Equivalent Dichotomies for Triangle Detection in Subgraph, Induced, and Colored H-Free Graphs

Amir Abboud
Weizmann Institute of Science, Rehovot, Israel

Ron Safier
Weizmann Institute of Science, Rehovot, Israel

Nathan Wallheimer
Weizmann Institute of Science, Rehovot, Israel

Abstract

A recent paper by the authors (ITCS'26) initiates the study of the Triangle Detection problem in graphs avoiding a fixed pattern H as a subgraph and proposes a *dichotomy hypothesis* characterizing which patterns H make the Triangle Detection problem easier in H-free graphs than in general graphs. In this work, we demonstrate that this hypothesis is, in fact, equivalent to analogous hypotheses in two broader settings that a priori seem significantly more challenging: *induced* H-free graphs and *colored* H-free graphs.

Our main contribution is a reduction from the induced H-free case to the non-induced H^+-free case, where H^+ preserves the structural properties of H that are relevant for the dichotomy, namely 3-colorability and triangle count. A similar reduction is given for the colored case.

A key technical ingredient is a self-reduction to Unique Triangle Detection that preserves the induced H-freeness property, via a new color-coding-like reduction.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Fine-grained complexity, Triangle Detection, H-free graphs

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.64

Related Version *Full Version*: https://arxiv.org/abs/2602.23196

Funding This work is part of the project CONJEXITY that has received funding from the European Research Council (ERC) under the European Union's Horizon Europe research and innovation programme (grant agreement No. 101078482). Supported by a research grant from the Center for New Scientists at the Weizmann Institute of Science.
Amir Abboud: Part of this work was done while visiting the Courant Institute of Mathematical Sciences at New York University.

1 Introduction

The Triangle Detection problem asks to decide if a given n-node graph G contains a triangle. It is arguably the simplest problem without a near-linear time algorithm (i.e., running in $m^{1+o(1)}$ time in connected graphs with m edges) and serves as the underlying "complexity atom" at the core of fine-grained hardness assumptions. For dense inputs (e.g. where $m = \Omega(n^2)$), the trivial algorithm runs in $O(n^3)$ time. A truly subcubic $n^{3-\Omega(1)}$ running time [41] can be achieved by algebraic techniques: $O(n^\omega)$ where $\omega < 2.371339$ via Fast Matrix Multiplication (FMM) [12] or a higher bound via Fast Fourier Transform (FFT) [54, 29]. The best bound achievable by other, so-called "combinatorial" methods [15, 18, 25, 60, 4] is mildly subcubic $n^3/2^{\log^{\Omega(1)} n}$. Faster, classical algorithms are known for sparse graphs [41, 14], achieving time $O(m^{3/2})$ combinatorially and $O(m^{\frac{2\omega}{\omega+1}})$ through FMM.

34th Annual European Symposium on Algorithms (ESA 2026).
Editors: Philip Bille, Seth Pettie, and Sabine Storandt; Article No. 64; pp. 64:1–64:22
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Recently, the authors [6] posed the following natural question:

Main Question

For which patterns H can we solve Triangle Detection in H-free graphs faster than in general graphs?

While this question can be asked with respect to each of the aforementioned classical bounds for general graphs, [6] suggests starting with the clean question of whether the H-freeness restriction allows one to break the cubic barrier with a combinatorial algorithm. Notably, due to a reduction by Vassilevska and Williams [59], breaking this barrier in general graphs is equivalent to refuting the famous Boolean Matrix Multiplication (BMM) conjecture.

► **Conjecture 1** (BMM). *There is no truly subcubic combinatorial Triangle Detection (equivalently, BMM) algorithm (in general n-node graphs).*

Thus, the main question above can be restated as follows:

► **Open Question 1.** *For which patterns H can Triangle Detection in H-free graphs be solved combinatorially in truly subcubic time?*

A major motivation for studying combinatorial algorithms for Triangle Detection generally, and Open Question 1 specifically, is the following. The algebraic $n^{3-\Omega(1)}$ algorithms for Triangle Detection (via FMM or FFT) are well-known to fail to generalize for important variants of the basic problem, and one may hope that a different, "combinatorial" algorithm would not suffer from the same limitations. Such an achievement would have groundbreaking consequences, even if the new algorithm only overcomes *some* of these limitations. For example, a truly subcubic algorithm that generalizes to the *minimum weight* setting would refute the All-Pairs Shortest-Paths (APSP) Conjecture, an algorithm that generalizes to the *listing* setting would refute the 3-SUM Conjecture, an algorithm for the *online* setting would refute the Online Matrix Vector Multiplication (OMv) hypothesis, and so on. For this reason, we follow the convention in the literature on combinatorial algorithms of leaving the notion flexible: as explained in [4], the goal is to find new techniques that would generalize to *any* of setting of interest. See [4, Section 1.1] for a broader discussion and additional motivations for studying combinatorial algorithms. Thus, the quest to develop a truly subcubic combinatorial algorithm for Triangle Detection is at the forefront of algorithmic attacks on the conjectures of fine-grained complexity.[1][2] A promising line of attack, related to modern algorithmic paradigms such as expander decompositions [56, 7, 8] and regularity lemmas [18, 4], is via a "structure vs. randomness" approach where a general graph is decomposed into random parts – for which the problem becomes easy – and structured parts (and a sparse part). The question becomes: which notion of structure can be used in such an approach? (See [1] for a novel notion recently employed successfully in the context of subgraph detection.) One natural candidate is H-freeness, and Open Question 1 aims to inform us on the cases for which it is useful.

[1] While fine-grained complexity is often presented as showing conditional lower bounds for a vast number of problems assuming the hardness of a small set of core problems, one should also keep in mind the interpretation that algorithmic research on a vast number of problems should be re-directed to the few core problems, because progress will be made only by finding new techniques that break the conjectures for the core problems.

[2] Note that a "combinatorial" algorithm is not guaranteed to achieve any of these consequences, but one hopes that it would achieve at least one of them.

In the same paper [6], the authors present both positive and negative results, leading them to suggest the following "dichotomy hypothesis" that offers the following crisp characterization as the answer to Open Question 1.

▶ **Hypothesis 1** (The Dichotomy Hypothesis for Triangle Detection in H-Free Graphs). *Triangle Detection in H-free graphs is:*

- *"easy", i.e., solvable in $n^{3-\Omega(1)}$ time via a combinatorial algorithm, if H is* 3-colorable and contains at most one triangle*; and*
- *"hard", i.e., requires $n^{3-o(1)}$ time combinatorially, otherwise.* [3]

The results of [6] confirm this hypothesis for any pattern of size up to 8 [6, Proposition 1.3]. Their hardness reductions apply if H is not 3-colorable or if it contains more than one triangle, i.e., to all the "hard" cases in the hypothesis. However, their algorithmic results can only handle a subset of the (supposedly) "easy" cases. In particular, a 3-colorable pattern H with one triangle is proven to be "easy" if it admits a *degenerate coloring* (see Definition 12).

1.1 What about *induced* H-free graphs?

While [6] start the investigation into Open Question 1 by focusing on the (simpler) non-induced *subgraph* setting of H-freeness, the investigation is equally natural and well-motivated for the (typically more complex) *induced* setting. Note that the class of induced H-free graphs generalizes H-free graphs: every H-free graph is also an induced H-free graph, but the converse does not hold. For instance, a clique of size n is induced P_k-free, where P_k is the k-vertex path, but it contains P_k as a subgraph (assuming $3 \leq k \leq n$).

The main question we ask is: *how does the answer to Open Question 1 change in the induced setting?* Because H-free graphs are a subset of induced H-free graphs, any lower bound for Triangle Detection in H-free graphs also extends to the induced setting, i.e., to *Triangle Detection in induced H-free graphs.* However, the same cannot be said about the upper bounds. In particular, the algorithmic techniques that were used in [6] completely fail for the induced case, as we explain next.

The primary algorithmic technique employed in [6] is the *embedding approach.* This method proceeds by iteratively embedding vertices of the pattern H into the host graph G, which is assumed to be 3-colored by color-coding [13]. A critical step in this recursion involves restricting the search space to the neighborhood of the currently embedded vertex. Specifically, when mapping a vertex $v \in V(H)$ to a vertex $u \in V(G)$, the algorithm restricts a specific color class of G to the neighbors of u, effectively making u universal to the remaining vertices in that color class. The paper shows that this reduces the density of H-free graphs for patterns that admit a degenerate coloring.

However, the embedding approach is fundamentally incompatible with the induced setting. In the induced context, if the pattern H contains a non-edge between v and some unembedded vertex w, we are forbidden from mapping w to a neighbor of u. Consequently, this technique is limited to a very restricted family of induced patterns.

As a result, even for simple patterns like paths, results for the induced setting remain elusive. For general k, a combinatorial $n^{3-\Omega(1)}$ algorithm for Triangle Detection in induced P_k-free graphs is unknown. This stands in stark contrast to the subgraph setting, where

[3] In [6], the authors define "hard" to mean that it is as hard as the general case, meaning that there is a BMM-based lower bound. This makes it technically possible that a pattern H is neither easy nor "hard" (e.g., if one can only show a lower bound under a different assumption). For simplicity, we choose the more relaxed statement.

