

# Counting edge-injective homomorphisms and matchings on restricted graph classes

Radu Curticapean · Holger Dell · Marc Roth

Received: date / Accepted: date

**Abstract** We consider the  $\#W[1]$ -hard problem of counting all matchings with exactly  $k$  edges in a given input graph  $G$ ; we prove that it remains  $\#W[1]$ -hard on graphs  $G$  that are line graphs or bipartite graphs with degree 2 on one side.

In our proofs, we use that  $k$ -matchings in line graphs can be equivalently viewed as edge-injective homomorphisms from the disjoint union of  $k$  length-2 paths 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 complement this result: If the graphs in  $\mathcal{H}$  have unbounded vertex-cover number even after deleting isolated edges, then counting edge-injective homomorphisms with patterns from  $\mathcal{H}$  is  $\#W[1]$ -hard.

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

## 1 Introduction

Since Valiant’s seminal  $\#P$ -hardness result for the permanent [37], various refinements of classical counting complexity have been studied, such as approximate [27], modular [2], and subexponential counting [18], with additional restrictions on the input classes [26, 41]. In this paper, we study counting problems through the lens of *parameterized complexity* [19], where the input comes with a parameter  $k \in \mathbb{N}$ , and we want to understand whether a problem admits a *fixed-parameter tractable* (*FPT*) algorithm, that is, one with running time  $f(k) \cdot \text{poly}(|x|)$  for some function  $f$ . The analogue of  $\#P$  in this setting is the complexity class  $\#W[1]$ , for which counting cliques of size  $k$  is a canonical complete problem.

In parameterized counting complexity, the problem of counting  $k$ -matchings plays an important role, because it captures the complexity inherent to the counting version of the subgraph isomorphism problem. Indeed if  $H$  is a graph with a maximum matching of size  $\nu$ , then we can count in time  $n^{O(\nu)}$  all occurrences of  $H$  as a (not necessarily induced) subgraph of a given  $n$ -vertex graph  $G$ . There is strong evidence that the dependency on  $\nu$  is necessary: For any class  $\mathcal{H}$  of graphs containing arbitrarily large matchings, it

---

Most of this work was done while the authors were visiting the Simons Institute for the Theory of Computing. Radu Curticapean is supported by ERC grants PARAMTIGHT (No. 280152) and SYSTEMATICGRAPH (No. 725978).

R. Curticapean  
Basic Algorithms Research Copenhagen (BARC) and ITU Copenhagen  
Denmark  
E-mail: radu.curticapean@gmail.com

H. Dell  
Cluster of Excellence (MMCI), Saarland University  
Germany  
E-mail: hdell@mmci.uni-saarland.de

M. Roth  
Cluster of Excellence (MMCI) and Graduate School of Computer Science, Saarland University  
Germany  
E-mail: mroth@mmci.uni-saarland.de

is  $\#W[1]$ -complete to count  $H$ -subgraphs [14] even when  $H$  is required to be from  $\mathcal{H}$ . Furthermore, an  $n^{o(\nu/\log \nu)}$  time algorithm for this restricted problem violates the exponential time hypothesis [13].

In this paper, we proceed from the  $\#W[1]$ -hardness result for counting  $k$ -matchings in two ways: First, we strengthen the 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 interpolate between the classical notions of homomorphisms and (subgraph) embeddings. In the second part of the paper, we study the parameterized complexity of counting such edge-injective homomorphisms as a topic in itself. This also relates to “graph motif parameters” [13], a recently introduced framework for pattern counting problems that was adapted from works by Lovász [30].

## 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 [35, 28], and several dichotomies show which  $\#P$ -hard counting versions of constraint satisfaction problems become polynomial-time solvable on planar graphs [6, 1].

Counting (not necessarily perfect) matchings has been studied by many authors [26, 16, 36], culminating in the work of Xia et al. [41] 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 [21], which rules out a parameterized analogue of the hardness result by Xia et al. [41]. On the other hand, counting  $k$ -matchings remains  $\#W[1]$ -complete on bipartite graphs, and this result was essential for a subsequent reduction to the general subgraph counting problem [14]. This reduction was recently superseded by [13].

### 1.1.1 Restricted bipartite graphs of high girth

In [14], the  $\#W[1]$ -completeness of counting  $k$ -matchings in bipartite graphs was first shown for an edge-colorful variant, which was then reduced to the uncolored version via inclusion–exclusion. In the edge-colorful variant, 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.

In this paper, we strengthen the  $\#W[1]$ -hardness result for counting edge-colorful  $k$ -matchings in bipartite graphs  $G$  by showing that the problem remains hard when we restrict one side of  $G$  to have maximum degree two. We may further 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 was known before [14] 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  $\#ETH$  [18]. That is, if such an algorithm existed, we could count satisfying assignments to  $n$ -variable 3-CNF formulas in time  $\exp(o(n))$ . Our result establishes the same consequence in the restricted case.

**Theorem 1.** *For every  $c \in \mathbb{N}$ , the problem of 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, if  $\#ETH$  holds, neither of these problems has an algorithm running in time  $f(k) \cdot n^{o(k/\log k)}$ , for any computable function  $f$ .*

We sketch the proof in §3 by extending the so-called Holant problems [38, 3] 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  $f_v : \{0, 1\}^{I(v)} \rightarrow \mathbb{C}$ , 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  $x$  contributes a weight  $\prod_{v \in V} f_v(x)$ .

In our edge-colored setting, the edges of  $G$  are colored with  $k$  colors and  $\text{Holant}(G)$  ranges only over assignments of Hamming weight  $k$ , picking exactly one edge from each color. We apply the technique of combined signatures [15] in this setting, an approach that is also implicit in [14]. This way, we will reduce 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 [15]. 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 prove that counting  $k$ -matchings in line graphs is  $\#W[1]$ -complete and we establish a lower bound under  $\#ETH$ .

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

Line graphs can be characterized by a finite set of forbidden induced subgraphs [25, 39]. They can be recognized in linear time [29], and several classical NP-complete problems are polynomial-time solvable in line graphs, such as finding a maximum independent set [34], a maximum cut [24], or a maximum clique [31]. In contrast, Theorem 2 shows that counting  $k$ -matchings remains  $\#W[1]$ -hard in line graphs.

To prove Theorem 2, one might try 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 via the framework of edge-colorful Holant problems introduced before.

Instead, we prove Theorem 2 in §4 by means of a delicate interpolation argument that is reminiscent of the first hardness proof for uncolored  $k$ -matchings [11]. We generate a linear system of equations such that one of the unknowns corresponds to a  $\#W[1]$ -hard problem. The right-hand side of the system can be evaluated by means of a gadget construction and an oracle for counting  $k$ -matchings in line graphs. It turns out that the system does not have full rank, yet a careful analysis shows that the  $\#W[1]$ -hard unknown we are interested in can still be uniquely determined in polynomial time.

### 1.1.3 Perfect matchings in line graphs

Complementing the above result, we show that the problem of counting *perfect* matchings is  $\#P$ -hard on line graphs. This holds even for line graphs of bipartite graphs. Such graphs are known to be perfect and play an important role in the proof of the strong perfect graph theorem [9].

**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 in 3-regular line graphs.*

The theorem can be shown by invoking known results for Holant problems [5], but in the present paper, we give a self-contained proof. To this end, we reduce the positive case of Theorem 3 to a known tractable case of counting constraint satisfaction problems [10], which admits a simple polynomial-time algorithm. The negative case of Theorem 3 follows by a relatively straightforward reduction from counting perfect matchings in 3-regular graphs, using specifically tailored gadgets which ensure that the resulting graphs are line graphs.

## 1.2 Counting edge-injective homomorphisms

In our proof of 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 equal to the number of  $k$ -matchings in the line graph  $L(G)$ , up to a simple factor depending only on  $k$ .

Starting from their relevance in our proof of Theorem 2, we observe that edge-injective homomorphisms are an interesting concept on its own, since they constitute an intermediate step 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  $\#EdgInj(\mathcal{H})$  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, 14], counting/deciding colorful subgraphs [14, 32, 23], counting/deciding induced subgraphs [8], counting/deciding (not necessarily edge-injective) homomorphisms [22, 17], and counting locally-injective homomorphisms [33]. 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  $\#W[1]$ -complete otherwise [17]. 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 [22].

Our main outcome is a similar result for counting edge-injective homomorphisms. 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.** *Let  $\mathcal{H}$  be any class of graphs. The problem  $\#\text{EdgInj}(\mathcal{H})$  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  $\#\text{EdgInj}(\mathcal{H})$  is  $\#W[1]$ -complete.*

We prove this theorem in §6. For the algorithm, we use ideas from the framework of graph motif parameters [13] to reduce the problem to subgraph counting. The latter has known  $n^{\text{vc}(H)+O(1)}$  time algorithms [40, 14], where  $\text{vc}(H)$  denotes the vertex-cover number of the pattern graph  $H$ . For the hardness result, 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  $\#\text{EdgInj}(\mathcal{H})$  on hereditary graph classes  $\mathcal{H}$ , but it leaves open the  $\#W[1]$ -hardness of non-hereditary classes  $\mathcal{H}$ . However, for the particular non-hereditary classes of paths and cycles, we obtain a separate hardness result.

**Theorem 5.** *The problem  $\#\text{EdgInj}(\mathcal{H})$  is  $\#W[1]$ -complete if  $\mathcal{H}$  is the class of all paths or the class of all cycles.*

## 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 time  $f(k) \cdot \text{poly}(n)$  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 [20].

The *counting exponential-time hypothesis (#ETH)* claims that there exists a constant  $\epsilon > 0$  for which there is no  $\exp(\epsilon n)$  time algorithm to compute the number of satisfying assignments for an  $n$ -variable 3-CNF formula. An algorithm with running time  $f(k) \cdot n^{o(k)}$  to count  $k$ -cliques in a given graph, for any function  $f$ , would refute the counting exponential-time hypothesis [7].

Graphs in this paper are undirected, loop-free, and simple, unless stated otherwise and we assume graphs to be encoded by their adjacency matrix. Let  $H$  and  $G$  be graphs. A function  $\varphi : V(H) \rightarrow V(G)$  is a *homomorphism* from  $H$  to  $G$  if  $\{\varphi(u), \varphi(v)\} \in E(G)$  holds for all  $\{u, v\} \in E(H)$ . 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 edges  $e, f \in E(H)$  with  $e \neq f$  satisfy  $\varphi(e) \neq \varphi(f)$ , where  $\varphi(\{u, v\}) = \{\varphi(u), \varphi(v)\}$  for any edge.  $\text{EdgInj}(H, G)$  denotes the set of all edge-injective homomorphisms from  $H$  to  $G$ . A homomorphism  $\varphi \in \text{Hom}(H, G)$  is an *embedding* of  $H$  in  $G$  if it is injective (on the vertices of  $H$ ). The set of all embeddings from  $H$  to  $G$  is denoted by  $\text{Emb}(H, G)$ .

