

# Counting Edge-Injective Homomorphisms and Matchings on Restricted Graph Classes<sup>\*†</sup>

Radu Curticapean<sup>1</sup>, Holger Dell<sup>2</sup>, and Marc Roth<sup>3</sup>

- 1 Institute for Computer Science and Control, Hungarian Academy of Sciences (MTA SZTAKI), Budapest, Hungary  
radu.curticapean@gmail.com
- 2 Saarland University, Saarbrücken, Germany; and  
Cluster of Excellence “Multimodal Computing and Interaction” (MMCI),  
Saarbrücken, Germany  
hdell@mmci.uni-saarland.de
- 3 Saarland University, Saarbrücken, Germany; and  
Cluster of Excellence “Multimodal Computing and Interaction” (MMCI),  
Saarbrücken, Germany  
mroth@mmci.uni-saarland.de

---

## Abstract

---

We consider the parameterized problem of counting all matchings with exactly  $k$  edges in a given input graph  $G$ . This problem is  $\#W[1]$ -hard (Curticapean, ICALP 2013), so it is unlikely to admit  $f(k) \cdot n^{O(1)}$  time algorithms. We show that  $\#W[1]$ -hardness persists even when the input graph  $G$  comes from restricted graph classes, such as line graphs and bipartite graphs of arbitrary constant girth and maximum degree two on one side.

To prove the result for line graphs, we observe that  $k$ -matchings in line graphs can be equivalently viewed as edge-injective homomorphisms from the disjoint union of  $k$  paths of length two into (arbitrary) host graphs. Here, a homomorphism from  $H$  to  $G$  is *edge-injective* if it maps any two distinct edges of  $H$  to distinct edges in  $G$ . We show that edge-injective homomorphisms from a pattern graph  $H$  can be counted in polynomial time if  $H$  has bounded vertex-cover number after removing isolated edges. For hereditary classes  $\mathcal{H}$  of pattern graphs, we obtain a full complexity dichotomy theorem by proving that counting edge-injective homomorphisms, restricted to patterns from  $\mathcal{H}$ , is  $\#W[1]$ -hard if no such bound exists.

Our proofs rely on an edge-colored variant of Holant problems and a delicate interpolation argument; both may be of independent interest.

**1998 ACM Subject Classification** F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Counting Problems, G.2.2 Graph Theory

**Keywords and phrases** matchings, homomorphisms, line graphs, counting complexity, parameterized complexity

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2017.25

## 1 Introduction

Since Valiant’s seminal  $\#P$ -hardness result for the permanent [35], the complexity theory of counting problems has advanced to a classical subfield of computational complexity. As

---

\* A full version of the paper is available at <http://arxiv.org/abs/1702.05447>.

† This work was done while the authors were visiting the Simons Institute for the Theory of Computing. Radu Curticapean is supported by ERC Starting Grant PARAMTIGHT (No. 280152)



© Radu Curticapean, Holger Dell, and Marc Roth;  
licensed under Creative Commons License CC-BY

34th Symposium on Theoretical Aspects of Computer Science (STACS 2017).

Editors: Heribert Vollmer and Brigitte Vallée; Article No. 25; pp. 25:1–25:15

Leibniz International Proceedings in Informatics



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

it turned out that many interesting counting problems are  $\#P$ -hard, various *relaxations* of the original problems were introduced, giving rise to approximate [26], modular [3], and subexponential counting [17], with additional restrictions on the input classes [25, 38].

In this paper, we focus on a recent relaxation of hard counting problems by studying their *parameterized* complexity [18]. In this paradigm, the input to a given counting problem comes with a parameter  $k \in \mathbb{N}$ , and we ask whether the problem is *fixed-parameter tractable (FPT)*: That is, can it be solved in time  $f(k) \cdot \text{poly}(|x|)$  for some computable function  $f$  that may grow super-polynomially? For instance, the  $\#P$ -complete problem of counting vertex-covers of size  $k$  in an  $n$ -vertex graph can be solved in time  $2^k \cdot \text{poly}(n)$  (and even faster) and is hence FPT [18]. For other parameterized problems however, such as counting cliques, cycles, paths, or matchings of size  $k$  in  $n$ -vertex graphs, the best known algorithms run in time  $n^{O(k)}$ , and FPT-algorithms are not believed to exist. To substantiate this belief, Flum and Grohe [18] introduced the class  $\#W[1]$  and identified the problem of counting  $k$ -cliques to be complete for  $\#W[1]$  under parameterized reductions. Hence this problem is not FPT, unless the classes FPT and  $\#W[1]$  coincide, which is considered unlikely. Subsequently,  $\#W[1]$ -completeness was also shown for counting  $k$ -cycles and  $k$ -paths [18], and later on for counting  $k$ -matchings [11, 13]. Interestingly, the decision versions of these last three problems are in fact FPT [1] (or even polynomial-time solvable in the case of matchings).

As it turns out, the problem of counting  $k$ -matchings plays a central role in parameterized counting. This is partially due to its obvious similarity to Valiant's classical problem of counting perfect matchings. More importantly however,  $k$ -matchings represent an important reduction source to prove the hardness of other problems. For example, they constitute the bottleneck problem for counting general small subgraph patterns: Given a graph class  $\mathcal{H}$ , we can define a problem  $\#Sub(\mathcal{H})$  that asks, given a pattern graph  $H \in \mathcal{H}$  and a host graph  $G$ , to count the occurrences of  $H$  as a subgraph in  $G$ . The problem  $\#Sub(\mathcal{H})$  can be solved in polynomial time if the graphs in  $\mathcal{H}$  have a constant upper bound on the size of their matchings, whereas classes  $\mathcal{H}$  with matchings of unbounded size make the problem  $\#W[1]$ -complete [13]. This shows in particular that counting  $k$ -matchings is the minimal hard case for  $\#Sub(\mathcal{H})$ .

In this paper, we proceed from the  $\#W[1]$ -hardness of counting  $k$ -matchings in two directions: First, we strengthen this particular hardness result by showing that counting  $k$ -matchings remains  $\#W[1]$ -complete even on natural *restricted* graph classes, such as line graphs and bipartite graphs where one side has maximum degree 2. As an instrument in our proofs, we introduce the notion of *edge-injective homomorphisms*, which interpolates between the classical notions of homomorphisms and (subgraph) embeddings. In the second part of the paper, we study the parameterized complexity of counting edge-injective homomorphisms as a topic in itself. The proofs of lemmas, claims and theorems marked with  $\star$  appear in the full version of this paper, which can be found at <http://arxiv.org/abs/1702.05447>.

## 1.1 Counting matchings in restricted graph classes

In non-parameterized counting complexity, restrictions of hard problems to planar and bounded-degree graphs were studied extensively: We can count *perfect* matchings on planar graphs in polynomial time by the FKT method [33, 27], and several dichotomies show which counting versions of constraint satisfaction problems become easy on planar graphs [6, 2].

For the particular problem of counting (not necessarily perfect) matchings, a line of research [25, 15, 34] culminated in the work of Xia et al. [38] who showed that the problem remains  $\#P$ -hard even on planar bipartite graphs whose left and right side have maximum degree 2 and 3, respectively. In the parameterized setting, counting  $k$ -matchings is FPT

in planar or bounded-degree graphs [20], which rules out a parameterized analogue of the hardness result by Xia et al. [38]. It was however shown that counting  $k$ -matchings remains  $\#W[1]$ -complete on bipartite graphs [13], which was essential for the subsequent reductions to the general subgraph counting problem. In the first part of the paper, we find additional restricted graph classes on which counting  $k$ -matchings remains  $\#W[1]$ -complete.