(non-induced) P_k-free graphs have bounded arboricity by the Erdős-Gallai theorem [35], allowing Triangle Detection to be solved trivially in $O(m) = O(n)$ time using the classic Chiba-Nishizeki algorithm [27]. This leads one to suspect that a classification that would answer Open Question 1 in the induced case would be very different from Hypothesis 1. At the very least, such a dichotomy appears significantly more challenging to establish.

Main Result

The main result of this work is a reduction from the induced case to the non-induced case, showing a surprising *equivalence* between the dichotomy hypotheses in the two settings.

▶ **Theorem 2** (Informal, see Theorem 5). *The Dichotomy Hypothesis for Triangle Detection in H-free graphs holds* if and only if *the same dichotomy hypothesis holds for induced H-free graphs.*

More explicitly, we show a reduction from Triangle Detection in induced H-free graphs to (non-induced) H^+-free graphs, such that if H falls into the "easy" cases of the Hypothesis 1, then so does H^+. Consequently, the same dichotomy hypothesis of [6] not only answers the non-induced case but also the induced case. In other words, Hypothesis 1 turns out to be tantamount to the following hypothesis (in which we only added the word "induced").

▶ **Hypothesis 2** (The Dichotomy in the Induced Setting). *Triangle Detection in* induced *H-free graphs is:*

- *"easy", i.e., solvable in $n^{3-\Omega(1)}$ time via a combinatorial algorithm, if H is* 3-colorable and contains at most one triangle*; and*
- *"hard", i.e., requires $n^{3-o(1)}$ time combinatorially, otherwise.*

The main message of this result is that the induced setting will not be any more challenging than the non-induced case. From a positive perspective, this means that all we have to do to answer Open Question 1 for both non-induced *and* induced is to find algorithms for the remaining "easy" cases of the (non-induced) hypothesis. From a negative perspective, to refute the dichotomy hypothesis proposed by [6], it is enough to find a hardness result for an *induced* pattern.

Our reduction can also be used to prove new positive results for specific patterns H that map to cases already solvable by the algorithms of [6] (i.e., those with a degenerate coloring). Interesting applications include the induced P_6 and induced C_7; see Section 6.

A main technical ingredient in our proof, which may be of independent interest, is a new "color-coding" style self-reduction for Triangle Detection that preserves the induced H-freeness.

Finally, we note that our reduction also applies to the generalization of the problem to k-Clique Detection for any fixed $k \geq 3$, showing that the generalized dichotomy hypotheses for k-Clique Detection in H-free graphs and in induced H-free graphs are also equivalent. We discuss this generalization in more detail in Appendix B.

1.2 Colored H-free graphs

Next, we consider another natural variant that appears incomparable to either the non-induced setting or the induced setting. The notion of *colored H-freeness* was introduced by [6] to exploit the structural properties of patterns that admit a degenerate coloring. Roughly speaking, a colored graph G is considered free of a colored pattern H if it contains no copy of H that preserves its colors. The formal definition for 3-colored graphs is provided below, though the concept generalizes naturally to general colorings.

▶ **Definition 3** (Colored copy of H in G)**.** *Given a* 3*-colored graph* G *with coloring* $c_G : V(G) \to \{red, blue, green\}$*, and a* 3*-colored pattern* H *with coloring* $c_H : V(H) \to \{red, blue, green\}$*, a* colored copy of H *is an injective homomorphism* $\phi: V(H) \to V(G)$ *that preserves colors, i.e.,* $c_G(\phi(v)) = c_H(v)$ *for every* $v \in V(H)$*. If* G *contains no such copy, we say that* G *is* colored H-free*.*

Much like induced H-freeness, colored H-freeness is a generalization of the standard subgraph setting. It is important to note that a colored graph G may be colored H-free even if it contains many copies of H in the uncolored sense, provided the coloring of G does not induce the specific coloring of H on those copies. For instance, consider the specific 3-coloring of a 6-cycle (denoted C_6) depicted in Figure 1.

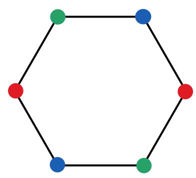

Figure 1 A specific 3-coloring of C_6.

It is a classical result that graphs excluding an uncolored C_6 as a subgraph must be sparse [20], allowing for an efficient Triangle Detection algorithm. In stark contrast, graphs avoiding a *colored* C_6 can be dense. For example, consider the colored blowup of a C_9 in Figure 2.

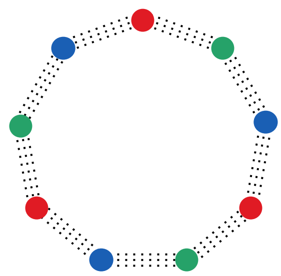

Figure 2 A colored blowup of C_9. This graph is obtained by replacing each vertex of C_9 with an independent set of size $n/9$ and every edge with a complete bipartite graph. All vertices within a single blown-up set share the same color.

This graph is free of the colored C_6 pattern shown in Figure 1, yet it contains $\Theta(n^2)$ edges. Consequently, it contains many copies of C_6 whose colorings fail to respect the color constraints. We remark that the embedding framework of [6] fails to yield a truly subcubic algorithm for the colored C_6 in Figure 1 because this coloring is not degenerate.

Furthermore, observe that the construction in Figure 2 exhibits high "colored density": every vertex has $\Omega(n)$ neighbors in every color class other than its own. This property is significant because the algorithms in [6] rely on efficiently eliminating vertices with low colored degree. For a dense graph like the colored C_9-blowup, where no such vertices exist, it is unclear how to detect triangles efficiently.

This suggests that the colored setting may pose strictly harder challenges than the uncolored setting, making an answer to Open Question 1 for the colored setting seem further away.

Our second result is another reduction showing that the colored case is *also equivalent* to the non-induced and induced cases, in the sense that their dichotomy hypotheses stand or fall together.

▶ **Theorem 4** (Informal, see Theorem 5). *The Dichotomy Hypothesis for Triangle Detection in H-free graphs holds* if and only if *the same dichotomy hypothesis holds for* colored *H-free graphs.*

Specifically, the Triangle Detection problem in colored H-free graphs is defined as follows: the input is a 3-colored H-free graph G, and the task is to decide whether G contains a triangle.

Just like in the induced case, we show a reduction from Triangle Detection in colored H-free graphs to (uncolored) $H^\star$-free graphs, such that if H falls into the "easy" cases of Hypothesis 1 (regardless of its coloring), then so does $H^\star$. Consequently, the same characterization suggested by [6] extends to the colored case.

▶ **Hypothesis 3** (The Dichotomy in the Colored Setting). *Triangle Detection in* colored *H-free graphs is:*

- *"easy", i.e., solvable in $n^{3-\Omega(1)}$ time via a combinatorial algorithm, if H* contains at most one triangle*; and*
- *"hard", i.e., requires $n^{3-o(1)}$ time combinatorially, otherwise.*

We note that the characterization in the colored setting is slightly different from the subgraph and induced settings, as it does not require H to be 3-colorable in the "easy" case. This is because H is already 3-colorable by definition.

To conclude, our results for the colored and induced settings imply our main result:

▶ **Theorem 5.** *The three hypotheses: 1, 2 and 3, are all equivalent.*

1.3 Technical overview

We begin by presenting the high-level idea behind our result for the induced setting. Given a pattern H, we begin with the following observation: If an induced H-free graph G contains H as a subgraph, the embedded copy of H in G must also induce a non-edge of H (otherwise, it would be an induced copy of H). We exploit this observation by augmenting H with additional structure that, together with a non-edge of H in G, implies the existence of many triangles in G. This makes the augmented pattern unlikely to exist in G: Intuitively, the "hardest" instances of detection problems like Triangle Detection have at most one witness.

However, proving this intuition formally in the setting of induced H-free graphs presents technical difficulties. Specifically, what we need is a self-reduction to the *unique* variant of the problem that preserves the induced H-freeness property. In the *Unique Triangle Detection* problem, the input graph is promised to contain at most one triangle.

The standard approach for proving self-reductions to the unique variant of Triangle Detection (and related problems like k-Clique Detection) relies on the input being color-coded. Standard color coding for Triangle Detection (e.g., [13]) is usually done either by having 3 independent copies of $V(G)$ and 3 copies of $E(G)$ appearing between them, or by randomly partitioning the vertices into three buckets and deleting edges within each bucket. However, such transformations do not necessarily preserve the induced H-freeness property of the graph. We provide an alternative color-coding approach that preserves the H-freeness property. Specifically, we show a self-reduction from Triangle Detection in induced H-free graphs, to Triangle Detection in 3-*colored* induced H-free graphs:

▶ **Theorem 6** (Color-coding induced H-free graphs)**.** *There is a deterministic algorithm that reduces Triangle Detection to* 3*-colored Triangle Detection, running in* $O(n^{2.5})$ *time. It either finds a triangle in* G*, or outputs* $O(n^{3/2})$ *instances, where each instance is a* 3*-colored induced subgraph of* G *on* $O(\sqrt{n})$ *vertices. Every triangle in* G *is present in exactly one instance.*

Note that, unlike standard color-coding, the output of this algorithm consists of *induced* subgraphs of G (i.e., no edge deletions). Hence, if G is induced H-free, then every instance in the output is induced H-free. Having this tool in hand, we can then apply a self-reduction to Unique Triangle Detection, such that every output instance is free from the augmented pattern of H.

