighbors in M, which would create many undesired induced occurrences of H with any two vertices anticomplete to such a clique. We circumvent this problem by "reducing" the number of columns of the central part to $h - 1$, and adding a vertex s_0 to the set T that is complete to M and anticomplete to $A \cup B$. This vertex guarantees property P3, at the price of achieving only a near-optimal lower bound[3] for $H = K_h + I_2$, except for the case $h = 1$, in which the lower bound is optimal under the ETH. For technical reasons discussed in the proof of Theorem 11, in our construction we need to assume that $h \geq 4$.

▶ **Theorem 11.** *Let $h \geq 1$ be an integer. Assuming the* ETH*, the $(K_h + I_2)$-IS-Deletion problem cannot be solved in time $\mathcal{O}^*(2^{o(t)})$ if $h \leq 3$, and in time $\mathcal{O}^*(2^{o(t^{h-1})})$ if $h \geq 4$, where t is the width of a given tree decomposition of the input graph.*

Proof. We first treat the cases $h \in \{1, 2, 3\}$ separately. As in the proof of Theorem 10, we present polynomial-time reductions from the Vertex Cover problem. Again, we will prove the stronger lower bound of $\mathcal{O}^*(2^{o(n)})$, which implies the lower bound $\mathcal{O}^*(2^{o(t)})$ corresponding to the cases where $h \leq 3$ claimed in the statement of the theorem.

Let first $h = 1$, and note that $K_1 + I_2 = I_3$. We will show that, under the ETH, K_3-IS-Deletion cannot be solved in time $\mathcal{O}^*(2^{o(n)})$, which implies, by taking the complement of the input graph, that I_3-IS-Deletion cannot be solved in time $\mathcal{O}^*(2^{o(n)})$, concluding the proof. Given an instance G of Vertex Cover, let G' be obtained from G by adding, for each edge $\{u, v\} \in E(G)$, a new vertex w and two edges $\{u, w\}$ and $\{v, w\}$. Note that $|V(G')| = |V(G)| + |E(G)|$, and recall that Vertex Cover cannot be solved in time $2^{o(n+m)}$ under the ETH. It can be easily verified that the size of a minimum vertex cover of G equals the minimum size of a vertex set of G' intersecting all its K_3's. Hence, the K_3-IS-Deletion problem cannot be solved in time $\mathcal{O}^*(2^{o(n)})$ under the ETH.

Let $h = 2$. Given an instance G of Vertex Cover, let G' be obtained from G by adding $|V(G)|$ isolated vertices. It can be easily verified that the size of a minimum vertex cover of G equals the minimum size of a vertex set of G' intersecting all the induced occurrences of $K_2 + I_2$.

---

[3] In the conference version of this article presented at MFCS 2020, we claimed a tight bound of $\mathcal{O}^*(2^{o(t^h)})$ for $H = K_h + I_2$, but the proof contained a flaw that we have fixed in the full version.

Finally, let $h = 3$. Given an instance G of Vertex Cover, let G' be obtained from G by adding $|V(G)|$ isolated vertices and, for each edge $\{u, v\} \in E(G)$, a new vertex w and two edges $\{u, w\}$ and $\{v, w\}$. It can be easily verified that the size of a minimum vertex cover of G equals the minimum size of a vertex set of G' intersecting all the induced occurrences of $K_3 + I_2$.

Suppose henceforth that $h \geq 4$, and let $H = K_{h'} + I_2$ for some $h' \geq 4$ (again, we relabel h as h' in H to keep the index h for the number of columns in the frame graph $F_{H,\varphi}$). We will present a reduction from the 3-Sat problem restricted to clean formulas. Given such a formula φ, let $F_{H,\varphi}$ be the frame graph described above the statement of the theorem, with $h = h' - 1$, $L = K_{h'}$, and $T = \{s_0\}$ where s_0 is a new vertex. We proceed to build, starting from $F_{H,\varphi}$, an instance $G_{H,\varphi}$ of H-IS-Deletion with budget $k = 5n - m$ satisfying properties P1, P2, and P3, and then Lemma 9 will imply the claimed lower bound.

We add an edge between any two vertices $w_{i,j}, w_{i',j'} \in M$ with $j \neq j'$. That is, we turn $G_{H,\varphi}[M]$ into a complete h-partite graph, where each part has size s. For each clause C and each literal ℓ in C, where $\ell \in \{x, \bar{x}\}$ for some variable x, we add the edges $\{a_{C,x,\ell}, w_{i,j}\}$ and $\{b_{C,\ell}, w_{i,j}\}$ for every $j \in [h]$ and every $i \in [h] \setminus \{f_{C,\ell}(j)\}$. That is, in this case, the function $f_{C,\ell}$ indicates the *unique* non-neighbor of $a_{C,x,\ell}$ and $b_{C,\ell}$ in the j-th column of M, for every $j \in [h]$. We make vertex s_0 complete to M and anticomplete to $A \cup B$. Finally, for every copy of L that we have attached in $G_{H,\varphi}$, let $v_1, \ldots, v_{h'-2}$ be the vertices of L distinct from the two attachment vertices, ordered arbitrarily. For $j \in [h' - 2]$, we make v_j complete to the j-th column of M (note that the last column of M is not used). We add these edges for two reasons. The first one is to prevent that the non-attachment vertices of L may play the role of the two isolated vertices in a potential occurrence of $K_{h'} + I_2$. The second one is to prevent that the non-attachment vertices of L may participate in a clique of size h' in an occurrence of $K_{h'} + I_2$. Here is where the hypothesis that $h' \geq 4$ is important. Indeed, since $h' \geq 4$, each copy of L contains at least two non-attachment vertices, and the fact that each such vertex is adjacent to a *distinct* column of M implies that, together with one of the "surviving" attachment vertices and some vertices of M, these non-attachment vertices cannot participate in a clique of size h' (for this, we also use that each column of M induces an independent set). This concludes the construction of $G_{H,\varphi}$, which clearly satisfies property P1, since the vertices in the variable and clauses gadgets have neighbors only in M and within those gadgets.

Let us now argue that $G_{H,\varphi}$ satisfies property P2. Assume for contradiction that there exists a hitting set X of size at most k and a variable x such that none of the pairs $(a_{x,C_1,\ell}, a_{x,C_3,\ell})$ and $(a_x, a_{x,C_2,\bar{\ell}})$ is entirely in X. The choice of k and the construction of the frame graph $F_{H,\varphi}$ imply that, in that case, there exists an entire copy of $L = K_{h'}$ in $G_{H,\varphi} \setminus X$. Since the vertices in the variable or clause gadget where this copy of L lies do not have neighbors in other variable or clause gadgets, in order for a copy of H not to occur in $G_{H,\varphi} \setminus X$, necessarily X must contain *all* vertices in *all* variable and clause gadgets except possibly two of them (the one containing L, and another one that may allow for the occurrence of $K_{h'} + I_1$), which clearly exceeds the budget k. An analogous argument applies if we assume that $|X \cap \{b_{C,\ell} \mid \ell \in C\}| \leq |C| - 2$ for some clause C. Thus, $G_{H,\varphi}$ satisfies property P2.

