




Counting Small Induced Subgraphs: Scorpions Are Easy but Not Trivial

Radu Curticapean   

University of Regensburg, Germany
IT University of Copenhagen, Denmark

Simon Döring  

Max Planck Institute for Informatics, Saarbrücken, Germany
Saarland University (SIC), Saarbrücken, Germany

Daniel Neuen  

Max Planck Institute for Informatics, Saarbrücken, Germany

Abstract

In the parameterized problem $\# \text{INDSUB}(\Phi)$ for fixed graph properties Φ , given as input a graph G and an integer k , the task is to compute the number of induced k -vertex subgraphs satisfying Φ . Dörfler et al. [Algorithmica 2022] and Roth et al. [SICOMP 2024] conjectured that $\# \text{INDSUB}(\Phi)$ is $\# \text{W}[1]$ -hard for all non-meager properties Φ , i.e., properties that are nontrivial for infinitely many k . This conjecture has been confirmed for several restricted types of properties, including all hereditary properties [STOC 2022] and all edge-monotone properties [STOC 2024].

We refute this conjecture by showing that induced k -vertex graphs that are *scorpions* can be counted in time $O(n^4)$ for all k . Scorpions were introduced more than 50 years ago in the context of the evasiveness conjecture. A simple variant of this construction results in graph properties that achieve arbitrary intermediate complexity assuming ETH.

Moreover, we formulate an updated conjecture on the complexity of $\# \text{INDSUB}(\Phi)$ that correctly captures the complexity status of scorpions and related constructions.

2012 ACM Subject Classification Theory of computation \rightarrow Fixed parameter tractability; Mathematics of computing \rightarrow Graph algorithms

Keywords and phrases induced subgraphs, counting complexity, parameterized complexity, scorpions

Digital Object Identifier 10.4230/LIPIcs.ESA.2025.96

Related Version *Full Version*: <https://arxiv.org/abs/2505.22300>

Funding The research is funded by the European Union (ERC, CountHom, 101077083). Views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.



1 Introduction

Counting small patterns in graphs is a fundamental problem in computer science, with applications in bioinformatics [39], network analysis [28, 38], databases [18], and other areas. In this paper, we focus on counting certain induced k -vertex graphs in a large n -vertex input graph G , for fixed k .

Counting Induced Copies of a Fixed Graph. The starting point of our investigation is the problem of counting induced H -copies for a fixed *individual* graph H . Formally, this problem asks for the number of sets $X \subseteq V(G)$ such that the induced subgraph $G[X]$ is isomorphic to H . We stress that H is considered fixed in this problem, and only G is the input. In particular, each such problem can be solved in time $O(n^k)$, which is polynomial in n for fixed k .



© Radu Curticapean, Simon Döring, and Daniel Neuen;
licensed under Creative Commons License CC-BY 4.0

33rd Annual European Symposium on Algorithms (ESA 2025).

Editors: Anne Benoit, Haim Kaplan, Sebastian Wild, and Grzegorz Herman; Article No. 96; pp. 96:1–96:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Some improvements over the trivial $O(n^k)$ running time are known: For example, triangles can be counted in $O(n^\omega)$ time [21], where $\omega < 2.372$ is the optimal exponent of $n \times n$ matrix multiplication [1]. This improvement can be lifted to cliques beyond K_3 : Denoting by $C(K_k)$ the optimal exponent for counting k -cliques in n -vertex graphs, similar to the exponent of matrix multiplication, we have $C(K_k) \leq \omega \cdot \lceil k/3 \rceil$ (see [30]). Under the Exponential-Time Hypothesis ETH, there exists a fixed constant α such that $C(K_k) \geq \alpha \cdot k$ (see [4]).

Writing $C_{\text{ind}}(H)$ to denote the optimal exponent of counting induced H -subgraphs [5] for fixed H , a straightforward reduction shows that $C_{\text{ind}}(H) = C(K_k)$ for all k -vertex graphs H . In other words, for fixed k , all induced H -counting problems with k -vertex H are equally hard, and they require an exponent of $\Omega(k)$ under ETH. Compare this to counting not necessarily induced subgraphs, where different patterns can yield different complexity exponents: The number of subgraph copies of the edgeless k -vertex graph IS_k in an n -vertex graph is $\binom{n}{k}$ and can thus be computed in linear time, while counting k -cliques is hard.

Counting Patterns From a Set. In recent years, counting occurrences of individual k -vertex patterns H has been generalized to counting pattern occurrences from a fixed *set* of patterns [22, 23]. In this setting, we fix a number $k \in \mathbb{N}$ and a set \mathcal{H} of k -vertex graphs. On input G , we wish to count the induced k -vertex subgraphs of G isomorphic to some $H \in \mathcal{H}$. This subsumes the problem of counting induced H -copies, but also allows us to address, e.g., the problem of counting connected k -vertex graphs [22].

Of course, for every set \mathcal{H} , this problem can be solved by counting induced H -copies for the individual graphs $H \in \mathcal{H}$, which readily implies an $O(n^{C(K_k)})$ time algorithm. Significantly faster algorithms were only known for *trivial* pattern sets \mathcal{H} , i.e., if the pattern set \mathcal{H} is empty or contains all k -vertex graphs. In these cases, the output is just 0 or $\binom{n}{k}$, respectively. Also, for some pattern sets \mathcal{H} , slight improvements can be obtained over the $O(n^{C(K_k)})$ time algorithm [9], but no nontrivial set \mathcal{H} of k -vertex graphs with exponent strictly less than $C(K_{\lfloor k/2 \rfloor})$ was known. In other words, counting induced patterns from a fixed set \mathcal{H} of k -vertex graphs appeared to be either trivial or very hard.

Parameterized Complexity. In the literature, pattern counting is often phrased in terms of graph properties Φ that may hold on infinitely many graphs rather than finite sets \mathcal{H} . More specifically, in the problem $\# \text{INDSUB}(\Phi)$ for a fixed graph property Φ , the input is a graph G and $k \in \mathbb{N}$, and we ask to count the induced k -vertex subgraphs of G satisfying Φ . Compared to the previous setting, the pattern size k is now part of the input rather than a fixed constant, so the problem $\# \text{INDSUB}(\Phi)$ may not be polynomial-time solvable. We say that $\# \text{INDSUB}(\Phi)$ is *fixed-parameter tractable* if it can be solved in time $f(k) \cdot n^{O(1)}$ for a computable function f , and we would like to understand which properties Φ render $\# \text{INDSUB}(\Phi)$ fixed-parameter tractable.

In prior literature, all known properties Φ with fixed-parameter tractable $\# \text{INDSUB}(\Phi)$ are essentially trivial: Formally, we call Φ *meager* if the restriction $\Phi^{(k)}$ of Φ to k -vertex graphs (i.e., the k -th slice of Φ) is trivial for all but finitely many $k \in \mathbb{N}$. Meager computable properties Φ trivially render $\# \text{INDSUB}(\Phi)$ fixed-parameter tractable. Complementing this, the problem $\# \text{INDSUB}(\Phi)$ was conjectured to be $\# \text{W}[1]$ -hard for all other computable properties Φ . Here, $\# \text{W}[1]$ is the parameterized analogue of $\# \text{P}$; it is known that ETH rules out fixed-parameter tractable algorithms for all $\# \text{W}[1]$ -hard problems (see, e.g., [15]).

► **Conjecture 1** ([12, 16, 37]). *For every property Φ that is computable and not meager, the problem $\# \text{INDSUB}(\Phi)$ is $\# \text{W}[1]$ -hard.*

This conjecture has been verified for general classes of properties Φ , e.g., properties that are closed under deleting edges [9, 13, 14] or deleting vertices [16], and other natural classes of properties [9, 22, 23, 35, 37]. This made it a plausible working hypothesis in the area.

Our Results. We show that **Conjecture 1 fails**. That is, we exhibit non-meager properties Φ such that $\#\text{INDSUB}(\Phi)$ is fixed-parameter tractable – in fact, even polynomial-time solvable.