Colored H-free graphs

In the setting of colored H-free graphs, we employ the following strategy. Our strategy is to embed uncolored copies of H within a larger 3-colorable pattern $H^\star$, but augment it to ensure that *every* 3-coloring of $H^\star$ forces the specific forbidden coloring of H embedded within $H^\star$. Thus, if a 3-colored graph G contains $H^\star$ as a subgraph, the coloring that G induces on $H^\star$ will imply the existence of a forbidden colored copy of H in G. This idea is implemented using various gadgets to enforce that different copies of H in $H^\star$ receive different colorings.

Roadmap

Basic preliminaries and notation appear in Section 2, and related work is discussed in Appendix A. In Section 3, we present our reduction from the induced setting to the subgraph setting. The reduction from the colored setting to the subgraph setting is stated in Section 4 but appears only in the full version. Section 5 combines these results to prove Theorem 5, establishing the equivalence of the three hypotheses. In Section 6, we discuss algorithmic applications of our reduction for the induced setting. Appendix B discusses the generalization of our results to k-Clique Detection for any fixed $k \geq 3$. We conclude with a summary and open questions in Section 7.

2 Preliminaries

Throughout, let $G = (V(G), E(G))$ be a graph with $n = |V(G)|$ vertices and $m = |E(G)|$ edges. An edge between vertices u and v is denoted by uv, and a triangle on vertices x, y, z is denoted by xyz. If $uv \notin E(G)$ then we say that uv is a *non-edge* of G. A graph is said to be *properly* 3*-colored* if each vertex is assigned one of the three colors from the set $\{red, blue, green\}$, such that no two adjacent vertices share the same color. This extends for k-colorings as well, where the set of colors is simply $\{1, 2, \ldots, k\}$. We omit the word "properly" when it is clear from the context that we are referring to a proper coloring. A *color class* of a k-coloring is the set of vertices assigned a particular color. For a vertex $v \in V(G)$, its neighborhood is denoted by $N(v)$.

A *copy of* H *in* G is an injective homomorphism $\phi\colon V(H) \to V(G)$; that is, ϕ is one-to-one and for every edge $xy \in E(H)$, we have $\phi(x)\phi(y) \in E(G)$. We say that G is H*-free* if it contains no copies of H. An *induced copy of* H *in* G is a mapping $\phi\colon V(H) \to V(G)$ such that ϕ is one-to-one and $xy \in E(H)$ if and only if $\phi(x)\phi(y) \in E(G)$. We say that G is *induced* H*-free* if it contains no induced copies of H.

Throughout this paper, we assume the standard Word-RAM model of computation with words of size $O(\log n)$. We use $\widetilde{O}(f(n))$ to hide polylogarithmic factors; that is, $\widetilde{O}(f(n)) = O(f(n) \log^c n)$ for some constant c.

3 Induced H-Free to H^+-Free

In this section, we prove the following theorem:

▶ **Theorem 7.** *For every 3-colorable pattern H containing at most one triangle, there exists a 3-colorable pattern H^+ containing the same number of triangles as H, such that Triangle Detection in induced H-free graphs reduces to Triangle Detection in H^+-free graphs. Specifically, suppose there is an algorithm for Triangle Detection in H^+-free graphs that runs in time $T(N)$ on N-vertex graphs and succeeds with probability at least $2/3$. In that case, there is an algorithm for Triangle Detection in induced H-free graphs running in time*

$$\widetilde{O}\left(n^{2.5} + n^{3/2} \cdot T(\sqrt{n})\right),$$

and succeeding with probability at least $1 - O(1/n)$.

Note that the $n^{3/2} \cdot T(\sqrt{n})$ term in the time complexity dominates the $n^{2.5}$ term whenever $T(N) = \widetilde{\Omega}(N^2)$ (which corresponds to linear time in the input size for dense graphs).

The definition of the augmented pattern H^+ is given below.

▶ **Definition 8.** *Given a pattern H, the* augmented pattern *H^+ is obtained by attaching two wedges (paths of length 2) to the endpoints of every non-edge of H. Formally, $V(H) \subseteq V(H^+)$, $E(H) \subseteq E(H^+)$, and for every $u, v \in V(H)$ such that $uv \notin E(H)$, we add two distinct vertices x_{uv}, y_{uv} to $V(H^+)$ and four edges $ux_{uv}, vx_{uv}, uy_{uv}, vy_{uv}$ to $E(H^+)$.*

Crucially, we observe that if H is 3-colorable and contains at most one triangle, then H^+ also satisfies these properties. That is because the additional vertices x_{uv} and y_{uv} do not form a triangle by definition. Moreover, since their only neighbors are two vertices in H, they can be assigned a color distinct from those of their neighbors. Another crucial property of H^+ is that it contains H as a subgraph. Consequently, if an induced H-free graph contains H^+ as a subgraph, the copy of H embedded within must contain at least one additional edge corresponding to a non-edge of H. By construction, this non-edge closes two triangles in H^+. See Figure 3 for an example.

The goal is, therefore, to show a self-reduction to the unique variant of Triangle Detection that preserves the induced H-freeness property. Specifically, we prove the following lemma:

▶ **Lemma 9** (Reduction to Unique Triangle Detection). *There is a randomized reduction that, given a graph G, either finds a triangle in G, or outputs $k = \widetilde{O}(n^{3/2})$ induced subgraphs $G_1, \ldots, G_k$, such that:*

1. *If G contains a triangle, then with probability $1 - O(1/n)$, there exists at least one G_i that contains exactly one triangle.*
2. *Each G_i has $O(\sqrt{n})$ vertices.*

The reduction runs in $\widetilde{O}(n^{2.5})$ time.

Note that since each G_i is an induced subgraph of G, the property of induced H-freeness is preserved. Namely, if G is induced H-free, then so is G_i for every $1 \leq i \leq k$. To prove this lemma, we use two standard techniques: color coding and sieving. We begin with our proof of the color-coding result:

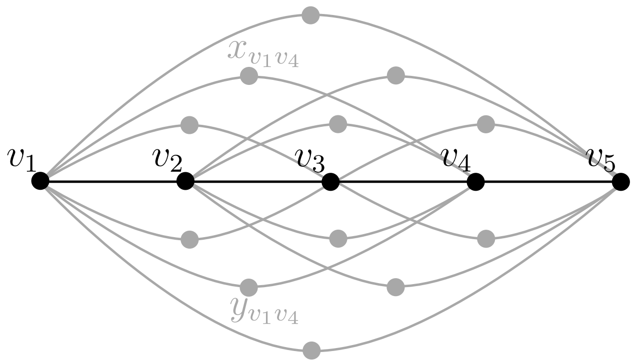

Figure 3 The augmented pattern H^+ corresponding to P_5. The original pattern P_5 is drawn in black, and the additional parts are drawn in gray. Observe that if an induced P_5-free graph contains H^+ as a subgraph, then the vertices corresponding to P_5 must be connected by at least one non-edge of P_5, e.g., the edge v_1v_4. In that case, the graph contains at least two triangles: $v_1v_4x_{v_1v_4}$ and $v_1v_4y_{v_1v_4}$.

▶ **Theorem 6** (Color-coding induced H-free graphs). *There is a deterministic algorithm that reduces Triangle Detection to 3-colored Triangle Detection, running in $O(n^{2.5})$ time. It either finds a triangle in G, or outputs $O(n^{3/2})$ instances, where each instance is a 3-colored induced subgraph of G on $O(\sqrt{n})$ vertices. Every triangle in G is present in exactly one instance.*

Proof of Theorem 6. The idea is to find a coloring of G with $O(\sqrt{n})$ colors, and then take every triplet of color classes as an instance. A paper by Adrian Dumitrescu [32] presents an algorithm for finding either a triangle in G or a $O(\sqrt{n})$-coloring in $O(m + n^{3/2}) = O(n^2)$ time [32, Theorem 4]. [4] If the algorithm does not find a triangle, we refine the provided coloring as follows. As long as some color class is larger than $2\sqrt{n}$, we extract a block of size $\sqrt{n}$, and give it a new color. This results in a finer partition where each color class has size $O(\sqrt{n})$. Note that the refinement procedure adds at most $O(\sqrt{n})$ additional colors because each block has size $\sqrt{n}$ and the blocks are disjoint.

Now, we create 3-colored Triangle Detection instances by taking the induced subgraph of every triplet of color classes. There are $O(n^{3/2})$ instances, and each instance has $O(\sqrt{n})$ vertices and $O(n)$ edges. To implement this efficiently, we can sort the adjacency lists of G according to the color classes, so that we can quickly extract the edges between any triplet of color classes. Then, it takes $O(n^{2.5})$ time to generate all the instances (which is linear in the output size).