### 1.1.1 Restricted bipartite graphs of high girth

In [13], the  $\#W[1]$ -completeness of counting  $k$ -matchings in bipartite graphs was actually shown for an edge-colorful variant where the edges of the bipartite graph are (not necessarily properly) colored with  $k$  colors and we wish to count  $k$ -matchings that pick exactly one edge from each color. This variant can be reduced to the uncolored one via inclusion–exclusion.

In this paper, we strengthen the  $\#W[1]$ -hardness result for counting edge-colorful  $k$ -matchings in bipartite graphs  $G$  and show that we may restrict one side of  $G$  to have maximum degree two. We may additionally assume any constant lower bound on the *girth* of  $G$ , that is, the length of the shortest cycle in  $G$ . For counting (edge-colorful)  $k$ -matchings, it is known that an algorithm with running time  $f(k) \cdot n^{o(k/\log k)}$  for any computable function  $f$  would refute the counting exponential-time hypothesis  $\#ETH$  [17]. That is, if such an algorithm existed, we could count satisfying assignments to 3-CNF formulas on  $n$  variables in time  $2^{o(n)}$ . Our result establishes the same lower bound in the restricted case.

► **Theorem 1** ( $\star$ ). *For every  $c \in \mathbb{N}$ , counting (edge-colorful or uncolored)  $k$ -matchings is  $\#W[1]$ -complete, even for bipartite graphs of girth at least  $c$  whose right side vertices have degree at most two. Furthermore, unless  $\#ETH$  fails, neither of these problems has an  $f(k) \cdot n^{o(k/\log k)}$ -time algorithm, for any computable function  $f$ .*

We sketch the proof in §3 by extending the so-called Holant problems [36, 4] to an edge-colored variant that proves to be useful for parameterized counting problems. In classical Holant problems, we are given as input a graph  $G = (V, E)$  with a signature  $f_v$  at each vertex  $v \in V$ . Here,  $f_v$  is a function  $f_v : \{0, 1\}^{I(v)} \rightarrow \mathbb{Z}$ , where  $\{0, 1\}^{I(v)}$  is the set of binary assignments to the edges incident with  $v$ . The problem is to compute  $\text{Holant}(G)$ , a sum over all binary assignments  $x \in \{0, 1\}^E$ , where each assignment  $x$  is weighted by  $\prod_{v \in V} f_v(x)$ .

In our edge-colored setting, the graph  $G$  is edge-colored and  $\text{Holant}(G)$  ranges only over assignments that pick exactly one edge from each color. We apply the recent technique of combined signatures [14] in this setting, an approach that is also implicit in [13]. This gives a reduction from counting edge-colorful  $k$ -matchings in general graphs to  $2^k$  instances of the restricted bipartite case. Previously, combined signatures were used only for problems with structural parameterizations, such as counting perfect matchings in graphs whose genus or apex number is bounded [14]. Our edge-colorful approach allows us to apply them also when the parameter is the solution size  $k$ .

### 1.1.2 Line graphs

Building upon Theorem 1, we then prove that counting  $k$ -matchings is  $\#W[1]$ -complete even when the input graph is a line graph. We also obtain a lower bound under  $\#ETH$ .

► **Theorem 2.** *The problem of counting  $k$ -matchings is  $\#W[1]$ -complete, even when restricted to line graphs. Furthermore, unless  $\#ETH$  fails, this problem does not have an  $f(k) \cdot n^{o(k/\log k)}$  time algorithm, for any computable function  $f$ .*

Line graphs are claw-free, that is, they exclude  $K_{1,3}$  as induced subgraphs. In fact, the class of line graphs can be characterized by a set  $S$  of nine (for large graphs seven) minimal subgraphs such that  $G$  is a line graph if and only if  $G$  contains none of the graphs from  $S$  as an induced subgraph [24, 37]. Line graphs can be recognized in linear time [28], and several classical NP-complete problems are polynomial-time solvable on line graphs, such as finding a maximum independent set [32], a maximum cut [23], or a maximum clique [30]. In contrast, Theorem 2 shows that counting  $k$ -matchings remains  $\#W[1]$ -hard on line graphs.

To prove Theorem 2, one might be tempted to first prove hardness of counting edge-colorful  $k$ -matchings in line graphs, and then reduce this problem via inclusion–exclusion to the uncolored case. This approach however fails: While the colored problem is easily shown to be  $\#W[1]$ -complete (even on complete graphs), we cannot use inclusion–exclusion to subsequently reduce to counting uncolored matchings, since doing so would lead to graphs that are not necessarily line graphs. Hence we do not know how to prove Theorem 2 in the framework of edge-colorful Holant problems directly, as we do for Theorem 1.

Instead, we prove Theorem 2 in §4 by means of a delicate interpolation argument, most similar to techniques used in the first hardness proof for uncolored  $k$ -matchings [11]. Using a simple gadget and  $k$ -matchings in line graphs as the oracle for our reduction, we generate a linear system of equations such that one of the unknowns is the number of  $k$ -matchings in a general input graph  $G$ , which is  $\#W[1]$ -complete to compute. The system turns out not to have full rank, but a careful analysis shows that the unknown we are interested in can still be uniquely determined in polynomial time.

### Perfect matchings in (perfect) line graphs

Completing the picture, we show that the non-parameterized problem of counting *perfect* matchings also remains  $\#P$ -hard on line graphs. This holds even for line graphs of bipartite graphs, which are known to be perfect, and which play an important role in the proof of the strong perfect graph theorem by Chudnovsky, Seymour, and Robertson [10].

► **Theorem 3** (\*). *The problem of counting perfect matchings is  $\#P$ -complete even for graphs that have maximum degree 4 and are line graphs of bipartite graphs. On the other hand, the problem is polynomial-time solvable on 3-regular line graphs.*

To prove this theorem, we invoke a dichotomy theorem for Holant problems by Cai, Lu, and Xia [7]: We show that the positive case of Theorem 3 can be reduced to a polynomial-time solvable Holant problem, while hardness in the negative case of Theorem 3 follows by reduction from a  $\#P$ -complete Holant problem. Due to space limitations the proof is deferred to the journal version of this paper.

## 1.2 Counting edge-injective homomorphisms

To prove Theorem 2, we actually prove the equivalent statement that counting edge-injective homomorphisms from the graph  $k \cdot P_2$  to host graphs  $G$  is  $\#W[1]$ -complete. Here, we write  $k \cdot P_2$  for the graph consisting of  $k$  disjoint copies of the path  $P_2$  with two edges. A homomorphism  $f$  from  $H$  to  $G$  is *edge-injective* if, for any distinct (but not necessarily disjoint) edges  $e = uv$  and  $e' = u'v'$  of  $H$ , the edges  $f(u)f(v)$  and  $f(u')f(v')$  in  $G$  are distinct (but not necessarily disjoint). The number of edge-injective homomorphisms from  $k \cdot P_2$  to  $G$  is easily seen to be equal to the number of  $k$ -matchings in  $L(G)$ , up to a factor of  $k! \cdot 2^k$ , which is the size of the automorphism group of a  $k$ -matching.