Finally, let X be a set as in property P3, let $\tilde{H}$ be an induced occurrence of H in $G_{H,\varphi} \setminus X$, and let K be the subgraph of $\tilde{H}$ isomorphic to K'_h. The choice of X and the discussion above about the edges between M and the copies of L imply that K cannot contain a non-attachment vertex of a copy of L. However, K may contain a variable or clause vertex $v \in A \cup B$ together with h of its neighbors in M, one in each column. Note that, by construction of $G_{H,\varphi}$, $|V(K) \cap (A \cup B)| \leq 1$. In order to complete K into $K_{h'} + I_2$, $\tilde{H}$ should contain two non-adjacent vertices in $A \cup B$ that are anticomplete to K. The construction of $G_{H,\varphi}$ forces that these two vertices must be $a_{x,C,\ell} \in A$ and $b_{C',\ell'} \in B$ with $(C, \ell) = (C', \ell')$. We distinguish two cases. If $|V(K) \cap (A \cup B)| = 0$, property P3 follows and we are done by Lemma 9. Otherwise, $V(K) \cap (A \cup B) = \{v\}$ for some vertex v. Note that, since v is not adjacent to s_0 and s_0 is complete to M, $s_0 \notin V(\tilde{H})$. We construct from $\tilde{H}$ another induced occurrence $\tilde{H}'$ of H in $G_{H,\varphi} \setminus X$ by defining $V(\tilde{H}') := V(\tilde{H}) \setminus \{v\} \cup \{s_0\}$. The subgraph $\tilde{H}'$ satisfies the conditions of property P3, and the theorem follows. ◀

Note that, in the proof of Theorem 10 for $H = K_{h+2} - e$, all the occurrences of H in $G_{H,\varphi}$ are induced, and therefore the lower bound also applies to the $(K_{h+2} - e)$-S-Deletion problem. On the other hand, for $H = K_h + I_2$ the proof of Theorem 11 strongly uses the fact that H cannot occur as an *induced* subgraph. The following lemma explains why the proof does not work for the subgraph version: it can be easily solved in single-exponential time. This points out an interesting difference between both problems.

▶ **Lemma 12.** *For every two integers $h \geq 1$ and $\ell \geq 0$, the $(K_h + I_\ell)$-S-Deletion problem can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(t)})$, where t is the width of a given tree decomposition of the input graph.*

Proof. We will use the fact that, as observed in [13], K_h-S-Deletion can be solved in time $\mathcal{O}^*(2^{\mathcal{O}(t)})$ for every $h \geq 1$. This proves the result for $\ell = 0$, so let now $\ell \geq 1$. Without loss of generality, assume that $n \geq h + \ell$, as otherwise the solution is zero. Given an n-vertex input graph G together with a tree decomposition of width t, we first solve K_h-S-Deletion on G in time $\mathcal{O}^*(2^{\mathcal{O}(t)})$. Let X be a smallest K_h-hitting set in G and let $|X| = k$. Notice that if $n \geq h + \ell$, whenever G contains a clique of size h then it contains $K_h + I_\ell$ as a subgraph as well. Thus, if $k \geq n - (h + \ell) + 1$, then $|V(G) \setminus X| < h + \ell$, so we can safely output $n - (h + \ell) + 1$ as the size of a smallest $(K_h + I_\ell)$-hitting set in G. Otherwise, if $k \leq n - (h + \ell)$, we output k. The algorithm is correct by the fact that a $(K_h + I_\ell)$-hitting set is not smaller than a K_h-hitting set. ◀

By Theorem 2, the lower bound presented in Theorem 10 for $H = K_{h+2} - e$ is tight under the ETH, and the one presented in Theorem 11 for $H = K_h + I_2$ is almost tight. These two graphs are very symmetric, in the sense that each of them contains two non-adjacent vertices that are either complete or anticomplete to a "central" clique K_h (cf. Figure 2). Unfortunately, for graphs without two such non-adjacent symmetric vertices, our framework described above is not capable of obtaining (almost) tight lower bounds. For instance, let $H = K_{h+1} + I_1$, that is, a clique of size $h + 1$ plus an isolated vertex. Let $a \in V(H)$ be any vertex in K_{h+1} and let b be the isolated vertex. The natural idea in order to obtain a tight lower bound of $\mathcal{O}^*(2^{o(t^h)})$ would be, starting from the frame graph $F_{H,\varphi}$ described above, to make the vertices $a_{x,C,\ell} \in A$ play the role of a, and vertices $b_{C,\ell} \in B$ that of b. Then, the functions $f_{C,\ell}$ would indicate, for each vertex $a_{x,C,\ell}$ (resp. $b_{C,\ell}$), its unique neighbor (resp. unique non-neighbor) in each of the columns of M. However, this idea does not work for the following reason: many undesired copies of H appear by interchanging the expected roles of vertices $a_{x,C,\ell}$ and $b_{C,\ell}$, and selecting, in each column of M, any of the $s - 1$ non-neighbors of a and any of the $s - 1$ neighbors of b. We overcome this problem by "pledging" one column of M and introducing a *sentinel* vertex $s_0 \in T$ that is complete to M and the vertices $a_{x,C,\ell}$, and anticomplete to the vertices $b_{C,\ell}$. This vertex "fixes" the roles of vertices in A and B at the price of losing one column of M, hence getting (by Lemma 9) a weaker lower bound of $\mathcal{O}^*(2^{o(t^{h-1})})$, as in Theorem 11. In the following theorem we formalize this idea and we extend it to a more general graph H, namely $K_{h+1} + v_x$ for $0 \leq x \leq h-1$, defined as the graph obtained from K_{h+1} by adding a vertex v adjacent to x vertices in the clique (cf. Figure 2). We will need an extra sentinel vertex in T for each of the neighbors of v in the clique, losing one column of the central part M for each of them.

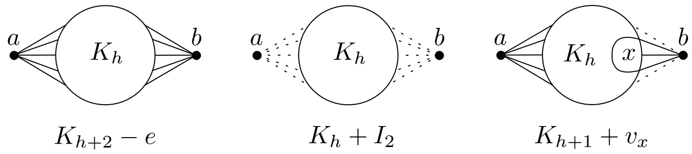

Figure 2 Graphs H considered in Theorem 10, Theorem 11, and Theorem 13, respectively.