For every triangle xyz in G, its vertices x, y, z must belong to different color classes (two vertices can't be in the same color class because an edge connects them). Since the color classes are disjoint, this specific triplet of color classes is unique. Thus, the triangle xyz appears in exactly one output instance. ◀

[4] In fact, the running time of the algorithm in [32] can be improved to $O(m + n)$ by a more careful implementation. Specifically, the bottleneck is Phase I of the algorithm, where checking whether a set $U \subseteq N(v)$ of size $\sqrt{n}$ is independent takes $O(n)$ time. This check can alternatively be performed by scanning the neighborhoods of the vertices in U in $O\left(\sum_{u\in U} \deg(u)\right)$ time. Because each vertex is successfully placed in an independent set (and thus colored) at most once, the sum of these scans over the entire algorithm is bounded by $O(m)$.

The second tool is a "sieving" lemma that reduces the problem to instances with a unique solution. This technique dates back to the Valiant-Vazirani theorem [58] and is commonly used in various settings (e.g., the approximate counting setting [37, 30]). We use the following formulation from [6]:

▶ **Lemma 10** ([6, Lemma 5.2])**.** *There is a near-linear time reduction that, given a 3-colored graph G with at most n vertices in each color class, outputs $r = \widetilde{O}(1)$ induced subgraphs of G, such that if G contains a triangle, then with probability at least $1 - O(1/n)$, one of the subgraphs contains a unique triangle.*

The high-level idea is to "guess" (e.g., via exponential search) the approximate number of vertices participating in a triangle within some color class, and then appropriately subsample. For example, if there are t red vertices that participate in a triangle, subsampling the red vertices with probability $\Theta(1/t)$ results in a graph that, with constant probability, has only one red vertex participating in a triangle. Repeating this for every color class gives the result.[5] It is crucial here that the input graph is properly 3-colored. If instead we merely partitioned $V(G)$ into three parts that need not be independent (e.g., by a random partition), then the sieving lemma would only ensure a unique *colorful* triangle in one of the instances. Other triangles using a monochromatic edge could still remain, so the resulting instance would not necessarily be a Unique Triangle Detection instance.

Proof of Lemma 9. Given the graph G, we first apply the color-coding reduction from Theorem 6 to generate $\ell = O(n^{3/2})$ induced subgraphs $G_1, \ldots, G_\ell$, where each G_i is 3-colored on $O(\sqrt{n})$ vertices. For each G_i, we apply the sieving reduction from Lemma 10 to generate $r = \widetilde{O}(1)$ induced subgraphs $G_{i,1}, \ldots, G_{i,r}$. The collection of all $G_{i,j}$ constitutes the output.

If G is triangle-free, then all subgraphs are triangle-free. If G has a triangle, Theorem 6 ensures it appears in some G_i. Conditioned on this, Lemma 10 ensures that with high probability, one of the $G_{i,j}$ contains a unique triangle. The generation of the instances takes $\widetilde{O}(n^{2.5})$ time, hence the overall time bound of the reduction is $\widetilde{O}(n^{2.5})$. ◀

We are now ready to prove the main theorem in this section.

Proof of Theorem 7. Assume there exists a Triangle Detection algorithm $\mathcal{A}$ for H^+-free graphs running in time at most $T(N)$ on N-vertex graphs, that succeeds with probability at least $2/3$. Given an induced H-free graph G, we apply the reduction from Lemma 9 to generate instances $G_1, \ldots, G_k$ where $k = \widetilde{O}(n^{3/2})$ and each G_i is an induced subgraph of G. The promise is that if G contains a triangle, then with probability $1 - O(1/n)$, at least one G_i contains a unique triangle. We run $\mathcal{A}$ on every instance G_i, and we amplify the success probability of $\mathcal{A}$ to be $1 - O(1/n^3)$. Standard amplification is done by running $\Theta(\log n)$ independent executions of $\mathcal{A}$ on G_i and taking the majority vote. If the majority of executions return YES, we return YES for that instance; otherwise, we return NO.

Correctness. We analyze the two cases depending on whether G contains a triangle.

- **Case 1: G is triangle-free (NO-instance).** Since every G_i is an induced subgraph of G, it is also induced H-free and triangle-free. For our algorithm to err, $\mathcal{A}$ must output YES on some G_i. This can happen only if:

---

[5] We remark here that [6] replaced two subsampling phases with one phase where they subsample the edges. This is not suitable for induced H-free graphs, but as we explained, it can be replaced with subsampling vertices at the cost of another $O(\log n)$ factor in the running time. In fact, this is the standard way to prove such a result.

1. The randomized algorithm $\mathcal{A}$ fails. This happens with probability $O(1/n^3)$, for a total error probability of $\widetilde{O}(n^{3/2}/n^3) = O(1/n)$.
2. The input G_i is not H^+-free, so the behavior of $\mathcal{A}$ on G_i is indeterminate.

We show that item 2 is impossible. Suppose, for contradiction, that G_i contains a copy of H^+. Since G_i is induced H-free, and H^+ contains H as a subgraph, this (non-induced) copy of H must induce an edge uv such that $uv \notin E(H)$. By the definition of the augmented pattern H^+, the vertices u, v are connected to a specific vertex x_{uv} in H^+. Consequently, the presence of edge uv implies that $\{u, v, x_{uv}\}$ forms a triangle in G_i. This contradicts the fact that G_i is triangle-free. Thus, all inputs are valid H^+-free graphs, and the algorithm returns NO with probability $1 - O(1/n)$.

- **Case 2: *G* contains a triangle (YES-instance).** By Lemma 9, with probability $1 - O(1/n)$, there exists an index i such that G_i contains a unique triangle. We claim that such a graph G_i is H^+-free. Suppose, for contradiction, that G_i contains a copy of H^+. As in Case 1, since G_i is induced H-free, this copy must use a non-edge uv of H as an edge in G_i. By the construction of H^+, the endpoints u, v are connected to *two* distinct vertices x_{uv} and y_{uv} in H^+. Therefore, the edge uv closes two distinct triangles: uvx_{uv} and uvy_{uv}. This contradicts the property that G_i contains exactly one triangle. Thus, the instance G_i is valid (H^+-free) and contains a triangle. The algorithm $\mathcal{A}$ will return YES with probability $1 - O(1/n^3)$.

Complexity analysis. The total time is the reduction time plus the oracle calls:

$$\underbrace{\widetilde{O}(n^{2.5})}_{\text{Reduction}} + \underbrace{\widetilde{O}(n^{3/2})}_{\#\text{ instances}} \times \underbrace{T(O(\sqrt{n}))}_{\text{Oracle time}}. \qquad \blacktriangleleft$$

4 Colored H-Free to $H^\star$-Free

The main theorem of this section is the following.

▶ **Theorem 11.** *For every 3-colored pattern H that contains at most one triangle, there exists a 3-colorable pattern $H^\star$ containing the same number of triangles as H, such that every 3-colored graph G that is colored H-free (with respect to the coloring of H) is also $H^\star$-free.*

The proof of this theorem is deferred to the full version of the paper.

5 Equivalence of the Three Hypotheses

In this section, we prove the main theorem in this paper, Theorem 5, showing that Hypotheses 1, 2 and 3 are all equivalent.

Proof of Theorem 5. We begin by showing an equivalence between Hypothesis 1 (the subgraph setting) and Hypothesis 2 (the induced setting), using Theorem 7.

Subgraph $\Longrightarrow$ Induced. Suppose that Hypothesis 1 is true. Recall that the lower bounds from [6] for the hard patterns naturally extend to the induced setting because the class of H-free graphs is a subset of the class of induced H-free graphs. Thus, if the problem is hard for H-free graphs, it is certainly hard for the superset.

It remains to show that the easy patterns (i.e., 3-colorable patterns with at most one triangle) also extend to the induced setting. Let H be a 3-colorable pattern containing at most one triangle. Since H^+ preserves the 3-coloring and triangle count, our assumption implies that Triangle Detection in H^+-free graphs is solvable in $n^{3-\varepsilon}$ time, for some $\varepsilon > 0$.

Consequently, by Theorem 7, there exists an algorithm for induced H-free graphs running in truly subcubic time $\widetilde{O}(n^{2.5} + n^{3/2}(\sqrt{n})^{3-\varepsilon}) = \widetilde{O}(n^{2.5} + n^{3-\varepsilon/2})$. Therefore, if Hypothesis 1 holds, then Hypothesis 2 holds as well.

Induced $\Longrightarrow$ Subgraph. Conversely, a confirmation of Hypothesis 2 implies a confirmation of Hypothesis 1. For the easy patterns, any algorithm that solves Triangle Detection in induced H-free graphs also works for H-free graphs, as the latter is a subset of the former. For the hard patterns, the lower bounds are already established in [6]. Thus, the two hypotheses are equivalent.

We next establish the equivalence between Hypothesis 1 (the subgraph setting) and Hypothesis 3 (the colored setting), using Theorem 11.

Subgraph $\Longrightarrow$ Colored. Suppose that Hypothesis 1 is true. First, consider the "hard" patterns. Note that the input for Triangle Detection in colored H-free graphs is a 3-colored graph G. Therefore, if H is not 3-colorable, G cannot contain a colored copy of H, and the problem is trivially hard. For 3-colorable patterns H that contain more than one triangle, the lower bounds from [6] extend naturally to the colored setting via color-coding. Specifically, take a hard instance G of the uncolored H-free Triangle Detection problem. We can color-code it (e.g., using random partitioning or a perfect hash family [13]) to produce a 3-colored subgraph G'. This subgraph G' is equivalently hard (preserving a triangle with constant probability) and satisfies the property of being colored H-free for every possible coloring of H (since G contained no copy of H at all). Thus, the hardness of the subgraph setting implies hardness in the colored setting.

