Parameterised and Fine-Grained Subgraph Counting, Modulo 2*

Leslie Ann Goldberg
Department of Computer Science, University of Oxford, UK

Marc Roth
Department of Computer Science, University of Oxford, UK

Abstract

Given a class of graphs $\mathcal{H}$, the problem $\oplus\textsc{Sub}(\mathcal{H})$ is defined as follows. The input is a graph $H \in \mathcal{H}$ together with an arbitrary graph G. The problem is to compute, modulo 2, the number of subgraphs of G that are isomorphic to H. The goal of this research is to determine for which classes $\mathcal{H}$ the problem $\oplus\textsc{Sub}(\mathcal{H})$ is fixed-parameter tractable (FPT), i.e., solvable in time $f(|H|) \cdot |G|^{O(1)}$.

Curticapean, Dell, and Husfeldt (ESA 2021) conjectured that $\oplus\textsc{Sub}(\mathcal{H})$ is FPT if and only if the class of allowed patterns $\mathcal{H}$ is *matching splittable*, which means that for some fixed B, every $H \in \mathcal{H}$ can be turned into a matching (a graph in which every vertex has degree at most 1) by removing at most B vertices.

Assuming the randomised Exponential Time Hypothesis, we prove their conjecture for (I) all hereditary pattern classes $\mathcal{H}$, and (II) all tree pattern classes, i.e., all classes $\mathcal{H}$ such that every $H \in \mathcal{H}$ is a tree. We also establish almost tight fine-grained upper and lower bounds for the case of hereditary patterns (I).

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Mathematics of computing → Discrete mathematics

Keywords and phrases modular counting, parameterised complexity, fine-grained complexity, subgraph counting

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.68

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `http://arxiv.org/abs/2301.01696`

Acknowledgements We want to thank Radu Curticapean, Holger Dell and Thore Husfeldt for insightful discussions on an early draft of this work.

1 Introduction

The last two decades have seen remarkable progress in the classification of subgraph counting problems: Given a small *pattern* graph H and a large *host* graph G, how often does H occur as a subgraph if G? Since it was discovered that subgraph counts from small patterns reveal global properties of complex networks [26, 27], subgraph counting has also found several applications in fields such as biology [2, 30] genetics [32], phylogeny [25], and data mining [33]. Moreover, the theoretical study of subgraph counting and related problems has led to many deep structural insights, establishing both new algorithmic techniques and tight lower bounds under the lenses of fine-grained and parameterised complexity theory [19, 16, 10, 14, 13, 6, 4].

* For the purpose of Open Access, the authors have applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission. All data is provided in full in the results section of this paper.

Without any additional restrictions, the subgraph counting problem is infeasible. The complexity class #W[1] is the parameterised complexity class analogous to NP (see Section 2 for more detail). Under standard assumptions, problems that are #W[1]-hard are not *fixed-parameter tractable* (FPT). The canonical complete problem for #W[1], the problem of counting k-cliques, corresponds to the special case of the subgraph counting problem where H is a clique of size k. This problem cannot be solved in time $f(k) \cdot n^{o(k)}$ for any function f unless the Exponential Time Hypothesis (ETH) fails [8, 9]. Due to this hardness result, the research focus in this area shifted to the question: Under which restrictions on the patterns H and the hosts G is algorithmic progress possible? More precisely, under which restrictions can the problem be solved in time $f(|H|) \cdot |G|^{O(1)}$, for some computable function f? Instances that can be solved within such a run time bound are called *fixed-parameter tractable* (FPT); allowing a potential super-polynomial overhead in the size of the pattern $|H|$ formalises the assumption that H is assumed to be (significantly) smaller than G.

If only the patterns are restricted, then the situation is fully understood. Formally, given a class $\mathcal{H}$ of patterns, the problem $\#\textsc{Sub}(\mathcal{H})$ asks, given as input a graph $H \in \mathcal{H}$ and an arbitrary graph G, to compute the number of subgraphs of G that are isomorphic to H. Following initial work by Flum and Grohe [19] and by Curticapean [11], Curticapean and Marx [14] proved that, under standard assumptions, $\#\textsc{Sub}(\mathcal{H})$ is FPT if and only if $\mathcal{H}$ has bounded matching number, that is, if there is a positive integer B such that the size of any matching in any graph in $\mathcal{H}$ is at most B. They also proved that all FPT cases are polynomial-time solvable.

In stark contrast, almost nothing is known for the decision version $\textsc{Sub}(\mathcal{H})$. Here, the task is to correctly decide whether there is a copy of $H \in \mathcal{H}$ in G, rather than to count the copies. It is known that $\textsc{Sub}(\mathcal{H})$ is FPT whenever $\mathcal{H}$ has bounded treewidth (see e.g. [20, Chapter 13]), and it is conjectured that those are all FPT cases. However, resolving this conjecture belongs to the "most infamous" open problems in parameterised complexity theory [18, Chapter 33.1].

1.1 Counting Modulo 2

To interpolate between the fully understood realm of (exact) counting and the barely understood realm of decision, Curticapean, Dell and Husfeldt proposed the study of counting subgraphs, modulo 2 [12]. Formally, they introduced the problem $\oplus\textsc{Sub}(\mathcal{H})$, which expects as input a graph $H \in \mathcal{H}$ and an arbitrary graph G, and the goal is to compute *modulo 2* the number of subgraphs of G isomorphic to H.

The study of counting modulo 2 received significant attention from the viewpoint of classical, structural, and fine-grained complexity theory. For example, one way to state Toda's Theorem [31] is $\mathrm{PH} \subseteq \mathrm{P}^{\oplus\mathrm{P}}$, implying that counting satisfying assignments of a CNF, modulo 2, is at least as hard as the polynomial hierarchy. Another example is the quest to classify the complexity of counting modulo 2 the homomorphisms to a fixed graph, which was very recently resolved by Bulatov and Kazeminia [7]. There has also been work by Abboud, Feller, and Weimann [1] on the fine-grained complexity of counting modulo 2 the number of triangles in a graph that satisfy certain weight constraints.

In their work [12], Curticapean, Dell and Husfeldt proved that the problem of counting k-matchings modulo 2, that is, the problem $\oplus\textsc{Sub}(\mathcal{H})$ where $\mathcal{H}$ is the class of all 1-regular graphs, is fixed-parameter tractable, where the parameter k is $|H|$. Since the exact counting version of this problem is #W[1]-hard [11], their result provides an example where counting modulo 2 is *strictly* easier than exact counting (subject to complexity assumptions). The complexity class $\oplus$W[1] can be defined via the complete problem of counting k-cliques modulo 2. Crucially, $\oplus$W[1]-hard problems are not fixed-parameter tractable, unless the randomised ETH (rETH) fails. Curticapean et al. [12] proved that counting k-paths modulo 2 is $\oplus$W[1]-hard. Since *finding* a k-path in a graph G is fixed-parameter tractable via colour-coding [3], this hardness result provides an example where counting modulo 2 is *strictly* harder than decision (subject to complexity assumptions). Combining those observations, it appears that counting subgraphs modulo 2 may lie strictly in between the complexity of decision and the complexity of exact counting.

A *matching* is a graph whose maximum degree is at most 1. The *matching-split number* of a graph H is the minimum size of a set $S \subseteq V(H)$ such that $H \setminus S$ is a matching. A class of graphs $\mathcal{H}$ is called *matching splittable* if there is a positive integer B such that the matching-split number of any $H \in \mathcal{H}$ is at most B. For example, the class of all matchings is matching splittable while the class of all cycles is not. Curticapean, Dell and Husfeldt extended their FTP algorithm for counting k-matchings modulo 2 to obtain an FPT algorithm for $\oplus\textsc{Sub}(\mathcal{H})$ for any matching-splittable class $\mathcal{H}$. On this basis, they then made the following conjecture.

▶ **Conjecture 1** ([12]). *$\oplus\textsc{Sub}(\mathcal{H})$ is FPT if and only if $\mathcal{H}$ is matching splittable.*

A class $\mathcal{H}$ of graphs is called *hereditary* if it is closed under vertex removal. Intriguingly, if Conjecture 1 is true, then the FPT criterion for counting subgraphs modulo 2 ($\oplus\textsc{Sub}(\mathcal{H})$) would coincide with the polynomial-time criterion for finding subgraphs ($\textsc{Sub}(\mathcal{H})$) for hereditary pattern classes $\mathcal{H}$ as established by Jansen and Marx.