▶ **Theorem 13.** *Let $h \geq 1$ and $0 \leq x \leq h-1$ be integers and let $K_{h+1} + v_x$ be the graph obtained from K_{h+1} by adding a vertex adjacent to x vertices in the clique. Then, unless the* ETH *fails, the $(K_{h+1} + v_x)$-IS-DELETION problem cannot be solved in time $\mathcal{O}^*(2^{o(t^{h-x-1})})$, where t is the width of a given tree decomposition of the input graph.*

Proof. Let $H = K_{h'+1} + v_x$ for $h' \geq 1$ and $0 \leq x \leq h'-1$ (again, we relabel h as h' in H to keep the index h for the number of columns in the frame graph $F_{H,\varphi}$). We will again present a reduction from the 3-SAT problem restricted to clean formulas. Given such a formula φ, let $F_{H,\varphi}$ be the frame graph described above with $h = h' - x - 1$, $T = \{s_0, s_1, \ldots, s_x\}$, and $L = K_{h'+1}$ if $x = 0$ and $L = H$ if $x \geq 1$. We proceed to build, starting from $F_{H,\varphi}$, an instance $G_{H,\varphi}$ of H-IS-DELETION with budget $k = 5n - m$ satisfying properties P1, P2, and P3.

We first add an edge between any two vertices $w_{i,j}, w_{i',j'} \in M$ with $j \neq j'$. That is, we turn $G_{H,\varphi}[M]$ into a complete h-partite graph, where each part has size s. We make $s_0 \in T$ complete to $A \cup M$ and anticomplete to B, and all vertices in $T \setminus \{s_0\}$ complete to $A \cup B \cup M$. We also turn $G_{H,\varphi}[T]$ into a clique. For each clause C and each literal ℓ in C, where $\ell \in \{x, \bar{x}\}$ for some variable x, we add the edges $\{a_{C,x,\ell}, w_{f_{C,\ell}(j),j}\}$ and $\{b_{C,\ell}, w_{i,j}\}$ for every $j \in [h]$ and every $i \in [h] \setminus \{f_{C,\ell}(j)\}$. That is, the functions $f_{C,\ell}$ indicate the neighbors of the vertices in A and the non-neighbors of the vertices in B.

Moreover, in the case $x = 0$ (that is, when $H = K_{h'+1} + I_1$, which is disconnected), for each copy of L in $G_{H,\varphi}[B]$, we do the following. Let $v_1, \ldots, v_{h'-1}$ be the vertices of L distinct from the two attachment vertices, ordered arbitrarily. For $j \in [h'-1]$, we make v_j complete to the j-th column of M. We add these edges to prevent that the non-attachment vertices of L may play the role of the isolated vertex in a potential occurrence of $K_{h'+1} + I_1$.

This concludes the construction of $G_{H,\varphi}$, which can be easily seen to satisfy properties P1 and P2 using arguments analogous to those used in the proof of Theorem 10. Let $X \subseteq V(G_{H,\varphi})$ be a set as in property P3, and let $\tilde{H}$ be an induced subgraph of $G_{H,\varphi} \setminus X$ isomorphic to H.

Suppose first that $x \geq 1$. Since $\omega(G_{H,\varphi}[M \cup T]) = h'$, $\omega(G_{H,\varphi}[(A \cup B) \setminus X]) \leq h'$, and $\omega(H) = h' + 1$, $\tilde{H}$ intersects both $M' = M \cup T$ and $A \cup B$. Moreover, since no two adjacent vertices in $(A \cup B) \setminus X$ both have neighbors in M, necessarily $|V(\tilde{H}) \cap (A \cup B)| = 2$ and $V(\tilde{H}) \cap M'$ induces a clique of size h', which implies that $\tilde{H}$ contains a vertex in each column of M' and the whole set T. By the definition of the functions $f_{C,\ell}$ and the construction of $G_{H,\varphi}$, the two vertices in $V(\tilde{H}) \cap (A \cup B)$ must be $a_{x,C,\ell} \in A$ and $b_{C',\ell'} \in B$ with $(C, \ell) = (C', \ell')$, and therefore property P3 follows and we are done by Lemma 9.

Finally, when $x = 0$, the same arguments yield that $V(\tilde{H}) \cap M'$ induces a clique K of size h'. Note that K must contain one vertex from each column of M and the whole set T. By construction of $G_{H,\varphi}$, the only vertices in $A \cup B$ that can be complete to K (resp. complete to $T \setminus \{s_0\}$ and anticomplete to $K \setminus (T \setminus \{s_0\})$) are the vertices $a_{x,C,\ell}$ (resp. $b_{C,\ell}$). Hence, the two vertices in $V(\tilde{H}) \cap (A \cup B)$ must be $a_{x,C,\ell} \in A$ and $b_{C',\ell'} \in B$ with $(C,\ell) = (C',\ell')$, and therefore property P3 also follows and we are done again by Lemma 9. ◀

It is worth mentioning that the lower bound given in Theorem 13 can be strengthened to $\mathcal{O}^*(2^{o(t^{h-\tilde{x}-1})})$, where $\tilde{x} = \min\{x, h-1-x\}$, by using the following trick. If the vertex not belonging to the clique K_{h+1} (vertex b in Figure 2) has more than $\frac{h-1}{2}$ neighbors in the clique (i.e., if $\tilde{x} = h-1-x$), we can interchange, for vertices $b_{C,\ell} \in B$, the roles of neighbors/non-neighbors of the set $T \setminus \{s_0\}$ and the vertices in M given by the functions $f_{C,\ell}$. Doing this modification, an analogous proof works and we can keep the number of columns of M to be $h - \tilde{x} - 1$, which is always at least $\frac{h-3}{2}$. We omit the details.

Another direction for transferring the lower bounds of Theorem 10 and Theorem 13 to other graphs H is to consider complete bipartite graphs.

▶ **Theorem 14.** *For any integer $h \geq 2$, the $K_{h,h}$-IS-Deletion problem cannot be solved in time $\mathcal{O}^*(2^{o(t^h)})$ unless the ETH fails, where t is the width of a given tree decomposition of the input graph.*

Proof. Let $H = K_{h',h'}$ for $h' \geq 2$. Given a clean 3-Sat formula φ, let $F_{H,\varphi}$ be the frame graph described above with $h = h'$, $T = \emptyset$, and $L = K_{h,h}$. In this reduction we will slightly change the budget and property P3 of the constructed instance $G_{H,\varphi}$ of H-IS-Deletion.