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

The *line graph*  $L(G)$  of  $G$  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 there exists a vertex in  $G$  to which both edges  $e$  and  $f$  are incident.

Polynomials in this paper are elements of the polynomial ring  $\mathbb{Q}[x_1, \dots, x_t]$ , where  $t$  is the number of variables. A polynomial is called *univariate* if  $t = 1$  and *bivariate* if  $t = 2$ . A univariate polynomial  $p(x)$  of degree  $d$  can be interpolated if the value of  $p(a)$  is known for at least  $d$  pairwise different values of  $a$ .

A bivariate polynomial  $p(x, y)$  of degree  $d$  can be interpolated if the values of  $p(a_i, b_i)$  for  $i = 1, \dots, s$  with  $s = (d + 1)^2$  are known and provided that the following bivariate Vandermonde matrix is invertible:

$$\begin{pmatrix} 1 & a_1 & b_1 & a_1 b_1 & \dots & a_1^i b_1^j & \dots & a_1^d b_1^d \\ 1 & a_2 & b_2 & a_2 b_2 & \dots & a_2^i b_2^j & \dots & a_2^d b_2^d \\ \vdots & & & & & & & \vdots \\ 1 & a_s & b_s & a_s b_s & \dots & a_s^i b_s^j & \dots & a_s^d b_s^d \end{pmatrix}$$

### 3 Matchings in restricted bipartite graphs

In this section, we prove Theorem 1. We use  $k$ -edge-colored graphs for  $k \in \mathbb{N}$ , which are graphs  $G$  with a (not necessarily proper) edge-coloring  $c : E(G) \rightarrow \{1, \dots, k\}$ . 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 which is parameterized by  $k$ ; this problem is known to be  $\#\text{W}[1]$ -hard.

**Theorem 6 ([14], Theorem 1.2).** *The problem  $\#\text{ColMatch}$  is  $\#\text{W}[1]$ -complete. If  $\#\text{ETH}$  holds, 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 yields a reduction from counting edge-colorful matchings to counting uncolored matchings (see, e.g., [12, Lemma 1.34] or [14, Lemma 2.7]).

**Lemma 7.** *There is an fpt Turing reduction from  $\#\text{ColMatch}$  for  $k$ -edge-colored graphs  $G$  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 *colorful Holant problems*. In the uncolored setting, the notion of a “Holant” was introduced by Valiant [38] and later developed to a general theory of Holant problems by Cai, Lu, Xia, and other authors [3, 4]. In §3.2, we 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  $G$  and a vertex  $v \in V(G)$ , we denote by  $I(v)$  the set of edges incident to  $v$ . For  $k \in \mathbb{N}$ , a  $k$ -edge-colored signature graph is a  $k$ -edge-colored graph  $\Omega$  that has a *signature*  $f_v : \{0, 1\}^{I(v)} \rightarrow \mathbb{Q}$  at each vertex  $v \in V(\Omega)$ . The graph underlying  $\Omega$  may feature parallel edges. We write  $E_i$  for the set of edges with color  $i \in \{1, \dots, k\}$ .

The task of Holant problems is to count, on input a signature graph  $\Omega$ , the Boolean-valued assignments to  $E(\Omega)$  that satisfy all local constraints given by the signatures. In our colorful setting, only colorful assignments will be counted.

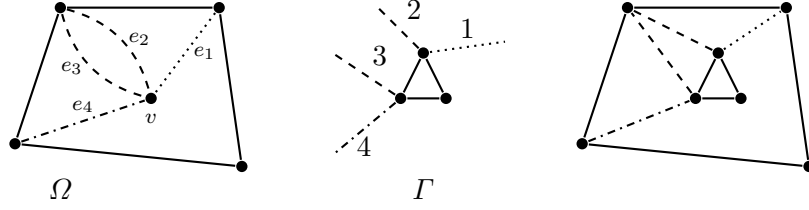
**Definition 9.** An assignment  $x \in \{0, 1\}^{E(\Omega)}$  is *colorful* if, for each  $i \in \{1, \dots, k\}$ , there is exactly one edge  $e \in E_i$  with  $x(e) = 1$ . Given a set  $S \subseteq E(\Omega)$ , we write  $x|_S$  for the restriction of  $x$  to  $S$ , which is the unique assignment in  $\{0, 1\}^S$  that agrees with  $x$  on  $S$ . We define  $\text{ColHolt}(\Omega)$  as the sum

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

Next, we express the number of edge-colorful matchings in a graph as a colorful Holant problem. If all signatures in  $\Omega$  map to  $\{0, 1\}$ , then  $\text{ColHolt}(\Omega)$  simply counts the edge-colorful assignments  $x$  that satisfy  $f_v(x|_{I(v)}) = 1$  for all  $v \in V(\Omega)$ . For assignments  $x \in \{0, 1\}^*$ , write  $\text{hw}(x)$  for the Hamming weight of  $x$ . For a proposition  $\varphi$ , let  $[\varphi]$  be defined to be 1 if  $\varphi$  holds and 0 otherwise.

**Fact 10.** Let  $k \in \mathbb{N}$  and  $G$  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

$$\begin{aligned} \text{hw}_{\leq 1} : \{0, 1\}^{I(v)} &\rightarrow \{0, 1\} \\ x &\mapsto [\text{hw}(x) \leq 1]. \end{aligned}$$



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

Then we have

$$\text{ColHolant}(\Omega) = \#\text{ColMatch}(G).$$

This fact can also be used in reverse: For any signature graph  $\Omega$  that has  $\text{hw}_{\leq 1}$  associated with every vertex, we can obtain a graph  $G$  such that  $\text{ColHolant}(\Omega) = \#\text{ColMatch}(G)$  by deleting the signatures from  $\Omega$ .

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 simpler signatures associated with its vertices, such as  $\text{hw}_{\leq 1}$ . Replacing all signatures by such graph fragments, we obtain a signature graph  $\Omega'$  featuring only  $\text{hw}_{\leq 1}$ . Then, by Fact 10, the quantity  $\text{ColHolant}(\Omega')$  can be expressed as a number of edge-colorful matchings. This will allow us to reduce the computation of  $\text{ColHolant}(\Omega)$  to an instance of  $\#\text{ColMatch}$ . The graph fragments required for this reduction are formally defined as *edge-colored matchgates*:

**Definition 11.** 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 in  $E(\Gamma) \setminus D$  are called *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 via

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

Let  $\Omega$  be a  $k$ -edge-colored signature graph and let  $\Gamma$  be an edge-colored matchgate with  $d \in \mathbb{N}$  dangling edges such that the internal colors of  $\Gamma$  are disjoint from the colors  $\{1, \dots, k\}$  present in  $\Omega$ . Then we can *insert*  $\Gamma$  at a vertex  $v \in V(\Omega)$  of degree  $d$  as follows, see Figure 1: First delete  $v$  from  $\Omega$ , but keep  $I(v)$  as dangling edges in  $\Omega$ , and choose an ordering  $e_1, \dots, e_d$  of  $I(v)$ . Then insert a disjoint copy of  $\Gamma$  into  $\Omega$ , and for all  $i \in [d]$ , identify the  $i$ -th dangling edge of  $\Gamma$  with  $e_i$ . That is, if  $e_i$  has endpoint  $u_i$  in  $\Omega$  and the  $i$ -th dangling edge of  $\Gamma$  has endpoint  $v_i$  in  $\Gamma$ , then form the edge  $u_i v_i$  in the resulting graph.<sup>1</sup>

**Remark 12.** 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  $\Omega$ .

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

**Fact 13.** Let  $\Omega$  be a  $k$ -edge-colored signature graph, and for each  $v \in V(\Omega)$ , let  $f_v$  be the signature associated with  $v$ . 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  $\text{ColHolant}(\Omega) = \#\text{ColMatch}(G)$ .

If the involved signatures cannot be realized by matchgates, then Fact 13 is not applicable. For such cases, Curticapean and Xia [15] define *combined signatures*: Even if a given signature  $f$  cannot be realized 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. In the following, we write  $[N] = \{1, \dots, N\}$ .

<sup>1</sup> We need to assume here that the colors of the two identified edges agree.





**Fig. 2** Edges of color  $i$  are shown on the left, and their corresponding edges in the graph  $\Omega_{\text{bip}}$  together with their annotations are shown on the right.

**Lemma 14.** *Let  $\Omega$  be a  $k$ -colored signature graph. Let  $s, t \in \mathbb{N}$  and let  $w_1, \dots, w_s$  be fixed 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 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*

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

*Proof.* In the following claim, we first prove the lemma for  $s = 1$ . That is, the signature of exactly one vertex  $w_1$  is expressed as a linear combination of signatures.

*Claim 15.* Let  $\Omega$  be a signature graph and let  $w \in V(\Omega)$  be a fixed vertex with signature  $f_w$ . Let  $g_1, \dots, g_t$  be signatures and  $c_1, \dots, c_t \in \mathbb{Q}$  be such that  $f_w = \sum_{i=1}^t c_i \cdot g_i$  holds point-wise. For  $i \in [t]$ , let  $\Omega_i$  be the signature graph obtained from  $\Omega$  by replacing  $f_w$  with  $g_i$ . Then  $\text{ColHolant}(\Omega) = \sum_{i=1}^t c_i \cdot \text{ColHolant}(\Omega_i)$ .

*Proof.* In the following, let  $x$  range over edge-colorful assignments in  $\{0, 1\}^{E(\Omega)}$ . By elementary manipulations, we have

$$\begin{aligned} \text{ColHolant}(\Omega) &= \sum_x f_w(x) \prod_{v \in V(\Omega) \setminus \{w\}} f_v(x) \\ &= \sum_x \left( \sum_{i=1}^t c_i \cdot g_i(x) \right) \prod_{v \in V(\Omega) \setminus \{w\}} f_v(x) \\ &= \sum_{i=1}^t \sum_x c_i \cdot g_i(x) \prod_{v \in V(\Omega) \setminus \{w\}} f_v(x) \\ &= \sum_{i=1}^t c_i \cdot \text{ColHolant}(\Omega_i). \end{aligned} \quad \square$$

The lemma follows by applying Claim 15 inductively for  $w_1, \dots, w_s$ . Each of the  $s$  involved steps reduces the number of combined signatures by one, and elementary algebraic manipulations imply (1). ■

Lemma 14 allows us to prove hardness results under fpt Turing reductions if  $\text{ColHolant}(\Omega)$  is  $\#W[1]$ -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.