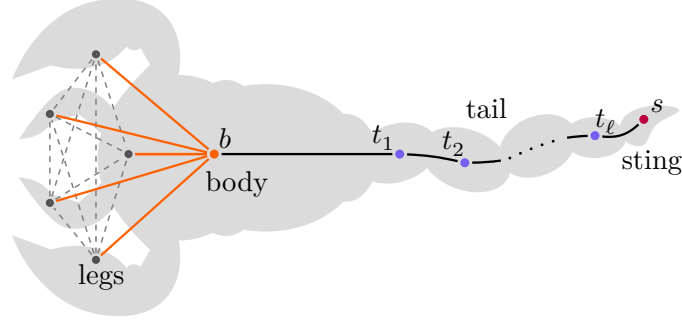
These counterexamples are derived from simple constructions that were introduced 50 years ago in the context of the *evasiveness conjecture* [27, 33, 26], a major open problem about the worst-case query complexity of graph properties. In the setting of the evasiveness conjecture, we are given a fixed graph property Φ , and we wish to decide whether an unknown k -vertex graph G is contained in Φ , where G can only be accessed via edge queries. More precisely, the vertex set $V(G)$ is known, and for two vertices v, w , we can query whether vw is an edge of G . Clearly, every property can be decided by querying all $\binom{k}{2}$ pairs of vertices, and the *evasiveness conjecture* postulates that this trivial worst-case upper bound is optimal for all (*edge-*)*monotone* properties.

However, when we drop the monotonicity requirement, properties Φ with linear query complexity $O(k)$ are known [27, 33], and it turns out that those examples can be used to refute Conjecture 1.

- As a simple counterexample, the property Ψ_{sink} of having a sink in a directed graph is nontrivial on graphs of fixed size $k \geq 2$, and hence not meager. Here, a graph D satisfies Ψ_{sink} if there is a vertex $s \in V(D)$ with $(v, s) \in E(D)$ for all $v \neq s$.¹ Nevertheless, $\#\text{INDSUB}(\Psi_{\text{sink}})$ can be solved in linear time on directed graphs without antiparallel edges. We invite the reader to discover the algorithm themselves before proceeding to Section 3.1. This property actually presented the first counterexample to a (too strong) version of the evasiveness conjecture on directed graphs [33].
- A marginally more involved construction also works for undirected graphs: The *scorpion property* Ψ is a non-meager property of undirected graphs such that $\#\text{INDSUB}(\Psi)$ can be solved in $O(n^4)$ time on general undirected graphs. Scorpions were the first counterexample to a (too strong) variant of the evasiveness conjecture for undirected graphs [27], prompting the restriction of this conjecture to monotone properties.
- We show more generally that $\#\text{INDSUB}(\Psi)$ can be made gradually harder: For every $\ell \in \mathbb{N}$, we construct a *generalized scorpion property* Ψ_ℓ such that $\#\text{INDSUB}(\Psi_\ell)$ can be solved in $O(n^{\ell+3})$ time, while ETH rules out $O(n^{\alpha \cdot \ell})$ time algorithms for a fixed constant $\alpha > 0$. Our construction also allows for ℓ to be a function in k . Under the Strong Exponential-Time Hypothesis SETH, we rule out $O(n^{\ell+2-\varepsilon})$ time algorithms.

Finally, informed by these counterexamples, we formulate a new hypothesis on the computational complexity of $\#\text{INDSUB}(\Phi)$ for properties Φ . This new hypothesis is more technical and explained in Section 4. In a nutshell, it is based around the well-established fact that sums of induced pattern counts like the k -th slices $\#\text{IndSub}(\Phi^{(k)} \rightarrow G)$ can be expressed as linear combinations of (not necessarily induced) subgraph counts [5], and that such basis changes may help in understanding the complexity of a problem [5, 6, 9, 12, 13, 14, 16, 17, 24, 35, 37, 36].

¹ Contrary to contemporary notation, in which a *sink* is a vertex of out-degree 0, we use the term *sink* to refer to a vertex $s \in V(D)$ such that $(v, s) \in E(D)$ for all $v \neq s$. This follows the terminology of the original paper [33], which is still common in the context of the evasiveness conjecture.



■ **Figure 1** A graph H is an ℓ -scorpion if it has the above form: Dashed edges *may* be present in H or not, solid edges *must* be present, and non-drawn edges *must not* be present.

More specifically, the new hypothesis postulates that a useful phenomenon occurs when expressing the graph parameter $\#\text{IndSub}(\Phi^{(k)} \rightarrow G)$ as a linear combination of subgraph counts: Any hard term in such a linear combination ensures hardness of the entire linear combination. General linear combinations of k -vertex subgraph counts do not enjoy this useful phenomenon. We demonstrate in Section 4 that our conjecture would imply useful new ways of proving hardness of $\#\text{IndSub}(\Phi^{(k)} \rightarrow G)$.

Further Connections to the Evasiveness Conjecture. Notably, this work is not the first to draw connections between the evasiveness conjecture and the computational complexity of $\#\text{INDSUB}(\Phi)$. Indeed, this connection was already made by Roth and Schmitt [35], who adapted the topological approach to the evasiveness conjecture, dating back to Kahn, Saks and Sturtevant [25], to prove lower bounds for $\#\text{INDSUB}(\Phi)$ for certain graph properties Φ . In a similar spirit, [9] adapts a lower bound of $\Omega(k^2)$ on the query complexity of monotone graph properties due to Rivest and Vuillemin [32] to obtain hardness of $\#\text{INDSUB}(\Phi)$ for monotone properties Φ . The present work complements these connections on the upper-bound side: We observe that standard examples of properties Φ with linear query complexity admit efficient algorithms for $\#\text{INDSUB}(\Phi)$.

Currently, these connections seem to be restricted to specific examples and proof techniques. We believe that it is an interesting open question whether a general connection can be established in either direction. Without going into details, we point out that the *Fourier degree* of a property Φ (when Φ is restricted to a slice and viewed as a Boolean function) provides a lower bound both for the query complexity of Φ [31], and for the computational complexity of $\#\text{INDSUB}(\Phi)$ [9]. However, in both cases, large Fourier degree is only known to give a *sufficient* criterion for hardness, so it does not establish a direct link between the query complexity of Φ and the complexity of $\#\text{INDSUB}(\Phi)$.

2 Preliminaries

We write $\mathbb{N} = \{1, 2, 3, \dots\}$ and $[n] := \{1, \dots, n\}$ for $n \in \mathbb{N}$. For a set A , we write $\binom{A}{k}$ for the set of all k -element subsets of A .

Graph Theory. We follow standard textbooks [11] for graph-theoretic notation. Unless stated otherwise, graphs are simple (i.e., without multiedges or self-loops) and undirected. We write $V(G)$ and $E(G)$ for the vertex and edge set of G , respectively, and we write $N_G(v) := \{u \mid uv \in E(G)\}$ for the neighborhood of $v \in V(G)$ in G .

A graph H is a *subgraph* of G , written $H \subseteq G$, if it can be obtained by deleting vertices and edges from G . For $X \subseteq V(G)$, we write $G[X]$ to denote the *induced subgraph* on X . For $S \subseteq E(G)$, we write $G[S]$ for the subgraph with vertex set $V(G)$ and edge set S . For $k, \ell \in \mathbb{N}$, we write K_ℓ for the complete ℓ -vertex graph, IS_ℓ for the edgeless ℓ -vertex graph, and $K_{\ell,k}$ for the complete bipartite graph on $\ell + k$ vertices.