Starting from $F_{H,\varphi}$, for each clause C and each literal ℓ in C, where $\ell \in \{x, \bar{x}\}$ for some variable x, we add the edges $\{a_{C,x,\ell}, w_{f_{C,\ell}(j),j}\}$ and $\{b_{C,\ell}, w_{f_{C,\ell}(j),j}\}$ for every $j \in [h]$. We duplicate $h-2$ times the subgraph $G_{H,\varphi}[A]$, obtaining $h-1$ copies overall, and each copy has the same neighborhood in $G_{H,\varphi}[M \cup B]$ as the original one. We relabel the vertices $a_{x,C,\ell}$ in each copy as $a^{\beta}_{x,C,\ell}$ for $\beta \in [h-1]$, and we call again A the set $V(G_{H,\varphi}) \setminus (M \cup B)$. This concludes the construction of $G_{H,\varphi}$, which clearly satisfies properties P1 and P2, where the latter one applies to all the pairs $(a^{\beta}_{x,C_1,\ell}, a^{\beta}_{x,C_3,\ell})$ and $(a^{\beta}_{x}, a^{\beta}_{x,C_2,\bar{\ell}})$ for $\beta \in [h-1]$. Hence, we update the budget accordingly to $k := 5n - m + 2(h-2)n = (2h+1)n - m$.

We proceed to prove the following modified version of property P3 adapted to the current construction:

P3′: Let $X \subseteq V(G_{H,\varphi})$ contain exactly one of $(a^{\beta}_{x,C_1,\ell}, a^{\beta}_{x,C_3,\ell})$ and $(a^{\beta}_{x}, a^{\beta}_{x,C_2,\bar{\ell}})$ for each variable x and $\beta \in [h-1]$, and $|C|-1$ vertices in $\{b_{C,\ell} \mid \ell \in C\}$ for each clause C. Then all occurrences of $K_{h,h}$ in $G_{H,\varphi} \setminus X$ as an induced subgraph contain a set $\{a^{\beta}_{x,C,\ell} \mid \beta \in [h-1]\} \subseteq A$, and exactly one vertex $b_{C',\ell'} \in B$, with $(C,\ell) = (C',\ell')$. Moreover, each such a vertex set gives rise to an occurrence of H in $G_{H,\varphi} \setminus X$.

Basically, property P3′ states that all the copies of the former set A behave in a similar way. With this in mind, it is easy to see that Lemma 9 still holds if we replace, for an instance $G_{H,\varphi}$ constructed in this reduction, property P3 by property P3′.

Consider a set $X \subseteq V(G_{H,\varphi})$ as in property P3′, and let $\tilde{H}$ be an induced subgraph of $G_{H,\varphi} \setminus X$ isomorphic to $K_{h,h}$. By construction of $G_{H,\varphi}$, one of the two parts of $\tilde{H}$ must lie entirely inside M, as there are no edges among distinct variable or clause gadgets. Since the other part of $\tilde{H}$ must lie entirely inside $A \cup B$, the choice of the functions $f_{C,\ell}$ implies that the only vertex sets of size h in $A \cup B$ that are complete to a set of h non-adjacent vertices are of the form $\{a^{\beta}_{x,C,\ell} \mid \beta \in [h-1]\} \cup \{b_{C,\ell}\}$ for some clause C and a literal $\ell \in \{x, \bar{x}\}$, hence property P3′ holds and the theorem follows. ◀

Note that the proof of Theorem 14 works for both $K_{h,h}$-IS-Deletion and $K_{h,h}$-S-Deletion, since all the occurrences of $K_{h,h}$ in the constructed graph $G_{H,\varphi}$ are induced. Hence, as the particular case of Theorem 14 for $h = 2$ we get the following corollary, which answers a question of Mi. Pilipczuk [28] about the asymptotic complexity of C_4-S-Deletion parameterized by treewidth.

▶ **Corollary 15.** *Neither C_4-IS-Deletion nor C_4-S-Deletion can be solved in time $\mathcal{O}^*(2^{o(t^2)})$ unless the* ETH *fails, where t is the width of a given tree decomposition of the input graph.*

As mentioned in [28], C_4-S-Deletion can be easily solved in time $\mathcal{O}^*(2^{\mathcal{O}(t^2)})$. This fact together with Theorem 2 imply that both lower bounds of Corollary 15 are tight.

We can obtain lower bounds for other graphs H that are "close" to a complete bipartite graph. Indeed, note that the lower bound of Theorem 14 also applies to the graph H obtained from $K_{h,h}$ by turning one of the two parts into a clique: the same reduction works similarly, and the only change in the construction is to turn the whole central part M into a clique. We can also consider complete bipartite graphs $K_{a,b}$ with parts of different sizes, by letting the number of columns of the central part M be equal to $\max\{a, b\}$, hence obtaining a lower bound of $\mathcal{O}^*(2^{o(t^{\max\{a,b\}})})$. Similarly, we can also turn one of the two parts of $K_{a,b}$ into a clique, and obtain the same lower bound. In particular, in this way we can obtain a lower bound of $\mathcal{O}^*(2^{o(t^h)})$ for the graph H obtained from K_{h+3} by removing the edges in a triangle.

5 Lower bounds for Colorful H-IS-Deletion

Our main reduction for the colored version is again strongly inspired by the corresponding reduction of Cygan et al. [13, Theorem 2] for the non-induced version, again based on a reduction from the 3-Sat problem restricted to clean formulas and the frame graph defined in Section 4. The main difference with respect to their reduction is that in the non-induced version, the graph H is required to contain a connected component that is neither a *clique* nor a *path*, while for the induced version we only require a component that is not a *clique*, and therefore we need extra arguments to deal with the case where all the connected components of H are paths. In particular, in the proof of Theorem 16, the definition of the graphs L_A and L_B where the graph H_0 is a path is inspired from the proof of [13, Theorem 22].

▶ **Theorem 16.** *Let H be a graph having a connected component on h vertices that is not a clique. Then Colorful H-IS-Deletion cannot be solved in time $\mathcal{O}^*(2^{o(t^{h-2})})$ unless the* ETH *fails, where t is the width of a given tree decomposition of the input graph.*

Proof. Let $H_0, H_1, \ldots, H_p$ be the connected components of H, where H_0 is not a clique. Hence, $|V(H_0)| \geq 3$. As in Section 4, we will again reduce from the 3-Sat problem restricted to clean formulas. Given such a formula φ with n variables and m clauses, we proceed to construct an instance $(G_{H,\varphi}, \sigma)$ of Colorful H-IS-Deletion such that φ is satisfiable if and only if G has a set $X \subseteq V(G)$ of size at most $k := 15n - 4m$ hitting all induced σ-H-subgraphs of G, satisfying properties P1, P2, and P3 (the latter one, concerning only colorful copies of H, of course), and then Lemma 9 will imply the claimed lower bound. The choice of the budget of the current reduction will become clear below, and does not affect the main properties of the reduction.