▶ **Theorem 2** ([24]). *Let $\mathcal{H}$ be a hereditary class of graphs and assume $\mathrm{P} \neq \mathrm{NP}$. Then $\textsc{Sub}(\mathcal{H})$ is solvable in polynomial time if and only if $\mathcal{H}$ is matching splittable.*

Jansen and Marx also conjecture that the condition of $\mathcal{H}$ being hereditary can be removed.

▶ **Conjecture 3** ([24]). *$\textsc{Sub}(\mathcal{H})$ is solvable in polynomial time if and only if $\mathcal{H}$ is matching splittable.*

Conjectures 1 and 3 have the remarkable consequence that $\oplus\textsc{Sub}(\mathcal{H})$ is FPT if and only if $\textsc{Sub}(\mathcal{H})$ is solvable in polynomial time. In the current work we establish this consequence for all hereditary pattern classes.

1.2 Our Contributions

We resolve Conjecture 1 for all hereditary classes $\mathcal{H}$, as well as for every class $\mathcal{H}$ consisting only of trees; note that the upper bounds were shown in [12] and that the lower bounds are the novel part.

▶ **Theorem 4.** *Let $\mathcal{H}$ be a hereditary class of graphs. If $\mathcal{H}$ is matching splittable, then $\oplus\textsc{Sub}(\mathcal{H})$ is fixed-parameter tractable. Otherwise, the problem is $\oplus$W[1]-complete and, assuming rETH, cannot be solved in time $f(|H|) \cdot |G|^{o(|V(H)|/\log|V(H)|)}$ for any function f.*

▶ **Theorem 5.** *Let $\mathcal{T}$ be a recursively enumerable class of trees. If $\mathcal{T}$ is matching splittable, then $\oplus\textsc{Sub}(\mathcal{T})$ is fixed-parameter tractable. Otherwise $\oplus\textsc{Sub}(\mathcal{T})$ is $\oplus$W[1]-complete.*

The requirement that the class of trees $\mathcal{T}$ needs to be recursively enumerable is a standard technicality - the reason for it is that the function f in the running time in the standard definition of an FPT algorithm is required to be computable. It turns out that having $\mathcal{T}$ recursively enumerable is enough for this.

In order to prove our classifications, we adapt the by-now-standard technique for analysing subgraph counting problems established by Curticapean, Dell and Marx [13]. Let $\#\mathsf{Sub}(H \to G)$ denote the number of subgraphs of a graph G that are isomorphic to a graph H and let $\#\mathsf{Hom}(F \to G)$ denotes the number of homomorphisms (edge-preserving mappings) from a graph F to a graph G. Given a graph H, there is a function a_H from graphs to rationals with finite support such that the following holds for any graph G:

$$\#\mathsf{Sub}(H \to G) = \sum_F a_H(F) \cdot \#\mathsf{Hom}(F \to G)\,, \tag{1}$$

where the sum is over all (isomorphism types of) graphs. Since a_H has finite support, $a_H(F) = 0$ for all but finitely-many graphs F. Thus, equation (1) allows us to express the solution to the *exact* counting problem as a finite linear combination of homomorphism counts. In a nutshell, the framework of [13] states that computing the function $G \mapsto \#\mathsf{Sub}(H \to G)$ is hard to compute if and only if there is a graph F of high treewidth with $a_H(F) \neq 0$. This translates the complexity of (exact) subgraph counting to the purely combinatorial problem of understanding the coefficients a_H. One might hope that this strategy transfers to counting modulo 2 as well. Unfortunately, this is not possible as Equation (1) might not be well-defined if arithmetic is done modulo 2. The reason for this is the fact that the coefficients $a_H(F)$ are of the form $\mu(F, H) \times |\mathsf{Aut}(H)|^{-1}$, where $\mu(F, H)$ is an integer, and $\mathsf{Aut}(H)$ is the automorphism group of the graph H [13]. Thus there is, a priori, no hope to extend the framework to counting modulo 2 for pattern graphs with an even number of automorphisms. In fact, according to Curticapean, Dell and Husfeldt [12], the absence of a comparable framework for counting modulo 2 is one of the main challenges for establishing the hardness part of Conjecture 1, and it is the main reason why the reductions in [12] use more classical, gadget-based reductions.

In this work, we solve the problem of patterns with an even number of automorphisms by considering a colourful intermediate problem. More concretely, we will equip each edge of the pattern H with a distinct colour and show that it will be sufficient to consider only automorphisms that preserve the colours. If H has no isolated vertices, then this is only the trivial automorphism. Formally, the coloured approach will be based on the notion of so-called *fractured* graphs introduced by Peyerimhoff et al. [29].

In what follows (Section 2), we will first introduced all required notions and previous results. In Section 3, we will prove the classification for hereditary pattern classes (Theorem 4). On a technical level, this proof can be considered a warm-up for the significantly harder challenge of establishing the classification for trees (Theorem 5), which we defer to the full version due to the space constraints.

2 Preliminaries

Let $f : A_1 \times A_2 \to B$ be a function. For each $a_1 \in A_1$ we write $f(a_1, \star) : A_2 \to B$ for the function that maps $a_2 \in A_2$ to $f(a_1, a_2)$.

Graphs in this work are undirected and without self loops. A *homomorphism* from a graph H to a graph G is a mapping φ from the vertices $V(H)$ of H to the vertices $V(G)$ of G such that for each edge $e = \{u, v\} \in E(H)$ of H, the image $\varphi(e) = \{\varphi(u), \varphi(v)\}$ is an edge of G. A homomorphism is called an *embedding* if it is injective. We write $\mathsf{Hom}(H \to G)$ and $\mathsf{Emb}(H \to G)$ for the sets of homomorphisms and embeddings, respectively, from H to G. An embedding $\varphi \in \mathsf{Emb}(H \to G)$ is called an *isomorphism* if it is bijective and $\{u, v\} \in E(H) \Leftrightarrow \{\varphi(u), \varphi(v)\} \in E(G)$. We say that H and G are *isomorphic*, denoted by $H \cong G$, if an isomorphism from H to G exists. A *graph invariant* ι is a function from graphs to rationals such that $\iota(H) = \iota(G)$ for each pair of isomorphic graphs H and G.

A *subgraph* of G is a graph G' with $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$. We write $\mathsf{Sub}(H \to G)$ for the set of all subgraphs of G that are isomorphic to H. Given a subset of vertices $S \subseteq V(G)$ of a graph G, we write $G[S]$ for the graph induced by S, that is, $G[S]$ has vertices S and edges $\{\{u,v\} \subseteq S \mid \{u,v\} \in E(G)\}$.

We denote by $\mathsf{tw}(G)$ the *treewidth* of the graph G. Since we will rely on treewidth purely in a black-box manner, we omit the technical definition and refer the reader to [15, Chapter 7].

Given any graph invariant ι (such as treewidth) and a class of graphs $\mathcal{G}$, we say that ι is *bounded* in $\mathcal{G}$ if there is a non-negative integer B such that, for all $G \in \mathcal{G}$, $\iota(G) \leq B$. Otherwise we say that ι is *unbounded* in $\mathcal{G}$.

Given a graph $H = (V, E)$, a *splitting set* of H is a subset of vertices S such that every vertex in $H[V \setminus S]$ has degree at most 1. The *matching-split number* of H is the minimum size of a splitting set of H. A class of graphs $\mathcal{H}$ is called *matching splittable* if the matching-split number of $\mathcal{H}$ is bounded.

2.1 Colour-Preserving Homomorphisms and Embeddings

A homomorphism c from a graph G to a graph Q is sometimes called a "Q-colouring" of G. A Q-*coloured* graph is a pair consisting of a graph G and a homomorphism c from G to Q. Note that the identity function id_Q on $V(Q)$ is a Q-colouring of Q. If a homomorphism c from G to Q is vertex surjective, then we call (G, c) a *surjectively* Q-coloured graph.