Induced Subgraph Counts. A *graph property* Φ is a function that maps each graph G to $\{0, 1\}$ and is invariant under isomorphisms, i.e., $\Phi(G) = \Phi(H)$ for all isomorphic graphs G, H . For $k \in \mathbb{N}$, the k -th *slice* of Φ , denoted by $\Phi^{(k)}$, is the restriction of Φ to k -vertex graphs. We implicitly identify $\Phi^{(k)}$ with the set of all k -vertex graphs G satisfying $\Phi(G) = 1$, and we use graph properties and sets of graphs interchangeably. For $k \in \mathbb{N}$ and a graph G , the number of k -vertex induced subgraphs $G[X]$ satisfying Φ will be denoted by

$$\#\text{IndSub}(\Phi^{(k)} \rightarrow G) := \sum_{\substack{X \subseteq V(G) \\ |X|=k}} \Phi(G[X]).$$

Moreover, we write $\#\text{IndSub}(\Phi^{(k)} \rightarrow \star)$ for the map $G \mapsto \#\text{IndSub}(\Phi^{(k)} \rightarrow G)$.

Complexity Theory. A *parameterized problem* consists of a function $P: \Sigma^* \rightarrow \mathbb{N}$ and a computable parameterization $\kappa: \Sigma^* \rightarrow \mathbb{N}$. It is *fixed-parameter tractable* (FPT) if there is a computable function f , a constant $c \in \mathbb{N}$, and a deterministic algorithm \mathbb{A} that computes $P(x)$ in time $O(f(\kappa(x)) \cdot |x|^c)$ for all $x \in \Sigma^*$. We write $\#\text{INDSUB}(\Phi)$ for the parametrized problem that gets as input a graph G and a *parameter* k , and computes $\#\text{IndSub}(\Phi^{(k)} \rightarrow G)$.

For lower bounds, we rely on the *Exponential-Time Hypothesis* (ETH) [20], which asserts the existence of some $\varepsilon > 0$ such that the Boolean satisfiability problem on n -variable 3-CNF formulas cannot be solved in time $O(2^{\varepsilon \cdot n})$ (see also [10, Conjecture 14.1]). The *Strong Exponential-Time Hypothesis* (SETH) [19] states that for all $\varepsilon > 0$, there is a $k \geq 3$ such that Boolean Satisfiability on k -CNF formulas cannot be solved in $O(2^{(1-\varepsilon) \cdot n})$ (see also [10, Conjecture 14.2]).

3 Main Result

To present the idea underlying the tractability of scorpions, we first consider a variant for directed graphs, where this idea becomes particularly simple. Then we introduce scorpions and their generalizations and prove the claimed upper and lower complexity bounds. Finally, we use scorpions to show that a useful and previously known complexity lower bound on the complexity of $\#\text{INDSUB}(\Phi)$ is tight.

3.1 Directed Graphs Containing a Sink

As a warm-up, we consider a property Ψ_{sink} of directed graphs that is nontrivial but yields a linear-time counting problem $\#\text{INDSUB}(\Psi_{\text{sink}})$. For this subsection, we momentarily consider directed graphs without antiparallel edges, i.e., at most one of the edges (u, v) and (v, u) may be present. The property Ψ_{sink} is defined to hold on H if there is a *sink* vertex $s \in V(H)$, i.e., a vertex s such that $(u, s) \in E(H)$ for all $u \in V(H) \setminus \{s\}$. This property is clearly nontrivial in every slice $k \geq 2$, as there are k -vertex graphs with a sink and k -vertex graphs without a sink (e.g., an in-star versus the edgeless graph).

Towards an algorithm, we observe crucially that every graph without antiparallel edges contains at most one sink, since two distinct sinks u, v would imply the presence of both edges (u, v) and (v, u) . Hence, for an input graph G , the set of k -vertex sets containing a sink

$$\mathcal{X} := \left\{ X \in \binom{V(G)}{k} \mid G[X] \in \Psi_{\text{sink}} \right\}$$

can be partitioned, according to the unique sink, into

$$\mathcal{X} = \bigcup_{v \in V(G)} \mathcal{X}_v \quad \text{with} \quad \mathcal{X}_v := \{X \in \mathcal{X} \mid v \text{ is the sink of } G[X]\}.$$

Finally, for fixed $v \in V(G)$, every set $X \in \mathcal{X}_v$ has the form $X = \{v, w_1, \dots, w_{k-1}\}$ with all w_1, \dots, w_{k-1} pairwise distinct, distinct from v , and incoming neighbors of v , i.e., they satisfy $(w_i, v) \in E(G)$. Writing $\text{in}_G(v)$ for the number of incoming neighbors of v , it follows that

$$|\mathcal{X}_v| = \binom{\text{in}_G(v)}{k-1}.$$

Combining the above equations, we readily obtain a linear-time algorithm for $\#\text{INDSUB}(\Psi_{\text{sink}})$ by computing $\text{in}_G(v)$ for all $v \in V(G)$ and evaluating the resulting formula

$$\#\text{IndSub}(\Psi_{\text{sink}}^{(k)} \rightarrow G) = \sum_{v \in V(G)} |\mathcal{X}_v| = \sum_{v \in V(G)} \binom{\text{in}_G(v)}{k-1}. \quad (1)$$

3.2 Generalized Scorpions

The algorithmic idea for counting k -vertex graphs with a sink vertex applies whenever the set \mathcal{X} of induced subgraphs to be counted admits a partition into few sets \mathcal{X}_i such that each $|\mathcal{X}_i|$ is easily determined. More specifically, we consider partitions in which each \mathcal{X}_i is determined by the manifestation of a special small set of uniquely identifiable vertices (e.g., the sink vertex), while the subgraph induced by the other vertices is irrelevant.

To apply this idea to undirected graphs, we use a construction that was first presented in [27] for the so-called *evasiveness conjecture*, and which has become a standard example in this context (see, e.g., [26, Section 13.1]): A graph H is a *scorpion* if it can be obtained from an arbitrary graph H' with $|V(H')| \geq 2$ by adding fresh vertices b, t, s , making b adjacent to all of H' , and then adding the edges bt and ts . The vertices b, t, s are usually called *body*, *tail* and *sting*, and it can be shown crucially (see Lemma 3 below) that these vertices are uniquely recoverable from H . Similarly to the arguments above, this allows us (in Theorem 4) to design an efficient algorithm to count induced scorpions.

In this paper, we prove these statements for a slightly generalized version of scorpions, since this allows us to obtain gradually harder properties. Towards this end, we replace the tail vertex t by a path of ℓ vertices t_1, \dots, t_ℓ . See also Figure 1.

► **Definition 2.** For $\ell \in \mathbb{N}$, an ℓ -scorpion is a graph H with $|V(H)| \geq \ell + 4$ that admits a tuple of pairwise distinct vertices

$$\left(\underbrace{b}_{\text{body of } H}, \underbrace{t_1, \dots, t_\ell}_{\text{tail of } H}, \underbrace{s}_{\text{sting of } H} \right) \in V(H)^{\ell+2},$$

such that the following holds: Writing $Q := \{b, t_1, \dots, t_\ell, s\}$ and calling the vertices in $V(H) \setminus Q$ the legs of H , we have that

- the graph $H[Q]$ is an induced path from the body b to the sting s ,
- the body b is adjacent to all legs, and
- the body b is the only vertex in Q adjacent to legs.

We define Ψ_ℓ as the class of all ℓ -scorpions.

The property Ψ_ℓ is non-meager for all $\ell \geq 1$: Indeed, for $k \geq \ell + 4$, at least one k -vertex scorpion exists (e.g., consisting of the sting, tail, body, and an independent set of legs), while the k -vertex graph K_k is not a scorpion.

Note that scorpion graphs are 1-scorpions. Also note that the definition speaks about “the” body, tail, and sting vertices, as if they were unique. Indeed, they are:

► **Lemma 3.** *If H is an ℓ -scorpion, then its body, tail, and sting are unique.*

Proof. Let $k = |V(H)|$ and recall that $k \geq \ell + 4$ by Definition 2. The body has degree $k - \ell - 1 \geq 3$. Since every leg has degree at most $k - \ell - 2$, we conclude that H contains exactly one vertex of degree $k - \ell - 1$; this uniquely identifies b .