We start with the frame graph $F_{H,\varphi}$ defined in Section 4, with $h = |V(H_0)| - 2 \geq 1$, $T = \emptyset$, and L to be specified later. Let the vertices of H_0 be labeled $z_0, z_1, \ldots, z_{h+1}$ such that $\{z_0, z_{h+1}\} \notin E(H_0)$; note that this is always possible since H_0 is not a clique. We define the H-coloring σ of $G_{H,\varphi}$ starting from the vertices of $F_{H,\varphi}$ except for the non-attachment vertices of the graphs L, whose coloring will be defined later together with the description of L. Namely, for each variable x and each clause C containing $\ell \in \{x, \bar{x}\}$, we define $\sigma(a_{x,C,\ell}) = \sigma(a_x) = z_0$ and $\sigma(b_{C,\ell}) = z_{h+1}$. For every $i \in [s]$ and $j \in [h]$ we define $\sigma(w_{i,j}) = z_j$. That is, the vertices $a_{x,C,\ell}$ (resp. $b_{C,\ell}$) are mapped to z_0 (resp. z_{h+1}), and the whole j-th column of M is mapped to z_j for $j \in [h]$. We now add edges among the already colored vertices as follows. Within M, the edges mimic those in H_0: for any two vertices $w_{i,j}, w_{i',j'} \in M$ we add the edge $\{w_{i,j}, w_{i',j'}\}$ in $G_{H,\varphi}$ if and only if $\{z_j, z_{j'}\} \in E(H_0)$. As for the edges between $A \cup B$ and M, the functions $f_{C,\ell}$ capture the existence or non-existence of the edges in H_0 between the corresponding vertices. Namely, for each clause C and each literal ℓ in C, where $\ell \in \{x, \bar{x}\}$ for some variable x, and for every $j \in [h]$, we do the following:

- If $\{z_0, z_j\} \in E(H_0)$, then we add the edge $\{a_{C,x,\ell}, w_{f_{C,\ell}(j),j}\}$. Otherwise (i.e., if $\{z_0, z_j\} \notin E(H_0)$), we add the edges $\{a_{C,x,\ell}, w_{i,j}\}$ for every $i \in [h] \setminus \{f_{C,\ell}(j)\}$.
- Similarly, if $\{z_{h+1}, z_j\} \in E(H_0)$, then we add the edge $\{b_{C,\ell}, w_{f_{C,\ell}(j),j}\}$. Otherwise (i.e., if $\{z_{h+1}, z_j\} \notin E(H_0)$), we add the edges $\{b_{C,\ell}, w_{i,j}\}$ for every $i \in [h] \setminus \{f_{C,\ell}(j)\}$.

We now proceed to describe the graph L together with its H-coloring. In fact, we define different (but very similar) graphs L for the copies to be attached in A and B; we call them L_A and L_B, respectively. We start with the definition of L_A, and we distinguish two cases according to H_0:

- Suppose first that H_0 is not a path. As observed in [13], it is easy to see that every graph that is not a path contains at least three vertices that are not separators. Let z_β, z_γ be two such non-separating vertices of H_0 different from z_0. We define L_A as the graph obtained from the disjoint union of three copies H_0^1, H_0^2, H_0^3 of H_0 by identifying the vertices z_β of H_0^1 and H_0^2, and the vertices z_γ of H_0^2 and H_0^3. See Figure 3(a) for an example.
- Suppose now that H_0 is a path. Let z_β be an endvertex of H_0 different from z_0, and let z_γ be an internal vertex of H_0 different from z_0. Note that the latter choice is always possible as $|V(H_0)| \geq 3$ and, if z_0 were the only internal vertex of H_0, then z_0 would not have a non-neighbor in H_0, contradicting the choice of z_0 and z_{h+1}. We define L_A as the graph obtained from the disjoint union of three copies H_0^1, H_0^2, H_0^3 of H_0 by identifying the vertices z_β of H_0^1 and H_0^2, identifying the vertices z_γ of H_0^2 and H_0^3, and adding an edge between every vertex in $V(H_0^2) \setminus \{z_\gamma\}$ and every vertex in $V(H_0^3) \setminus \{z_\gamma\}$. See Figure 3(b) for an example.

In both the above cases, when we attach a copy of L_A between two vertices $u, v \in A$, we identify u and v with the vertices z_0 of the first and third copies of H_0, respectively. The H-coloring σ of L_A is defined naturally, that is, each vertex gets its original color in H_0.

The graph L_B is defined in a completely symmetric way, just by replacing vertex $z_0 \in V(H_0)$ in the above definition of L_A by vertex $z_{h+1} \in V(H_0)$.

▷ Claim 17. In the graph L_A (resp. L_B) defined above, where u and v are the attachment vertices, there are exactly two vertex sets $X_1, X_2 \subseteq V(L_A)$ (resp. $V(L_B)$) of minimum size hitting all induced σ-H_0-subgraphs of L_A (resp. L_B), $|X_1| = |X_2| = 2$, $X_1 \cap \{u, v\} = \{u\}$, and $X_2 \cap \{u, v\} = \{v\}$.

Proof. By symmetry, it suffices to present the proof for L_A. In order to prove the claim, it is enough to prove that there are exactly three induced σ-H_0-subgraphs in L_A, corresponding to the three copies H_0^1, H_0^2, H_0^3 of H_0. Indeed, once this is proved, there are exactly two

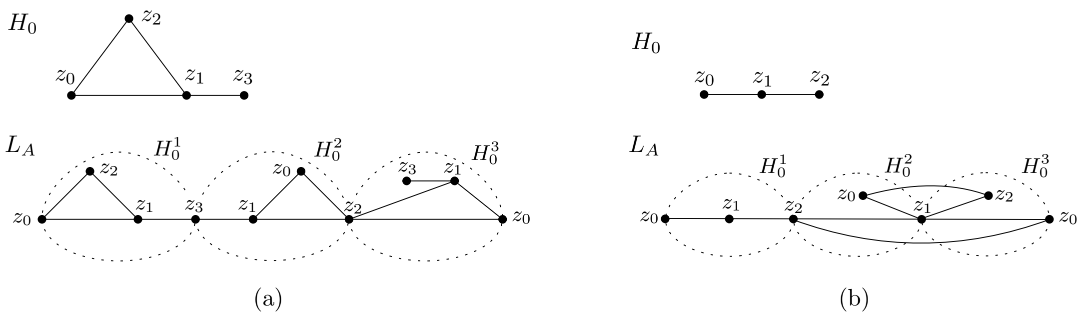

Figure 3 Examples of the construction of L_A: (a) A graph H_0 that is not a path, with $z_\beta = z_3$ and $z_\gamma = z_2$. (b) A path H_0, with $z_\beta = z_2$ and $z_\gamma = z_1$. For better visibility, the edges $\{z_0, z_0\}$ and $\{z_2, z_2\}$ between H_0^2 and H_0^3 in L_A are not shown in the figure.

minimum-sized hitting sets in L_A: u together with vertex $z_\gamma \in V(H_0^2) \cap V(H_0^3)$, and v together with vertex $z_\beta \in V(H_0^1) \cap V(H_0^2)$.