▶ **Definition 6** (c_E)**.** *A Q-colouring c of a graph G induces a (not necessarily proper) edge-colouring $c_E\colon E(G) \to E(Q)$ given by $c_E(\{u,v\}) = \{c(u), c(v)\}$.*

Notation. Given a Q-coloured graph (G, c) and a vertex $u \in V(Q)$, we will use the capitalised letter U to denote the subset of vertices of G that are coloured by c with u, that is, $U := c^{-1}(u) \subseteq V(G)$.

Given two Q-coloured graphs (H, c_H) and (G, c_G), we call a homomorphism φ from H to G *colour-preserving* if for each $v \in V(H)$ we have $c_G(\varphi(v)) = c_H(v)$. We note the special case in which $Q = H$ and c_H is the identity id_Q; then the condition simplifies to $c_G(\varphi(v)) = v$. A colour-preserving embedding of (H, c_H) in (G, c_G) is a vertex injective colour-preserving homomorphism from (H, c_H) to (G, c_G). We write $\mathsf{Hom}((H, c_H) \to (G, c_G))$ and $\mathsf{Emb}((H, c_H) \to (G, c_G))$ for the sets of all colour-preserving homomorphisms and embeddings, respectively, from (H, c_H) to (G, c_G).

Let k be a positive integer, let H be a graph with k edges, and let (G, γ) be a pair consisting of a graph G and a function that maps each edge of G to one of k distinct colours. We refer to γ as a "k-edge colouring" of G. For example, in most of our applications we will fix a graph Q with k edges and a Q-colouring c of G and we will take γ to be the edge-colouring c_E from Definition 6. We write $\mathsf{ColSub}(H \to (G, \gamma))$ for the set of all subgraphs of G that are isomorphic to H and that contain each of the k edge colours precisely once.

2.2 Fractures and Fractured Graphs

In this work, we will crucially rely on and extend the framework of *fractured* graphs as introduced in [29].

▶ **Definition 7** (Fractures)**.** *Let Q be a graph. For each vertex v of Q, let $E_Q(v)$ be the set of edges of Q that are incident to v. A* fracture *of Q is a tuple $\rho = (\rho_v)_{v \in V(Q)}$, where for each vertex v of Q, ρ_v is a partition of $E_Q(v)$.*

Figure 1 Illustration of the construction of a fractured graph from [29]. The left picture shows a vertex v of a graph Q with incident edges $E_Q(v) = \{{\color{red}\bullet}, {\color{green}\bullet}, {\color{blue}\bullet}, {\color{orange}\bullet}, \bullet, {\color{cyan}\bullet}\}$. The right picture shows the splitting of v in the construction of the fractured graph $Q\#\sigma$ for a fracture σ satisfying that the partition σ_v contains two blocks $B_1 = \{{\color{red}\bullet}, {\color{green}\bullet}, {\color{blue}\bullet}\}$, and $B_2 = \{{\color{orange}\bullet}, \bullet, {\color{cyan}\bullet}\}$.

Note that a fracture describes how to split (or how to *fracture*) each vertex of a given graph: for each vertex v, create a vertex v^B for each block B in the partition ρ_v; edges originally incident to v are made incident to v^B if and only if they are contained in B. We call the resulting graph the *fractured graph* $H\#\rho$; a formal definition is given in Definition 8, a visualisation is given in Figure 1.

▶ **Definition 8** (Fractured Graph $Q\#\rho$)**.** *Given a graph Q, we consider the matching M_Q containing one edge for each edge of Q; formally,*

$$V(M_Q) := \bigcup_{e=\{u,v\}\in E(Q)} \{u_e, v_e\} \quad \textit{and} \quad E(M_Q) := \{\{u_e, v_e\} \mid e = \{u,v\} \in E(Q)\}.$$

For a fracture ρ of Q, we define the graph $Q\#\rho$ to be the quotient graph of M_Q under the equivalence relation on $V(M_Q)$ which identifies two vertices v_e, w_f of M_Q if and only if $v = w$ and e, f are in the same block B of the partition ρ_v of $E_Q(v)$. We write v^B for the vertex of $Q\#\rho$ given by the equivalence class of the vertices v_e (for which $e \in B$) of M_Q.

▶ **Definition 9** (Canonical Q-colouring c_ρ)**.** *Let Q be a graph and let ρ be a fracture of Q. The* canonical Q-colouring *of the fractured graph $Q\#\rho$ maps v^B to v for each $v \in V(Q)$ and block $B \in \rho_v$, and is denoted by c_ρ.*

Observe that c_ρ is the identity in $V(Q)$ if ρ is the coarsest fracture (that is, each partition ρ_v only contains one block, in which case $Q\#\rho = Q$).

2.3 Parameterised and Fine-grained Computation

A *parameterised computational problem* is a pair consisting of a function $P : \Sigma^* \to \{0,1\}$ and a computable parameterisation $\kappa : \Sigma^* \to \mathbb{N}$. A *fixed-parameter tractable* (FPT) *algorithm* for (P, κ) is an algorithm that computes P and runs, on input $x \in \Sigma^*$, in time $f(\kappa(x)) \cdot |x|^{O(1)}$ for some computable function f. We call (P, κ) *fixed-parameter tractable* (FPT) if an FPT algorithm for (P, κ) exists.

A parameterised Turing-reduction from (P, κ) to (P', κ') is an FPT algorithm for (P, κ) that is equipped with oracle access to P' and for which there is a computable function g such that, on input x, each oracle query y satisfies $\kappa'(y) \le g(\kappa(x))$. We write $(P, \kappa) \le^{\mathrm{fpt}}_{\mathrm{T}} (P', \kappa')$ if a parameterised Turing-reduction from (P, κ) to (P', κ') exists. This guarantees that fixed-parameter tractability of (P', κ') implies fixed-parameter tractability of (P, κ). For a more comprehensive introduction, we refer the reader the standard textbooks [15] and [20].

Counting modulo 2 and the rETH

The lower bounds in this work will rely on the hardness of the parameterised complexity class $\oplus\mathsf{W}[1]$, which can be considered a parameterised equivalent of $\oplus\mathrm{P}$. Following [12], we define $\oplus\mathsf{W}[1]$ via the complete problem $\oplus$Clique: Given as input a graph G and a positive integer k, the goal is to compute the number of k-cliques in G modulo 2, i.e., to compute $\oplus\mathsf{Sub}(K_k \to G)$. The problem is parameterised by k. A parameterised problem (P,κ) is called $\oplus\mathsf{W}[1]$-*hard* if $\oplus\textsc{Clique} \leq^{\mathrm{fpt}}_{\mathrm{T}} (P,\kappa)$, and it is called $\oplus\mathsf{W}[1]$-*complete* if, additionally, $(P,\kappa) \leq^{\mathrm{fpt}}_{\mathrm{T}} \oplus\textsc{Clique}$.

Modifications of the classical Isolation Lemma (see e.g. [5] and [34]) yield a *randomised* parameterised Turing reduction from finding a k-clique to computing the parity of the number of k-cliques. In combination with existing fine-grained lower bounds for finding a k-clique [8, 9], it can then be shown that $\oplus$Clique cannot be solved in time $f(k)\cdot|G|^{o(k)}$ for any function f, unless the randomised Exponential Time Hypothesis fails:

▶ **Definition 10** (rETH, [23]). *The* randomised Exponential Time Hypothesis *(rETH) asserts that 3-SAT cannot be solved by a randomised algorithm in time* $\exp o(n)$*, where* n *is the number of variables of the input formula.*

As an immediate consequence, the rETH implies that $\oplus\mathsf{W}[1]$-hard problems are not fixed-parameter tractable.

For the lower bounds in this work, we won't reduce from $\oplus$Clique directly, but instead from the following, more general problem:

▶ **Definition 11** ($\oplus$cp-Hom). *Let* $\mathcal{H}$ *be a class of graphs. The problem* $\oplus\textsc{cp-Hom}(\mathcal{H})$ *has as input a graph* $H \in \mathcal{H}$ *and a surjectively* H*-coloured graph* (G,c)*. The goal is to compute* $\oplus\mathsf{Hom}((H,\mathsf{id}_H) \to (G,c))$*. The problem is parameterised by* $|H|$*.*

The following lower bound was proved independently in [28, 29] and [12].