The vertices not adjacent to b are precisely t_2, \dots, t_ℓ, s . Among these, s is the only vertex of degree 1 and is thus uniquely identified. The vertices t_1, \dots, t_ℓ are uniquely identified by their distance from s , since vertex t_i is the only vertex at distance $\ell - i + 1$ from s . ◀

We use this lemma to show that $\#\text{IndSub}(\Psi_\ell)$ can be solved in polynomial time for every fixed $\ell \geq 1$, similarly to the algorithm presented in Section 3.1.

► **Theorem 4.** *There is an algorithm that, given $k \geq \ell + 4$ and an n -vertex graph G as input, computes $\#\text{IndSub}(\Psi_\ell^{(k)} \rightarrow G)$ in time $O(\ell \cdot n^{\ell+3})$.*

Proof. Let \mathcal{X} be the set of induced k -vertex graphs in G that are ℓ -scorpions. In our proof, we use the uniqueness of body, tail, and sting to partition \mathcal{X} into classes $\mathcal{X}_{\mathbf{q}}$ for $\mathbf{q} \in V(G)^{\ell+2}$ and then determine each $|\mathcal{X}_{\mathbf{q}}|$ in linear time. Then the algorithm follows from

$$\#\text{IndSub}(\Psi_\ell^{(k)} \rightarrow G) = |\mathcal{X}| = \sum_{\mathbf{q} \in V(G)^{\ell+2}} |\mathcal{X}_{\mathbf{q}}|. \quad (2)$$

More specifically, let $\mathcal{P} \subseteq V(G)^{\ell+2}$ denote the tuples inducing a path in G . Given $\mathbf{q} \in \mathcal{P}$ with $\mathbf{q} = (b, t_1, \dots, t_\ell, s)$, let $\mathcal{X}_{\mathbf{q}} \subseteq \mathcal{X}$ denote the set of k -vertex ℓ -scorpions in G with body b , tail t_1, \dots, t_ℓ and sting s . By Lemma 3, the sets $\mathcal{X}_{\mathbf{q}}$ for $\mathbf{q} \in \mathcal{P}$ partition \mathcal{X} , so (2) holds with $\mathcal{X}_{\mathbf{q}} = \emptyset$ for $\mathbf{q} \notin \mathcal{P}$. It remains to determine $|\mathcal{X}_{\mathbf{q}}|$ for $\mathbf{q} = (b, t_1, \dots, t_\ell, s) \in \mathcal{P}$. We show that

$$|\mathcal{X}_{\mathbf{q}}| = \binom{|X_G(\mathbf{q})|}{k - \ell + 2} \quad \text{with } X_G(\mathbf{q}) := \left| N_G(b) \setminus \bigcup_{v \in \{t_1, \dots, t_\ell, s\}} N_G(v) \right|. \quad (3)$$

Indeed, since the $\ell + 2$ vertices in \mathbf{q} induce a path, the set $\mathcal{X}_{\mathbf{q}}$ consists of all k -vertex sets X that contain $\{b, t_1, \dots, t_\ell, s\}$ and $k - \ell - 2$ additional vertices (the legs) that are adjacent to b and not adjacent to any t_i or s . The number of such sets X is precisely $|\mathcal{X}_{\mathbf{q}}| = \binom{|X_G(\mathbf{q})|}{k - \ell + 2}$.

An algorithm with the claimed running time follows from evaluating (2) term by term while using (3) to determine $|\mathcal{X}_{\mathbf{q}}|$. Indeed, \mathcal{P} can be enumerated in time $O(n^{\ell+2})$ and $|\mathcal{X}_{\mathbf{q}}|$ for $\mathbf{q} \in \mathcal{P}$ can be computed via (3) in time $O(\ell n)$. ◀

Since Ψ_ℓ is non-meager for every $\ell \geq 1$, Theorem 4 refutes Conjecture 1 even with $\ell = 1$.

3.3 Lower Bounds Based on ETH

Next, we show that the running time obtained in Theorem 4 is essentially optimal under ETH. In particular, by choosing ℓ as a function of k , we can use ℓ -scorpions to obtain properties with varying computational difficulty.

To obtain the lower bound, we rely on a result from [6, 9, 37] that provides a lower bound based on the set of possible Hamming weights attained by a property Φ : Given a graph property Φ and $k \in \mathbb{N}$, we say that Φ *attains weight ℓ on slice k* if there is a graph $H \in \Phi$ with k vertices and ℓ edges. It *avoids weight ℓ on slice k* if it does not attain it. Note that every property attains between 0 and $\binom{k}{2} + 1$ different weights on slice k , since the attained weights form a subset of $\{0, \dots, \binom{k}{2}\}$.

► **Theorem 5** ([9, Lemma 5.1] & [6, Theorem 7.1]). *Assuming ETH, there are $N_0, \delta > 0$ such that the following holds: If $k \geq N_0$ and $0 < d \leq k/2$ and Φ avoids at least $d \cdot k$ distinct weights and attains at least one weight, all on slice k , then no algorithm computes $\#\text{IndSub}(\Phi^{(k)} \rightarrow G)$ in time $O(n^{\delta \cdot d})$.*

For $k \geq \ell + 4$, we can easily determine the number of weights attained by Ψ_ℓ on slice k : Since there is no freedom in choosing edges incident to body, tail and sting, this number is precisely $\binom{k-\ell-2}{2} + 1$. Thus, the number of weights avoided by Ψ_ℓ on slice k is

$$\binom{k}{2} - \binom{k-\ell-2}{2} - 1 = k(\ell+2) - \frac{\ell(\ell+5)}{2} - 4.$$

For $k \geq 2\ell + 4$, it follows that Ψ_ℓ avoids at least $\ell/2 \cdot k$ weights on slice k . Being nontrivial, it attains at least one weight. Theorem 5 readily implies:

► **Corollary 6.** *Assuming ETH, there are $N_0, \delta > 0$ such that the following holds: If $k \geq N_0$ and $0 < \ell \leq (k-4)/2$, then no algorithm computes $\#\text{IndSub}(\Psi_\ell^{(k)} \rightarrow G)$ in time $O(n^{\delta \cdot \ell})$.*

► **Remark 7.** The algorithm from Theorem 4 also demonstrates that the lower bound in Theorem 5 cannot be further improved. Indeed, Ψ_ℓ avoids at most $(\ell+2) \cdot k$ weights on slice k , but $\#\text{IndSub}(\Phi^{(k)} \rightarrow G)$ can be solved in $O(n^{\ell+3})$ time. Therefore, ℓ -scorpions show that the bound in Theorem 5 is essentially tight.

With the lower bound of Corollary 6 at hand, we can obtain properties with varying computational complexity, by choosing ℓ dependent on k . For example, setting $\ell \approx \sqrt{k}$ yields a property $\Psi_{\text{sqr}}t$ for which $\#\text{INDSUB}(\Psi_{\text{sqr}}t)$ takes $n^{\Theta(\sqrt{k})}$ time under ETH. More generally, consider a monotone increasing $f: \mathbb{N} \rightarrow \mathbb{N}$ with $1 \leq f(k) \leq (k-4)/2$ for all $k \in \mathbb{N}$. We define $\Psi_f = \bigcup_{k \in \mathbb{N}} \Psi_{f(k)}^{(k)}$ to contain exactly the k -vertex $f(k)$ -scorpions for all $k \in \mathbb{N}$.

► **Corollary 8.** *Let $f: \mathbb{N} \rightarrow \mathbb{N}$ be monotone increasing with $1 \leq f(k) \leq (k-4)/2$. Then $\#\text{INDSUB}(\Psi_f)$ can be solved in time $O(k \cdot n^{f(k)+3})$, and assuming ETH, not in time $O(n^{\alpha \cdot f(k)})$ for a fixed constant $\alpha > 0$.*

In the full version of the paper, we combine results from Section 4 with a result from [3] to obtain the following lower bound under the Strong Exponential-Time Hypothesis SETH.