Starting from their application in the proof of Theorem 2, we observe that edge-injective homomorphisms are an interesting concept on its own, since they constitute a natural interpolation between homomorphisms and subgraph embeddings (which are vertex-injective homomorphisms). To study the complexity of counting edge-injective homomorphisms from general patterns, we define the problems  $\#\text{Hom}^*(\mathcal{H})$  for fixed graph classes  $\mathcal{H}$ : Given graphs  $H \in \mathcal{H}$  and  $G$ , the problem is to count the edge-injective homomorphisms from  $H$  to  $G$ . Similar frameworks exist for counting subgraphs [13], counting/deciding colorful subgraphs [13, 31, 22], counting/deciding induced subgraphs [9], and counting/deciding (not necessarily edge-injective) homomorphisms [21, 16]. In all of these cases, precise dichotomies are known for the parameterized complexity of the problem when the pattern is chosen from a fixed class  $\mathcal{H}$  and the parameter is  $|V(H)|$ . For instance, homomorphisms from  $\mathcal{H}$  can be counted in polynomial time if  $\mathcal{H}$  has bounded treewidth, and the problem is  $\#\text{W}[1]$ -complete otherwise [16]. A similar statement holds for the decision version of this problem, but here only the cores of the graphs in  $\mathcal{H}$  need to have bounded treewidth [21].

Our main outcome is a similar result for counting edge-injective homomorphisms: To state it, let the *weak vertex-cover number* of a graph  $G$  be defined as the size of the minimum vertex-cover in the graph obtained from  $G$  by deleting all *isolated edges*, that is, connected components with two vertices. Furthermore, a graph class  $\mathcal{H}$  is *hereditary* if  $H \in \mathcal{H}$  implies  $F \in \mathcal{H}$  for all induced subgraphs  $F$  of  $H$ .

► **Theorem 4** ( $\star$ ). *Let  $\mathcal{H}$  be any class of graphs. The problem  $\#\text{Hom}^*(\mathcal{H})$  can be solved in polynomial time if there is a constant  $c \in \mathbb{N}$  such that the weak vertex-cover number of all graphs in  $\mathcal{H}$  is bounded by  $c$ . If no such constant exists and  $\mathcal{H}$  additionally is hereditary, then  $\#\text{Hom}^*(\mathcal{H})$  is  $\#\text{W}[1]$ -complete.*

To prove this theorem in §5, we first adapt an algorithm for counting subgraphs of bounded vertex-cover number [13] to the setting of edge-injective homomorphisms. Then we use a Ramsey argument to show that any graph class with unbounded weak vertex-cover number contains one of six hard classes as induced subgraphs. This gives a full dichotomy for the complexity of  $\#\text{Hom}^*(\mathcal{H})$  on hereditary graph classes  $\mathcal{H}$ , however leaving out several interesting non-hereditary classes such as paths, cycles, and  $P_c$ -packings. Here, a  $P_c$ -packing for  $c \in \mathbb{N}$  is a disjoint union of paths  $P_c$ , that is, paths consisting of  $c$  edges. We handle these specific classes individually.

► **Theorem 5** ( $\star$ ). *The problem  $\#\text{Hom}^*(\mathcal{H})$  is  $\#\text{W}[1]$ -complete if  $\mathcal{H}$  is the class of paths, the class of cycles, or the class of  $P_c$ -packings, for any  $c \geq 2$ .*

We conclude this introduction with a possible future application of edge-injective homomorphisms. A wide open problem in parameterized complexity lies in classifying the subgraph patterns whose *existence* is easy to decide. The problem is known to be FPT on patterns of bounded treewidth [1], and it seems reasonable to believe that all classes  $\mathcal{H}$  of unbounded treewidth are  $\text{W}[1]$ -hard. However, even  $\text{W}[1]$ -hardness for the class of complete bipartite graphs was only shown recently in a major breakthrough [29]. On the other hand, the complexity is much better understood for deciding the existence of *homomorphisms* from a pattern class  $\mathcal{H}$ : As stated above, the treewidth of the cores is the criterion for the complexity dichotomy [21]. Since subgraph embeddings are vertex-injective homomorphisms, the notion of edge-injective homomorphisms interpolates between the solved case of homomorphisms and the unsolved case of subgraphs. In light of this fact, we also consider our results on edge-injective homomorphisms as an initial investigation of a potential avenue towards a dichotomy for deciding subgraph patterns.

## 2 Preliminaries

A *parameterized counting problem* is a function  $\Pi : \{0, 1\}^* \rightarrow \mathbb{N}$  that is endowed with a computable *parameterization*  $\kappa : \{0, 1\}^* \rightarrow \mathbb{N}$ ; it is *fixed-parameter tractable (FPT)* if there is a computable function  $f : \mathbb{N} \rightarrow \mathbb{N}$  and an  $f(k) \cdot \text{poly}(n)$ -time algorithm to compute  $\Pi(x)$ , where  $n = |x|$  and  $k = \kappa(x)$ .

An *fpt Turing reduction* is a Turing reduction from a problem  $(\Pi, \kappa)$  to a problem  $(\Pi', \kappa')$ , such that the reduction runs in  $f(k) \cdot \text{poly}(n)$ -time and each query  $y$  to the oracle satisfies  $\kappa'(y) \leq g(k)$ . Here, both  $f$  and  $g$  are computable functions. A problem is *#W[1]-hard* if there is an fpt Turing reduction from the problem of counting the cliques of size  $k$  in a given graph; since it is believed that the latter does not have an FPT-algorithm, #W[1]-hardness is a strong indicator that a problem is not FPT. For more details, see [19].

The *counting exponential-time hypothesis (#ETH)* is the claim that there exists a constant  $\epsilon > 0$  for which there is no  $2^{\epsilon n}$ -time algorithm to compute the number of satisfying assignments for an  $n$ -variable 3-CNF formula. For the counting  $k$ -cliques problem, #ETH implies that there is no  $f(k) \cdot n^{o(k)}$ -time algorithm [8]. Whenever an fpt Turing reduction from counting  $k$ -cliques (or any source problem) to another parameterized counting problem increases the parameter  $k$  by at most a constant factor, then the same running time lower bound under #ETH holds for the target problem as well.

Let  $H$  and  $G$  be graphs. A function  $\varphi : V(H) \rightarrow V(G)$  is a *homomorphism* from  $H$  to  $G$  if  $\varphi(e) \in E(G)$  holds for all  $e \in E(H)$ , where  $\varphi(\{u, v\}) = \{\varphi(u), \varphi(v)\}$ . The set of all homomorphisms from  $H$  to  $G$  is denoted by  $\text{Hom}(H, G)$ . A homomorphism  $\varphi \in \text{Hom}(H, G)$  is called *edge-injective* if all  $e, f \in E(H)$  with  $e \neq f$  satisfy  $\varphi(e) \neq \varphi(f)$ . We denote the set of all edge-injective homomorphisms from  $H$  to  $G$  by  $\text{Hom}^*(H, G)$ . A homomorphism  $\varphi \in \text{Hom}(H, G)$  is an *embedding* of  $H$  in  $G$  if it is a (vertex-)injective homomorphism from  $H$  to  $G$ . The set of all embeddings from  $H$  to  $G$  is denoted by  $\text{Emb}(H, G)$ .

For a class  $\mathcal{H}$  of graphs, let  $\#\text{Hom}^*(\mathcal{H})$  denote the following computational problem: Given  $H \in \mathcal{H}$  and a simple graph  $G$ , compute the number  $\#\text{Hom}^*(H, G)$ , parameterized by  $|V(H)|$ . The problems  $\#\text{Hom}(\mathcal{H})$  and  $\#\text{Emb}(\mathcal{H})$  are defined analogously.