▶ **Theorem 12.** *Let* $\mathcal{H}$ *be a recursively enumerable class of graphs. If the treewidth of* $\mathcal{H}$ *is unbounded then* $\oplus\textsc{cp-Hom}(\mathcal{H})$ *is* $\oplus\mathsf{W}[1]$*-hard and, assuming the rETH, it cannot be solved in time* $f(|H|)\cdot|G|^{o(\mathsf{tw}(H)/\log\mathsf{tw}(H))}$ *for any function* f*.*

Next is the central problem in this work.

▶ **Definition 13** ($\oplus$Sub). *Let* $\mathcal{H}$ *be a class of graphs. The problem* $\oplus\textsc{Sub}(\mathcal{H})$ *has as input a graph* $H \in \mathcal{H}$ *and a graph* G*. The goal is to compute* $\oplus\mathsf{Sub}(H \to G)$*. The problem is parameterised by* $|H|$*.*

For example, writing $\mathcal{K}$ for the set of all complete graphs, the problem $\oplus\textsc{Sub}(\mathcal{K})$ is equivalent to $\oplus$Clique.

Complexity Monotonicity and Inclusion-Exclusion

Throughout this work, we will rely on two important tools introduced in [29]. For the sake of being self-contained, we encapsulate them below in individual lemmas.

The first tool is an adaptation of the so-called Complexity Monotonicity principle to the realm of fractured graphs and modular counting (see [29, Sections 4.1 and 6.3] for a detailed treatment and for a proof). Intuitively, the subsequent lemma states that evaluating, modulo 2, a linear combination of colour-prescribed homomorphism counts from fractured graphs, is as hard as evaluating its hardest term with an odd coefficient.

▶ **Lemma 14** ([29]). *There is a deterministic algorithm $\mathbb{A}$ and a computable function f such that the following conditions are satisfied:*

1. *$\mathbb{A}$ expects as input a graph Q and a Q-coloured graph (G, c).*
2. *$\mathbb{A}$ is equipped with oracle access to a function*
$$(G', c') \mapsto \sum_{\rho} a(\rho) \cdot \oplus\mathsf{Hom}((Q\#\rho, c_\rho) \to (G', c')) \mod 2\,,$$
where the sum is over all fractures of Q and a is a function from fractures of Q to integers.
3. *Each oracle query (G', c') is of size at most $f(|Q|) \cdot |G|$.*
4. *$\mathbb{A}$ computes $\oplus\mathsf{Hom}((Q\#\rho, c_\rho) \to (G, c))$ for each fracture ρ with $a(\rho) \neq 0 \mod 2$.*
5. *The running time of $\mathbb{A}$ is bounded by $f(|Q|) \cdot |G|^{O(1)}$.*

The second tool is a standard application of the inclusion-exclusion principle (see e.g. [29, Sections 4.2 and 6.3]). It will be used in the final steps of our reductions to remove the colourings.

▶ **Lemma 15** ([29]). *There is a deterministic algorithm $\mathbb{A}$ that satisfies the following conditions:*

1. *$\mathbb{A}$ expects as input a graph H with k edges, a graph G and a k-edge colouring γ of G.*
2. *$\mathbb{A}$ is equipped with oracle access to the function $\oplus\mathsf{Sub}(H \to \star)$, and each oracle query G' satisfies $|G'| \leq |G|$.*
3. *$\mathbb{A}$ computes $\oplus\mathsf{ColSub}(H \to (G, \gamma))$.*
4. *The running time of $\mathbb{A}$ is bounded by $2^{|H|} \cdot |G|^{O(1)}$.*

3 Classification for Hereditary Graph Classes

In this section, we will completely classify the complexity of $\oplus\textsc{Sub}(\mathcal{H})$ for hereditary classes. Let us start by restating the classification theorem.

▶ **Theorem 4.** *Let $\mathcal{H}$ be a hereditary class of graphs. If $\mathcal{H}$ is matching splittable, then $\oplus\textsc{Sub}(\mathcal{H})$ is fixed-parameter tractable. Otherwise, the problem is $\oplus\mathsf{W}[1]$-complete and, assuming rETH, cannot be solved in time $f(|H|) \cdot |G|^{o(|V(H)|/\log |V(H)|)}$ for any function f.*

The proof of Theorem 4 is split in four cases, which stem from a structural property of non matching splittable hereditary graph classes $\mathcal{H}$ due to Jansen and Marx [24]. For the statement, we need to consider the following classes:

- $\mathcal{F}_\omega$ is the class of all complete graphs.
- $\mathcal{F}_\beta$ is the class of all complete bipartite graphs.
- $\mathcal{F}_{P_2}$ is the class of all P_2-packings, that is, disjoint unions of paths with two edges.[1]
- $\mathcal{F}_{K_3}$ is the class of all triangle packings, that is, disjoint unions of the complete graph of size 3.

▶ **Theorem 16** (Theorem 3.5 in [24]). *Let $\mathcal{H}$ be a hereditary class of graphs. If $\mathcal{H}$ is not matching splittable then at least one of the following are true: (1.) $\mathcal{F}_\omega \subseteq \mathcal{H}$, (2.) $\mathcal{F}_\beta \subseteq \mathcal{H}$, (3.) $\mathcal{F}_{P_2} \subseteq \mathcal{H}$, or (4.) $\mathcal{F}_{K_3} \subseteq \mathcal{H}$.*

[1] To avoid confusion, we remark that [24] uses P_3 to denote the path of two edges (and three vertices). In the current work, it will be more convenient to use the number of edges of a path as index.

Thus, it suffices to consider cases 1. − 4. to prove Theorem 4. We start with the easy cases of cliques and bicliques; they follow implicitly from previous works [12, 17, 28] and we only include a proof for completeness. Note that a tight bound under rETH is known for those cases:

▶ **Lemma 17.** *Let $\mathcal{H}$ be a hereditary class of graphs. If $\mathcal{F}_\omega \subseteq \mathcal{H}$ or $\mathcal{F}_\beta \subseteq \mathcal{H}$ then $\oplus\textsc{Sub}(\mathcal{H})$ is $\oplus$W[1]-hard and, assuming rETH, cannot be solved in time $f(|H|) \cdot |G|^{o(|V(H)|)}$ for any function f.*

Proof. If $\mathcal{F}_\omega \subseteq \mathcal{H}$ then $\oplus$W[1]-hardness follows immediately from the fact that $\oplus\textsc{Clique}$ is the canonical $\oplus$W[1]-complete problem [12]. For the rETH lower bound, we can reduce from the problem of *deciding* the existence of a k-clique via a (randomised) reduction using a version of the Isolation Lemma due to Williams et al. [34, Lemma 2.1]. This reduction does not increase k or the size of the host graph and is thus tight with respect to the well-known lower bound for the clique problem due to Chen et al. [8, 9]: Deciding the existence of a k-clique in an n-vertex graph cannot be done in time $f(k) \cdot n^{o(k)}$ for any function f, unless ETH fails. Our lower bound under rETH follows since the reduction is randomised.

If $\mathcal{F}_\beta \subseteq \mathcal{H}$, then the claim holds by [17, Theorem 5], which established the problem of counting, modulo 2, the induced copies of a k-by-k-biclique in an n-vertex bipartite graph to be $\oplus$W[1]-hard and not solvable in time $f(k) \cdot n^{o(k)}$ for any function f, unless rETH fails. Since a copy of a biclique (with at least one edge) in a bipartite graph must always be induced, the claim follows. This concludes the proof of Lemma 17. ◀

The more interesting cases are $\mathcal{F}_{P_2} \subseteq \mathcal{H}$ and $\mathcal{F}_{K_3} \subseteq \mathcal{H}$. One reason for this is that, in contrast to cliques and bicliques, the decision version of those instances are fixed-parameter tractable. Hence a reduction from the decision version via e.g. an isolation lemma does not help. In other words, establishing hardness for those cases requires us to rely on the full power of counting modulo 2. More precisely, we will rely on the framework of fractures graphs (see Section 2). Both cases can be considered simpler applications of the machinery used in the later sections, so we will present all steps in great detail. While this might seem unnecessary given the simplicity of the constructions, we hope that it enables the reader to make themselves familiar with the general reduction strategies which will be used throughout the later sections of this work.