► **Proposition 9.** *Assuming SETH, for every $\ell \geq 3$ and $\varepsilon > 0$, there is no algorithm solving $\#\text{INDSUB}(\Psi_\ell)$ in time $O(n^{\ell+2-\varepsilon})$.*

We stress that, in contrast to Corollaries 6 and 8, the SETH lower bound from Proposition 9 only holds for fixed ℓ , and not if ℓ is chosen as a function of k .

4 An Updated Conjecture Using the Subgraph Basis

Seeing what scorpions have done to Conjecture 1, it is natural to update the conjecture and reconsider the question when $\# \text{INDSUB}(\Phi)$ is fixed-parameter tractable. Towards this, we first observe that our counterexamples can be generalized in a number of ways while keeping their key features. This suggests that a dichotomy theorem for $\# \text{INDSUB}(\Phi)$ will have to encompass various tractable cases, as the following facts indicate:

- The key to the efficient algorithm in Theorem 4 is that body, tail and sting are uniquely identified in every ℓ -scorpion, and that no restrictions are imposed on the subgraph induced by the legs. As long as there is a constant-sized set of uniquely identifiable vertices and no restrictions on the remaining vertices, an efficient algorithm for $\# \text{INDSUB}(\Phi)$ follows.
- This however is not the final word: Observe that $\# \text{INDSUB}(\Phi)$ has the same complexity as $\# \text{INDSUB}(\neg\Phi)$, where $\neg\Phi$ contains exactly those graphs that are not contained in Φ (see, e.g., [37, Fact 2.3]). Notably, the non-scorpion property $\neg\Psi_\ell$ contains the complete graph on k vertices for every $k \geq \ell + 4$, which arguably has no constant-sized set of “uniquely identifiable” vertices.
- Beyond, we can also “nest” easy properties: For example, let Λ be the graph property containing all graphs H with $|V(H)| \geq 9$ so that (a) H is a 1-scorpion, with some set of legs X , and (b) $H[X]$ is not a 2-scorpion. Then $\# \text{INDSUB}(\Lambda)$ can be solved in polynomial time using similar arguments as in Theorem 4.

The diversity of these examples suggests that a complexity classification for all properties Φ may be quite intricate. To obtain a new classification conjecture, we first express the induced subgraph counts arising in $\# \text{INDSUB}(\Phi)$ as linear combinations of (not necessarily induced) subgraph counts. Such basis changes among counting problems have already been used for $\# \text{INDSUB}(\Phi)$ before [6, 9, 12, 13, 14, 16, 35, 37].²

In the following, recall that $\Phi^{(k)}$ for $k \in \mathbb{N}$ is the restriction of Φ to k -vertex graphs and let $\# \text{IndSub}(\Phi^{(k)} \rightarrow \star)$ be the number of induced k -vertex graphs satisfying $\Phi^{(k)}$ in an input graph. Likewise let $\# \text{Sub}(H \rightarrow \star)$ denote the number of not necessarily induced H -subgraph copies in an input graph. Then there exists a finite set \mathcal{H} of unlabeled graphs on exactly k vertices, and coefficients α_H for $H \in \mathcal{H}$, such that

$$\# \text{IndSub}(\Phi^{(k)} \rightarrow \star) = \sum_{H \in \mathcal{H}} \alpha_H \cdot \# \text{Sub}(H \rightarrow \star); \quad (4)$$

see [2, 5]. In fact, an explicit formula for the coefficients can be found through inclusion-exclusion: Given a graph property Φ and graph H , the coefficient α_H in (4) is given by the so-called *alternating enumerator* $\widehat{\Phi}(H)$ (see [9, 35]), sometimes defined without the $(-1)^{|E(H)|}$ factor below [12, 13, 14],

$$\widehat{\Phi}(H) := (-1)^{|E(H)|} \sum_{S \subseteq E(H)} (-1)^{|S|} \Phi(H[S]).$$

Evaluating fixed finite linear combinations of k -vertex graphs as in (4) is asymptotically no harder than evaluating the individual H -subgraph counts with $\widehat{\Phi}(H) \neq 0$. For fixed graphs H , individual H -subgraph counts in turn can be evaluated in time $O(n^{\tau(H)+1})$, where

² Going further, the problem $\# \text{INDSUB}(\Phi)$ has already been expressed in the similar basis of *homomorphism counts*, and a dichotomy criterion was shown in terms of this basis [5]. However, the transformation into the homomorphism basis renders combinatorial interpretations of Φ opaque and only yields an implicit dichotomy criterion. In particular, all complexity results listed above were achieved using the *subgraph* basis rather than the homomorphism basis.

$\tau(H)$ is the vertex-cover number of H (see [5, Theorem 1.1]). In particular, this implies that $\#\text{INDSUB}(\Phi)$ is fixed-parameter tractable when only graphs H of small vertex-cover number satisfy $\widehat{\Phi}(H) \neq 0$. More quantitatively, writing $\tau_\Phi(k)$ for the maximal vertex cover number among k -vertex graphs H with $\widehat{\Phi}(H) \neq 0$, we have:

► **Theorem 10** (see, e.g., [8, Theorem 3.6]). *For every computable graph property Φ , the problem $\#\text{INDSUB}(\Phi)$ can be solved in time $f(k) \cdot n^{\tau_\Phi(k)+1}$ for a computable function f .*

Conversely, under certain conditions, a k -vertex graph H with $\widehat{\Phi}(H) \neq 0$ that is hard on its own also implies hardness of the entire linear combination. For example, under the guarantee that all patterns in the linear combination have the same number k of vertices, this occurs when some graph H with $\widehat{\Phi}(H) \neq 0$ has large treewidth (see, e.g., [12]). Large treewidth is however only a sufficient criterion, since counting k -matchings (and more generally, patterns of large vertex-cover number) is hard as well [7].

Knowing about a similar phenomenon for the related case of homomorphism counts, one³ might be led to believe that linear combinations of k -vertex subgraph counts are hard if at least one pattern has large vertex-cover number. However, this hypothesis is too optimistic, as shown by k -partial determinants: These are weighted counts of k -vertex cycle covers C in a graph G , where C is counted with weight -1 if it contains an odd number of cycles, and with weight 1 otherwise. The patterns in the subgraph count expansion of k -partial determinants are precisely the cycle covers on k vertices. These do have large vertex-cover number, yet k -partial determinants can be computed in time $O(n^{\omega+1})$ due to the same cancellations that render usual determinants tractable.

In our updated version of Conjecture 1, we assert that such cancellations cannot occur for $\#\text{INDSUB}(\Phi)$. That is, we assert that graphs of large vertex-cover number in the subgraph expansion of $\#\text{INDSUB}(\Phi)$ indeed render the linear combination hard. Recall that $\tau_\Phi(k)$ denotes the maximal vertex cover number among k -vertex graphs H with $\widehat{\Phi}(H) \neq 0$.

► **Conjecture 11.** *If Φ is a computable property and the function τ_Φ is unbounded, then the problem $\#\text{INDSUB}(\Phi)$ is $\#\text{W}[1]$ -hard.*

Note that, by Theorem 10, the problem $\#\text{INDSUB}(\Phi)$ is fixed-parameter tractable if τ_Φ is bounded. Hence, Conjecture 11 formulates a complete characterization in terms of the subgraph basis expression of a property: For a computable property Φ , the problem $\#\text{INDSUB}(\Phi)$ is fixed-parameter tractable if τ_Φ is bounded, and $\#\text{W}[1]$ -hard otherwise.

To the best of our knowledge, no counterexamples are known for Conjecture 11. Since k -partial determinants can be negative, they cannot be written as $\#\text{INDSUB}(\Phi)$ for a property Φ , so they do not form a counterexample to our updated conjecture. Moreover, we verify that none of the generalized scorpion properties Ψ_ℓ for fixed $\ell \in \mathbb{N}$ refute it. More specifically, we prove that, for every $k \geq \ell + 4$,

$$\tau_{\Psi_\ell}(k) = \ell + 2. \quad (5)$$

Thus $\tau_{\Psi_\ell}(k) \in O(1)$ for fixed ℓ , so Ψ_ℓ does not satisfy the premises of Conjecture 11.