It remains to show that the "easy" patterns also extend. Let H be a 3-colored pattern containing at most one triangle. By Theorem 11, there exists a 3-colorable pattern $H^\star$ (containing the same number of triangles as H) such that every colored H-free graph is also $H^\star$-free. If Hypothesis 1 holds, Triangle Detection in $H^\star$-free graphs is solvable in $O(n^{3-\varepsilon})$ time. Given a colored H-free graph G, we know G is $H^\star$-free; thus, we can run the algorithm for $H^\star$-free graphs on G. Consequently, if Hypothesis 1 holds, Hypothesis 3 holds as well.

Colored $\Longrightarrow$ Subgraph. Conversely, suppose that Hypothesis 3 is true. For the hard patterns, the lower bounds from [6] apply directly. For the easy patterns, let H be a 3-colorable pattern with at most one triangle. Given an uncolored H-free graph G, we use color-coding to obtain a 3-colored subgraph G' of G (which preserves a triangle with constant probability). Since G is H-free, G' contains no colored copies of H with respect to any valid 3-coloring of H. Thus, G' is a valid input for the algorithm guaranteed by Hypothesis 3. Assuming that Hypothesis 3 holds, there is a truly subcubic algorithm that decides if G' contains a triangle. Thus, H is an easy pattern in the subgraph setting, and the two hypotheses are equivalent.

This completes the proof, showing that Hypotheses 1, 2 and 3 are all equivalent. ◀

6 Applications for Specific Patterns

Notably, our reduction for induced subgraphs (Theorem 7) serves not only as a bridge between Hypothesis 1 and 2, but also as an algorithmic tool for induced patterns that do not a priori seem to admit a truly subcubic algorithm. We observe that while it is unclear how to efficiently detect a triangle in an induced P_6-free graph or an induced C_7-free graph, their augmented patterns admit a *degenerate coloring*.

▶ **Definition 12.** *A* 3*-colored graph H containing at most one triangle is* degenerately colored *if, in every induced subgraph of H which is not a triangle, there exists a vertex with a monochromatic neighborhood.*

It was shown in [6, Theorem 2] that if a triangle-free pattern H admits a degenerate coloring, then Triangle Detection in H-free graphs is solvable in $\widetilde{O}(n^{3-2^{-(k-3)}})$ time, where $k = |V(H)|$. By applying the reduction from Theorem 7, we obtain the following result:

▶ **Proposition 13.** *Let $a = |V(P_6^+)| = 26$ and $b = |V(C_7^+)| = 35$. There is a combinatorial algorithm for Triangle Detection in induced P_6-free graphs and induced C_7-free graphs, running in time*

$$\widetilde{O}\left(n^{3/2} \cdot \sqrt{n^{3-2^{-(a-3)}}}\right) = \widetilde{O}(n^{3-2^{-(a-2)}}),$$

and

$$\widetilde{O}\left(n^{3/2} \cdot \sqrt{n^{3-2^{-(b-3)}}}\right) = \widetilde{O}(n^{3-2^{-(b-2)}}),$$

respectively.

The degenerate colorings of P_6^+ and C_7^+ are illustrated in Figures 4 and 5, respectively.

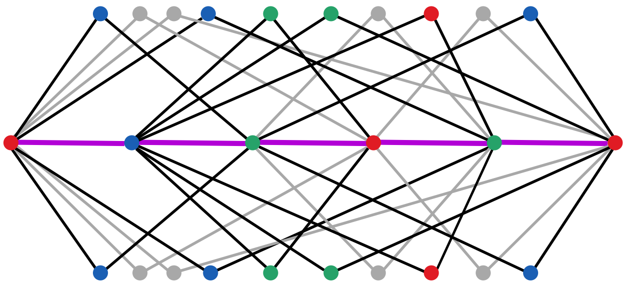

Figure 4 The augmented pattern P_6^+ with a degenerate coloring. The edges of the original P_6 are represented by highlighted purple lines. Wedges connecting the endpoints of non-edges are shown in gray whenever the endpoints share the same color; this is because such paths do not obstruct the degenerate coloring. From this coloring, it is evident that P_6^+ admits a degenerate coloring. Consequently, Triangle Detection in induced P_6-free graphs can be solved in truly subcubic time using [6, Theorem 2].

While our reduction to induced subgraphs yields some unexpected algorithmic results, the reduction to colored subgraphs does not achieve similar results. The reason for this is that the augmented pattern $H^\star$ contains a subgraph that does not admit a degenerate coloring. Specifically, the equality gadget is derived from the Grötzsch graph by removing an edge, which is not degenerately colorable. Therefore, we cannot use [6, Theorem 2] in conjunction with Theorem 11 to obtain new algorithms for colored patterns.

7 Conclusions

We have shown that the three hypotheses: 1, 2 and 3, are all equivalent. Consequently, the central open question remains the validity of these hypotheses. Our results suggest a strategic approach for resolving this: to prove the hypotheses, it suffices to focus on the restricted subgraph setting, which is arguably simpler. Conversely, to refute them, one can target the more general colored or induced settings. For instance, demonstrating a cubic conditional lower bound for induced P_{100}-free graphs would refute the dichotomy hypothesis for the subgraph setting as well.

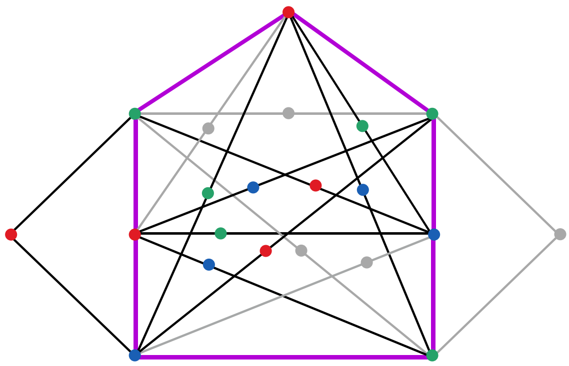

Figure 5 The augmented pattern C_7^+ with a degenerate coloring. The edges of the original C_7 are represented by highlighted purple lines. Paths connecting the endpoints of non-edges are shown in gray whenever the endpoints share the same color; this is because such paths do not obstruct the degenerate coloring. (Note: For visual clarity, only one wedge is shown for these connections, rather than the two required by the formal definition of H^+). From this coloring, it is evident that C_7^+ admits a degenerate coloring. Consequently, Triangle Detection in induced C_7-free graphs can be solved in truly subcubic time using [6, Theorem 2].

Even if the hypotheses hold true, we can ask a more refined question regarding exact running times. In other words, while our work shows that the hypotheses are equivalent in classifying patterns as "easy" or "hard", it does not provide a complete picture of the exact running times for specific patterns. Specifically, for an "easy" pattern H, are induced H-free graphs strictly harder to process than their subgraph counterparts? Our reduction implies a slowdown due to color-coding and the increased size of the augmented pattern H^+. For example, Triangle Detection in (subgraph) C_7-free graphs is solvable in $\widetilde{O}(m + n^{5/3})$ time [6, Theorem 6], but the running time we obtain for induced C_7-free graphs is significantly larger: $\widetilde{O}(n^{3-2^{-33}})$.

Note that even if the hypotheses are false and a different dichotomy emerges, our reduction may still have the potential to prove equivalence between the induced and non-induced settings. Because the augmented pattern H^+ is structurally simple, it may still preserve the properties relevant to an alternative dichotomy.

Finally, another open question is whether we can obtain a deterministic reduction from the induced setting to the non-induced setting. Our current reduction is based on the random sieve approach, and derandomizing this technique remains an open problem.

References

1 Amir Abboud, Shyan Akmal, and Nick Fischer. A truly subcubic combinatorial algorithm for induced 4-cycle detection. In *Proceedings of the 2026 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3562–3599. SIAM, 2026. doi:10.1137/1.9781611978971.130.

2 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the current clique algorithms are optimal, so is valiant's parser. *SIAM J. Comput.*, 47(6):2527–2555, 2018. doi:10.1137/16M1061771.

3 Amir Abboud, Karl Bringmann, Seri Khoury, and Or Zamir. Hardness of approximation in p via short cycle removal: cycle detection, distance oracles, and beyond. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1487–1500, 2022. doi:10.1145/3519935.3520066.

4 Amir Abboud, Nick Fischer, Zander Kelley, Shachar Lovett, and Raghu Meka. New graph decompositions and combinatorial boolean matrix multiplication algorithms. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 935–943, 2024. `doi:10.1145/3618260.3649696`.

5 Amir Abboud, Nick Fischer, and Yarin Shechter. Faster combinatorial k-clique algorithms. In José A. Soto and Andreas Wiese, editors, *LATIN 2024: Theoretical Informatics – 16th Latin American Symposium, Puerto Varas, Chile, March 18-22, 2024, Proceedings, Part I*, Lecture Notes in Computer Science, pages 193–206. Springer, 2024. `doi:10.1007/978-3-031-55598-5_13`.