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

We prove Theorem 1 by a reduction from  $\#\text{ColMatch}$ , which is  $\#W[1]$ -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 the number  $\#\text{ColMatch}(G)$ . To this end, we first construct a bipartite signature graph  $\Omega_{\text{bip}}$  such that  $\text{ColHolant}(\Omega_{\text{bip}}) = \#\text{ColMatch}(G)$  holds.

**Lemma 16.** *Given a  $k$ -edge-colored graph  $G$ , let  $\Omega_{\text{bip}} = \Omega_{\text{bip}}(G)$  denote the signature graph with edge-colors  $[k] \times [2]$  that is constructed as follows: Initially,  $\Omega_{\text{bip}}$  is  $G$ , where each vertex is associated with the signature  $\text{hw}_{\leq 1}$ . Then, for each  $i \in [k]$ :*

1. Add a fresh vertex  $w_i$  to  $\Omega_{\text{bip}}$ .

2. For each  $e \in E(G)$  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  $\text{ColHolant}(\Omega_{\text{bip}}) = \#\text{ColMatch}(G)$ .

*Proof.* It is clear that  $\Omega_{\text{bip}}$  is bipartite, since every edge of  $\Omega_{\text{bip}}$  has exactly one of the vertices  $w_i$  for  $i \in [k]$  as an endpoint.

Let us call an assignment  $x \in \{0, 1\}^{E(\Omega_{\text{bip}})}$  satisfying for  $\Omega_{\text{bip}}$  if none of the signatures in  $\Omega_{\text{bip}}$  vanishes on  $x$ . The edge-colorful satisfying assignments  $x$  correspond bijectively to the edge-colorful matchings of  $G$ : In any such  $x$ , the vertex  $w_i$  for  $i \in [k]$  is incident with two edges  $e_i$  and  $e'_i$  that are assigned 1 under  $x$  and have the same annotation  $h_i = \pi(e_i) = \pi(e'_i)$ , for some  $h_i \in E(G)$ . We can hence contract  $e_i$  and  $e'_i$  to one edge  $h_i$ . The resulting edge set is an edge-colorful matching in  $G$  due to the signature  $\text{hw}_{\leq 1}$  at non-subdivision vertices. Reversing this contraction operation, every edge-colorful matching in  $G$  can be extended to a unique satisfying assignment  $x \in \{0, 1\}^{E(\Omega_{\text{bip}})}$ , and all signatures evaluate to 1 on this assignment. ■

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  $i$ -colored edges in  $G$ . Let  $m_i = |E_i(G)|$  and order the edges in  $E_i(G)$  in some arbitrary fixed way.

**Definition 17.** Recall the definition of  $\Omega_{\text{bip}}$  from Lemma 16. 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:

1. Create independent sets  $a_1, \dots, a_m$  and  $b_1, \dots, b_m$ , which we call “external” vertices.
2. 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  $a_j c_j$  of color  $(i, 3)$  and an edge  $c_j b_j$  of color  $(i, 4)$ .

**Lemma 18.** We can express the signature  $f_i$  from Lemma 16 as the linear combination

$$f_i = (m^2 - 3m + 3) \cdot \text{ColSig}(\Gamma_{i,1}) - \text{ColSig}(\Gamma_{i,2}).$$

*Proof.* Observe first that, for all edge-colorful  $x \in \{0, 1\}^{I(w_i)}$ , we have

$$\text{ColSig}_{\text{col}}(\Gamma_{i,1}, x) = 1.$$

This is because  $x$  trivially is the only satisfying assignment that extends  $x$ , since there are no edges other than  $I(w_i)$  in  $\Gamma_{i,1}$ .

Concerning  $\Gamma_{i,2}$ , let  $x \in \{0, 1\}^{I(w_i)}$  be a colorful assignment and let  $e_1, e_2$  be the edges that are assigned 1 under  $x$ . We show

$$\text{ColSig}_{\text{col}}(\Gamma_{i,2}, x) = \begin{cases} m^2 - 3m + 2 & \text{if } \pi(e_1) = \pi(e_2), \\ m^2 - 3m + 3 & \text{otherwise,} \end{cases} \quad (2)$$

which implies the claim of the lemma. In the following, we calculate the two cases in (2) separately. To this end, let us say that a path in  $\Gamma_{i,2}$  is *hit* by  $x$  if at least one of the edges assigned 1 in  $x$  is incident with an endpoint of the path.

- If  $\pi(e_1) = \pi(e_2)$ , then there are  $m - 1$  paths in  $\Gamma_{i,2}$  not hit by  $x$ , each on the same two colors. This gives  $(m - 1)(m - 2) = m^2 - 3m + 2$  matchings.
- Otherwise, there are  $m - 2$  paths not hit by  $x$ . Each matching may contain
  - 2 edges from the intact paths, yielding  $(m - 2)(m - 3) = m^2 - 5m + 6$  matchings, or
  - 1 such edge, yielding  $2(m - 2)$  matchings, or
  - 0 such edges, yielding 1 matching.

By summing over the disjoint possible choices, we obtain (2). ■



Using Lemmas 14, 16 and 18, we can now reduce counting edge-colorful matchings in graphs  $G$  to the same problem in subdivisions of  $G$ . For a  $k$ -colored graph  $G$  and  $t \in \mathbb{N}$ , a  $t$ -subdivision of  $G$  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 19.** *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 of the form  $\#\text{ColMatch}(G')$  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.*

*Proof.* Recall that  $\Omega_{\text{bip}} = \Omega_{\text{bip}}(G)$  is bipartite, with the vertices  $\{w_i\}_{i \in [k]}$  on one side. If we insert  $\Gamma_{i,2}$  at  $w_i$ , for every  $i \in [k]$ , the resulting signature graph  $\Omega^*$  is a 3-subdivision of  $G$ . It has at most  $4(n + m)$  vertices and edges and at most  $4k$  colors. For  $\theta \in [2]^k$ , consider the graph  $\Omega_\theta$  obtained from  $\Omega_{\text{bip}}$  by inserting  $\Gamma_{i,\theta(i)}$  at  $w_i$  for every  $i \in [k]$ . We observe that  $\Omega_\theta$  is a subgraph of  $\Omega^*$ .

Invoking Lemmas 14 and 18, we can hence write  $\text{ColHolant}(\Omega_{\text{bip}})$  as a linear combination of the  $2^k$  quantities  $\text{ColHolant}(\Omega_\theta)$  where each  $\Omega_\theta$  features only the signature  $\text{hw}_{\leq 1}$ . By Fact 10, the value  $\text{ColHolant}(\Omega_\theta)$  can hence be computed as  $\#\text{ColMatch}(G')$  for the edge-colored graph  $G'$  obtained by removing all signatures from  $\Omega_\theta$ . As observed above, this is a subgraph of a 3-subdivision of  $G$ . ■

Theorem 1 now follows easily from the hardness of  $\#\text{ColMatch}$  and repeated applications of Lemma 19:

*Proof of Theorem 1.* Let  $c \in \mathbb{N}$  be an arbitrary constant and let  $G$  be a  $k$ -edge-colored graph for which we wish to compute  $\#\text{ColMatch}(G)$ . We first prove Theorem 1 for the edge-colorful case. To this end, we reduce the computation of  $\#\text{ColMatch}(G)$  to instances  $\#\text{ColMatch}(F)$ , where  $F$  is obtained from  $G$  by repeating  $c$  times the operation of taking a subgraph of the 3-subdivision.

Note that this indeed proves the edge-colorful part of Theorem 1: If  $F$  is obtained as above, then  $F$  is bipartite, with the maximum degree of one side bounded by 2: Put the first and third vertex (if present) of all subdivided edges on one side. Concerning the girth, each cycle  $C$  in  $G$  will appear in  $F$  either (i) as a subdivision of  $C$ , or (ii) not at all, because vertices of  $C$  or its subdivisions were deleted in the process. No other cycles can be created by the operation. In other words, every cycle that is contained in the graph obtained from  $G$  by applying Lemma 19  $c$  times is a  $3c$ -subdivision of a cycle in  $G$ .

A single application of Lemma 19 on a  $[t]$ -edge-colored graph  $F$  creates  $2^t$  new instances for  $\#\text{ColMatch}$  that are  $[4t]$ -edge-colored subgraphs of 3-subdivisions of  $F$ . Iteratively applying Lemma 19 on  $G$  for a total of  $c$  times yields  $2^{O(4^c k)}$  instances  $\#\text{ColMatch}(F)$  where each  $F$  is a  $[4^c k]$ -edge-colored graph on  $O(4^c(n + m))$  vertices and edges that is obtained by repeatedly taking subgraphs of subdivisions. In particular, since  $c$  is constant, we reduce counting  $k$ -matchings in  $G$  to counting  $O(k)$ -matchings in the graphs  $F$ , hence proving the claimed lower bound under  $\#\text{ETH}$ .

Finally, we show  $\#\text{W}[1]$ -completeness of the uncolored variant of the problem. To this end, we use Lemma 7 to reduce from the edge-colorful variant: To compute  $\#\text{ColMatch}(F)$  for a  $[t]$ -colored graph  $G$ , we only need to count  $t$ -matchings in uncolored subgraphs of  $F$ . This preserves the properties required on  $F$  and the claimed lower bound under  $\#\text{ETH}$ . ■

#### 4 Matchings in line graphs

We now sketch the proof of Theorem 2, which asserts that counting  $k$ -matchings in line graphs is  $\#\text{W}[1]$ -hard. In our proof, we will use an equivalent characterization of this problem: 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 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 counting edge-injective homomorphisms from wedge packings. The following lemma encapsulates the interpolation argument used in the reduction. For  $t \in \mathbb{N}$ , let  $(x)_t$  denote the falling factorial, where

$$(x)_t = (x) \cdot (x - 1) \cdots (x - t + 1).$$

**Lemma 20.** *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  $r \leq O(k)$ ,<sup>2</sup> computes the numbers  $a_{t,k-t}$  for all  $t \in \{0, \dots, k\}$ .