4.1 Scorpions in the Subgraph Basis

We prove (5) in this section. Recall the definition of tail, sting, body, and legs of an ℓ -scorpion S . We call an ℓ -scorpion S a *skeleton* if its legs form an independent set. Moreover, an ℓ -scorpion *fossil* is any graph S' obtained from an ℓ -scorpion skeleton S by adding an arbitrary

³ Some did, see a superseded arXiv version of a related paper: <https://arxiv.org/abs/2407.07051v1>

number of edges uv with $u, v \in V(S)$ such that at least one of u, v is not a leg. (Considering Figure 1, a scorpion skeleton is obtained by removing all dashed edges. A scorpion fossil is obtained from a scorpion skeleton by adding arbitrary edges, but not between legs.)

To show (5), we prove the stronger statement that the graphs H occurring with non-zero coefficients $\widehat{\Psi}_\ell(H) \neq 0$ in the subgraph expansion of Ψ_ℓ are precisely the ℓ -scorpion fossils. This is indeed stronger: The vertex-cover number of ℓ -scorpion fossils is at most $\ell + 2$, since they retain the independent set on the legs. On the other hand, the augmented biclique $K_{\ell+2, k-\ell-2}^+$ obtained from a complete bipartite graph by turning the left side (one $\ell + 2$ vertices) into a clique is an ℓ -scorpion fossil of vertex-cover number $\ell + 2$.

► **Lemma 12.** *For every $\ell \in \mathbb{N}$ and every k -vertex graph H with $k \geq \ell + 4$, we have $\widehat{\Psi}_\ell(H) \neq 0$ if and only if H is an ℓ -scorpion fossil.*

Proof. Let $\mathcal{S} := \{S \subseteq E(H) \mid H[S] \in \Psi_\ell\}$. Similar to Theorem 4, we use the uniqueness of body, tail, and sting to partition \mathcal{S} into classes $\mathcal{S}_{\mathbf{q}}$ for $\mathbf{q} \in V(H)^{\ell+2}$. More specifically, for a tuple $\mathbf{q} = (b, t_1, \dots, t_\ell, s) \in V(H)^{\ell+2}$, let $\mathcal{S}_{\mathbf{q}}$ denote the set of all $S \in \mathcal{S}$ such that the scorpion $H[S]$ has body b , tail t_1, \dots, t_ℓ and sting s . Note that we may have $\mathcal{S}_{\mathbf{q}} = \emptyset$.

By Lemma 3, the sets $\mathcal{S}_{\mathbf{q}}$ for $\mathbf{q} \in V(H)^{\ell+2}$ partition \mathcal{S} , which implies

$$\widehat{\Psi}_\ell(H) = (-1)^{|E(H)|} \cdot \sum_{S \in \mathcal{S}} (-1)^{|S|} = (-1)^{|E(H)|} \cdot \sum_{\mathbf{q} \in V(H)^{\ell+2}} \sum_{S \in \mathcal{S}_{\mathbf{q}}} (-1)^{|S|}. \quad (6)$$

In the following, for a tuple $\mathbf{q} \in V(H)^{\ell+2}$, we write $\text{set}(\mathbf{q})$ for the set of entries of \mathbf{q} .

▷ **Claim 13.** If $\mathbf{q} \in V(H)^{\ell+2}$ is such that $\text{set}(\mathbf{q})$ is not a vertex cover of H , then

$$\sum_{S \in \mathcal{S}_{\mathbf{q}}} (-1)^{|S|} = 0.$$

Proof. Let us first observe that the statement is trivial if $\mathcal{S}_{\mathbf{q}} = \emptyset$.

Let $E_{\mathbf{q}} := \{e \in E(H) \mid e \cap \text{set}(\mathbf{q}) = \emptyset\}$ be the edges of H without an endpoint in $\text{set}(\mathbf{q})$. Since $\text{set}(\mathbf{q})$ is not a vertex cover, we get $E_{\mathbf{q}} \neq \emptyset$. Observe that $F_{\mathbf{q}} := S \setminus E_{\mathbf{q}} = S' \setminus E_{\mathbf{q}}$ for all $S, S' \in \mathcal{S}_{\mathbf{q}}$, because edges incident to body, tail or sting (i.e., edges incident to the vertices in \mathbf{q}) are fixed. Since edges between leg vertices can be chosen arbitrarily, we conclude that

$$\mathcal{S}_{\mathbf{q}} = \{F_{\mathbf{q}} \cup E'_{\mathbf{q}} \mid E'_{\mathbf{q}} \subseteq E_{\mathbf{q}}\}.$$

In particular, we obtain

$$\sum_{S \in \mathcal{S}_{\mathbf{q}}} (-1)^{|S|} = \sum_{E'_{\mathbf{q}} \subseteq E_{\mathbf{q}}} (-1)^{|F_{\mathbf{q}}| + |E'_{\mathbf{q}}|} = (-1)^{|F_{\mathbf{q}}|} \sum_{E'_{\mathbf{q}} \subseteq E_{\mathbf{q}}} (-1)^{|E'_{\mathbf{q}}|} = 0$$

since $E_{\mathbf{q}} \neq \emptyset$. ◁

Now, for the forward direction of the lemma, suppose that $\widehat{\Psi}_\ell(H) \neq 0$. Then there is some $\mathbf{q} = (b, t_1, \dots, t_\ell, s)$ such that $\sum_{S \in \mathcal{S}_{\mathbf{q}}} (-1)^{|S|} \neq 0$. This implies that

- (a) $\text{set}(\mathbf{q})$ is a vertex cover of H by Claim 13, so the vertices in $V(H) \setminus \text{set}(\mathbf{q})$ form an independent set in H , and
- (b) $\mathcal{S}_{\mathbf{q}} \neq \emptyset$, i.e., there is some $S \subseteq E(H)$ such that $H[S]$ is an ℓ -scorpion with body b , tail t_1, \dots, t_ℓ and sting s . Thus, the vertices in $V(H) \setminus \text{set}(\mathbf{q})$ form the legs of $H[S]$.

These two properties together directly imply that H is an ℓ -scorpion fossil.

For the backward direction, suppose H is an ℓ -scorpion fossil, and let $\mathbf{q} = (b, t_1, \dots, t_\ell, s)$ denote the tuple of body, tail, and sting of an underlying ℓ -scorpion skeleton. Then

- (i) $\mathcal{S}_{\mathbf{q}} \neq \emptyset$,
- (ii) $\text{set}(\mathbf{q})$ is a vertex cover of H .

The first item holds because the underlying ℓ -scorpion skeleton is a witness to $\mathcal{S}_{\mathbf{q}} \neq \emptyset$, while the second item holds from the definition of ℓ -scorpion skeletons.