3.1 Triangle Packings

The goal of this subsection is to establish hardness of $\oplus\textsc{Sub}(\mathcal{F}_{K_3})$. To this end, let Δ be an infinite computable class of cubic bipartite expander graphs, and let $\mathcal{Q} = \{L(H) \mid H \in \Delta\}$ where $L(H)$ is constructed as follows: Each $v \in V(H)$ becomes a triangle with vertices v_x, v_y, and v_z corresponding to the three neighbours x, y, and z of v. Finally, for every edge $\{u, v\} \in E(H)$ we identify v_u and u_v. In fact, $L(H)$ is just the *line graph* of H: Every edge of H becomes a vertex in $L(H)$, and two vertices of $L(H)$ are made adjacent if and only if the corresponding edges in H are incident. Since all $H \in \Delta$ are bipartite (and thus triangle-free), we can easily observe the following.[2]

▶ **Observation 18.** *The mapping $v \mapsto (v_x, v_y, v_z)$ is a bijection from vertices of H to triangles in $L(H)$.*

[2] Observation 18 is also an immediate consequence of Whitney's Isomorphism Theorem implying that a triangle of a line graph corresponds to either a claw or to a triangle in its primal graph.

We also consider the fracture of $L(H)$ that splits $L(H)$ back into $|V(H)|$ triangles; consider Figure 2 for an illustration.

▶ **Definition 19** ($\tau(H)$). *Let $H \in \Delta$ and recall that each vertex w of $L(H)$ is obtained by identifying v_u and u_v for some edge $\{u,v\} \in E(H)$. Moreover, w has four incident edges e_x, e_y, e_a, e_b, to v_x, v_y, u_a, u_b, respectively, where x,y,u are the neighbours of v in H and v,a,b are the neighbours of u in H. We define $\tau(H)_w := \{\{e_x,e_y\},\{e_a,e_b\}\}$, and we proceed similar for all vertices of $L(H)$.*

Next, we use that $\mathsf{tw}(L(H)) = \Omega(\mathsf{tw}(H))$ (see e.g. [22]). Moreover, $\mathsf{tw}(L(H)) \leq |V(L(H))|$ since the treewidth of a graph is always bounded by the number of its vertices. Additionally, $|V(L(H))| = |E(H)|$ by construction. Since the graphs in Δ are cubic, we further have that $|E(H)| = \Theta(|V(H)|)$ for $H \in \Delta$. We combine those bounds with the fact that expander graphs have treewidth linear in the number of vertices (see e.g. [21]); therefore Δ and thus $\mathcal{Q}$ have unbounded treewidth. Putting these facts together, we obtain the following.

▶ **Fact 20.** *$\mathcal{Q}$ has unbounded treewidth and $\mathsf{tw}(L(H)) = \Theta(|V(L(H))|) = \Theta(|V(H)|)$ for $H \in \Delta$.*

We are now able to establish hardness of $\oplus\textsc{Sub}(\mathcal{F}_{K_3})$. The proof will heavily rely on the transformation from edge-coloured subgraphs to homomorphisms established in [29].

▶ **Lemma 21.** *The problem $\oplus\textsc{Sub}(\mathcal{F}_{K_3})$ is $\oplus\mathsf{W}[1]$-hard. Furthermore, on input kK_3 and G, the problem cannot be solved in time $f(k)\cdot|G|^{o(k/\log k)}$ for any function f, unless rETH fails.*

Proof. We reduce from $\oplus\textsc{cp-Hom}(\mathcal{Q})$, which, by Fact 20 and Theorem 12, is $\oplus\mathsf{W}[1]$-hard and for $L(H) \in \mathcal{Q}$, it cannot be solved in time $f(|L(H)|)\cdot|G|^{o(|V(L(H))|/\log|V(L(H))|)}$, unless rETH fails.

Let L and (G,c) be an input instance to $\oplus\textsc{cp-Hom}(\mathcal{Q})$. Recall that Δ is computable – that is, there is an algorithm that takes a graph H and determines whether it is in Δ. Thus, there is an algorithm that takes input $L \in \mathcal{Q}$ and finds a graph $H \in \Delta$ with $L = L(H)$. The run time of this algorithm depends on $|L|$ but clearly not on (G,c). Let $k = |V(H)|$ and note that $|E(L(H))| = 3k$, since, by construction, each vertex v of H becomes a triangle of $L(H)$. We consider the graph G as a $3k$-edge-coloured graph, coloured by c_E. That is, each edge $e = \{x,y\}$ of G is assigned the colour $c_E(e) = \{c(x),c(y)\}$ which is an edge of L (see Figure 2 for an illustration).

Now, for *any* L-coloured graph (G',c') recall that $\mathsf{ColSub}(kK_3 \to (G',c'_E))$ is the set of subgraphs of G' that are isomorphic to kK_3 and that include each edge colour (each edge of L) precisely once. We will see later that $\oplus\mathsf{ColSub}(kK_3 \to (G',c'_E))$ can be computed using our oracle for $\oplus\textsc{Sub}(\mathcal{F}_{K_3})$ using the principle of inclusion and exclusion.

It was shown in [29, Lemma 4.1] that there is a unique function a such that for every L-coloured graph (G',c') we have[3]

$$\#\mathsf{ColSub}(kK_3 \to (G',c'_E)) = \sum_{\rho} a(\rho)\cdot \mathsf{Hom}(L\#\rho \to (G',c'))\,. \tag{2}$$

where the sum is over all fractures of L. Additionally, it was shown in [29, Corollary 4.3] that

$$a(\top) = \sum_{\rho\in\mathsf{F}(kK_3,L)} \prod_{w\in V(L)} (-1)^{|\rho_w|-1}\cdot(|\rho_w|-1)!\,, \tag{3}$$

[3] In the language of [29], Equation (2) is obtained by choosing Φ as the property of being isomorphic to kK_3.

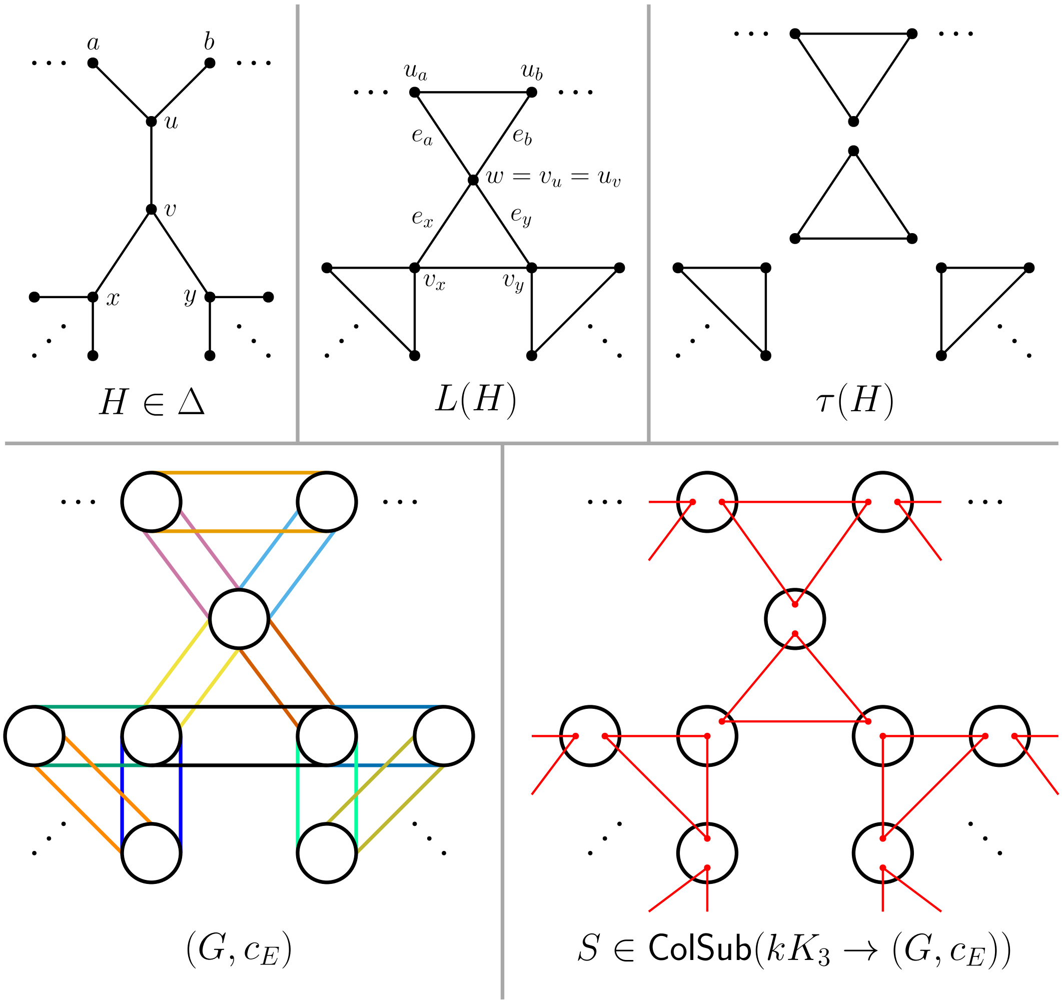

Figure 2 (*Top*:) A cubic bipartite graph $H \in \Delta$, its line graph $L(H)$, and the fractured graph induced by $\tau(H)$. (*Below*:) An $L(H)$-coloured graph (G, c); emphasised in distinct colours is the edge-colouring c_E of G induced by the mapping $\{u, v\} \mapsto \{c(u), c(v)\}$. Additionally we depict an element $S \in \mathsf{ColSub}(kK_3 \to (G, c_E))$, that is, a subgraph of G isomorphic to kK_3 that contains each edge colour of G precisely once.