*Proof.* Let  $t \in \{0, \dots, k\}$ . For all  $k, i \in \mathbb{N}$ , let  $I_{k,i}$  be defined as

$$I_{k,i} = \sum_{t=0}^k a_{t,k-t} \cdot t^i.$$

As an intermediate step, we construct a polynomial-time algorithm that allows us, given the coefficients of  $P_r(y)$  and a number  $m \in \mathbb{N}$ , to compute  $I_{k,i}$  for all  $k, i \in \mathbb{N}$  with  $2k + i \leq m$ .

If  $m = 0$ , then  $I_{0,0}$  is the only number we need to compute. We obtain it by observing that  $P_0(y) = I_{0,0} = a_{0,0}$ . Now suppose that  $m > 0$  and that we inductively already computed the values  $I_{k,i}$  for all  $k, i \in \mathbb{N}$  with  $2k + i \leq m$ . We will compute the values  $I_{k,i}$  with  $2k + i = m + 1$ .

Let  $r$  be an integer that satisfies  $2r - (m + 1) \geq 0$ . Furthermore, let  $C^r$  denote the coefficient of  $y^{2r-(m+1)}$  in  $P_r(y)$ , which is given as input. We want to describe  $C^r$  as an expression in terms of the unknowns  $a_{g,b}$ . To this end, we investigate which of the summands  $a_{t,k-t} \cdot \binom{r}{k} \cdot (y-t)_{2(r-k)}$  contribute to  $C^r$ .

*Claim 21.* If  $k > \lfloor \frac{m+1}{2} \rfloor$  then  $2(r-k) < 2r - (m+1)$ .

*Proof.* We have  $2(r-k) = 2r - 2k < 2r - 2\lfloor \frac{m+1}{2} \rfloor$ . If  $m+1$  is even, we get  $2(r-k) < 2r - (m+1)$  as claimed. Otherwise  $m+1$  is odd, and we only get  $2(r-k) < 2r - m$ . However, since  $m$  and  $2(r-k)$  are both even, we actually get  $2(r-k) < 2r - (m+1)$  as claimed.  $\square$

It follows that the summands with  $k > \lfloor \frac{m+1}{2} \rfloor$  do not contribute to  $C^r$ .

Let us view  $(y-t)_{2(r-k)}$  as a bivariate polynomial in  $y$  and  $t$  for a moment. Then, by expanding this polynomial in powers of  $y$ , there exist univariate polynomials  $\sigma_i(t)$  for all  $i \in \mathbb{N}$  with  $i \leq 2(r-k)$  such that

$$(y-t)_{2(r-k)} = \sum_{i=0}^{2(r-k)} \sigma_i(t) \cdot y^{2(r-k)-i}.$$

Using bivariate interpolation, we can easily compute all coefficients of  $\sigma_i(t)$  for all  $i \leq 2(r-k)$ . Note that the coefficient of  $t^{m+1-2k}$  in  $\sigma_{m+1-2k}$  is  $(-1)^{m+1} \cdot \binom{2(r-k)}{m+1-2k}$ . Let  $c_0, \dots, c_{m+1-2k-1}$  be the remaining coefficients.

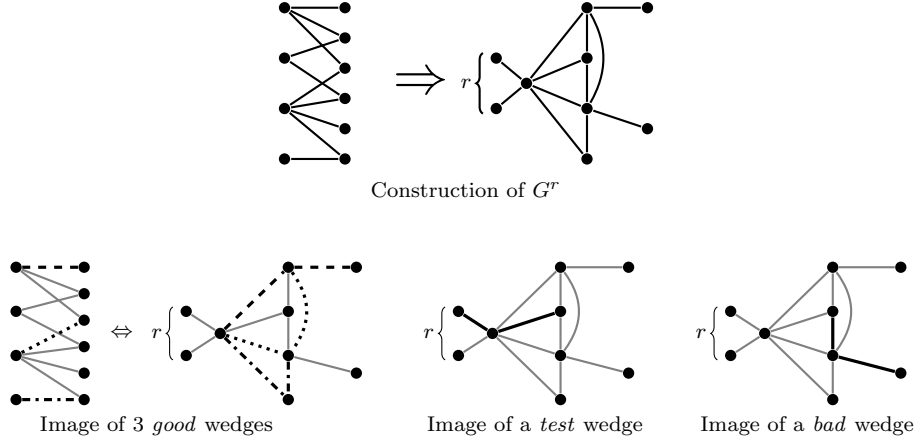
Since only terms with  $k \leq \lfloor (m+1)/2 \rfloor$  contribute to  $C^r$ , we obtain the following.

$$\begin{aligned} C^r &= \sum_{k=0}^{\lfloor \frac{m+1}{2} \rfloor} \sum_{t=0}^k a_{t,k-t} \cdot \binom{r}{k} \cdot \sigma_{m+1-2k}(t) \\ &= \sum_{k=0}^{\lfloor \frac{m+1}{2} \rfloor} \sum_{t=0}^k a_{t,k-t} \cdot \binom{r}{k} \cdot \left[ (-1)^{m+1} \binom{2(r-k)}{m+1-2k} t^{m+1-2k} + \sum_{j=0}^{m+1-2k-1} c_j t^j \right] \\ &= (-1)^{m+1} \cdot \sum_{k=0}^{\lfloor \frac{m+1}{2} \rfloor} \left[ \sum_{t=0}^k a_{t,k-t} \cdot \binom{r}{k} \cdot \binom{2(r-k)}{m+1-2k} \cdot t^{m+1-2k} + \sum_{t=0}^k a_{t,k-t} \cdot \binom{r}{k} \cdot \sum_{j=0}^{m-2k} c_j \cdot t^j \right] \\ &= (-1)^{m+1} \cdot \left[ \sum_{k=0}^{\lfloor \frac{m+1}{2} \rfloor} \binom{2(r-k)}{m+1-2k} \binom{r}{k} \cdot \sum_{t=0}^k a_{t,k-t} \cdot t^{m+1-2k} \right] + \left[ \sum_{k=0}^{\lfloor \frac{m+1}{2} \rfloor} \sum_{j=0}^{m-2k} c_j \binom{r}{k} \sum_{t=0}^k a_{t,k-t} \cdot t^j \right] \\ &= (-1)^{m+1} \cdot \left[ \sum_{k=0}^{\lfloor \frac{m+1}{2} \rfloor} \binom{2(r-k)}{m+1-2k} \cdot \binom{r}{k} \cdot I_{k,m+1-2k} \right] + \left[ \sum_{k=0}^{\lfloor \frac{m+1}{2} \rfloor} \sum_{j=0}^{m-2k} c_j \cdot \binom{r}{k} \cdot I_{k,j} \right]. \end{aligned}$$

Now consider the  $I_{k,j}$  from the last sum. Since  $2k + j \leq 2k + m - 2k = m$ , we have already computed these  $I_{k,j}$  recursively. We also know all of the  $c_j$ , so we can compute the number  $C'^r$  for any  $r \geq \frac{m+1}{2}$ , where

$$C'^r = \sum_{k=0}^{\lfloor \frac{m+1}{2} \rfloor} \binom{2(r-k)}{m+1-2k} \cdot \binom{r}{k} \cdot I_{k,m+1-2k}.$$

<sup>2</sup> That is,  $r$  is bounded by  $dk$  for some overall constant  $d$ .



**Fig. 3** Example of the construction of  $G^r$  as used in the proof of Theorem 22. 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.

Finally, consider the matrix  $A$  corresponding to the above system of equations such that

$$A_{j,i} = \binom{2(r_j - i)}{m + 1 - 2i} \cdot \binom{r_j}{i}$$

for  $i = 0, \dots, \lfloor \frac{m+1}{2} \rfloor$  and pairwise distinct and  $r_0, \dots, r_{\lfloor \frac{m+1}{2} \rfloor}$  large enough such that the binomial coefficient does not become zero. Column  $i$  is an evaluation vector of the polynomial  $Q_i(r) = \binom{2(r-i)}{m+1-2i} \cdot \binom{r}{i}$ . Each  $Q_i$  has degree  $m + 1 - 2i + i = m + 1 - i$ ; in particular, the degree of  $Q_i$  is different for different  $i$ . This implies that the set  $\{Q_i : i \in \mathbb{N}\}$  is a set of linearly independent polynomials, and thus the column vectors of  $A$  are linearly independent and  $A$  is invertible. This allows us to compute the unique solution for the  $I_{k,m+1-2k}$  for all  $k \in \mathbb{N}$  with  $k \leq m/2$ .

Finally, we argue how to compute the  $a_{t,k-t}$  from the  $I_{k,i}$ . By definition, we have the following set of linear equations:

$$\begin{aligned} \sum_{t=0}^k t^0 \cdot a_{t,k-t} &= I_{k,0} \\ \sum_{t=0}^k t^1 \cdot a_{t,k-t} &= I_{k,1} \\ &\dots \\ \sum_{t=0}^k t^k \cdot a_{t,k-t} &= I_{k,k} \end{aligned}$$

The corresponding matrix  $B$  where  $(B)_{i,j} = j^i$  for  $i, j = 0, \dots, k$  is a Vandermonde matrix and thus invertible. Therefore we can compute the unique solution for the  $a_{t,k-t}$ . ■

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

**Theorem 22.** *If  $\mathcal{H}$  is the class of all wedge packings, then the problem  $\# \text{EdgInj}(\mathcal{H})$  is  $\# \text{W}[1]$ -hard. If  $\# \text{ETH}$  holds, it 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 3):

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  $\varphi \in \text{EdgInj}(H, G^0)$ , 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  $\varphi \in \text{EdgInj}(H, G^0)$  for which there are  $g$  test wedges,  $g$  good wedges, and  $b$  bad wedges.

*Claim 23.* The number of  $k$ -matchings in  $G$  is equal to  $\alpha_{k,0}/(2^k \cdot k!)$ .

*Proof.* The integer  $\alpha_{k,0}$  is the number of all  $\varphi \in \text{EdgInj}(H, G^0)$  such that the image of every  $H_i$  consists of a wedge that uses exactly one edge incident to 0.