So suppose for contradiction that there exists an induced σ-H_0-subgraph $\tilde{H}_0$ in L_A containing vertices in $V(H_0^i) \setminus V(H_0^j)$ and $V(H_0^j) \setminus V(H_0^i)$ for some distinct $i, j \in [3]$. (For notational simplicity, we interpret an induced σ-H_0-subgraph as an induced subgraph of L_A isomorphic to H_0, with matching colors.) We distinguish two cases depending on H_0.

If H_0 is not a path, the existence of such an $\tilde{H}_0$ in L_A would imply, by the construction of L_A and since H_0 is connected, that at least one of z_β and z_γ is a separator in H_0, contradicting their choice.

Otherwise, if H_0 is a path, first note that since the vertex $z_\beta \in V(H_0^1) \cap V(H_0^2)$ is an endvertex of H_0, $\tilde{H}_0$ cannot contain vertices in $V(H_0^1) \setminus V(H_0^2)$ and $V(H_0^2) \setminus V(H_0^1)$. Hence, necessarily $\tilde{H}_0$ contains vertices in both $V(H_0^2) \setminus V(H_0^3)$ and $V(H_0^3) \setminus V(H_0^2)$. In particular, note that $z_\gamma = V(H_0^2) \cap V(H_0^3) \in V(\tilde{H}_0)$. Since in L_A, vertex z_γ was chosen as an internal vertex of H_0, let z_i and z_j be the two neighbors of z_γ in $\tilde{H}_0$. Recall that in L_A we added all edges between $V(H_0^2) \setminus \{z_\gamma\}$ and $V(H_0^3) \setminus \{z_\gamma\}$. We distinguish two cases:

- Suppose first that both $z_i, z_j \in V(H_0^2) \setminus V(H_0^3)$ or $z_i, z_j \in V(H_0^3) \setminus V(H_0^2)$. Assume that the former case holds –the other one being symmetric– and let z_r be a vertex of $\tilde{H}_0$ in $V(H_0^3) \setminus V(H_0^2)$, which exists by hypothesis. Then $\{z_\gamma, z_i, z_j, z_r\}$ induces a C_4 in $\tilde{H}_0$, a contradiction since $\tilde{H}_0$ is a path.
- Otherwise, suppose without loss of generality that $z_i \in V(H_0^2) \cap V(H_0^3)$ and $z_j \in V(H_0^3) \cap V(H_0^2)$. Then $\{z_\gamma, z_i, z_j\}$ induces a C_3 in $\tilde{H}_0$ (for example, in Figure 3(b), the vertices $\{z_1, z_0, z_2\}$ with $z_1 \in V(H_0^2) \cap V(H_0^3)$ induce a C_3), a contradiction again. ◀

Note that Claim 17 justifies the budget k of our eventual instance $(G_{H,\varphi}, \sigma)$ of Colorful H-IS-Deletion: any optimal solution X needs to contain one of the pairs $(a_{x,C_1,\ell}, a_{x,C_3,\ell})$ and $(a_x, a_{x,C_2,\bar{\ell}})$ for each variable x, and $|C| - 1$ vertices in $\{b_{C,\ell} \mid \ell \in C\}$ for each clause C. Moreover, X contains an extra internal vertex for each of the gadgets L_A and L_B. Taking into account that the number of clauses in φ with exactly three (resp. two) literals equals $3n-2m$ (resp. $3m-3n$), the number of gadgets L_A or L_B equals $4n+(3m-3n)+3(3n-2m)$. Therefore, this amounts to a budget of

$$2n + \sum_{C \in \varphi} (|C| - 1) + 4n + (3m - 3n) + 3(3n - 2m) = 15n - 4m = k.$$

Finally, for every $i \in [p]$, add $k+1$ disjoint copies of H_i, and color their vertices according to their colors in H. This concludes the construction of $(G_{H,\varphi}, \sigma)$, which clearly satisfies property P1. Note that since for every $i \in [p]$ the number of copies of H_i in $G_{H,\varphi}$ exceeds the budget, hitting all colorful induced copies of H in $G_{H,\varphi}$ with at most k vertices is equivalent to hitting all colorful induced copies of component H_0. Therefore, Claim 17 implies that $(G_{H,\varphi}, \sigma)$ satisfies property P2 as well.

Consider a set $X \subseteq V(G_{H,\varphi})$ as in property P3, and let $\tilde{H}$ be an induced σ-H-subgraph of $G_{H,\varphi} \setminus X$. Since for every $j \in [h]$ the only vertices of $G_{H,\varphi}$ colored z_j by σ are those in the j-th column of M, we conclude that, for every $j \in [h]$, $\tilde{H}$ contains exactly one vertex from the j-column of M. Since H_0 is connected and the only vertices in $G_{H,\varphi}$ colored z_0 (resp. z_{h+1}) with neighbors in M are those of type $a_{x,C,\ell}$ (resp. $b_{C,\ell}$), it follows that $\tilde{H}$ contains exactly one vertex $a_{x,C,\ell}$ and exactly one vertex $b_{C',\ell'}$. The properties of the functions $f_{C,\ell}$, which define the edges between $A \cup B$ and M, and the fact that $\tilde{H}$ needs to be an *induced* σ-H-subgraph, imply that necessarily $(C, \ell) = (C', \ell')$, and therefore $(G_{H,\varphi}, \sigma)$ satisfies property P3 and the theorem follows by Lemma 9. ◀

When H is a connected graph, the lower bound of Theorem 16 together with the algorithms given by Proposition 5 and Theorem 3 completely settle, under the ETH, the asymptotic complexity of Colorful H-IS-Deletion parameterized by treewidth. Note that, in particular, Theorem 16 applies when H is path, in contrast to the subgraph version that can be solved in polynomial time [13].

Therefore, what remains is to obtain tight lower bounds when H is disconnected. In particular, Theorem 16 cannot be applied at all when all the connected components of H are cliques, since the machinery that we developed (inspired by Cygan et al. [13]) using the framework graph $F_{H,\varphi}$ crucially needs two non-adjacent vertices in the same connected component. Let us now focus on those graphs, sometimes called *cluster* graphs in the literature.

As mentioned in Section 3, both Colorful K_2-IS-Deletion and Colorful I_2-IS-Deletion can be solved in polynomial time. In our next result we show that if H is slightly larger than these two graphs (namely, K_2 or I_2), then Colorful H-IS-Deletion becomes hard. Namely, we provide a single-exponential lower bound for the following three graphs H on three vertices that are *not* covered by Theorem 16: K_3, I_3, and $K_2 + K_1$. Note that these lower bounds are tight by the algorithm of Theorem 3.