where $\top$ is the fracture in which each partition consists only of one block (that is, $L\sharp\top = L$), and $\mathsf{F}(kK_3, L)$ is the set of all fractures ρ of L such that $L\sharp\rho \cong kK_3$. However, note that, by Observation 18, there is only way to fracture L into k disjoint triangles, and this fracture is given by $\tau(H)$. Thus, (3) simplifies to

$$a(\top) = \prod_{w \in V(L)} (-1)^{|\tau(H)_w|-1} \cdot (|\tau(H)_w| - 1)!\,, \tag{4}$$

which is odd since each partition of $\tau(H)$ consists of precisely two blocks (so in fact the expression in (4) is $(-1)^{|V(L)|}$).

Note that the algorithm for $\oplus$CP-HOM($\mathcal{Q}$) is supposed to compute $\oplus\mathsf{Hom}((L, \mathsf{id}_L) \to (G, c))$ which is equal to $\oplus\mathsf{Hom}(L\sharp\top \to (G, c_\top))$. Since $a(\top)$ is odd, we can invoke Lemma 14 to recover this term by evaluating the entire linear combination (2), that is, by evaluating the function $\oplus\mathsf{ColSub}(kK_3 \to \star)$. More concretely, this means that we need to compute

$\oplus\mathsf{ColSub}(kK_3 \to (G', c'_E))$ for some L-coloured graphs (G', c') of size at most $f(|L|) \cdot |G|$ for some computable function f (see 3. in Lemma 14). This can easily be done using Lemma 15 since we have oracle access to the function $\oplus\mathsf{Sub}(kK_3 \to \star)$. We emphasise that, by condition 2. of Lemma 15, each oracle query $\hat{G}$ satisfies $|\hat{G}| \le |G'|$, where (G', c') is the L-coloured graph for which we wish to compute $\oplus\mathsf{ColSub}(kK_3 \to (G', c'_E))$. Since $|(G', c')| \le f(|L|) \cdot |G|$, we obtain that $|\hat{G}| \le f(|L|) \cdot |G|$ as well.

Since, by Fact 20, $k = \Theta(|kK_3|) = \Theta(|V(L)|) = \Theta(\mathsf{tw}(L))$, our reduction yields $\oplus\mathsf{W[1]}$-hardness and transfers the conditional lower bound under rETH as desired. ◀

3.2 P_2-packings

Next we establish hardness for the case of P_2-packings. The strategy will be similar in spirit to the construction for triangle packings; however, rather then identifying a unique fracture for which the technique applies, we will encounter an *odd* number of possible fractures in the current section.

Let Δ be a computable infinite class of 4-regular expander graphs, and let $\mathcal{Q}$ be the class of all subdivisions of graphs in Δ, that is $\mathcal{Q} = \{H^2 \mid H \in \Delta\}$, where H^2 is obtained from H by subdividing each edge once.

We start by establishing an easy but convenient fact on the treewidth of the graphs in $\mathcal{Q}$.

▶ **Lemma 22.** *$\mathcal{Q}$ has unbounded treewidth and $\mathsf{tw}(H^2) = \Theta(|V(H)|)$ for $H \in \Delta$.*

Proof. As in Section 3.1, $\mathsf{tw}(H) = \Theta(|V(H)|)$ for $H \in \Delta$, since expanders have treewidth linear in the number of vertices. Since H is a minor of H^2, and since taking minors cannot increase treewidth (see [15, Exercise 7.7]), we thus have that $\mathsf{tw}(H^2) = \Omega(|V(H)|)$. Finally, we have $\mathsf{tw}(H^2) \le |V(H^2)|$ since the treewidth is at most the number of vertices, and $|V(H^2)| = O(|V(H)|)$ since H is 4-regular. In combination, we obtain $\mathsf{tw}(H^2) = \Theta(|V(H)|)$ for $H \in \Delta$. Note that this also implies that $\mathcal{Q}$ has unbounded treewidth (as Δ is infinite). ◀

For what follows, given a subdivision H^2 of a graph H, it will be convenient to assume that $V(H^2) = V(H) \cup S_E$, where $S_E = \{s_e \mid e \in E(H\})$ is the set of the subdivision vertices.

▶ **Definition 23** (Odd Fractures). *Let $H \in \Delta$ and let τ be a fracture of H^2. We say that τ is* odd *if the following two conditions are satisfied:*

1. *For each $s \in S_E$ the partition τ_s consists of two singleton blocks.*
2. *For each $v \in V(H)$ the partition τ_v consists of two blocks of size 2.*

Consider Figure 3 for a depiction of an odd fracture.

The following two lemmas are crucial for our construction.

▶ **Lemma 24.** *Let $H \in \Delta$. The number of odd fractures of H^2 is odd.*

Proof. The first condition in Definition 23 leaves only one choice for subdivision vertices. Let us thus consider a vertex $v \in V(H) = V(H^2) \setminus S_E$. Since H is 4-regular, there are 4 incident edges to v. Now note that there are precisely 3 partitions of a 4-element set with two blocks of size 2. Thus the total number of odd fractures of H^2 is $3^{|V(H)|}$, which is odd. ◀

▶ **Lemma 25.** *Let $H \in \Delta$, let $k = 2|V(H)|$ and let τ be a fracture of H^2 such that τ_v consists of at most 2 blocks for each $v \in V(H^2)$. Then $H^2 \sharp \tau \cong kP_2$ if and only if τ is odd.*