Given any such  $\varphi$ , we construct a  $k$ -matching  $M_\varphi$  of  $G$  as follows. For each  $i$ , consider the wedge  $\varphi(H_i)$ : It uses an edge  $\{0, v\}$  for  $v \in L(G)$  and an edge  $\{v, w\}$  with  $w \neq 0$ . If  $w \in R(G)$ , then  $N_G(w) = \{v\}$ , and we add the edge  $e_i$  with  $e_i = \{v, w\} \in E(G)$  to the matching. Otherwise, we have  $w \in L(G)$ , and so the edge  $\{v, w\} \in E(G^0)$  corresponds to a vertex  $u \in R(G)$  with  $N_G(u) = \{v, w\}$ , from which it was constructed. In this case, we add the edge  $e_i$  with  $e_i = \{v, u\} \in E(G)$  to the matching. Note that  $e_i$  and  $e_j$  for  $i$  and  $j$  with  $i \neq j$  are disjoint; for if they shared a vertex  $v \in L(G)$ , the edge  $\{0, v\}$  would be used by both  $\varphi(H_i)$  and  $\varphi(H_j)$ , and if they shared a vertex  $v \in R(G)$ , then either  $N_G(v)$  or  $N_G(v) \cup \{v\}$  would be an edge in  $G^0$ , which would be used by both  $\varphi(H_i)$  and  $\varphi(H_j)$ . Thus the constructed set  $M_\varphi$  is indeed a  $k$ -matching.

On the other hand, for each  $k$ -matching  $M$ , we have that there are exactly  $2^k \cdot k!$  edge-injective homomorphisms  $\varphi \in \text{EdgInj}(H, G^0)$  with  $M = M_\varphi$  since the automorphism group of  $H$  has this size.  $\square$

We aim at determining the number  $\alpha_{k,0}$  by using an oracle for  $\#\text{EdgInj}(\mathcal{H})$ . 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 20. To apply the lemma, we observe the following identity.

*Claim 24.* Let  $k, r \in \mathbb{N}$ . Then  $\beta_k(G^r) := \#\text{EdgInj}(H, G^r)$  satisfies

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

*Proof.* We construct an element  $\varphi$  of  $\text{EdgInj}(H_1 \dot{\cup} \dots \dot{\cup} H_k, G^r)$  whose image consists of  $g$  good wedges,  $b$  bad wedges, and  $t$  test wedges, where  $g+b+t=k$ . There are  $\binom{k}{g+b}$  possibilities to select the set of  $H_i$  that will be mapped to a good or a bad wedge. Once this selection has been done, there are  $\alpha_{g,b}$  edge-injective homomorphisms that map the selected  $H_i$  to  $g$  good and  $b$  bad wedges; to see this, note that  $G^0$  and  $G^r$  have exactly the same good and bad wedges. Finally, the test wedges can only be mapped to the edges incident to 0, for which reason only the star with center 0 is relevant for the test wedges. Each good wedge that has already been placed blocks one edge of the star. Hence the  $t$  wedges map into a star  $S_{n+r-g}$  with  $n+r-g$  leaves. The number of edge-injective homomorphisms that map  $t$  wedges into a star with  $\ell$  leaves is  $(\ell)_{2t}$ .  $\square$

Note that  $\beta_k(G^r)$  is a polynomial in  $r$  of degree at most  $2k$ . Setting  $y = n+r$ , Claim 24 yields a polynomial identity that is exactly of the form required by Lemma 20, 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  $\#\text{ETH}$ .  $\blacksquare$

## 5 Perfect matchings in line graphs of bipartite graphs

In this section, we prove Theorem 3, which asserts that it is  $\#\text{P}$ -hard to count perfect matchings on line graphs of maximum degree 4, whereas this is polynomial-time solvable on 3-regular line graphs. We use a characterization of 3-regular line graphs that was established in [42].

**Fact 25.** Every 3-regular line graph  $G$  with  $|V(G)| \geq 5$  is the union of two edge-disjoint graphs  $M$  and  $T$  on the same vertex set  $V(G)$ , where  $M$  is a perfect matching and  $T$  is a *perfect triangle packing*. That is,  $T$  is a vertex-disjoint union of triangles that covers all vertices of  $G$ . Since all triangles of  $G$  are contained in  $T$ , the decomposition into  $M$  and  $T$  is unique.

In the following, fix a graph  $G$  with a partition into  $M$  and  $T$  as above, and let  $G_\downarrow$  be the graph obtained by contracting each triangle in  $T$  to a single vertex without a self-loop. Then  $G_\downarrow$  is 3-regular and it turns out that the perfect matchings of  $G$  correspond bijectively to the odd edge-sets of  $G_\downarrow$ . For the purposes of this section, we say that an edge-set  $S \subseteq E(G_\downarrow)$  is *odd* if, for all  $v \in V(G_\downarrow)$ , the degree of  $v$  in the subgraph  $(V(G_\downarrow), S)$  is odd.

**Fact 26.** For all  $t \in \{0, \dots, |E(G_\downarrow)|\}$ , the odd edge-sets  $S \subseteq E(G_\downarrow)$  of cardinality  $t$  correspond bijectively to the perfect matchings of  $G$  that contain exactly  $t$  edges from the matching  $M$ .

*Proof.* By construction of  $G_\downarrow$ , every set  $S \subseteq E(G_\downarrow)$  corresponds to a set  $S' \subseteq E(M)$  of the same size. Clearly  $S$  is odd if and only if  $S'$  is incident to exactly one or three vertices in each triangle of  $G$ . In turn, the latter holds if and only if  $S'$  can be extended with edges of  $T$  to obtain a perfect matching of  $G$ . Since the extension is unique if it exists, this defines a bijective mapping from odd edge-sets of  $G_\downarrow$  to perfect matchings of  $G$ . ■

Since odd edge-sets can be counted in polynomial time [10], we obtain the algorithmic result of Theorem 3 as follows.

**Lemma 27.** *Given as input a 3-regular line graph  $G$ , the number of perfect matchings in  $G$  can be computed in polynomial time.*

*Proof.* If  $|V(G)| < 5$ , we apply brute-force. Otherwise we decompose  $G$  into a perfect matching  $M$  and a perfect triangle packing  $T$  as in Fact 25. We obtain this decomposition in polynomial time by greedily removing triangles from  $G$ ; as all triangles of  $G$  are contained in the vertex-disjoint collection of triangles  $T$ , this procedure does indeed recover  $T$ .

As a consequence of Fact 26, the number of perfect matchings in  $G$  equals the number of odd edge-sets in  $G_\downarrow$ . By the algorithmic part of Theorem 4.4 in [10], counting odd edge-sets admits a polynomial-time algorithm. ■

For the hardness result, we reduce from counting perfect matchings in 3-regular graphs, which is #P-hard [16]. In the remainder, let  $G$  be a 3-regular graph. Let  $G'$  be the graph obtained from  $G$  by replacing every vertex  $v \in V(G)$  by a triangle  $T_v$  and attaching the  $i$ -th edge incident with  $v$ , for  $i \in [3]$ , to the  $i$ -th vertex of  $T_v$ .

By construction, the graph  $G'$  admits a decomposition into a matching  $M$  and a triangle packing  $T$  as in Fact 25, and we have  $G = G'_\downarrow$ . Since the perfect matchings of any graph  $A$  are precisely its odd edge-sets of cardinality  $|V(A)|/2$ , we obtain the following corollary of Fact 26:

**Fact 28.** The perfect matchings of  $G$  correspond bijectively to the perfect matchings of  $G'$  that contain exactly  $|V(G)|/2$  edges from the matching  $M$ .

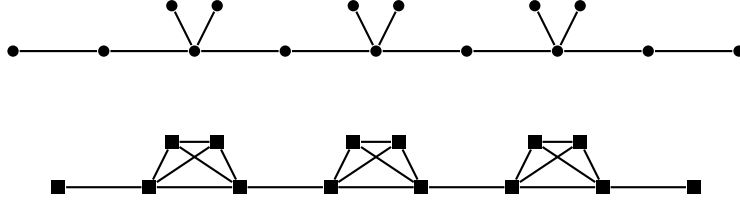
In order to establish the reduction from counting perfect matchings in  $G$ , it remains to count those perfect matchings of  $G'$  that contain the desired number of edges from  $M$ . To this end, we replace each edge of  $M$  by a “collar” gadget from Figure 4 so as to obtain a new graph  $B$ .

**Definition 29.** A *collar* of length  $\ell \in \mathbb{N}$  with end vertices  $u, v$  is the graph obtained as follows: Start with  $\ell$  vertex-disjoint copies  $C_1, \dots, C_\ell$  of  $K_4$ . For each  $i$ , let  $a_i, b_i$  be two arbitrary distinct vertices of  $C_i$ . For each  $i$  with  $1 \leq i < \ell$ , add the edge  $b_i a_{i+1}$ . Add a vertex  $u$  and the edge  $u a_1$ , and add a vertex  $v$  and the edge  $b_\ell v$ .

In the graph  $B$ , each collar gadget intersects the remainder of  $B$  only in its end vertices. Perfect matchings of  $B$  may block zero, one, or both ends of a collar by edges outside of the collar. In each case, there is a simple closed expression for the number of perfect matchings in the remaining part of the collar.

**Fact 30.** A collar  $X_\ell$  of length  $\ell \in \mathbb{N}$  with end vertices  $u, v$  has exactly one perfect matching. (It contains the two edges incident to the ends, the  $\ell - 1$  edges between  $K_4$ -copies and, for each  $i$ , the unique edge in  $C_i$  that is disjoint from  $a_i b_i$ .) The graphs  $X_\ell - u$  and  $X_\ell - v$  have an odd number of vertices and thus no perfect matching. Finally, the graph  $X_\ell - \{u, v\}$  has exactly  $3^\ell$  perfect matchings, as we can independently choose one of three perfect matchings in each  $K_4$ -copy.

We can now prove the hardness result in Theorem 3.



**Fig. 4** A barbed wire of length 3 (*top*) and a collar of length 3 (*bottom*).

**Lemma 31.** *Counting perfect matchings in line graphs of bipartite graphs is #P-hard, even when the input graph has maximum degree 4.*

*Proof.* Let  $G$  be a 3-regular graph for which we wish to determine the number of perfect matchings. Recall that  $G'$  is obtained by inserting triangles at vertices and that  $G'$  decomposes into a perfect matching  $M$  and a perfect triangle packing  $T$ . By Fact 28, the number of perfect matchings of  $G$  is equal to the number of perfect matchings of  $G'$  that contain exactly  $|V(G)|/2$  edges from the matching  $M$ .