▶ **Theorem 18.** *Let $H \in \{K_3, I_3, K_2 + K_1\}$. Then, unless the* ETH *fails, the* Colorful *H-IS-Deletion problem cannot be solved in time $\mathcal{O}^*(2^{o(t)})$, where t is the width of a given tree decomposition of the input graph.*

Proof. We will prove that none of the considered problems can be solved in time $2^{o(n)}$ under the ETH, which clearly implies the statement of the theorem. For this, we reduce from the Vertex Cover problem restricted to input graphs with maximum degree at most three. To see that this problem cannot be solved in time $2^{o(n)}$ under the ETH, where n is the number of vertices of the input graph, one can apply the classical NP-hardness reduction [20] from 3-Sat to Vertex Cover, but restricting the input formulas to be *clean*. Then the result follows from Lemma 1.

We first present a reduction for $H = K_3$. Given an instance G of Vertex Cover, with $|V(G)| = n$, $|E(G)| = m$, and $\Delta(G) \leq 3$, we proceed to construct an instance (G_{K_3}, σ) of H-IS-Deletion with $|V(G_{K_3})| = \mathcal{O}(n)$ such that G has a vertex cover of size at most k if and only if G_{K_3} has a set of size at most $k + m$ hitting all (induced) σ-K_3-subgraphs. Note that this will prove the desired result, as $\mathsf{tw}(G_{K_3}) \leq |V(G_{K_3})| = \mathcal{O}(n)$, and a tree decomposition of G_{K_3} achieving that width consists of just one bag containing all vertices.

Let $V(K_3) = \{z_1, z_2, z_3\}$ and let L be the graph obtained from three disjoint copies of K_3 by identifying vertices z_2 of the first and second copies, and vertices z_3 of the second and third copies. We define G_{K_3} as the graph obtained from G be replacing each edge $\{u, v\} \in E(G)$ by the graph L, identifying vertex u (resp. v) with vertex z_1 of the first (resp. third) copy of K_3 in L. We define the K_3-coloring σ of G_{K_3} in the natural way, that is, each vertex of G_{K_3} gets the color of its corresponding vertex in the gadget L. Note that all vertices that were originally in G get color z_1. Since $\Delta(G) \leq 3$, it follows that $|V(G_{K_3})| = |V(G)| + |E(G)| \cdot (|V(L)| - 2) = |V(G)| + 5|E(G)| \leq 17|V(G)|/2 = \mathcal{O}(n)$.

By construction of G_{K_3}, each edge of G gives rise to exactly three (induced) σ-K_3-subgraphs in G_{K_3}, and all σ-K_3-subgraphs in G_{K_3} are of this type. For each gadget L, there are exactly two vertex sets of minimum size hitting its three σ-K_3-subgraphs, of size two, each of them containing exactly one of the original vertices of G. Therefore, a vertex cover of G of size at most k can be easily transformed into a set $X \subseteq V(G_{K_3})$ of size at most $k + m$ hitting all σ-K_3-subgraphs of G_{K_3}, and vice versa.

Let now $H = I_3$, with $V(I_3) = \{z_1, z_2, z_3\}$. Given an instance G of Vertex Cover, with $|V(G)| = n$ and $\Delta(G) \leq 3$, we start with the instance (G_{K_3}, σ) of K_3-IS-Deletion defined above, and we construct an instance (G_{I_3}, σ') of I_3-IS-Deletion such that $V(G_{I_3}) = V(G_{K_3})$, $\sigma' = \sigma$ (by associating the labels of $V(K_3)$ and $V(I_3)$), and $E(G_{I_3})$ defined as the tripartite complement of $E(G_{K_3})$, that is, for every pair of vertices $u, v \in V(G_{I_3})$, $\{u, v\} \in E(G_{I_3})$ if and only if $\sigma'(u) \neq \sigma'(v)$ and $\{u, v\} \notin E(G_{K_3})$. Since $|V(G_{I_3})| = |V(G_{K_3})| = \mathcal{O}(n)$ and there is a one-to-one correspondence between induced σ-K_3-subgraphs in G_{K_3} and induced σ-I_3-subgraphs in G_{I_3}, the result follows.

Finally, let now $H = K_2 + K_1$, with $V(H) = \{z_1, z_2, z_3\}$ such that z_1 and z_2 are adjacent. Similarly, we construct an instance $(G_{K_2+K_1}, \sigma)$ of $(K_2 + K_1)$-IS-Deletion starting from (G_{K_3}, σ), but in this case we only complement the neighborhood of the vertices $u \in V(G_{K_2+K_1})$ with $\sigma(u) = z_3$, keeping the set of vertices colored z_3 an independent set. Again, there is a one-to-one correspondence between induced σ-K_3-subgraphs in G_{K_3} and induced σ-$(K_2 + K_1)$-subgraphs in $G_{K_2+K_1}$, and the proof is complete. ◀

The proof of Theorem 18 can be easily adapted to $H = P_3$ by complementing the appropriate neighborhoods, hence obtaining a lower bound of $\mathcal{O}^*(2^{o(t)})$ for Colorful P_3-IS-Deletion. Note, however, that this lower for P_3 bound already follows from Theorem 16.

It is also easy to adapt the proof of Theorem 18 to larger graphs, but then the lower bound of $\mathcal{O}^*(2^{o(t)})$ is not tight anymore. For example, for $H = 2K_2$ (the disjoint union of two edges), with $V(H) = \{z_1, z_2, z_3, z_4\}$ such that the edges are $\{z_1, z_2\}$ and $\{z_3, z_4\}$, it suffices to take the instance $(G_{K_2+K_1}, \sigma)$ of $(K_2 + K_1)$-IS-Deletion defined above and to add a private neighbor colored z_4 for every vertex of $G_{K_2+K_1}$ colored z_3. Also, for $H = K_h$ with $h \geq 4$, in the gadget L we just replace the triangles by cliques of size h, and for $H = I_h$ with $h \geq 4$, we take the h-partite complement of the previous instance of K_h-IS-Deletion.

6 Further research

Concerning H-IS-Deletion, the complexity gap is still quite large for most graphs H, as our lower bounds (Theorems 10, 11, 13, and 14) only apply to graphs H that are "close" to cliques or complete bipartite graphs. In particular, Theorem 10 provides tight bounds for P_3 or $K_4 - e$ (the *diamond*), but we do not know the tight function $f_H(t)$ for other small graphs H on four vertices such as P_4, $K_{1,3}$ (the *claw*), or $2K_2$.