The *line graph*  $L(G)$  of a simple graph  $G$  is the graph whose vertex set satisfies  $V(L(G)) = E(G)$  such that  $e, f \in E(G)$  with  $e \neq f$  are adjacent in  $L(G)$  if and only if the edges  $e$  and  $f$  are incident to the same vertex in  $G$ . A line graph is called *line-perfect* if it is the line graph of a bipartite graph.

## 3 Matchings in restricted bipartite graphs

In this section, we prove Theorem 1. Our arguments make heavy use of  *$k$ -edge-colored graphs*, for  $k \in \mathbb{N}$ , which are graphs  $G$  with a (not necessarily proper) edge-coloring  $c : E(G) \rightarrow [k]$ . A matching in  $G$  is *colorful* if it contains exactly one edge from each color. We let  $\#\text{ColMatch}(G)$  be the number of such matchings and  $\#\text{ColMatch}$  be the corresponding computational problem; this problem is #W[1]-hard by the following theorem.

► **Theorem 6** ([13], Theorem 1.2). *The problem #ColMatch is #W[1]-complete. Unless #ETH fails, it cannot be solved in time  $f(k) \cdot n^{o(k/\log k)}$  for any computable  $f$ .*

A straightforward application of the inclusion–exclusion principle reduces the edge-colorful version to the uncolored one (see, e.g., [12, Lemma 1.34] or [13, Lemma 2.7]).

► **Lemma 7.** *There is an fpt Turing reduction from #ColMatch for  $k$ -edge-colored graphs to the problem of counting  $k$ -matchings in uncolored subgraphs of  $G$ ; the reduction makes at most  $2^k$  queries, each query is a subgraph of  $G$ , and the parameter of each query is  $k$ .*

### 3.1 Colorful Holant problems

We first adapt Holant problems to an edge-colorful setting by introducing *edge-colored signature graphs* and *colorful Holant problems*. In the uncolored setting, the notion of a “Holant” was introduced by Valiant [36] and later developed to a general theory of Holant problems by Cai, Lu, Xia, and various other authors [4, 5]. In Section 3.2, we will use colorful Holants to prove Theorem 1 by a reduction from  $\#\text{ColMatch}$ . A more general exposition of this material appears in the first author’s PhD thesis [12, Chapters 2 and 5.2].

► **Definition 8.** For a graph  $G$ , we denote the edges incident with a given vertex  $v \in V(G)$  by  $I(v)$ . For  $k \in \mathbb{N}$ , a *k-edge-colored signature graph* is a  $k$ -edge-colored graph  $\Omega$  that has a *signature*  $f_v : \{0, 1\}^{I(v)} \rightarrow \mathbb{Q}$  associated with each vertex  $v \in V(\Omega)$ . The graph underlying  $\Omega$  may feature parallel edges.

Given such an  $\Omega$ , denote its color classes by  $E_1, \dots, E_k \subseteq E(\Omega)$ . An assignment  $x \in \{0, 1\}^{E(\Omega)}$  is *colorful* if, for each  $i \in [k]$ , there is exactly one edge  $e \in E_i$  with  $x(e) = 1$ . Given a set  $S \subseteq E(\Omega)$ , we write  $x|_S$  for the restriction of  $x$  to  $S$ , which is the unique assignment in  $\{0, 1\}^S$  that agrees with  $x$  on  $S$ . We then define  $\text{ColHolant}(\Omega)$  as the sum

$$\text{ColHolant}(\Omega) = \sum_{\substack{x \in \{0, 1\}^{E(\Omega)} \\ \text{colorful}}} \prod_{v \in V(\Omega)} f_v(x|_{I(v)}).$$

If all signatures in  $\Omega$  map to  $\{0, 1\}$ , then  $\text{ColHolant}(\Omega)$  simply counts those edge-colorful assignments  $x$  with  $f_v(x|_{I(v)}) = 1$  for all  $v \in V(\Omega)$ . In the following, we will use this simple fact to rephrase  $\#\text{ColMatch}$  as  $\text{ColHolant}(\Omega)$  for an edge-colored signature graph  $\Omega$ .

For assignments  $x \in \{0, 1\}^*$ , write  $\text{hw}(x)$  for the Hamming weight of  $x$ . For a statement  $\varphi$ , let  $[\varphi]$  be defined to be 1 if  $\varphi$  holds and 0 otherwise.

► **Fact 9.** Let  $k \in \mathbb{N}$  and let  $G$  be a  $k$ -edge-colored graph. Define the *k-edge-colored signature graph*  $\Omega = \Omega(G)$  by associating with each vertex  $v \in V(G)$  the signature  $\text{hw}_{\leq 1} : \{0, 1\}^{I(v)} \rightarrow \{0, 1\}$ ; for any  $x \in \{0, 1\}^*$ , this signature is defined as  $\text{hw}_{\leq 1}(x) = [\text{hw}(x) \leq 1]$ . Then we can verify that  $\text{ColHolant}(\Omega) = \#\text{ColMatch}(G)$  holds.

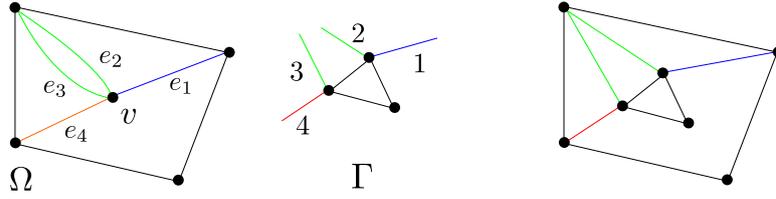
If a signature graph  $\Omega$  has a vertex  $v$  with some complicated signature  $f$  associated with it, we can sometimes simulate the effect of  $f$  by replacing  $v$  with a graph fragment that has only the signature  $\text{hw}_{\leq 1}$  associated with its vertices. Since  $\text{ColHolant}(\Omega)$  of signature graphs  $\Omega$  featuring only  $\text{hw}_{\leq 1}$  can be expressed as a number of edge-colorful matchings via Fact 9, this will allow us to reduce from  $\text{ColHolant}(\Omega)$  to  $\#\text{ColMatch}$ . The graph fragments we are looking for are formalized as *edge-colored matchgates*:

► **Definition 10.** An *edge-colored matchgate* is an edge-colored signature graph  $\Gamma$  that contains a set  $D \subseteq E(\Gamma)$  of *dangling* edges. These are edges with only one endpoint in  $V(\Gamma)$ , and we consider them to be labeled with  $1, \dots, |D|$ . Furthermore, we require the signature  $\text{hw}_{\leq 1}$  to be associated with all vertices in  $\Gamma$ . The colors on edges  $E(\Gamma) \setminus D$  will be denoted as *internal* colors.

We say that an assignment  $y \in \{0, 1\}^{E(\Gamma)}$  extends an assignment  $x \in \{0, 1\}^D$  if  $y$  agrees with  $x$  on  $D$ . The signature  $\text{ColSig}(\Gamma) : \{0, 1\}^D \rightarrow \mathbb{Q}$  of  $\Gamma$  is defined as

$$\text{ColSig}(\Gamma, x) = \sum_{\substack{y \in \{0, 1\}^{E(\Gamma)} \\ \text{colorful, extends } x}} \prod_{v \in V(\Gamma)} f_v(y|_{I(v)}).$$

If  $\Omega$  is a  $k$ -edge-colored signature graph and  $\Gamma$  is an edge-colored matchgate with internal colors disjoint from  $[k]$ , then we can *insert*  $\Gamma$  at a vertex  $v \in V(\Omega)$  as follows (see Figure 1):



■ **Figure 1** A matchgate  $\Gamma$  is inserted into a signature graph  $\Omega$  at vertex  $v$ .