Let  $B$  be obtained from  $G'$  by replacing each edge  $uv \in M$  with a fresh collar of length  $|E(G')| + 1$  with ends  $u$  and  $v$ . Write

$$R := 3^{|E(G')|+1}.$$

For  $0 \leq t \leq |E(M)|$ , let  $m_t$  be the number of perfect matchings of  $G'$  that contain exactly  $t$  edges from  $M$ . By Fact 30, the number of perfect matchings in  $B$  satisfies

$$\#\text{PerfMatch}(B) = \sum_{t=0}^{|E(M)|} m_t \cdot R^{|E(M)|-t}. \quad (3)$$

Clearly  $m_t < R$  holds for all  $t \in \mathbb{N}$ . Thus (3) can be viewed as a representation of the integer  $\#\text{PerfMatch}(B)$  in base  $R$ , and then the  $(|E(M)| - t)$ -th digit in this representation is precisely  $m_t$ . Given the value of  $\#\text{PerfMatch}(B)$ , the values of  $m_t$  are uniquely determined and can be recovered by elementary arithmetic. This way, we obtain  $m_{|V(G)|/2}$ , the number we wish to compute.

It is clear  $B$  has maximum degree 4, since  $G'$  is 3-regular and replacing edges with collars increases the maximum degree to 4. It remains to prove that  $B$  is the line graph of a bipartite graph. To this end, we construct a bipartite graph  $S$  with  $B = L(S)$  as follows: Starting from  $G$ , replace each edge  $uv \in E(G)$  by a *barbed wire* of length  $\ell = |E(G')| + 1$ . This is a  $(u, v)$ -path with  $2\ell + 2$  edges, in which each of the  $\ell$  odd-numbered internal vertices has two additional leaf-edges attached (see Figure 4 for an example with  $\ell = 3$ .) The line graph of a barbed wire of length  $\ell \in \mathbb{N}$  is a collar of length  $\ell \in \mathbb{N}$ . From this fact, it can be verified that  $B$  is the line graph of  $S$ .  $\blacksquare$

Together, Lemmas 27 and 31 prove Theorem 3.

## 6 Edge-injective homomorphisms

In this section we prove Theorem 4, our complexity dichotomy theorem for counting edge-injective homomorphisms. Recall that  $\#\text{EdgInj}(H, G)$  is the number of edge-injective homomorphisms from  $H$  to  $G$ . 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* for  $\mathcal{H}$ .

### 6.1 Polynomial-time algorithm for bounded weak vertex-cover number

In this section, we present a polynomial-time algorithm for counting edge-injective homomorphisms from pattern graphs with bounded weak vertex-cover number. To improve readability, we drop the cardinality signifier  $\#$  in this section.

**Theorem 32.** *For any constant  $c \in \mathbb{N}$ , given a graph  $H$  with a weak vertex-cover of size at most  $c$  and a graph  $G$ , we can compute  $\text{EdgInj}(H, G)$  in time  $O(n^{c'})$ , where  $c'$  is a constant depending only on  $c$ .*



In a preprocessing step, our algorithm removes isolated vertices and edges by exhaustively applying the following reduction rules.

**Lemma 33 (Deleting isolated vertices and edges).** *Let  $H$  and  $G$  be graphs.*

- *If  $v$  is an isolated vertex in  $H$  and  $H - v$  is obtained by deleting  $v$  from  $H$ , then*

$$\text{EdgInj}(H, G) = |V(G)| \cdot \text{EdgInj}(H - v, G).$$

- *If  $e = \{u, v\}$  is an isolated edge in  $H$ , then*

$$\text{EdgInj}(H, G) = 2(|E(G)| - |E(H)| + 1) \cdot \text{EdgInj}(H - u - v, G).$$

*Proof.* The first item is trivial. To prove the second item, first note that every edge-injective homomorphism  $h$  from  $H - e$  to  $G$  has exactly  $|E(H)| - 1$  edges of  $G$  in its image. To extend  $h$  to an edge-injective homomorphism from  $H$  to  $G$ , we have to map  $e$  to an edge that is distinct (but not necessarily disjoint) from the edges in the image of  $h$ . There are  $|E(G)| - |E(H)| + 1$  candidates for the image of  $e$ , and once an image  $e'$  of  $e$  has been determined, we can independently choose one of the two orientations to map  $u$  and  $v$  to the endpoints of  $e'$ . Since every edge-injective homomorphism from  $H$  to  $G$  is obtained in this way exactly once, the claim follows. ■

By this preprocessing, we can assume that  $H$  contains neither isolated vertices nor isolated edges. We then reduce the counting of edge-injective homomorphisms to the counting of embeddings. We achieve this by writing  $\text{EdgInj}(H, \cdot)$  as a linear combination of embedding numbers  $\text{Emb}(F, \cdot)$  for suitable graphs  $F$ . Let  $\text{Part}(H)$  be the set of all partitions  $\rho$  of the vertex set  $V(H)$ . Given a partition  $\rho \in \text{Part}(H)$ , let  $H/\rho$  be the quotient graph of  $H$ , which is obtained by merging each block of  $\rho$  into a vertex.

**Lemma 34.** *Let  $H$  and  $G$  be graphs. Call a partition  $\rho \in \text{Part}(H)$  edge-injective if, for every block  $B \in \rho$ , there is no edge between two vertices in  $B$  and, for every pair of blocks  $B, B' \in \rho$ , there is at most one edge between  $B$  and  $B'$  in  $H$ . Then*

$$\text{EdgInj}(H, G) = \sum_{\substack{\rho \in \text{Part}(H) \\ \text{edge-injective}}} \text{Emb}(H/\rho, G). \quad (4)$$

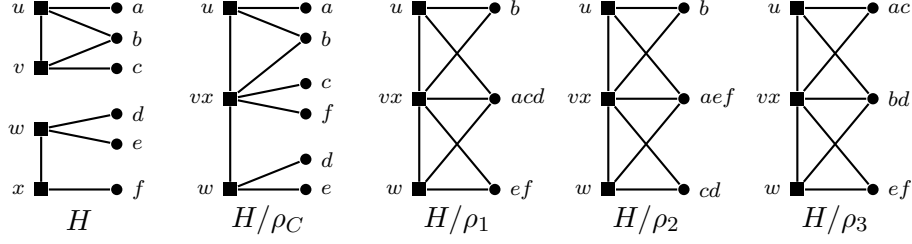
*Proof.* There exists a bijection between edge-injective homomorphisms  $h$  from  $H$  to  $G$  and pairs  $(\rho, g)$  where  $\rho \in \text{Part}(H)$  is edge-injective and  $g$  is an injective homomorphism from  $H/\rho$  to  $G$ . To define  $(\rho, g)$  from  $h$ , put vertices of  $H$  into the same block  $B$  of  $\rho$  if and only if they map to the same vertex  $v \in V(G)$  under  $h$ ; then  $g$  maps that block  $B$ , which is a vertex in  $H/\rho$ , to  $v$ . Conversely, if  $(\rho, g)$  is a given pair, the canonical homomorphism  $f$  that maps  $H$  to  $H/\rho$  is edge-injective, and we set  $h = g \circ f$ . It is easy to check that this is indeed the required bijection. ■

Each quantity  $\text{Emb}(H/\rho, G)$  on the right side of (4) can be computed in time  $n^{\text{vc}(H/\rho) + O(1)}$  by known algorithms [40, 14], where  $\text{vc}(H/\rho)$  is the vertex-cover number of  $H/\rho$ . Since every vertex-cover of  $H$  is also a vertex-cover of  $H/\rho$ , we can combine the preprocessing rules for isolated vertices and edges with (4) to obtain an  $n^{c+O(1)}$  time algorithm for  $\text{EdgInj}(H, \cdot)$  for any fixed graph  $H$  of weak vertex-cover number  $c$ . However, when taking the size of  $H$  into account as  $k = |V(H)|$ , there are up to  $k^{\Omega(k)}$  quotient graphs  $H/\rho$ . Thus, for patterns  $H$  of constant weak vertex-cover number, the algorithm via (4) is fixed-parameter tractable in the parameter  $k$ , but it does not run in polynomial time in  $k$ .

To obtain a polynomial-time algorithm and thus prove Theorem 32, we collect terms for isomorphic quotients in (4) in such a way that the resulting reduced linear combination has polynomial length and its coefficients can be computed in polynomial time. More precisely, we define an equivalence relation on  $\text{Part}(H)$  that has only polynomially many equivalence classes and we show that the size of each equivalence class can be computed efficiently. The definition of this equivalence relation is somewhat technical, so we interleave it with a discussion of a running example in Figure 5 to facilitate reading.

**Definition 35.** Let  $H$  be a graph with a fixed vertex-cover  $C$ . Let  $\rho \in \text{Part}(H)$  be an edge-injective partition.

- The *vertex-cover sub-partition* of  $\rho$  is the set  $\rho_C \subseteq \rho$  of all blocks  $B \in \rho$  that intersect  $C$ . We write  $H/\rho_C$  for the graph obtained from  $H$  by contracting each block  $B \in \rho_C$  to a single vertex, and we speak of the resulting vertices as the  *$C$ -image vertices* of  $\rho$ . We define  $I := V(H) \setminus \bigcup \rho_C$ ; these are the vertices of  $H$  not covered by  $\rho_C$ .



**Fig. 5** The figure shows a graph  $H$  with a vertex cover  $C = \{u, v, w, x\}$ , depicted as squares. There is a partition  $\rho_C = \{\{u\}, \{v, x\}, \{w\}\}$  of the vertices in the vertex cover, and  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  are (edge-injective) extensions of  $\rho_C$  such that the resulting graphs are isomorphic. The color allocations  $\mathcal{K}_{\rho_1}$  and  $\mathcal{K}_{\rho_2}$  are equal and hence  $\rho_1$  and  $\rho_2$  are equivalent. On the other hand, the color allocations  $\mathcal{K}_{\rho_1}$  and  $\mathcal{K}_{\rho_3}$  are not equal and hence  $\rho_1$  and  $\rho_3$  are not equivalent.

All three partitions  $\rho_1, \rho_2, \rho_3 \in \text{Part}(H)$  in Figure 5 have the same vertex-cover sub-partition  $\rho_C$ . The  $C$ -image vertices of these partitions are shown on the left sides of the graphs. To simplify the figure, we have chosen the partitions in such a way that, for any  $\rho \in \{\rho_1, \rho_2, \rho_3\}$ , each block in  $\rho_C$  is in fact fully contained in  $C$ . That is,  $\rho_C$  is a partition of  $C$ . In general however,  $\rho_C$  could also contract vertices from  $V(H) \setminus C$  into  $C$ .