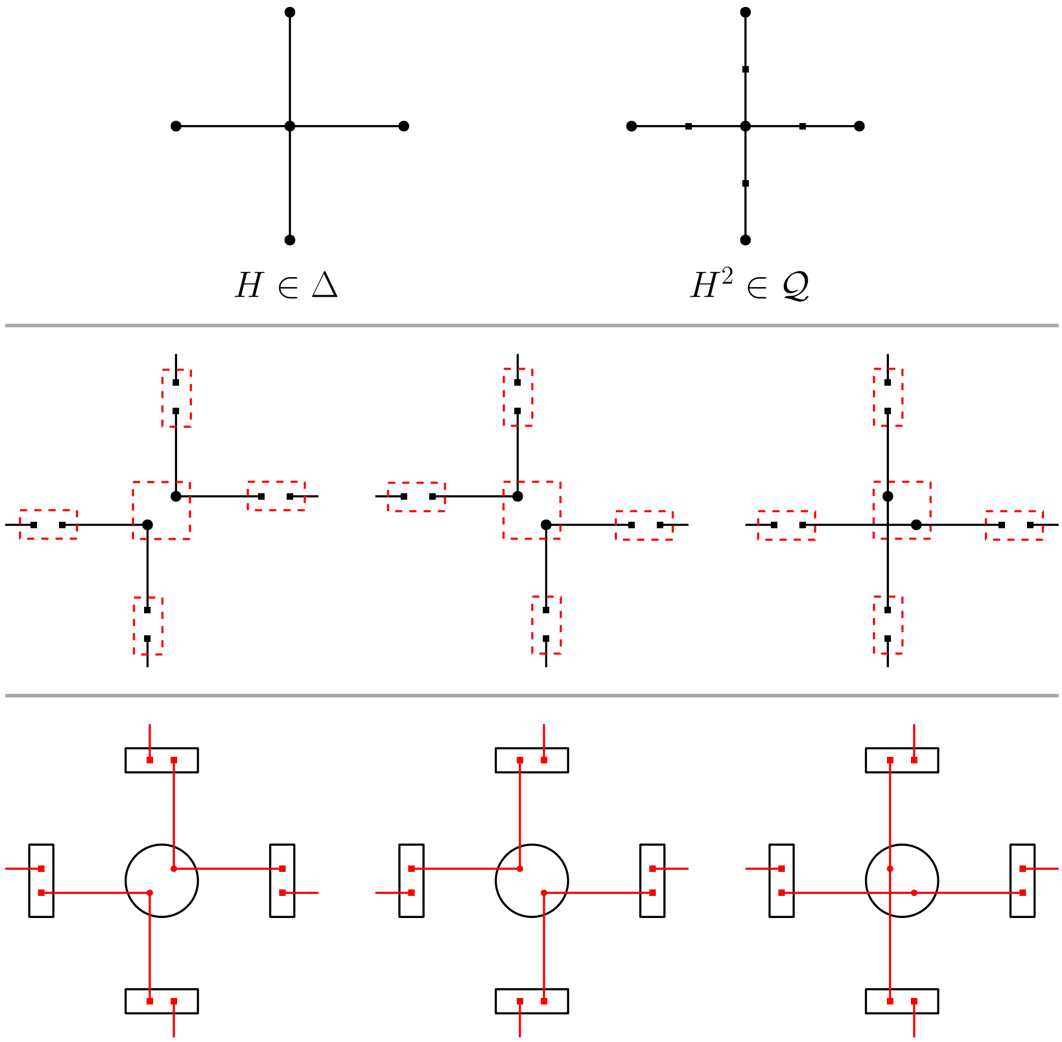

Figure 3 (*Top*:) Subdividing a 4-regular expander in Δ depicted by the neighbourhood of an individual vertex. (*Centre*:) Illustrations of odd fractures (Definition 23). For each non-subdivision vertex, there are only three ways to satisfy 2. in Definition 23. This observation is used in Lemma 24 to show that the number of odd fractures is a power of 3. (*Bottom*:) Elements of $\mathsf{ColSub}(kP_2 \to (G, c_E))$ inducing fractures of H^2 such that each partition has at most two blocks. Lemma 25 shows that those are precisely the odd fractures of H^2.

Proof. First observe that $|E(H^2)| = 2|E(H)| = 4|V(H)| = 2k$. Thus the number of edges of $H^2 \sharp \tau$ is equal to $2k$ (for each fracture τ of H^2), which is also equal to the number of edges of kP_2.

Thus, $H^2 \sharp \tau$ is isomorphic to kP_2 if and only if each connected component of $H^2 \sharp \tau$ is a path of length 2. It follows immediately by Definition 23 that τ being odd implies that $H^2 \sharp \tau$ consists only of disjoint P_2. It thus remains to show the other direction.

Assume for contradiction that there is a subdivision vertex $s \in S_E$ of H^2 such that τ_s consists of only one block (recall that s has degree 2, thus τ_s either consists of two singleton blocks, or of one block of size 2). Let $e = \{u, v\} \in E(H)$ be the edge corresponding to s, that is, s was created by subdividing e. Since $H^2 \sharp \tau$ is a union of P_2, we can infer that τ_v and τ_u contain a singleton block (otherwise we would have created a connected component which is not isomorphic to P_2). Now recall that both u and v have degree 4, since H is 4-regular. We obtain a contradiction as follows: By assumption of the lemma, we know that τ_v and τ_u can have at most two blocks. Since we have just shown that both contain a singleton block, it follows that both τ_v and τ_u contain one further block of size 3. However, a block of size 3 yields a vertex of degree 3 in the fractured graph $H^2 \# \tau$, contradicting the fact that $H^2 \# \tau$ consists only of disjoint P_2.

Thus we have established that, for each $s \in S_E$, the partition τ_s consists of two singleton blocks. Given this fact, the only way for $H^2 \# \tau$ being a disjoint union of P_2 is that each partition τ_v, for $v \in V(H) = V(H^2) \setminus S_E$, consists of two blocks of size 2. ◀

We are now able to prove our hardness result.

▶ **Lemma 26.** *The problem $\oplus\textsc{Sub}(\mathcal{F}_{P_2})$ is $\oplus$W[1]-hard. Furthermore, on input kP_2 and G, the problem cannot be solved in time $f(k) \cdot |G|^{o(k/\log k)}$ for any function f, unless rETH fails.*

Proof. We reduce from $\oplus\textsc{cp-Hom}(\mathcal{Q})$, which, by Lemma 22 and Theorem 12, is $\oplus$W[1]-hard and for $H' \in \mathcal{Q}$, it cannot be solved in time $f(|H'|) \cdot |G|^{o(|V(H')|/\log|V(H')|)}$, unless rETH fails.

Let H' and (G, c) be an input instance to $\oplus\textsc{cp-Hom}(\mathcal{Q})$. There is an algorithm that takes as input a graph $H' \in \mathcal{Q}$ and finds a graph $H \in \Delta$ with $H' = H^2$ – this is basically 2-colouring. The run time of this algorithm depends on $|H'|$ but clearly not on (G, c). Let $k = 2|V(H)|$ and note that $|E(H^2)| = 2|E(H)| = 4|V(H)| = 2k$. We consider the graph G as a $2k$-edge-coloured graph, coloured by c_E. That is, each edge $e = \{x, y\}$ of G is assigned the colour $c_E(e) = \{c(x), c(y)\}$ which is an edge of $H' = H^2$.

Now, for *any* H^2-coloured graph (G', c') recall that $\mathsf{ColSub}(kP_2 \to (G', c'_E))$ is the set of subgraphs of G' that are isomorphic to kP_2 and that include each edge colour (each edge of H^2) precisely once. We will see later that $\oplus\mathsf{ColSub}(kP_2 \to (G', c'_E))$ can be computed using our oracle for $\oplus\textsc{Sub}(\mathcal{F}_{P_2})$ using the principle of inclusion and exclusion.

It was shown in [29, Lemma 4.1] that there is a unique function a such that, for every H^2-coloured graph (G', c'),

$$\#\mathsf{ColSub}(kP_2 \to (G', c'_E)) = \sum_{\rho} a(\rho) \cdot \mathsf{Hom}(H^2 \# \rho \to (G', c')) \,. \tag{5}$$

where the sum is over all fractures of H^2. As in Section 3.1 from [29, Corollary 4.3] we know that

$$a(\top) = \sum_{\rho \in \mathsf{F}(kP_2, H^2)} \prod_{w \in V(H^2)} (-1)^{|\rho_w|-1} \cdot (|\rho_w| - 1)! \,, \tag{6}$$

where $\top$ is the fracture in which each partition consists only of one block and $\mathsf{F}(kP_2, H^2)$ is the set of all fractures ρ of H^2 such that $H^2 \# \rho \cong kP_2$.

Our next goal is to show that $a(\top) = 1 \mod 2$. First, suppose that a fracture ρ contains a partition ρ_w with at least three blocks. Then $(|\rho_w| - 1)! = 0 \mod 2$. Thus such fractures do not contribute to $a(\top)$ if arithmetic is done modulo 2. Next, note that if, for each w, the partition ρ_w contains at most 2 blocks, then

$$\prod_{w \in V(H^2)} (-1)^{|\rho_w|-1} \cdot (|\rho_w| - 1)! = 1 \mod 2.$$

Let $\mathsf{Odd}(kP_2, H^2)$ be the set of all fractures ρ of H^2 such that $H^2\sharp\rho \cong kP_2$ and each partition of ρ consists of at most 2 blocks. Our analysis then yields $a(\top) = |\mathsf{Odd}(kP_2, H^2)| \mod 2$. Finally, Lemma 25 states that $\mathsf{Odd}(kP_2, H^2)$ is precisely the set of odd fractures, and Lemma 24 thus implies that $|\mathsf{Odd}(kP_2, H^2)| = 1 \mod 2$. Consequently, $a(\top) = 1 \mod 2$ as well, and we have achieved the goal.