First delete  $v$  from  $\Omega$ , but keep  $I(v)$  as dangling edges in  $\Omega$ . Then insert a disjoint copy of  $\Gamma$  into  $\Omega$  and identify its dangling edges with  $I(v)$ . That is, if  $e$  is a dangling edge with endpoint  $u$  in  $\Omega$  and  $e$  is identified with a dangling edge of the same color with endpoint  $v$  in  $\Gamma$ , then we consider  $e$  as an edge  $uv$  in the resulting graph.<sup>1</sup>

► **Remark.** When inserting  $\Gamma$  into a signature graph  $\Omega$ , we implicitly assume that the edge-colors of dangling edges are a subset of the edge-colors in  $\Omega$ . Furthermore, note that the insertion of matchgates can result in multigraphs.

A simple calculation shows that inserting a matchgate  $\Gamma$  at a vertex  $v$  with signature  $f_v$  preserves  $\text{ColHolant}(\Omega)$ , provided that  $\text{ColSig}(\Gamma) = f_v$ . Applying this insertion operation repeatedly, we obtain the following fact, as proved in Fact 2.17 and Lemma 5.16 of [12].

► **Fact 11.** *Let  $\Omega$  be a  $k$ -edge-colored signature graph such that each  $v \in V(\Omega)$  is associated with some signature  $f_v$ . If there is a matchgate  $\Gamma_v$  with  $\text{ColSig}(\Gamma_v) = f_v$  for every vertex  $v$ , then we can efficiently construct an edge-colored graph  $G$  on  $O(\sum_v |V(\Gamma_v)| + \sum_v |E(\Gamma_v)|)$  vertices and edges such that  $\text{ColHolant}(\Omega) = \#\text{ColMatch}(G)$ .*

In some cases, Fact 11 may not be applicable, since the involved signatures cannot be realized by matchgates. For such cases, Curticapean and Xia [14] define *combined signatures*: Rather than realizing a given signature  $f$  via matchgates, we may be able to express  $f$  as a linear combination of  $t \in \mathbb{N}$  signatures that do admit matchgates. If there are  $s \in \mathbb{N}$  occurrences of such signatures in  $\Omega$ , then we can compute  $\text{ColHolant}(\Omega)$  as a linear combination of  $t^s$  colorful Holants, where all involved signatures can be realized by matchgates.

► **Lemma 12** ( $\star$ ). *Let  $\Omega$  be a  $k$ -colored signature graph. Let  $s, t \in \mathbb{N}$  and let  $w_1, \dots, w_s$  be distinct vertices of  $\Omega$  such that the following holds: For all  $\kappa \in [s]$ , the signature  $f_\kappa$  at  $w_\kappa$  admits coefficients  $c_{\kappa,1}, \dots, c_{\kappa,t} \in \mathbb{Q}$  and signatures  $g_{\kappa,1}, \dots, g_{\kappa,t}$  such that  $f_\kappa = \sum_{i=1}^t c_{\kappa,i} \cdot g_{\kappa,i}$  holds. Here, the linear combination is to be understood point-wise.*

*Given a tuple  $\theta \in [t]^s$ , let  $\Omega_\theta$  be the edge-colored signature graph defined by replacing, for each  $\kappa \in [s]$ , the signature  $f_\kappa$  at  $w_\kappa$  with  $g_{\kappa,\theta(\kappa)}$ . Then we have*

$$\text{ColHolant}(\Omega) = \sum_{\theta \in [t]^s} \left( \prod_{\kappa=1}^s c_{\kappa,\theta(\kappa)} \right) \cdot \text{ColHolant}(\Omega_\theta). \quad (1)$$

Lemma 12 allows us to prove hardness results under fpt Turing reductions if  $\text{ColHolant}(\Omega)$  is  $\#\text{W}[1]$ -hard to compute and the values  $\text{ColHolant}(\Omega_\theta)$  for all  $\theta$  can be computed by reductions to the target problem. This is our approach in the remainder of this section.

<sup>1</sup> We assume  $I(v)$  to be ordered as  $e_1, \dots, e_{d(v)}$  in some arbitrary way; then  $e_b$  is identified with dangling edge  $b$  for all  $b \in [d(v)]$ . This also requires  $e_b$  and dangling edge  $b$  to have the same color.

### 3.2 $k$ -Matchings in bipartite graphs

In the following, we use the techniques from Section 3.1 to prove Theorem 1. We reduce from  $\#\text{ColMatch}$ , which is  $\#\text{W}[1]$ -complete by Theorem 6. Let  $k \in \mathbb{N}$  and let  $G$  be a simple  $k$ -edge-colored graph for which we want to compute  $\#\text{ColMatch}(G)$ . We first construct a certain bipartite signature graph  $\Omega_{\text{bip}}$  with  $\text{ColHolant}(\Omega_{\text{bip}}) = \#\text{ColMatch}(G)$ .

► **Lemma 13** ( $\star$ ). *Given a  $k$ -edge-colored graph  $G$ , let  $\Omega_{\text{bip}} = \Omega_{\text{bip}}(G)$  denote the signature graph on edge-colors  $[k] \times [2]$  constructed as follows: Initially,  $\Omega_{\text{bip}}$  is  $G$ , where each vertex is associated with the signature  $\text{hw}_{\leq 1}$ . Then, for each  $i \in [k]$ , perform the following:*

1. Add a fresh vertex  $w_i$  to  $\Omega_{\text{bip}}$ .
2. For  $e \in E(G)$ , of color  $i$  and with  $e = uv$ , delete  $e$  and insert an edge  $uw_i$  of color  $(i, 1)$  and an edge  $w_iv$  of color  $(i, 2)$ . Annotate the added edges with  $\pi(uw_i) = \pi(w_iv) = e$ .
3. Note that every colorful assignment  $x \in \{0, 1\}^{I(w_i)}$  at a vertex  $w_i$  has precisely two edges  $e_1(x)$  and  $e_2(x)$  that are incident to  $w_i$  and assigned 1 by  $x$ . We associate  $w_i$  with the signature  $f_i$  that maps  $x \in \{0, 1\}^{I(w_i)}$  to  $f_i(x) = [\pi(e_1(x)) = \pi(e_2(x))]$ .

The constructed signature graph  $\Omega_{\text{bip}}$  satisfies  $\text{ColHolant}(\Omega_{\text{bip}}) = \#\text{ColMatch}(G)$ .

We now realize the signatures  $f_i$  in  $\Omega_{\text{bip}}$  by linear combinations of the signatures of edge-colored matchgates. For  $i \in [k]$ , let  $E_i(G)$  denote the edges of color  $i$  in  $G$ . Let  $m_i = |E_i(G)|$  and consider the edges in  $E_i(G)$  to be ordered in some arbitrary fixed way.

► **Lemma 14** ( $\star$ ). *Recall the definition of  $\Omega_{\text{bip}}$  from Lemma 13. For  $i \in [k]$ , let  $m = m_i$  and let  $\Gamma_{i,1}$  denote the matchgate on dangling edges  $I(w_i)$  that consists of  $2m$  vertices and is defined as follows: First, create independent sets  $a_1, \dots, a_m$  and  $b_1, \dots, b_m$ , which we call “external” vertices. Then, for all  $j \in [m]$  and all edges  $e, e' \in E(\Omega_{\text{bip}})$  of colors  $(i, 1)$  and  $(i, 2)$  with  $\pi(e) = \pi(e')$ : If  $\pi(e)$  is the  $j$ -th edge in the ordering of  $E_i(G)$ , for  $j \in \mathbb{N}$ , then attach  $e$  as dangling edge to  $a_j$  and  $e'$  as dangling edge to  $b_j$ .*

Let  $\Gamma_{i,2}$  be defined likewise, with the following addition: For all  $j \in [m]$ , add an extra vertex  $c_j$ , an edge  $a_jc_j$  of color  $(i, 3)$  and an edge  $c_jb_j$  of color  $(i, 4)$ .