Now, let $A \subseteq V(H)^{\ell+2}$ denote the set of all tuples \mathbf{q}' satisfying (i) and (ii). Then, for every $\mathbf{q}' \in A$ and every $S \in \mathcal{S}_{\mathbf{q}'}$, the legs of $H[S]$ form an independent set, since the vertices in $\mathbf{q}' \in A$ form a vertex-cover. This implies that $H[S]$ is an ℓ -scorpion skeleton, so S contains exactly $k-1$ edges, and we obtain

$$\sum_{S \in \mathcal{S}_{\mathbf{q}'}} (-1)^{|S|} = (-1)^{k-1} |\mathcal{S}_{\mathbf{q}'}| \quad \text{for all } \mathbf{q}' \in A. \quad (7)$$

On the other hand, for $\mathbf{q}' \in V(H)^{\ell+2} \setminus A$, the tuple \mathbf{q}' violates (i), or \mathbf{q}' violates (ii). We obtain directly (in the first case) or via Claim 13 (in the second case) that

$$\sum_{S \in \mathcal{S}_{\mathbf{q}'}} (-1)^{|S|} = 0 \quad \text{for all } \mathbf{q}' \in V(H)^{\ell+2} \setminus A. \quad (8)$$

Let us abbreviate $X_{\mathbf{q}'} := \sum_{S \in \mathcal{S}_{\mathbf{q}'}} (-1)^{|S|}$. Then it follows that

$$\begin{aligned} \widehat{\Psi}_{\ell}(H) &= (-1)^{|E(H)|} \sum_{\mathbf{q}' \in V(H)^{\ell+2}} X_{\mathbf{q}'} \\ &= (-1)^{|E(H)|} \left(\sum_{\mathbf{q}' \in V(H)^{\ell+2} \setminus A} X_{\mathbf{q}'} + \sum_{\mathbf{q}' \in A} X_{\mathbf{q}'} \right) \\ &= (-1)^{|E(H)|} \left(0 + \sum_{\mathbf{q}' \in A} (-1)^{k-1} |\mathcal{S}_{\mathbf{q}'}| \right) = (-1)^{|E(H)|+k-1} \sum_{\mathbf{q}' \in A} |\mathcal{S}_{\mathbf{q}'}| \neq 0, \end{aligned}$$

where we used (7) and (8) in the third equality, and where the last inequality holds since $A \neq \emptyset$ and $\mathcal{S}_{\mathbf{q}'} \neq \emptyset$ for all $\mathbf{q}' \in A$. \blacktriangleleft

As discussed above, the lemma implies Equation (5) on the maximum vertex-cover number in the subgraph expansion of the scorpion property. Together with Theorem 10, the upper bound on the vertex-cover number gives an alternative proof that $\#\text{INDSUB}(\Psi_{\ell})$ is fixed-parameter tractable for $\ell \in \mathbb{N}$, with the same polynomial degree as in Theorem 4. We stress that similar arguments also work for the adaptations of the scorpion property discussed in the beginning of Section 4. For example, for the property Λ defined there, we get $\tau_{\Lambda}(k) \leq 7$ for every $k \in \mathbb{N}$.

4.2 Implications of the Conjecture

To conclude the paper and as a potential avenue for future work, we show that our refined Conjecture 11 would easily imply generalizations and variants of known hardness results.

Most of the recent works (see, e.g., [9, 12, 13, 14, 37]) prove hardness of $\#\text{INDSUB}(\Phi)$ by identifying graphs with non-zero alternating enumerator and unbounded *treewidth* rather than vertex-cover number; this is a stronger property. More concretely, let $\eta_{\Phi}(k)$ denote the maximal treewidth among k -vertex graphs H with $\widehat{\Phi}(H) \neq 0$. If Φ is computable and η_{Φ} is unbounded, then $\#\text{INDSUB}(\Phi)$ is $\#\text{W}[1]$ -hard (see, e.g., [9, Theorem 3.1]). However, showing that η_{Φ} is unbounded is often quite a challenge. Conjecture 11 postulates that it suffices for τ_{Φ} to be unbounded to obtain $\#\text{W}[1]$ -hardness, which is often much easier to show.

As a concrete example, previous work [12] established hardness of $\# \text{INDSUB}(\Phi)$ in the presence of infinitely many primes p with $\Phi(\text{IS}_{2p}) \neq \Phi(K_{p,p})$, where IS_{2p} and $K_{p,p}$ are the edge-less and complete bipartite graphs, respectively. Similarly, for edge-monotone properties Φ , which are studied in [9, 13, 14], we typically obtain $\Phi(\text{IS}_k) \neq \Phi(K_k)$ for all sufficiently large k . As we show below, Conjecture 11 implies that infinitely many *arbitrary* prime-order vertex-transitive graphs H_1, H_2 with $\Phi(H_1) \neq \Phi(H_2)$ suffice for hardness. A graph H is called *vertex-transitive* if for all vertices $u, v \in V(H)$ there exists an automorphism $\varphi \in \text{Aut}(H)$ with $\varphi(u) = v$.

► **Proposition 14.** *Let Φ be a graph property such that, for infinitely many primes p , there are vertex-transitive graphs H_p^1, H_p^2 with p vertices and $\Phi(H_p^1) \neq \Phi(H_p^2)$. Then $\# \text{INDSUB}(\Phi)$ is $\# \text{W}[1]$ -hard assuming Conjecture 11.*

Proof. For a prime p , we say that a group Γ is a p -group if its order is a power of p , and we establish some facts about groups and graphs:

- (I) Each transitive group that operates on a set of p elements contains a transitive p -subgroup. This holds since Γ contains a cyclic group of order p , which is transitive (see the discussion before [29, Theorem 1] for more details).
- (II) Each vertex-transitive graph is regular. This is trivial.
- (III) Every d -regular k -vertex graph H with $d > 1$ has vertex-cover number $\tau(H) \geq k/2 - 1$: By [34], the independence number $\alpha(H)$ of H is at most $k/2 + 1$, so $\tau(H) = k - \alpha(H) \geq k/2 - 1$.

By the requirements of the proposition, there are infinitely many primes p such that there are vertex-transitive graphs H_p^1, H_p^2 with p vertices and $\Phi(H_p^1) \neq \Phi(H_p^2)$. Hence, there is an $i \in \{1, 2\}$ with $\Phi(H_p^i) \neq \Phi(\text{IS}_p)$. Let $H_p := H_p^i$ be this graph. Then $\text{Aut}(H_p)$ contains a transitive p -subgroup Γ_p by (I). We say a subgraph F_p of H_p is a *fixed point of Γ_p in H_p* if $V(F_p) = V(H_p)$ and $\gamma(E(F_p)) = E(F_p)$ for all $\gamma \in \Gamma_p$, where $\gamma(E(F_p)) := \{\gamma(u)\gamma(v) \mid uv \in E(F_p)\}$. Let $\text{FP}(\Gamma_p, H_p)$ denote the set of fixed points of Γ_p in H_p (see [14, Appendix A] for more details on fixed points of group actions in graphs).

We show that there is a fixed point $F_p \in \text{FP}(\Gamma_p, H_p)$ that has a non-zero alternating enumerator and $\tau(F_p) \geq p/2 - 1$. For this, let F_p be a fixed point in $\text{FP}(\Gamma_p, H_p)$ with $\Phi(F_p) \neq \Phi(\text{IS}_p)$ and $\Phi(F_p') = \Phi(\text{IS}_p)$ for all $F_p' \in \text{FP}(\Gamma_p, H_p)$ with $F_p' \subsetneq F_p$ (i.e., F_p' is a proper subgraph of F_p). Such a graph F_p exists since $H_p \in \text{FP}(\Gamma_p, H_p)$ and $\Phi(H_p) \neq \Phi(\text{IS}_p)$. Now, [13, Lemma 4.8] implies that $\widehat{\Phi}(F_p) \neq 0$. Moreover, [13, Lemma 3.1] implies that $\Gamma_p \subseteq \text{Aut}(F_p)$, and hence F_p is vertex-transitive. Lastly, (II) and (III) yield that $\tau(F_p) \geq p/2 - 1$.

This means that we can find a sequence of graphs F_p with unbounded vertex cover number and $\widehat{\Phi}(F_p) \neq 0$, and thus $\# \text{INDSUB}(\Phi)$ is $\# \text{W}[1]$ -hard by Conjecture 11. ◀

References

- 1 Josh Alman, Ran Duan, Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. More asymmetry yields faster matrix multiplication. In Yossi Azar and Debmalaya Panigrahi, editors, *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12-15, 2025*, pages 2005–2039. SIAM, 2025. doi:10.1137/1.9781611978322.63.
- 2 Christian Borgs, Jennifer Chayes, László Lovász, Vera T. Sós, and Katalin Vesztegombi. Counting graph homomorphisms. In *Topics in discrete mathematics*, volume 26 of *Algorithms Combin.*, pages 315–371. Springer, Berlin, 2006. doi:10.1007/3-540-33700-8_18.