6 Amir Abboud, Ron Safier, and Nathan Wallheimer. Triangle Detection in H-Free Graphs. In Shubhangi Saraf, editor, *17th Innovations in Theoretical Computer Science Conference (ITCS 2026)*, volume 362 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:19, Dagstuhl, Germany, 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. Full version: `https://arxiv.org/abs/2511.17224`. `doi:10.4230/LIPIcs.ITCS.2026.1`.

7 Amir Abboud and Nathan Wallheimer. Worst-Case to Expander-Case Reductions. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:23, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ITCS.2023.1`.

8 Amir Abboud and Nathan Wallheimer. Worst-Case to Expander-Case Reductions: Derandomized and Generalized. In Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman, editors, *32nd Annual European Symposium on Algorithms (ESA 2024)*, volume 308 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:18, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ESA.2024.4`.

9 Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster alignment of sequences. In *International Colloquium on Automata, Languages, and Programming*, pages 39–51. Springer, 2014. `doi:10.1007/978-3-662-43948-7_4`.

10 Tara Abrishami, Maria Chudnovsky, Cemil Dibek, and Paweł Rzążewski. Polynomial-time algorithm for maximum independent set in bounded-degree graphs with no long induced claws. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1448–1470. SIAM, 2022. `doi:10.1137/1.9781611977073.61`.

11 Vladimir Evgen'evich Alekseev and DV Korobitsyn. Complexity of some problems on hereditary classes of graphs. *Diskretnaya Matematika*, 4(4):34–40, 1992.

12 Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2005–2039. SIAM, 2025. `doi:10.1137/1.9781611978322.63`.

13 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM (JACM)*, 42(4):844–856, 1995. `doi:10.1145/210332.210337`.

14 Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997. `doi:10.1007/BF02523189`.

15 Vladimir L Arlazarov, EA Dinic, MA Kronrod, and IA Faradzev. On economical construction of the transitive closure of a directed graph. In *Dokl. Akad. Nauk SSSR*, volume 194, pages 1209–1210, 1970.

16 Arturs Backurs, Nishanth Dikkala, and Christos Tzamos. Tight hardness results for maximum weight rectangles. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, Rome, Italy, July 11-15, 2016*, volume 55 of *LIPIcs*, pages 81:1–81:13. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. `doi:10.4230/LIPIcs.ICALP.2016.81`.

17 Arturs Backurs and Christos Tzamos. Improving viterbi is hard: Better runtimes imply faster clique algorithms. In *International Conference on Machine Learning*, pages 311–321. PMLR, 2017. URL: `http://proceedings.mlr.press/v70/backurs17a.html`.

18 Nikhil Bansal and Ryan Williams. Regularity lemmas and combinatorial algorithms. *Theory Comput.*, 8(1):69–94, 2012. `doi:10.4086/toc.2012.v008a004`.

19 Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 1998. `doi:10.1007/978-1-4612-0619-4`.

20 John A Bondy and Miklós Simonovits. Cycles of even length in graphs. *Journal of Combinatorial Theory, Series B*, 16(2):97–105, 1974.

21 Édouard Bonnet, Nicolas Bousquet, Pierre Charbit, Stéphan Thomassé, and Rémi Watrigant. Parameterized complexity of independent set in H-free graphs. *Algorithmica*, 82(8):2360–2394, 2020. `doi:10.1007/s00453-020-00730-6`.

22 Karl Bringmann, Paweł Gawrychowski, Shay Mozes, and Oren Weimann. Tree edit distance cannot be computed in strongly subcubic time (unless apsp can). *ACM Transactions on Algorithms (TALG)*, 16(4):1–22, 2020. `doi:10.1145/3381878`.

23 Karl Bringmann, Allan Grønlund, and Kasper Green Larsen. A dichotomy for regular expression membership testing. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 307–318. IEEE, 2017. `doi:10.1109/FOCS.2017.36`.

24 Karl Bringmann and Philip Wellnitz. Clique-based lower bounds for parsing tree-adjoining grammars. In Juha Kärkkäinen, Jakub Radoszewski, and Wojciech Rytter, editors, *28th Annual Symposium on Combinatorial Pattern Matching, CPM 2017, Warsaw, Poland, July 4-6, 2017*, volume 78 of *LIPIcs*, pages 12:1–12:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. `doi:10.4230/LIPIcs.CPM.2017.12`.

25 Timothy M. Chan. Speeding up the four russians algorithm by about one more logarithmic factor. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 212–217, USA, 2015. Society for Industrial and Applied Mathematics. `doi:10.1137/1.9781611973730.16`.

26 Yi-Jun Chang. Hardness of rna folding problem with four symbols. *Theoretical Computer Science*, 757:11–26, 2019. `doi:10.1016/j.tcs.2018.07.010`.

27 Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM Journal on computing*, 14(1):210–223, 1985. `doi:10.1137/0214017`.

28 Maria Chudnovsky, Marcin Pilipczuk, Michał Pilipczuk, and Stéphan Thomassé. Quasi-polynomial time approximation schemes for the maximum weight independent set problem in H-free graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2260–2278. SIAM, 2020. `doi:10.1137/1.9781611975994.139`.

29 Henry Cohn, Robert D. Kleinberg, Balázs Szegedy, and Christopher Umans. Group-theoretic algorithms for matrix multiplication. In *46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005, Pittsburgh, PA, USA, October 23-25, 2005, Proceedings*, pages 379–388. IEEE Computer Society, 2005. `doi:10.1109/SFCS.2005.39`.

30 Holger Dell, John Lapinskas, and Kitty Meeks. Approximately counting and sampling small witnesses using a colorful decision oracle. *SIAM Journal on Computing*, 51(4):849–899, 2022. `doi:10.1137/19m130604x`.

31 Guillaume Ducoffe, Michel Habib, and Laurent Viennot. Diameter, eccentricities and distance oracle computations on h-minor free graphs and graphs of bounded (distance) vapnik–chervonenkis dimension. *SIAM Journal on Computing*, 51(5):1506–1534, 2022. `doi:10.1137/20m136551x`.

32 Adrian Dumitrescu. Finding triangles or independent sets; and other dual pair approximations. *Studia Scientiarum Mathematicarum Hungarica*, 62(1):47–62, 2025. `doi:10.48550/arXiv.2105.01265`.

33 Lech Duraj, Filip Konieczny, and Krzysztof Potępa. Better Diameter Algorithms for Bounded VC-Dimension Graphs and Geometric Intersection Graphs. In Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman, editors, *32nd Annual European Symposium on Algorithms (ESA 2024)*, volume 308 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 51:1–51:18, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.ESA.2024.51`.

34 Tala Eagling-Vose, David C. Kutner, Felicia Lucke, Dániel Marx, Barnaby Martin, Daniël Paulusma, and Erik Jan van Leeuwen. Steiner forest for h-subgraph-free graphs, 2026. `doi:10.48550/arXiv.2602.21859`.

35 Paul Erdős and Tibor Gallai. On maximal paths and circuits of graphs. *Acta Math. Acad. Sci. Hungar*, 10:337–356, 1959.

36 Peter Gartland, Daniel Lokshtanov, Tomáš Masařík, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. Maximum weight independent set in graphs with no long claws in quasi-polynomial time. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, pages 683–691, New York, NY, USA, 2024. Association for Computing Machinery. `doi:10.1145/3618260.3649791`.

37 Oded Goldreich. On coarse and fine approximate counting of t-cliques. In *Computational Complexity and Local Algorithms: On the Interplay Between Randomness and Computation*, pages 52–79. Springer, 2025. `doi:10.1007/978-3-031-88946-2_4`.

38 Petr A Golovach, Daniël Paulusma, and Bernard Ries. Coloring graphs characterized by a forbidden subgraph. In *International Symposium on Mathematical Foundations of Computer Science*, pages 443–454. Springer, 2012. `doi:10.1007/978-3-642-32589-2_40`.

39 Martin Grohe. Fixed-point definability and polynomial time on graphs with excluded minors. *Journal of the ACM (JACM)*, 59(5):1–64, 2012. `doi:10.1145/2371656.2371662`.

40 Edin Husić, Stéphan Thomassé, and Nicolas Trotignon. The Independent Set Problem Is FPT for Even-Hole-Free Graphs. In Bart M. P. Jansen and Jan Arne Telle, editors, *14th International Symposium on Parameterized and Exact Computation (IPEC 2019)*, volume 148 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:12, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.IPEC.2019.21`.

41 Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. In *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing*, STOC '77, pages 1–10, New York, NY, USA, 1977. Association for Computing Machinery. `doi:10.1145/800105.803390`.

42 Matthew Johnson, Barnaby Martin, Jelle J. Oostveen, Sukanya Pandey, Daniël Paulusma, Siani Smith, and Erik Jan van Leeuwen. Complexity framework for forbidden subgraphs i: The framework. *Algorithmica*, 87(3):429–464, January 2025. `doi:10.1007/s00453-024-01289-2`.

43 Marcin Kamiński. Max-cut and containment relations in graphs. *Theoretical Computer Science*, 438:89–95, 2012. `doi:10.1016/j.tcs.2012.02.036`.