Recall the signature  $f_i$  from Lemma 13. We can express  $f_i$  as a linear combination of  $\text{ColSig}(\Gamma_{i,1})$  and  $\text{ColSig}(\Gamma_{i,2})$  by  $f_i = (m^2 - 3m + 3) \cdot \text{ColSig}(\Gamma_{i,1}) - \text{ColSig}(\Gamma_{i,2})$ .

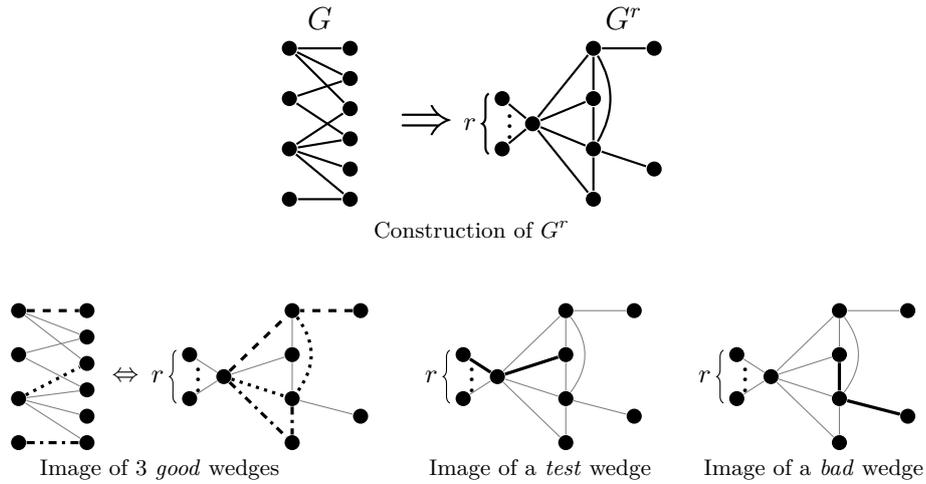
Using Lemmas 12, 13 and 14, we can now reduce counting edge-colorful matchings in graphs  $G$  to the same problem in subdivisions of  $G$ . If  $G$  is a  $k$ -colored graph and  $t \in \mathbb{N}$  is some number, then a  $t$ -subdivision of  $G$  for  $t \in \mathbb{N}$  is obtained by replacing each edge of  $G$  by a path with exactly  $t$  inner vertices. We may assign any colors to the new edges.

► **Lemma 15** ( $\star$ ). *Let  $G$  be a  $k$ -edge-colored graph on  $n$  vertices and  $m$  edges. Then we can compute  $\#\text{ColMatch}(G)$  with  $O(2^k)$  oracle calls  $\#\text{ColMatch}(G')$  for graphs  $G'$  that are subgraphs of a 3-subdivision of  $G$ . Furthermore,  $G'$  has at most  $4(n + m)$  vertices and edges and at most  $4k$  colors.*

Theorem 1 now follows easily from the hardness of  $\#\text{ColMatch}$  and repeated applications of Lemma 15. The full proof is given in the full version of this paper.

## 4 Matchings in line graphs

We now sketch the proof of Theorem 2, stating that counting  $k$ -matchings in line graphs is  $\#\text{W}[1]$ -hard. A *wedge* is any graph isomorphic to  $P_2$ , the path with two edges, and a *wedge packing*  $k \cdot P_2$  is the vertex-disjoint union of  $k$  wedges. For any graph  $G$ , we observe that the number of embeddings of a  $k$ -matching in  $L(G)$  is equal to the number of edge-injective



■ **Figure 2** Example of the construction of  $G^r$  as used in the proof of Theorem 17. The second row illustrates the correspondence between a 3-matching in  $G$  and the image of an edge-injective homomorphism from a wedge packing of size 3 such that all wedges are *good*. Furthermore we give examples for the image of a *test* wedge and a *bad* wedge.

homomorphisms from a wedge packing  $k \cdot P_2$  to  $G$ . To prove Theorem 2, we reduce from the  $k$ -matching problem in well-structured bipartite graphs to the latter problem. The following technical lemma encapsulates the delicate interpolation argument used in the reduction. For  $t \in \mathbb{N}$ , let  $(x)_t = (x) \cdot (x-1) \cdots (x-t+1)$  denote the falling factorial.

▶ **Lemma 16** ( $\star$ ). For all  $g, b \in \mathbb{N}$ , let  $a_{g,b} \in \mathbb{Q}$  be unknowns, and for all  $r \in \mathbb{N}$ , let  $P_r(y)$  be the univariate polynomial such that

$$P_r(y) = \sum_{k=0}^r \sum_{t=0}^k a_{t,k-t} \cdot \binom{r}{k} \cdot (y-t)_{2(r-k)}.$$

There is a polynomial-time algorithm that, given a number  $k$  and the coefficients of  $P_r(y)$  for all  $r \in \mathbb{N}$  with  $r \leq O(k)$ , computes the numbers  $a_{t,k-t}$  for all  $t \in \{0, \dots, k\}$ .

We then prove Theorem 2 by showing the following equivalent theorem.

▶ **Theorem 17.** If  $\mathcal{H}$  is the class of all wedge packings, then  $\#\text{Hom}^*(\mathcal{H})$  is  $\#\text{W}[1]$ -hard. Furthermore, unless  $\#\text{ETH}$  fails, the problem cannot be solved in time  $f(k) \cdot n^{o(k/\log k)}$ .

**Proof.** We reduce from the problem of counting  $k$ -matchings in bipartite graphs whose right-side vertices have degree  $\leq 2$  and where any two distinct left-side vertices have at most one common neighbor. For this problem, Theorem 1 for bipartite graphs with girth greater than 4 implies  $\#\text{W}[1]$ -hardness and the desired bound under  $\#\text{ETH}$ . Let  $(G, k)$  be an instance of this problem, and let  $L(G)$  and  $R(G)$  be the left and right vertex sets, respectively. For  $r \in \mathbb{N}$ , we construct a graph  $G^r$  as follows (see Figure 2):

1. Insert a vertex 0 that is adjacent to all vertices of  $L(G)$ .
2. Add  $r$  special vertices  $1, \dots, r$  as well as the edges  $01, 02, \dots, 0r$ .
3. For every vertex  $v \in R(G)$  with  $\deg(v) = 2$ , remove  $v$  and add the set  $N(v)$  as an edge to  $G^r$ . Note that  $|N(v)| = 2$ , so  $N(v)$  can indeed be considered as an edge.

Since  $G$  is a simple graph and any two distinct vertices  $u, v \in L(G)$  have at most one common neighbor in  $G$ , the graph  $G^r$  is again simple. Let  $H = H_1 \dot{\cup} \dots \dot{\cup} H_k$  be the graph that consists of  $k$  vertex-disjoint copies of  $P_2$ . For  $\varphi \in \text{Hom}^*(H, G^0)$ , we say that a wedge  $H_i$  is

- *test* if  $\varphi(H_i)$  contains two edges incident to 0,
- *good* if  $\varphi(H_i)$  contains exactly one edge incident to 0, and
- *bad* if  $\varphi(H_i)$  uses no edge incident to 0.