- 3 Karl Bringmann and Jasper Slusallek. Current algorithms for detecting subgraphs of bounded treewidth are probably optimal. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12–16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPICs*, pages 40:1–40:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ICALP.2021.40.
- 4 Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006. doi:10.1016/J.JCSS.2006.04.007.
- 5 Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19–23, 2017*, pages 210–223. ACM, 2017. doi:10.1145/3055399.3055502.
- 6 Radu Curticapean, Simon Döring, Daniel Neuen, and Jiaheng Wang. Can you link up with treewidth? In Olaf Beyersdorff, Michal Pilipczuk, Elaine Pimentel, and Kim Thang Nguyen, editors, *42nd International Symposium on Theoretical Aspects of Computer Science, STACS 2025, March 4–7, 2025, Jena, Germany*, volume 327 of *LIPICs*, pages 28:1–28:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPICs.STACS.2025.28.
- 7 Radu Curticapean and Dániel Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18–21, 2014*, pages 130–139. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.22.
- 8 Radu Curticapean and Daniel Neuen. Counting small induced subgraphs: Hardness via Fourier analysis. *CoRR*, abs/2407.07051, 2024. doi:10.48550/arXiv.2407.07051.
- 9 Radu Curticapean and Daniel Neuen. Counting small induced subgraphs: Hardness via Fourier analysis. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12–15, 2025*, pages 3677–3695. SIAM, 2025. doi:10.1137/1.9781611978322.122.
- 10 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshantov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 11 Reinhard Diestel. *Graph Theory*. Springer Berlin, 5 edition, 2017. doi:10.1007/978-3-662-53622-3.
- 12 Julian Dörfler, Marc Roth, Johannes Schmitt, and Philip Wellnitz. Counting induced subgraphs: An algebraic approach to $\#W[1]$ -hardness. *Algorithmica*, 84(2):379–404, 2022. doi:10.1007/S00453-021-00894-9.
- 13 Simon Döring, Dániel Marx, and Philip Wellnitz. Counting small induced subgraphs with edge-monotone properties. In Bojan Mohar, Igor Shinkar, and Ryan O’Donnell, editors, *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24–28, 2024*, pages 1517–1525. ACM, 2024. doi:10.1145/3618260.3649644.
- 14 Simon Döring, Dániel Marx, and Philip Wellnitz. From graph properties to graph parameters: Tight bounds for counting on small subgraphs. In Yossi Azar and Debmalya Panigrahi, editors, *Proceedings of the 2025 ACM-SIAM Symposium on Discrete Algorithms, SODA 2025, New Orleans, LA, USA, January 12–15, 2025*, pages 3637–3676. SIAM, 2025. doi:10.1137/1.9781611978322.121.
- 15 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2006. doi:10.1007/3-540-29953-X.
- 16 Jacob Focke and Marc Roth. Counting small induced subgraphs with hereditary properties. *SIAM J. Comput.*, 53(2):189–220, 2024. doi:10.1137/22M1512211.
- 17 Leslie Ann Goldberg and Marc Roth. Parameterised and fine-grained subgraph counting, modulo 2. *Algorithmica*, 86(4):944–1005, 2024. doi:10.1007/S00453-023-01178-0.

- 18 Martin Grohe, Thomas Schwentick, and Luc Segoufin. When is the evaluation of conjunctive queries tractable? In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 657–666. ACM, 2001. doi:10.1145/380752.380867.
- 19 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/JCSS.2000.1727.
- 20 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001. doi:10.1006/JCSS.2001.1774.
- 21 Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978. doi:10.1137/0207033.
- 22 Mark Jerrum and Kitty Meeks. The parameterised complexity of counting connected subgraphs and graph motifs. *J. Comput. Syst. Sci.*, 81(4):702–716, 2015. doi:10.1016/J.JCSS.2014.11.015.
- 23 Mark Jerrum and Kitty Meeks. Some hard families of parameterized counting problems. *ACM Trans. Comput. Theory*, 7(3):11:1–11:18, 2015. doi:10.1145/2786017.
- 24 Emily Jin, Michael M. Bronstein, İsmail İlkan Ceylan, and Matthias Lanzinger. Homomorphism counts for graph neural networks: All about that basis. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL: <https://openreview.net/forum?id=zRrzSLwNHQ>.
- 25 Jeff Kahn, Michael E. Saks, and Dean Sturtevant. A topological approach to evasiveness. *Comb.*, 4(4):297–306, 1984. doi:10.1007/BF02579140.
- 26 Dmitry N. Kozlov. *Combinatorial Algebraic Topology*, volume 21 of *Algorithms and computation in mathematics*. Springer, 2008. doi:10.1007/978-3-540-71962-5.
- 27 Hendrik Lenstra, Marc R. Best, and Peter van Emde Boas. A sharpened version of the Aanderaa-Rosenberg conjecture. *Report 30/74, Mathematisch Centrum Amsterdam (1974)*, pages 1–20, 1974. URL: <https://hdl.handle.net/1887/3792>.
- 28 Ron Milo, Shai S. Shen-Orr, Shalev Itzkovitz, Nadav Kashtan, Dmitri B. Chklovskii, and Uri Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002. doi:10.1126/science.298.5594.824.
- 29 Peter M. Neumann. Transitive permutation groups of prime degree. In *Proceedings of the Second International Conference on the Theory of Groups*, volume Vol. 372 of *Lecture Notes in Math.*, pages 520–535. Springer Berlin Heidelberg, 1974. doi:10.1007/978-3-662-21571-5_55.
- 30 Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Comment. Math. Univ. Carolin.*, 26(2):415–419, 1985. URL: <http://dml.cz/dmlcz/106381>.
- 31 Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Comput. Complex.*, 4:301–313, 1994. doi:10.1007/BF01263419.
- 32 Ronald L. Rivest and Jean Vuillemin. On recognizing graph properties from adjacency matrices. *Theor. Comput. Sci.*, 3(3):371–384, 1976. doi:10.1016/0304-3975(76)90053-0.
- 33 Arnold L. Rosenberg. On the time required to recognize properties of graphs: a problem. *SIGACT News*, 5(4):15–16, 1973. doi:10.1145/1008299.1008302.
- 34 Moshe Rosenfeld. Independent sets in regular graphs. *Israel J. Math.*, 2:262–272, 1964. doi:10.1007/BF02759743.
- 35 Marc Roth and Johannes Schmitt. Counting induced subgraphs: A topological approach to #W[1]-hardness. *Algorithmica*, 82(8):2267–2291, 2020. doi:10.1007/S00453-020-00676-9.
- 36 Marc Roth, Johannes Schmitt, and Philip Wellnitz. Detecting and counting small subgraphs, and evaluating a parameterized tutte polynomial: Lower bounds via toroidal grids and cayley graph expanders. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021, July 12-16, 2021, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPICs*, pages 108:1–108:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.ICALP.2021.108.

- 37 Marc Roth, Johannes Schmitt, and Philip Wellnitz. Counting small induced subgraphs satisfying monotone properties. *SIAM J. Comput.*, 53(6):FOCS20–139–FOCS20–174, 2024. doi:10.1137/20M1365624.
- 38 Benjamin Schiller, Sven Jager, Kay Hamacher, and Thorsten Strufe. StreaM - A stream-based algorithm for counting motifs in dynamic graphs. In Adrian-Horia Dediu, Francisco Hernández Quiroz, Carlos Martín-Vide, and David A. Rosenblueth, editors, *Algorithms for Computational Biology - Second International Conference, AlCoB 2015, Mexico City, Mexico, August 4-5, 2015, Proceedings*, volume 9199 of *Lecture Notes in Computer Science*, pages 53–67. Springer, 2015. doi:10.1007/978-3-319-21233-3_5.
- 39 Falk Schreiber and Henning Schwöbbermeyer. Frequency concepts and pattern detection for the analysis of motifs in networks. *Trans. Comp. Sys. Biology*, 3:89–104, 2005. doi:10.1007/11599128_7.