We think that for most graphs H on h vertices, the upper bound $f_H(t) = 2^{\mathcal{O}(t^{h-2})}$ given by Theorem 2 is the asymptotically tight function, and that the single-exponential algorithms for cliques and independent sets are isolated exceptions. The reason is that, in contrast to the subgraph version, when hitting induced subgraphs, edges and non-edges behave essentially in the same way when performing dynamic programming, as one has to keep track of both the existence and the non-existence of edges in order to construct the tables, and storing this information seems to be unavoidable.

Concerning the algorithm for I_h-IS-Deletion running in time $2^{\mathcal{O}(t)} \cdot n^h$ (Theorem 8), it would be interesting to find, if it exists, an FPT algorithm parameterized by both t and h while keeping the dependency on t single-exponential, maybe even being linear in n.

As for Colorful H-IS-Deletion, in view of Theorems 3, 5, 16, and 18, only the cases where H is a disjoint union of at least two cliques and $|V(H)| \geq 4$ remain open. In particular, we do not know the tight function when H is an independent set or a matching with $|V(H)| \geq 4$.

Acknowledgement. We would like to thank the anonymous reviewers for helpful remarks that improved the presentation of the manuscript, in particular for pointing out a previous flaw in the proof of Theorem 11.

References

1. Isolde Adler, Frederic Dorn, Fedor V. Fomin, Ignasi Sau, and Dimitrios M. Thilikos. Faster parameterized algorithms for minor containment. *Theoretical Computer Science*, 412(50):7018–7028, 2011. doi:10.1016/j.tcs.2011.09.015.
2. Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. A complexity dichotomy for hitting connected minors on bounded treewidth graphs: the chair and the banner draw the boundary. In *Proc. of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 951–970, 2020. doi:10.1137/1.9781611975994.57.
3. Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. I. General upper bounds. *SIAM Journal on Discrete Mathematics*, 34(3):1623–1648, 2020. doi:10.1137/19M1287146.
4. Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. II. Single-exponential algorithms. *Theoretical Computer Science*, 814:135–152, 2020. doi:10.1016/j.tcs.2020.01.026.
5. Julien Baste, Ignasi Sau, and Dimitrios M. Thilikos. Hitting minors on bounded treewidth graphs. III. Lower bounds. *Journal of Computer and System Sciences*, 109:56–77, 2020. doi:10.1016/j.jcss.2019.11.002.
6. Hans L. Bodlaender, Marek Cygan, Stefan Kratsch, and Jesper Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. *Information and Computation*, 243:86–111, 2015. doi:10.1016/j.ic.2014.12.008.
7. Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michal Pilipczuk. A $c^k n$ 5-Approximation Algorithm for Treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016. doi:10.1137/130947374.
8. Hans L. Bodlaender, Pinar Heggernes, and Daniel Lokshtanov. Graph modification problems (dagstuhl seminar 14071). *Dagstuhl Reports*, 4(2):38–59, 2014. doi:10.4230/DagRep.4.2.38.
9. Flavia Bonomo-Braberman, Julliano R. Nascimento, Fabiano de S. Oliveira, Uéverton S. Souza, and Jayme Luiz Szwarcfiter. Linear-time algorithms for eliminating claws in graphs. In *Proc. of the 26th International Conference on Computing and Combinatorics (COCOON)*, volume 12273 of *LNCS*, pages 14–26, 2020. doi:10.1007/978-3-030-58150-3_2.
10. Bruno Courcelle. The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs. *Information and Computation*, 85(1):12–75, 1990. doi:10.1016/0890-5401(90)90043-H.

11 Christophe Crespelle, Pål Grønås, Drange, Fedor V. Fomin, and Petr A. Golovach. A survey of parameterized algorithms and the complexity of edge modification. *CoRR*, abs/2001.06867, 2013. URL: https://arxiv.org/abs/2001.06867.

12 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

13 Marek Cygan, Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. Hitting forbidden subgraphs in graphs of bounded treewidth. *Information and Computation*, 256:62–82, 2017. doi:10.1016/j.ic.2017.04.009.

14 Marek Cygan, Jesper Nederlof, Marcin Pilipczuk, Michal Pilipczuk, Johan M. M. van Rooij, and Jakub Onufry Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proc. of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 150–159, 2011. doi:10.1109/FOCS.2011.23.

15 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012. URL: https://dblp.org/rec/books/daglib/0030488.bib.

16 Frederic Dorn, Eelko Penninkx, Hans L. Bodlaender, and Fedor V. Fomin. Efficient Exact Algorithms on Planar Graphs: Exploiting Sphere Cut Decompositions. *Algorithmica*, 58(3):790–810, 2010. doi:10.1007/s00453-009-9296-1.

17 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

18 Fedor V. Fomin, Daniel Lokshtanov, Fahad Panolan, and Saket Saurabh. Efficient computation of representative families with applications in parameterized and exact algorithms. *Journal of the ACM*, 63(4):29:1–29:60, 2016. doi:10.1145/2886094.

19 Fedor V. Fomin, Saket Saurabh, and Neeldhara Misra. Graph modification problems: A modern perspective. In *Proc. of the 9th International Frontiers in Algorithmics Workshop (FAW)*, volume 9130 of *LNCS*, pages 3–6, 2015. doi:10.1007/978-3-319-19647-3_1.

20 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979. URL: https://dblp.org/rec/books/fm/GareyJ79.bib.

21 Russell Impagliazzo and Ramamohan Paturi. On the Complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

22 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which Problems Have Strongly Exponential Complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001. doi:10.1006/jcss.2001.1774.

23 Bart M. P. Jansen, Daniel Lokshtanov, and Saket Saurabh. A Near-Optimal Planarization Algorithm. In *Proc. of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1802–1811, 2014. doi:10.1137/1.9781611973402.130.

24 Ton Kloks. *Treewidth. Computations and Approximations*. Springer-Verlag LNCS, 1994. doi:10.1007/BFb0045375.

25 John M. Lewis and Mihalis Yannakakis. The Node-Deletion Problem for Hereditary Properties is NP-Complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

26 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the Exponential Time Hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.

27 Marcin Pilipczuk. A tight lower bound for Vertex Planarization on graphs of bounded treewidth. *Discrete Applied Mathematics*, 231:211–216, 2017. doi:10.1016/j.dam.2016.05.019.

28 Michal Pilipczuk. Problems parameterized by treewidth tractable in single exponential time: A logical approach. In *Proc. of the 36th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, volume 6907 of *LNCS*, pages 520–531, 2011. doi:10.1007/978-3-642-22993-0_47.

29 Juanjo Rué, Ignasi Sau, and Dimitrios M. Thilikos. Dynamic programming for graphs on surfaces. *ACM Transactions on Algorithms*, 10(2):8:1–8:26, 2014. doi:10.1145/2556952.