44 Adam Karczmarz and Da Wei Zheng. Subquadratic algorithms in minor-free digraphs:(weighted) distance oracles, decremental reachability, and more. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4338–4351. SIAM, 2025. `doi:10.1137/1.9781611978322.147`.

45 Dmitry V Korobitsin. On the complexity of domination number determination in monogenic classes of graphs. *Diskretnaya Matematika*, 1992.

46 Alexandr V Kostochka. The minimum hadwiger number for graphs with a given mean degree of vertices. *Metody Diskret. Analiz.*, 38:37–58, 1982.

47 Tamás Kovári, Vera Sós, and Pál Turán. On a problem of k. zarankiewicz. In *Colloquium Mathematicum*, volume 3, pages 50–57. Polska Akademia Nauk. Instytut Matematyczny PAN, 1954.

48 Daniel Král', Jan Kratochvíl, Zsolt Tuza, and Gerhard J Woeginger. Complexity of coloring graphs without forbidden induced subgraphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 254–262. Springer, 2001. `doi:10.1007/3-540-45477-2_23`.

49 Hung Le and Christian Wulff-Nilsen. Vc set systems in minor-free (di) graphs and applications. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 5332–5360. SIAM, 2024. doi:10.1137/1.9781611977912.192.

50 Andrea Lincoln, Virginia Vassilevska Williams, and Ryan Williams. Tight hardness for shortest cycles and paths in sparse graphs. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1236–1252. SIAM, 2018. doi:10.1137/1.9781611975031.80.

51 Konrad Majewski, Tomáš Masařík, Jana Masaříková, Karolina Okrasa, Marcin Pilipczuk, Paweł Rzążewski, and Marek Sokołowski. Max weight independent set in graphs with no long claws: An analog of the gyárfás' path argument. *ACM Transactions on Computation Theory*, 16(2):1–18, 2024. doi:10.1145/3636422.

52 Jelle J. Oostveen, Daniël Paulusma, and Erik Jan van Leeuwen. The complexity of diameter on H-free graphs. *SIAM Journal on Discrete Mathematics*, 39(2):1213–1245, 2025. doi:10.1137/24M1677885.

53 Ilia Nikolaevich Ponomarenko. Isomorphism problem for classes of graphs closed under contractions. *Zapiski Nauchnykh Seminarov POMI*, 174:147–177, 1988.

54 Kevin Pratt, Yahel Uffenheimer, and Omri Weinstein. (approximate) matrix multiplication via convolutions. *CoRR*, abs/2510.22193, 2025. doi:10.48550/arXiv.2510.22193.

55 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 515–524, 2013. doi:10.1145/2488608.2488673.

56 Thatchaphol Saranurak and Di Wang. Expander decomposition and pruning: Faster, stronger, and simpler. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2616–2635. SIAM, 2019. doi:10.1137/1.9781611975482.162.

57 Andrew Thomason. An extremal function for contractions of graphs. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 95, pages 261–265. Cambridge University Press, 1984.

58 L G Valiant and V V Vazirani. NP is as easy as detecting unique solutions. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing*, STOC '85, pages 458–463, New York, NY, USA, 1985. Association for Computing Machinery. doi:10.1145/22145.22196.

59 Virginia Vassilevska Williams and Ryan Williams. Subcubic equivalences between path, matrix and triangle problems. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 645–654. IEEE, 2010. doi:10.1109/FOCS.2010.67.

60 Huacheng Yu. An improved combinatorial algorithm for boolean matrix multiplication. *Information and Computation*, 261:240–247, 2018. doi:10.1016/j.ic.2018.02.006.

A Related Work

Various notions of H-freeness have been widely studied to understand the complexity of fundamental problems across different computational regimes. The three primary notions prevalent in the literature are H-minor-freeness, H-freeness (the subgraph setting), and induced H-freeness. Observe that the classes of graphs avoiding these patterns form a strict hierarchy: H-minor-free $\subset$ H-free $\subset$ induced H-free.

Because H-minor-free graphs constitute a highly restricted family, many fundamental problems that are presumed hard on general graphs become significantly easier in H-minor-free graphs for every fixed H (yielding trivial dichotomies where all patterns are classified as "easy"). Examples in the coarse complexity regime include Max-Cut and Subgraph Isomorphism, which become solvable in polynomial time [43, 39, 53]. In the fine-grained context, there are fast algorithms for Diameter, Distance Oracles, and Decremental

Reachability [31, 33, 49, 44]. For Triangle Detection (and, more generally, k-Clique Detection), H-minor-free graphs have bounded degeneracy [57, 46]. Consequently, the problem can be solved in linear time for every pattern H using the Chiba-Nishizeki algorithm, yielding a trivial dichotomy.

In the intermediate setting of H-free graphs, dichotomies are known for Max-Cut, Maximum Independent Set, List-Coloring, and (very recently) Steiner Forest [43, 11, 38, 34]. In the fine-grained context, problems such as Diameter and APSP in dense graphs admit trivial dichotomies between cubic and truly subcubic time. This arises because hard instances for these problems include bipartite graphs via well-known reductions [55, 59]. Therefore, these problems remain hard for any non-bipartite pattern H, whereas for bipartite patterns, the problems become easy due to the sparsity of H-free graphs [47]. However, other dichotomies between linear and non-linear time can be established. Recently, Johnson et al. [42] established a dichotomy for Diameter in H-free graphs between patterns solvable in linear time and those requiring quadratic time. We remark here that in the context of Triangle Detection, a dichotomy between linear and non-linear time is conditionally established: Abboud et al. [3] show a $m^{1+\Omega(1)}$ lower bound for Triangle Detection in C_4-free graphs, and they pose as an open question an extension to C_k-free graphs for every fixed $k \geq 4$. Assuming this holds, the "easy" cases (i.e., linear) are all forests, since such graphs have bounded arboricity by the Erdős-Gallai theorem (so the Chiba-Nishizeki algorithm solves them in linear time), and the "hard" cases (i.e., non-linear) are all patterns that contain a cycle (i.e., non-forests).

In the most general setting of induced H-free graphs within the coarse complexity regime, dichotomies are known for Coloring, Dominating Set, and Max-Cut [48, 45, 43]. A long line of work exists regarding the Maximum Independent Set problem in induced H-free graphs [11, 10, 36, 51], but a P/NP dichotomy has yet to be fully established. The problem was also considered in approximation and parameterized settings [28, 21]. In the fine-grained regime, the work of Oostveen et al. [52] implies a dichotomy for Diameter between linear-time and non-linear-time patterns, assuming the optimality of current algorithms for the Orthogonal Vectors problem in the high-dimensional setting (i.e., where $d = \Theta(n)$). They also demonstrate an $m^{2\omega/(\omega+1)-\varepsilon}$ (or combinatorial $m^{3/2-\varepsilon}$) lower bound that holds for almost every "hard" pattern, conditioned on the conjecture that finding a simplicial vertex requires this amount of time. Note that in weighted graphs, problems such as Diameter and APSP admit trivial dichotomies where all patterns are classified as "hard". This is because edges with infinite weight can be added to effectively form a clique (additionally, since these problems admit hard bipartite instances, they remain hard even when excluding small cliques). Weighted Max-Cut also admits a trivial dichotomy for the same reason; furthermore, it is known that the problem remains NP-hard even on triangle-free graphs.[6] For k-Clique Detection, the problem is known to be hard even for very small forbidden patterns. This follows from results on the dual k-Independent Set problem [21] (obtained via graph complementation), although several positive algorithmic results have been achieved for specific cases [40, 21].

B Generalization to k-Clique Detection

In this section, we show how our results can be extended from Triangle Detection to k-Clique Detection, i.e., from $k = 3$ to any $k \geq 3$. The motivation for studying k-Clique in H-free graphs closely parallels the $k = 3$ case discussed in the introduction. The brute-force $O(n^3)$

[6] That is because a 2-subdivision preserves the maximum cut up to an additive term of $2m$.

algorithm for Triangle Detection extends to a brute-force $O(n^k)$ algorithm for k-Clique. It has been a major open question to break this barrier without using FMM, and the conjecture that this is impossible was put forth in 2015 by Abboud, Backurs, and Vassilevska Williams [2] and subsequently used as the basis for many conditional lower bounds (e.g., [23, 26, 50, 24]).

▶ **Conjecture 14** (Combinatorial k-Clique Detection)**.** *For any $\varepsilon > 0$, there is no combinatorial k-Clique Detection algorithm that runs in $O(n^{k-\varepsilon})$ time.*

Similarly to the $k = 3$ case, one hopes that breaking the n^k barrier with a new algorithm that avoids FMM would lead to a breakthrough for one of the generalizations of the problem, such as the weighted case where no known algorithm (even with FMM) can find the minimum weight k-Clique in $O(n^{k-\varepsilon})$ time. The latter was shown to be a barrier towards faster algorithms for several problems (e.g., [9, 17, 16, 22]). Understanding the complexity of k-Clique in H-free graphs could also inform algorithmic attacks on k-Clique that employ the structure-vs-randomness paradigm. We refer the reader to [5] for further discussion on the k-Clique conjecture.

The natural extension of the dichotomy hypothesis of [6] from the $k = 3$ case to k-Clique is the following.