- (ii) Sets  $K \subseteq \rho_C$  are called *colors*. These are the possible neighborhoods of vertices in  $I$ . That is, each vertex  $v \in I$  has a *color*  $K(v) \subseteq \rho_C$ , which is the set of blocks  $B \in \rho_C$  that  $v$  is adjacent to in  $H$ . (Here, we say that  $v$  is adjacent to a block  $B$  if  $v$  is adjacent to some  $w \in B$  in  $H$ .)

In Figure 5, the colors of vertices in  $I$  can be recognized well in the graph labeled with  $H/\rho_C$ : The color of a vertex in  $I$  is its neighborhood in  $H/\rho_C$ , and this neighborhood is fully contained among the  $C$ -image vertices of  $\rho$ . For example, the color of  $a$  is  $\{\{u\}\}$ , and the color of  $b$  is  $\{\{u\}, \{v, x\}\}$ .

- (iii) Consider the set  $\rho \setminus \rho_C$ ; this is a partition of the set  $I$ . When partitioning into isomorphic quotient graphs, we will not need know  $\rho \setminus \rho_C$  in its entirety. Rather, it is sufficient to know how many vertices of each color were identified by  $\rho \setminus \rho_C$ . To this end, we define the *color allocation*  $\mathcal{K}_\rho$  of  $\rho$  is the multiset

$$\mathcal{K}_\rho := \{\{K(v) : v \in B\} : B \in \rho \setminus \rho_C\}.$$

That is, for each block  $B \in \rho \setminus \rho_C$ , the color allocation contains a copy of the set of all colors appearing in  $B$ .

In Figure 5, the color allocations of  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  satisfy

$$\begin{aligned} \mathcal{K}_{\rho_1} &= \left\{ \{K(b)\}, \{K(a), K(c), K(d)\}, \{K(e), K(f)\} \right\}, \\ \mathcal{K}_{\rho_2} &= \left\{ \{K(b)\}, \{K(a), K(e), K(f)\}, \{K(d), K(c)\} \right\}, \\ \mathcal{K}_{\rho_3} &= \left\{ \{K(a), K(c)\}, \{K(b), K(d)\}, \{K(e), K(f)\} \right\}. \end{aligned}$$

Since  $K(e) = K(d)$  and  $K(c) = K(f)$ , the color allocations for  $\rho_1$  and  $\rho_2$  are actually equal. On the other hand,  $\rho_1$  and  $\rho_3$  have distinct color allocations, since  $\rho_1$  and  $\rho_3$  cannot be obtained from each other by swapping vertices of the same color for each other.

If two partitions of  $V(H)$  have the same vertex-cover sub-partition and the same color-allocation, then we will consider them to be equivalent for our purpose of collecting isomorphic quotient graphs.

- (iv) Two partitions  $\rho, \rho' \in \text{Part}(H)$  are called *equivalent* if  $\rho_C = \rho'_C$  and  $\mathcal{K}_\rho = \mathcal{K}_{\rho'}$  hold. This defines an equivalence relation on  $\text{Part}(H)$ ; the equivalence class of  $\rho$  is uniquely determined by the pair  $(\rho_C, \mathcal{K})$ . We write  $\rho_{\rho_C, \mathcal{K}}$  for an arbitrary fixed representative of this class.

In the following, we collect some properties on the interplay of the above definitions that we will ultimately put to use to obtain a polynomial-time algorithm.

**Lemma 36.** *Let  $H$  be a graph without isolated vertices, let  $C$  be a vertex-cover of  $H$ , and let  $k = |V(H)|$ . Let  $\rho \in \text{Part}(H)$  be an edge-injective partition.*

- (i) *We have  $|\rho_C| \leq |C|$ , that is, there are at most  $|C|$  vertices that are  $C$ -image vertices of  $\rho$ . We also have  $|\bigcup \rho_C| \leq |C|^2$ , that is, at most  $|C|^2$  vertices are contracted to obtain the  $C$ -image vertices of  $\rho$ .*

- (ii) Recall the color allocation  $\mathcal{K}_\rho$ . Every element  $\beta \in \mathcal{K}_\rho$  is a partition of the set  $\bigcup \beta$ , which is a set of colors. Writing  $\mathcal{K}_\rho(\beta)$  for the number of times  $\beta$  occurs as an element in the multiset  $\mathcal{K}_\rho$ , we have  $\mathcal{K}_\rho(\beta) \leq k$ .
- (iii) If  $\rho' \in \text{Part}(H)$  is edge-injective and equivalent to  $\rho$ , as defined above, then the quotient graphs  $H/\rho$  and  $H/\rho'$  are isomorphic.
- (iv) Consider the equivalence class of partitions in  $\text{Part}(H)$  corresponding to the pair  $(\rho_C, \mathcal{K}_\rho)$ . The size of this equivalence class is

$$N(\rho_C, \mathcal{K}_\rho) := \left( \prod_{K \subseteq \rho_C} \binom{k_K}{\mathcal{K}_\rho(\beta_1), \dots, \mathcal{K}_\rho(\beta_\ell)} \right) \cdot \left( \prod_{\beta} (\mathcal{K}_\rho(\beta)!)^{|\beta|-1} \right).$$

Here,  $\beta_1, \dots, \beta_\ell$  is an enumeration of all sets  $\beta \in \mathcal{K}_\rho$  with  $K \in \beta$ , and  $k_K$  is the number of vertices in  $I = V(H) \setminus \bigcup \rho_C$  that have color  $K$ .

*Proof.* (i) The first inequality follows directly from the definition. For the second inequality, let  $D = \bigcup \rho_C \subseteq V(H)$  be the set of all vertices occurring in  $\rho_C$ ; this is exactly the set of vertices that get merged into  $C$  in  $H/\rho$ . For edge-injective partitions  $\rho$ , the graph  $H[D]$  contains at most  $\binom{\rho_C}{2}$  edges. Furthermore, since  $|\rho_C| \leq |C|$  and since every vertex in  $D \setminus C$  is adjacent to  $C$ , we have  $|D| \leq |C|^2$ .

(ii) Let  $\beta \in \mathcal{K}_\rho$ . By definition there is a block  $B \in \rho \setminus \rho_C$  with  $\beta = \{K(v) : v \in B\}$ . Recall that  $K(v)$  is the set of blocks of  $\rho_C$  (or equivalently, the set of  $C$ -image vertices of  $\rho$ ) adjacent to  $v$  in the graph  $H$ . Since  $\rho$  is edge-injective, any two distinct vertices  $v, v' \in B$  have  $K(v) \cap K(v') = \emptyset$ . Thus  $\beta$  is indeed a partition of the set  $\bigcup \beta = \bigcup_{v \in B} K(v)$ . Finally,  $\mathcal{K}_\rho(\beta) \leq k$  follows immediately from the fact that  $|V(H)| = k$ .

(iii) Let  $\rho$  and  $\rho'$  be equivalent. Then  $\rho_C = \rho'_C$  holds, and so the quotients  $H/\rho$  and  $H/\rho'$  are identical on the vertex set  $\rho_C$ . Moreover,  $\rho_C$  is a vertex cover for both quotient graphs, and so the quotient graphs are isomorphic if and only if, for each set  $K \subseteq \rho_C$ , the two graphs have the same number of vertices  $v$  outside of  $\rho_C$  with  $N(v) = K$ . We prove that this is the case.

Let  $K \subseteq \rho_C$  be a color. We want to prove that  $H/\rho$  and  $H/\rho'$  have the same number of vertices outside of  $\rho_C$  whose neighborhood is exactly  $K$ . Recall that  $\mathcal{K}_\rho(\beta)$  is equal to the number of blocks  $B \in \rho \setminus \rho_C$  with  $\{K(v) : v \in B\} = \beta$ . Note that  $B$  is a vertex in the quotient graph  $H/\rho$  with neighborhood  $\bigcup \beta \subseteq \rho_C$ . As a result, the value  $\sum_{\beta \in \text{Part}(K)} \mathcal{K}_\rho(\beta)$  is equal to the number of blocks  $B \in \rho \setminus \rho_C$  such that the vertex  $B$  has neighborhood exactly  $K$  in the quotient graph  $H/\rho$ . Since these values are equal for  $\rho$  and  $\rho'$ , the claim follows, and the two quotient graphs are isomorphic.

(iv) To see this, consider a color  $K \subseteq \rho_C$ . By definition, there are exactly  $k_K$  vertices in  $I = V(H) \setminus \bigcup \rho_C$  that have color  $K$ . Let us call this set  $V_K$ . Furthermore let  $\beta_1, \dots, \beta_\ell$  be an enumeration of all sets in  $\mathcal{K}_\rho$  that contain  $K$ . All partitions  $\rho'$  that are equivalent to  $\rho$  use  $\mathcal{K}_\rho(\beta_1)$  vertices from  $V_K$  to form vertices with neighborhood  $\beta_1$  in  $H/\rho$ . There are  $\binom{k_K}{\mathcal{K}_\rho(\beta_1)}$  choices for these vertices. Similarly, there are  $\binom{k_K - \mathcal{K}_\rho(\beta_1)}{\mathcal{K}_\rho(\beta_2)}$  choices for selecting those vertices from  $V_K$  that form vertices with neighborhood  $\beta_2$  in  $H/\rho$ . Continuing this process inductively yields the claimed multinomial coefficient for the number of possibilities to allocate the vertices of  $V_K$  to the  $\beta_i$ . Since these choices are independent for independent colors  $K$ , multiplying the multinomial coefficients yields the total number of possibilities to allocate the vertices of  $I$  to the sets  $\beta \in \mathcal{K}_\rho$ .

Once vertices of  $I$  have been allocated to the  $\beta$ , we are still free to choose, for each fixed  $\beta$ , which vertices to put in the same block of  $\rho' \setminus \rho_C$ . We assembled sets  $S_1, \dots, S_{|\beta|} \subseteq I$ , each of size  $\mathcal{K}_\rho(\beta)$ . Each  $S_i$  contains vertices whose color is the  $i$ -th color of  $\beta$ . We need to construct  $\mathcal{K}_\rho(\beta)$  blocks in  $\rho' \setminus \rho_C$ , such that each block contains exactly one element from every  $S_i$ . For the first block, we have  $\mathcal{K}_\rho(\beta)^{|\beta|}$  choices. Once the first block is fixed, we have  $(\mathcal{K}_\rho(\beta) - 1)^{|\beta|}$  choices for the second one, and so on. Thus the overall number of choices to enumerate the elements of  $\rho' \setminus \rho_C$  is  $(\mathcal{K}_\rho(\beta)!)^{|\beta|}$ . Finally, since  $\rho' \setminus \rho_C$  is a set and does not care about the order of the blocks, we divide by  $\mathcal{K}_\rho(\beta)!$ . This results in  $(\mathcal{K}_\rho(\beta)!)^{|\beta|-1}$  choices to construct the set of blocks for  $\beta$ . These choices are independent for different  $\beta$ , so their product yields the number of  $\rho$  that are consistent with  $\rho_C$  and  $\mathcal{K}_\rho$ . ■

**Lemma 37 (Collecting terms).** Let  $H$  and  $G$  be graphs such that  $H$  has no isolated vertices. Let  $C$  be a fixed vertex-cover of  $H$ . Then we have

$$\text{EdgInj}(H, G) = \sum_{(\rho_C, \mathcal{K})} N(\rho_C, \mathcal{K}) \cdot \text{Emb}\left(H/(\rho_C, \mathcal{K}), G\right), \quad (5)$$

where the sum is over all equivalence classes  $(\rho_C, \mathcal{K})$  of  $\text{Part}(H)$ .