Let  $\alpha_{g,b}$  be the number of edge-injective homomorphisms  $\varphi \in \text{Hom}^*(H, G^0)$  for which there are 0 test wedges,  $g$  good wedges, and  $b$  bad wedges.

► **Claim 18** ( $\star$ ). *The number of  $k$ -matchings in  $G$  is equal to  $\alpha_{k,0}/(2^k \cdot k!)$ .*

We aim at determining the number  $\alpha_{k,0}$  by using an oracle for  $\#\text{Hom}^*(\mathcal{H})$ . Since we cannot directly ask the oracle to only count homomorphisms with a given number of bad and good wedges, we query the oracle multiple times and recover these numbers via a very specific form of interpolation fueled by Lemma 16. To apply the lemma, we observe the following identity.

► **Claim 19** ( $\star$ ). *Let  $k, r \in \mathbb{N}$ . Then  $\beta_k(G^r) := \#\text{Hom}^*(H, G^r)$  satisfies*

$$\beta_k(G^r) = \sum_{\substack{t, g, b \in \mathbb{N} \\ t+g+b=k}} \alpha_{g,b} \cdot \binom{k}{g+b} \cdot (n+r-g)_{2t}.$$

Note that  $\beta_k(G^r)$  is a polynomial in  $r$  of degree at most  $2k$ . Setting  $y = n+r$ , Claim 19 yields a polynomial identity that is exactly of the form required by Lemma 16, and thus we can compute the unknowns  $\alpha_{g,b}$  for all  $g, b \in \mathbb{N}$  with  $g+b \leq k$  from the polynomials  $\beta_0, \dots, \beta_{O(k)}$ . Overall, the reduction runs in polynomial time, makes at most  $O(k^2)$  queries to the oracle, and the parameter of each query is at most  $O(k)$ . This proves the  $\#\text{W}[1]$ -hardness and the lower bound under  $\#\text{ETH}$ . ◀

## 5 Edge-injective homomorphisms

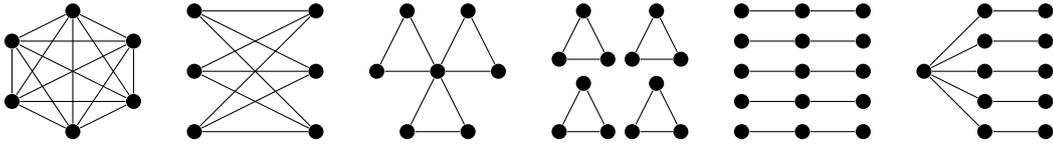
We sketch the proof of Theorem 4, our dichotomy theorem for counting edge-injective homomorphisms. Let  $H$  be a graph. Recall that a set  $S \subseteq V(H)$  is a *weak vertex-cover* if every edge  $e \in E(H)$  either has a non-empty intersection with  $S$  or  $e$  does not have any other edges incident to it. The *weak vertex-cover number* of  $G$  is the minimum size of a weak vertex-cover of  $G$ . A family of graphs  $\mathcal{H}$  has *bounded* weak vertex-cover number if this number can be uniformly bounded by a constant  $c = c(\mathcal{H})$  for all graphs  $H \in \mathcal{H}$ ; otherwise this number is *unbounded*.

### 5.1 Polynomial-time counting for bounded weak vertex-cover number

The polynomial-time cases of our dichotomy are established in the following theorem.

► **Theorem 20** ( $\star$ ). *If  $\mathcal{H}$  is a family of graphs with bounded weak vertex-cover number, then  $\#\text{Hom}^*(\mathcal{H})$  is polynomial-time computable.*

The algorithm is based on dynamic programming. Let  $H \in \mathcal{H}$  and  $G$  be the input for the algorithm. Isolated edges of  $H$  can be removed easily, as their contribution to the number of edge-injective homomorphisms admits a closed formula. The basic idea now is to guess which  $c$  vertices in  $G$  the vertex-cover of  $H$  maps to; after this part of the homomorphism is fixed, all vertices of  $H$  outside of the vertex-cover form an independent set, and they



■ **Figure 3** Example graphs from each of the six minimal graph classes that do not have bounded weak vertex-cover number according to Lemma 21:  $K_6$ ,  $K_{3,3}$ ,  $W_3$ ,  $4 \cdot K_3$ ,  $5 \cdot P_2$ , and  $SS_5$ .

can only be distinguished if their neighborhoods are distinct. Since there are at most  $2^c$  different neighborhoods, the graph  $H$  has a very simple structure, and a surprisingly technical dynamic programming algorithm achieves a running time of  $n^{O(2^c)}$ .

## 5.2 Hardness for hereditary graph classes

We now consider graph classes  $\mathcal{H}$  that do *not* have bounded weak vertex-cover number, and we prove that  $\#\text{Hom}^*(\mathcal{H})$  is  $\#\text{W}[1]$ -complete if  $\mathcal{H}$  is also hereditary. To do so, we first show that every class of unbounded weak vertex-cover number contains one of six basic graph classes (depicted in Figure 3) as induced subgraphs.

For the purposes of this paper, we say that  $W_k$  is a windmill of size  $k$  if it is a matching of size  $k$  with an additional *center vertex* adjacent to every other vertex. Moreover, the *subdivided star*  $SS_k$  is a  $k$ -matching with a center vertex that is adjacent to exactly one vertex of each edge in the matching. A *triangle packing*  $k \cdot K_3$  is the disjoint union of  $k$  triangles, a *wedge* is a path  $P_2$  that consists of two edges, and a *wedge packing*  $k \cdot P_2$  is the disjoint union of  $k$  wedges.

► **Lemma 21** ( $\star$ ). *Let us say that a class  $\mathcal{H}$  contains another class  $\mathcal{C}$  as induced subgraphs if, for every  $C \in \mathcal{C}$ , there is some  $H \in \mathcal{H}$  such that  $H$  contains  $C$  as induced subgraph. If  $\mathcal{H}$  is a class of graphs with unbounded weak vertex-cover number, then  $\mathcal{H}$  contains at least one of the following classes as induced subgraphs:*

- (i) *the class of all cliques,*
- (ii) *the class of all bicliques,*
- (iii) *the class of all subdivided stars,*
- (iv) *the class of all windmills,*
- (v) *the class of all triangle packings, or*
- (vi) *the class of all wedge packings.*

Since hereditary classes  $\mathcal{H}$  are closed under induced subgraphs, Lemma 21 guarantees that any hereditary class  $\mathcal{H}$  with unbounded weak vertex-cover number contains at least one of the six graph families defined above as an actual subset of  $\mathcal{H}$ . We prove hardness for each of these six families in the journal version of this paper; the hardness for hereditary classes  $\mathcal{H}$  and Theorem 4 then follows.

## 5.3 Hardness for some non-hereditary graph classes

The dichotomy for  $\#\text{Hom}^*(\mathcal{H})$  with hereditary graph classes  $\mathcal{H}$  leaves open some non-hereditary graph classes of interest. In the final part of the paper, we investigate  $\#\text{Hom}^*(\mathcal{H})$  for several such classes, namely those of cycles, paths, and packings of constant-length paths. It turns out that the problem of counting edge-injective homomorphisms is  $\#\text{W}[1]$ -hard in all of these cases (excluding the class of matchings, which are packings of length-1 paths).