Next we can proceed similarly to the case of triangle packings. As in that case, the goal is to compute $\oplus\mathsf{Hom}((H^2, \mathsf{id}_{H^2}) \to (G, c)))$ which is equal to $\oplus\mathsf{Hom}((H^2\sharp\top, c_\top) \to (G, c))$. Since $a(\top)$ is odd, we can invoke Lemma 14 to recover this term by evaluating the entire linear combination (5), that is, if we can evaluate the function $\oplus\mathsf{ColSub}(kP_2 \to \star)$. This can be done by using Lemma 15. Each call to the oracle is of the form $\oplus\mathsf{Sub}(kP_2 \to \hat{G})$ where $|\hat{G}|$ is bounded by $f(k) \cdot |G|$.

Now recall that $k \in \Theta(|V(H)|)$. By Lemma 22, we thus have $k = \Theta(\mathsf{tw}(H^2))$. Hence our reduction yields $\oplus\mathsf{W[1]}$-hardness and transfers the conditional lower bound under rETH as desired. ◀

We can now conclude the treatment of hereditary pattern classes by proving Theorem 4, which we restate for convenience.

▶ **Theorem 4.** *Let $\mathcal{H}$ be a hereditary class of graphs. If $\mathcal{H}$ is matching splittable, then $\oplus\textsc{Sub}(\mathcal{H})$ is fixed-parameter tractable. Otherwise, the problem is $\oplus\mathsf{W[1]}$-complete and, assuming rETH, cannot be solved in time $f(|H|) \cdot |G|^{o(|V(H)|/\log|V(H)|)}$ for any function f.*

Proof. The fixed-parameter tractability result was shown in [12]. For the hardness result, using the fact that $\mathcal{H}$ is not matching splittable and Theorem 16 we obtain four cases.

- If $\mathcal{H}$ contains all cliques or all bicliques, then hardness follows from Lemma 17.
- If $\mathcal{H}$ contains all triangle packings, then hardness follows from Lemma 21.
- If $\mathcal{H}$ contains all P_2-packings, then hardness follows from Lemma 26.

Since the case distinction is exhaustive, the proof is concluded. ◀

References

1 Amir Abboud, Shon Feller, and Oren Weimann. On the fine-grained complexity of parity problems. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 5:1–5:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.5.

2 Noga Alon, Phuong Dao, Iman Hajirasouliha, Fereydoun Hormozdiari, and S. Cenk Sahinalp. Biomolecular network motif counting and discovery by color coding. *Bioinformatics*, 24(13):i241–i249, 2008. doi:10.1093/bioinformatics/btn163.

3 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

4 Suman K. Bera, Lior Gishboliner, Yevgeny Levanzov, C. Seshadhri, and Asaf Shapira. Counting subgraphs in degenerate graphs. *J. ACM*, 69(3):23:1–23:21, 2022. doi:10.1145/3520240.

5 Andreas Björklund, Holger Dell, and Thore Husfeldt. The parity of set systems under random restrictions with applications to exponential time problems. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, volume 9134 of *Lecture Notes in Computer Science*, pages 231–242. Springer, 2015. doi:10.1007/978-3-662-47672-7_19.

6 Marco Bressan. Faster algorithms for counting subgraphs in sparse graphs. *Algorithmica*, 83(8):2578–2605, 2021. doi:10.1007/s00453-021-00811-0.

7 Andrei A. Bulatov and Amirhossein Kazeminia. Complexity classification of counting graph homomorphisms modulo a prime number. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1024–1037. ACM, 2022. doi:10.1145/3519935.3520075.

8 Jianer Chen, Benny Chor, Mike Fellows, Xiuzhen Huang, David W. Juedes, Iyad A. Kanj, and Ge Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, 201(2):216–231, 2005. doi:10.1016/j.ic.2005.05.001.

9 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.

10 Yijia Chen, Marc Thurley, and Mark Weyer. Understanding the Complexity of Induced Subgraph Isomorphisms. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 587–596. Springer, 2008. doi:10.1007/978-3-540-70575-8_48.

11 Radu Curticapean. Counting matchings of size k is w[1]-hard. In Fedor V. Fomin, Rusins Freivalds, Marta Z. Kwiatkowska, and David Peleg, editors, *Automata, Languages, and Programming - 40th International Colloquium, ICALP 2013, Riga, Latvia, July 8-12, 2013, Proceedings, Part I*, volume 7965 of *Lecture Notes in Computer Science*, pages 352–363. Springer, 2013. doi:10.1007/978-3-642-39206-1_30.

12 Radu Curticapean, Holger Dell, and Thore Husfeldt. Modular counting of subgraphs: Matchings, matching-splittable graphs, and paths. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms, ESA 2021, September 6-8, 2021, Lisbon, Portugal (Virtual Conference)*, volume 204 of *LIPIcs*, pages 34:1–34:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ESA.2021.34.

13 Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 210–223. ACM, 2017. doi:10.1145/3055399.3055502.

14 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.

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

16 Víctor Dalmau and Peter Jonsson. The complexity of counting homomorphisms seen from the other side. *Theoret. Comput. Sci.*, 329(1-3):315–323, 2004. doi:10.1016/j.tcs.2004.08.008.

17 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.

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

19 Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM J. Comput.*, 33(4):892–922, 2004. doi:10.1137/S0097539703427203.

20 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.

21 Martin Grohe and Dániel Marx. On tree width, bramble size, and expansion. *J. Comb. Theory, Ser. B*, 99(1):218–228, 2009. doi:10.1016/j.jctb.2008.06.004.

22 Daniel J. Harvey and David R. Wood. The treewidth of line graphs. *J. Comb. Theory, Ser. B*, 132:157–179, 2018. doi:10.1016/j.jctb.2018.03.007.

23 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.

24 Bart M. P. Jansen and Dániel Marx. Characterizing the easy-to-find subgraphs from the viewpoint of polynomial-time algorithms, kernels, and turing kernels. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 616–629. SIAM, 2015. `doi:10.1137/1.9781611973730.42`.

25 Oleksii Kuchaiev, Tijana Milenković, Vesna Memišević, Wayne Hayes, and Nataša Pržulj. Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, 7(50):1341–1354, 2010. `doi:10.1098/rsif.2010.0063`.

26 R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, 2002. `doi:10.1126/science.298.5594.824`.

27 Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663):1538–1542, 2004. `doi:10.1126/science.1089167`.

28 Norbert Peyerimhoff, Marc Roth, Johannes Schmitt, Jakob Stix, and Alina Vdovina. Parameterized (modular) counting and cayley graph expanders. In Filippo Bonchi and Simon J. Puglisi, editors, *46th International Symposium on Mathematical Foundations of Computer Science, MFCS 2021, August 23-27, 2021, Tallinn, Estonia*, volume 202 of *LIPIcs*, pages 84:1–84:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.MFCS.2021.84`.

29 Norbert Peyerimhoff, Marc Roth, Johannes Schmitt, Jakob Stix, Alina Vdovina, and Philip Wellnitz. Parameterized Counting and Cayley Graph Expanders. *SIAM J. Discrete Math.*, to appear.

30 Benjamin Schiller, Sven Jager, Kay Hamacher, and Thorsten Strufe. StreaM – A Stream-Based Algorithm for Counting Motifs in Dynamic Graphs. In *Proceedings of the 2nd International Conference on Algorithms for Computational Biology (AlCoB)*, pages 53–67. Springer International Publishing, 2015. `doi:10.1007/978-3-319-21233-3_5`.

31 Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput.*, 20(5):865–877, 1991. `doi:10.1137/0220053`.

32 Ngoc Hieu Tran, Kwok Pui Choi, and Louxin Zhang. Counting motifs in the human interactome. *Nature communications*, 4(1):1–8, 2013. `doi:10.1038/ncomms3241`.

33 Charalampos E. Tsourakakis, Jakub Pachocki, and Michael Mitzenmacher. Scalable motif-aware graph clustering. In *Proceedings of the 26th International Conference on World Wide Web (WWW)*, pages 1451–1460, 2017. `doi:10.1145/3038912.3052653`.

34 Virginia Vassilevska Williams, Joshua R. Wang, Richard Ryan Williams, and Huacheng Yu. Finding four-node subgraphs in triangle time. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1671–1680. SIAM, 2015. `doi:10.1137/1.9781611973730.111`.