▶ **Hypothesis 4** (Dichotomy for k-Clique Detection in H-Free Graphs)**.** *The k-Clique Detection problem in H-free graphs is:*

- *solvable in $O(n^{k-\varepsilon})$ time via a combinatorial algorithm for some $\varepsilon > 0$, if H is k-colorable and contains at most one k-clique; and*
- *as hard as k-Clique Detection in general graphs, otherwise.*

By simple extensions of the results of [6], one arrives at a similar situation for this hypothesis compared to the $k = 3$ case: the hardness part can be proven (i.e., patterns that are not k-colorable or contain more than one k-clique are hard), whereas the algorithmic part is only partially known (several positive results can be obtained, but they do not cover all patterns).

The main result of this section is to show that the primary message of this paper applies to k-Clique as well: the *induced* variant of the dichotomy hypothesis holds if and only if the non-induced variant holds.

Formally, the main result in this section is the following theorem:

▶ **Theorem 15.** *For every k-colorable pattern H containing at most one k-clique, there exists a k-colorable pattern H_k^+ containing the same number of k-cliques as H, such that k-Clique Detection in induced H-free graphs reduces to k-Clique Detection in H_k^+-free graphs. Specifically, suppose there is an algorithm for k-Clique Detection in H_k^+-free graphs that runs in time $T(N)$ on N-vertex graphs and succeeds with probability at least $2/3$. In that case, there is an algorithm for k-Clique Detection in induced H-free graphs running in time*

$$\widetilde{O}\left(n^{k-\frac{k-2}{k-1}} + n^{k(1-\frac{1}{k-1})} \cdot T(n^{\frac{1}{k-1}})\right),$$

and succeeding with probability at least $1 - O(1/n)$.

Note that the theorem implies that if $T(N) \le N^{k-\varepsilon}$, then the running time we get is polynomially smaller than n^k: $n^{k-\varepsilon/(k-1)}$.

The generalization of the pattern H^+ to H_k^+ is as follows:

▶ **Definition 16.** *Given a pattern H, the* augmented pattern *H_k^+ is obtained by attaching two $(k-2)$-cliques to the endpoints of every non-edge of H. Formally, $V(H) \subseteq V(H_k^+)$, $E(H) \subseteq E(H_k^+)$, and for every $u, v \in V(H)$ such that $uv \notin E(H)$, we add two distinct $(k-2)$-cliques X_{uv}, Y_{uv} and edges xu, xv, yu, yv for every $x \in X_{uv}$ and $y \in Y_{uv}$.*

Note that H_k^+ admits the following structural properties: If H is k-colorable, so is H_k^+, since X_{uv} and Y_{uv} see at most two colors in a k-coloring of H and thus can be colored with the remaining $k-2$ colors. The number of k-cliques in H_k^+ is preserved (i.e., it is the same as in H). This is because X_{uv} and Y_{uv} are not adjacent to any edge in H.

To prove the theorem, we will need a reduction to Unique k-Clique Detection that preserves induced H-freeness.

▶ **Lemma 17** (Reduction to Unique k-Clique Detection). *There is a randomized reduction that, given a graph G with n vertices and m edges, outputs $\widetilde{O}(n^{k(1-\frac{1}{k-1})})$ induced subgraphs $G_1, \ldots, G_k$, such that:*

1. *If G contains a k-clique, then with probability $1 - O(1/n)$, there exists at least one G_i that contains exactly one k-clique.*
2. *Each G_i has $O(n^{\frac{1}{k-1}})$ vertices.*

The reduction runs in $\widetilde{O}(mn^{(k-2)(1-\frac{1}{k-1})})$ time.

As with triangles, the implementation is via color-coding that preserves induced H-freeness, followed by sieving. We present a color-coding algorithm below which is essentially a generalization of [32] to k-cliques.

▶ **Lemma 18** (k-Clique or $O(n^{1-\frac{1}{k-1}})$-Coloring). *For every $k \geq 2$, there is an algorithm that runs in time $O(n+m)$ and finds either a k-clique or an $O(n^{1-\frac{1}{k-1}})$-coloring.*

Proof. The algorithm is recursive. For $k = 2$ it is trivial: we just have to find a single edge in the graph. If no edge is found, it means the graph is an independent set of size n and can be colored with one color. For $k > 2$, let $\Delta = n^{1-\frac{1}{k-1}}$. As long as there is some vertex v whose degree is at least Δ, we apply the following procedure. We execute a recursive call on $G[N[v]]$ with $k' = k-1$. If a $(k-1)$-clique is found, we are done because, together with v, it forms a k-clique. Otherwise, we obtain a coloring of $N[v]$ with at most $\Delta^{1-\frac{1}{k-2}}$ distinct colors. We remove $N[v]$ from the graph and repeat this process (each time using new colors). This phase lasts at most n/Δ iterations because in each iteration we remove more than Δ vertices from the graph. Hence, the number of colors produced in this phase is at most

$$\frac{n}{\Delta} \cdot \Delta^{1-\frac{1}{k-2}} = \frac{n}{\Delta^{\frac{1}{k-2}}} = \frac{n}{(n^{1-\frac{1}{k-1}})^{\frac{1}{k-2}}} = n^{1-\frac{1}{k-1}}.$$

The running time of this phase is $O(m+n)$: The time to process $G[N[v]]$ is $O(\sum_{u \in N[v]} \deg(u))$, and a vertex is processed at most once before being colored and deleted. Hence, the total time is bounded by $O(n + \sum_{u \in V} \deg(u)) = O(n+m)$.

After the first phase is over, the maximum degree is at most $\Delta - 1$. Then, the greedy coloring algorithm produces at most Δ distinct colors (again using a fresh set of colors) in linear time [19, p. 147]. In total, the number of colors used in both phases is at most $2n^{1-\frac{1}{k-1}}$. ◀

Given Lemma 18, color-coding that preserves H-freeness follows:

▶ **Lemma 19** (Color-coding for k-Clique Detection). *There is a deterministic algorithm that reduces k-Clique Detection to k-colored k-Clique Detection, running in $O(mn^{(k-2)(1-\frac{1}{k-1})})$ time. It either finds a k-clique in G, or outputs $O(n^{k(1-\frac{1}{k-1})})$ instances, where each instance is a k-colored induced subgraph of G on $O(n^{\frac{1}{k-1}})$ vertices. Every k-clique in G is present in exactly one instance.*

Proof. The proof is similar to the case of $k = 3$: We compute a coloring using Lemma 18, refine the partition to reduce large color classes to $O(n^{\frac{1}{k-1}})$ vertices, and then produce instances for each k-tuple of colors. The number of output instance is $O(n^{k(1-\frac{1}{k-1})})$. The running time is governed by the output size which is $O(mn^{(k-2)(1-\frac{1}{k-1})})$ because each edge participates in $O((n^{1-\frac{1}{k-1}})^{k-2})$ k-tuples. ◀

The next ingredient is the sieving lemma for k-Cliques:

▶ **Lemma 20** (Sieving k-colored graphs). *There is a near-linear time reduction that, given a k-colored graph G with at most n vertices in each part, outputs $O(\log^k(n))$ induced subgraphs of G, such that if G contains a k-clique, then with probability at least $1 - O(1/n)$, one of the subgraphs contains a unique k-clique.*

This lemma is folklore and we depicted its proof for $k = 3$ in Section 3. For larger k, the generalization is straightforward: we subsample each color class (except for one of them) independently with probabilities taken from $\{1, 1/2, 1/4, \ldots, 1/n\}$. It is then possible to show that with constant probability, some subgraph obtained by the sampling procedure contains exactly one k-clique if G contains a k-clique. With another $O(\log n)$ repetitions, we can boost the success probability to $1 - O(1/n)$.

Proof of Lemma 17. By applying Lemma 19 and then Lemma 20, we obtain a reduction to Unique k-Clique Detection that preserves the induced H-freeness property (as each output instance is an induced subgrpah of G). The running time is that of color-coding times a $\widetilde{O}(1)$-factor for sieving every graph in the output. Thus, the running time is $\widetilde{O}(mn^{(k-2)(1-\frac{1}{k-1})})$, thus proving Lemma 17. ◀

Proof of Theorem 15. We apply the reduction to Unique k-Clique Detection from Lemma 17. Then, induced H-freeness of G implies that all in the output are H_k^+-free, as the existence of H_k^+ in an induced H-free graph implies the existence of at least two triangles. Moreover, if G contains a triangle, there exists some subgraph G_i which contains a triangle with high probability. Thus, we run the supposed algorithm on each G_i and output YES if at least one execution returns YES. The total running time of the reduction is

$$O(mn^{(k-2)(1-\frac{1}{k-1})} + n^{k(1-\frac{1}{k-1})}T(n^{\frac{1}{k-1}})).$$

Plugging in $m = O(n^2)$ gives the result. This completes the proof of Theorem 15 (more details appear in the proof for $k = 3$ in Section 3). ◀

Finally, the formal equivalence between Hypothesis 4 and its induced analog is established: The hard patterns in the subgraph setting extend to the induced setting naturally, while the easy patterns extend via Theorem 15. The proof of the other direction (induced $\implies$ subgraph) is straightforward. The details are similar to the $k = 3$ case (see Section 5).