► **Theorem 22** ( $\star$ ). For the classes  $\mathcal{C}$  and  $\mathcal{P}$  of all cycles and paths, respectively, the problems  $\#\text{Hom}^*(\mathcal{C})$  and  $\#\text{Hom}^*(\mathcal{P})$  are  $\#\text{W}[1]$ -hard. Furthermore, the problem of counting all edge-disjoint  $s$ - $t$ -walks in a given graph is  $\#\text{W}[1]$ -hard.

► **Theorem 23** ( $\star$ ). For  $c \in \mathbb{N}$ , let  $\mathcal{PP}_c$  be the class of packings of the path  $P_c$ . Then  $\#\text{Hom}^*(\mathcal{PP}_c)$  is  $\#\text{W}[1]$ -hard for  $c \geq 2$  and computable in polynomial time otherwise.

**Acknowledgments.** The authors thank Cornelius Brand and Markus Bläser for interesting discussions, and Johannes Schmitt for pointing out a proof of Lemma 16.

---

## References

- 1 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.
- 2 Jin-Yi Cai and Zhiguo Fu. Holographic algorithm with matchgates is universal for planar  $\#\text{CSP}$  over boolean domain. *CoRR*, abs/1603.07046, 2016. URL: <http://arxiv.org/abs/1603.07046>.
- 3 Jin-Yi Cai and Lane A. Hemachandra. On the power of parity polynomial time. *Mathematical Systems Theory*, 23(2):95–106, 1990. doi:10.1007/BF02090768.
- 4 Jin-Yi Cai and Pinyan Lu. Holographic algorithms: From art to science. In *Proceedings of the 39th ACM Symposium on Theory of Computing, STOC*, pages 401–410, 2007. doi:10.1145/1250790.1250850.
- 5 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms by Fibonacci gates and holographic reductions for hardness. In *Proc. of the 49th Annual Symposium on Foundations of Computer Science, FOCS*, pages 644–653, 2008. doi:10.1109/FOCS.2008.34.
- 6 Jin-yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms with matchgates capture precisely tractable planar  $\#\text{CSP}$ . In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 427–436, 2010. doi:10.1109/FOCS.2010.48.
- 7 Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Computational complexity of holant problems. *SIAM Journal on Computing*, 40(4):1101–1132, 2011. doi:10.1137/100814585.
- 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. *Information and Computation*, 201(2):216–231, 2005. doi:10.1016/j.ic.2005.05.001.
- 9 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, 2008. doi:10.1007/978-3-540-70575-8\_48.
- 10 Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of mathematics*, pages 51–229, 2006.
- 11 Radu Curticapean. Counting matchings of size  $k$  is  $\#\text{W}[1]$ -hard. In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming, ICALP*, pages 352–363, 2013. doi:10.1007/978-3-642-39206-1\_30.
- 12 Radu Curticapean. *The simple, little and slow things count: On parameterized counting complexity*. PhD thesis, Saarland University, August 2015.
- 13 Radu Curticapean and Dániel Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *Proceedings of the 55th Annual Symposium on Foundations of Computer Science, FOCS*, pages 130–139. IEEE, 2014. doi:10.1109/FOCS.2014.22.

- 14 Radu Curticapean and Mingji Xia. Parameterizing the permanent: Genus, apices, minors, evaluation mod  $2^k$ . In *Proceedings of the 56th Annual Symposium on Foundations of Computer Science, FOCS*, pages 994–1009, 2015. doi:10.1109/FOCS.2015.65.
- 15 Paul Dagum and Michael Luby. Approximating the permanent of graphs with large factors. *Theoretical Computer Science*, 102(2):283–305, 1992. doi:10.1016/0304-3975(92)90234-7.
- 16 Víctor Dalmau and Peter Jonsson. The complexity of counting homomorphisms seen from the other side. *Theoretical Computer Science*, 329(1-3):315–323, 2004. doi:10.1016/j.tcs.2004.08.008.
- 17 Holger Dell, Thore Husfeldt, Dániel Marx, Nina Taslaman, and Martin Wahlen. Exponential time complexity of the permanent and the Tutte polynomial. *ACM Transactions on Algorithms*, 10(4):21, 2014. doi:10.1145/2635812.
- 18 Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM Journal of Computing*, 33(4):892–922, 2004. doi:10.1137/S0097539703427203.
- 19 Jörg Flum and Martin Grohe. *Parameterized complexity theory*. Springer, 2006.
- 20 Markus Frick. Generalized model-checking over locally tree-decomposable classes. *Theoretical Computer Science*, 37(1):157–191, 2004. doi:10.1007/s00224-003-1111-9.
- 21 Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1, 2007. doi:10.1145/1206035.1206036.
- 22 Martin Grohe, Thomas Schwentick, and Luc Segoufin. When is the evaluation of conjunctive queries tractable? In *Proceedings of the 33rd ACM Symposium on Theory of Computing, STOC*, pages 657–666, 2001. doi:10.1145/380752.380867.
- 23 Venkatesan Guruswami. Maximum cut on line and total graphs. *Discrete Applied Mathematics*, 92(2-3):217–221, 1999. doi:10.1016/S0166-218X(99)00056-6.
- 24 Frank Harary. *Graph theory*, 1969.
- 25 Mark Jerrum. Two-dimensional monomer-dimer systems are computationally intractable. *Journal of Statistical Physics*, 48(1-2):121–134, 1987. doi:10.1007/BF01010403.
- 26 Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM*, 51(4):671–697, 2004. doi:10.1145/1008731.1008738.
- 27 Pieter W. Kasteleyn. *Graph theory and crystal physics*. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.
- 28 Philippe G. H. Lehot. An optimal algorithm to detect a line graph and output its root graph. *Journal of the ACM*, 21(4):569–575, 1974. doi:10.1145/321850.321853.
- 29 Bingkai Lin. The parameterized complexity of  $k$ -biclique. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 605–615, 2015. doi:10.1137/1.9781611973730.41.
- 30 Vadim V. Lozin and Raffaele Mosca. Independent sets in extensions of  $2K_2$ -free graphs. *Discrete Applied Mathematics*, 146(1):74–80, 2005. doi:10.1016/j.dam.2004.07.006.
- 31 Kitty Meeks. The challenges of unbounded treewidth in parameterised subgraph counting problems. *Discrete Applied Mathematics*, 198:170–194, 2016. doi:10.1016/j.dam.2015.06.019.
- 32 Najiba Sbihi. Algorithme de recherche d’un stable de cardinalité maximum dans un graphe sans étoile. *Discrete Mathematics*, 29:53–76, 1980. doi:10.1016/0012-365X(90)90287-R.
- 33 Harold N. V. Temperley and Michael E. Fisher. Dimer problem in statistical mechanics – an exact result. *Philosophical Magazine*, 6(68):1478–6435, 1961.
- 34 Salil P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM Journal of Computing*, 31(2):398–427, 2001. doi:10.1137/S0097539797321602.
- 35 Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

- 36 Leslie G. Valiant. Holographic algorithms. *SIAM Journal of Computing*, 37(5):1565–1594, 2008. doi:10.1137/070682575.
- 37 L. Šoltés. Forbidden induced subgraphs for line graphs. *Discrete Mathematics*, 132(1):391–394, 1994.
- 38 Mingji Xia, Peng Zhang, and Wenbo Zhao. Computational complexity of counting problems on 3-regular planar graphs. *Theoretical Computer Science*, 384(1):111–125, 2007. Theory and Applications of Models of Computation. doi:10.1016/j.tcs.2007.05.023.