*Proof.* We start with (4) in Lemma 34 and collect terms for equivalent  $\rho \in \text{Part}(H)$ . Since the collected terms lead to isomorphic quotient graphs by Lemma 36(iii), the numbers  $\text{Emb}(H/\rho, G)$  are identical in each equivalence class, and they are equal to  $\text{Emb}(H/(\rho_{\rho_C, \mathcal{K}}), G)$ . The number of collected terms for each equivalence class is equal to  $N(\rho_C, \mathcal{K}_\rho)$  by Lemma 36(iv). This implies (5). ■

*Proof of Theorem 32.* The following algorithm computes  $\text{EdgInj}(H, G)$  via (5).

**Algorithm A** ( $\text{EdgInj}$ ) *Given  $H$  and  $G$ , this algorithm computes  $\text{EdgInj}(H, G)$ .*

- A1 Exhaustively apply the reduction rules from Lemma 33. [Now  $H$  does not have isolated vertices or edges.]
- A2 Compute a minimum vertex-cover  $C$  of  $H$  via exhaustive search.
- A3 Iterate over all equivalence classes of  $\text{Part}(H)$ ; this can be achieved by iterating over pairs  $(\rho_C, \mathcal{K})$  where  $\rho_C$  is a valid vertex-cover sub-partition and  $\mathcal{K}$  is a valid color allocation. For each equivalence class:
  - Compute  $N(\rho_C, \mathcal{K})$ .
  - Query the oracle for the value  $\text{Emb}(H/\rho_{\rho_C, \mathcal{K}}, G)$ .
- A4 Output the sum on the right side of (5).

Clearly, the steps A1 and A4 take polynomial time. Step A2 takes polynomial time, since we assumed  $|C| \leq c \leq O(1)$ . Moreover, the number  $N(\rho_C, \mathcal{K})$  and the graph  $H/\rho_{\rho_C, \mathcal{K}}$  can be computed in polynomial time. In place of the oracle for  $\text{Emb}$ , we use the known  $n^{\text{vc}(H)+O(1)}$  time algorithm [40, 14]. It remains to clarify how to iterate over the equivalence classes in A3, why there are not too many of them, and how to compute a representative  $\rho_{\rho_C, \mathcal{K}}$ .

To iterate over all candidates for  $\rho_C$ , note that  $\rho_C$  is a partition of  $\bigcup \rho_C$  and satisfies the bound  $|\bigcup \rho_C| \leq c^2$  from Lemma 36(i). Thus our algorithm exhaustively iterates over all sets  $D \subseteq V(H)$  with  $|D| \leq c^2$ . For each  $D$ , it exhaustively iterates over all  $\rho_C \in \text{Part}(D)$  with the property that every block of  $\rho_C$  intersects  $C$ . Let  $k = |V(H)|$ . The number of candidates for  $\rho_C$  is bounded by  $\binom{k}{\leq c^2} \cdot B_{c^2}$ , where the first factor reflects the possible choices for  $D$  and the second is the *Bell number*, the number of partitions of a  $c^2$ -element set. Thus there are only polynomially many choices for the candidates of  $\rho_C$ .

By Lemma 36(ii), all color allocations  $\mathcal{K}$  are multisets that contain at most  $k$  duplicates of each member. Moreover, each member of  $\mathcal{K}$  is a partition  $\beta \in \text{Part}(K)$  of some set  $K \subseteq \rho_C$ . Since there are at most  $2^c$  subsets  $K$ , each with at most  $B_c$  partitions  $\beta$ , the number of *distinct* elements of  $\mathcal{K}$  is at most  $2^c B_c \leq c'$  for some large enough constant  $c'$ . Each of these elements can occur between 0 and  $k$  times in  $\mathcal{K}$ , so once a candidate for  $\rho_C$  has been fixed, the number of distinct candidates for  $\mathcal{K}$  is bounded by  $k^{c'}$ , which is a polynomial.

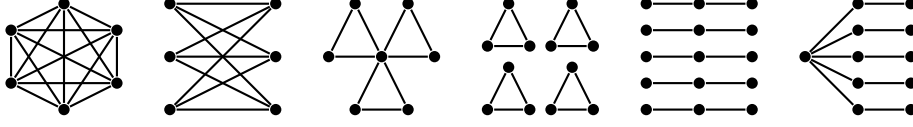
We conclude that A3 can be executed in a polynomial number of iterations. The candidates for  $\rho_C$  are partitions of size- $(\leq c^2)$  subsets of  $V(H)$  and the candidates for  $\mathcal{K}$  are multisets of partitions of subsets of  $\rho_C$ . Once a candidate  $(\rho_C, \mathcal{K})$  has been fixed, it remains to argue that we can construct a representative  $\rho_{\rho_C, \mathcal{K}}$  if it exists.

Given  $(\rho_C, \mathcal{K})$ , we construct  $\rho \supseteq \rho_C$  as follows. For each  $K \subseteq \rho_C$  and each  $\beta \in \text{Part}(K)$ , we do the following  $\mathcal{K}(\beta)$  times: Pick arbitrary vertices  $v_1, \dots, v_{|\beta|} \in V(H) \setminus \bigcup \rho_C$  such that  $\beta = \{K(v_i) : 1 \leq i \leq |\beta|\}$ , add the set  $\{v_1, \dots, v_{|\beta|}\}$  to  $\rho$ , and mark the vertices as used. If we run out of vertices in  $V(H) \setminus \bigcup \rho_C$  when doing so, or vertices are left unused at the end, then there is no  $\rho \in \text{Part}(H)$  with vertex-cover sub-partition  $\rho_C$  and color allocation  $\mathcal{K}$  (and thus  $N(\rho_C, \mathcal{K}) = 0$  holds for this candidate, since it did not represent an equivalence class). Otherwise we have constructed a partition  $\rho \in \text{Part}(H)$ . If  $\rho$  is edge-injective, we output it. Otherwise, we again have  $N(\rho_C, \mathcal{K}) = 0$  (in this case, the candidate  $\rho_C$  was not edge-injective to begin with). ■

## 6.2 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  $\#\text{EdgInj}(\mathcal{H})$  is  $\#W[1]$ -complete if  $\mathcal{H}$  has the additional property of being hereditary. To this end, we first show that every graph class of unbounded weak vertex-cover number contains one of the six basic graph classes depicted in Figure 6 as induced subgraphs.

For the purposes of this paper, we say that a graph is a windmill  $W_k$  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.



**Fig. 6** Example graphs from each of the six minimal graph classes that do not have bounded weak vertex-cover number according to Lemma 38:  $K_6$ ,  $K_{3,3}$ ,  $W_3$ ,  $4 \cdot K_3$ ,  $5 \cdot P_2$ , and  $SS_5$ .

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.

In what follows, we 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.

**Lemma 38.** *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.

*Proof.* Let  $\mathcal{H}$  be a graph class of unbounded weak vertex-cover number, and let  $C \in \mathbb{N}$  be a constant such that all cliques, bicliques, subdivided stars, windmills, and triangle packings that occur as induced subgraphs in  $\mathcal{H}$  have size at most  $C$ . To prove the lemma, we argue that  $\mathcal{H}$  contains induced  $P_2$ -packings of unbounded size. To simplify the argument, we assume without loss of generality that  $\mathcal{H}$  is closed under taking induced subgraphs. Let  $\mathcal{H}' \subseteq \mathcal{H}$  be the class of all graphs  $H \in \mathcal{H}$  that do not contain isolated edges. Since  $\mathcal{H}$  has unbounded weak vertex-cover number, the vertex-cover number of  $\mathcal{H}'$  is unbounded. Curticapean and Marx [14, Lemma 5.2] prove that, in this situation,  $\mathcal{H}'$  contains arbitrarily large cliques, induced bicliques, or induced matchings. By our assumption, the size of every clique and biclique is at most  $C$ . Thus for every  $k$ , there is a graph  $H_k \in \mathcal{H}'$  such that  $H_k$  contains a size- $k$  matching  $M_k \subseteq E(H_k)$  as an induced subgraph.

For every  $k$  and every  $e \in M_k$ , we choose an arbitrary vertex  $v_e \in V(H_k) \setminus V(M_k)$  that is adjacent in  $H_k$  to one or both endpoints of  $e$ . These vertices exist since  $e$  is not an isolated edge in  $H_k$  and  $M_k$  is an induced matching in  $H_k$ . Let  $N_k = \{v_e : e \in M_k\}$  and note that  $v_e$  and  $v_{e'}$  may coincide for distinct  $e, e' \in M_k$ . Let  $A_v \subseteq M_k$  be the set of all  $e \in M_k$  such that exactly one endpoint of  $e$  is adjacent to  $v$ , and let  $B_v \subseteq M_k$  be the set of all  $e \in M_k$  that have both their endpoints adjacent to  $v$ . If  $A_v \neq \emptyset$ , the graph  $H_k[V(A_v) \cup \{v\}]$  is an induced subdivided star of size  $|A_v|$ . Similarly, if  $B_v \neq \emptyset$ , the graph  $H_k[V(B_v) \cup \{v\}]$  is an induced windmill of size  $|B_v|$ . By our assumption on  $\mathcal{H}$ , the sets  $A_v$  and  $B_v$  have size at most  $C$  for all  $v \in N_k$  and all  $k$ .

Before we argue that arbitrarily large  $P_2$ -packings exist as induced subgraphs, we apply Ramsey's theorem to obtain more structure. Since  $M_k = \bigcup_{v \in N_k} (A_v \cup B_v)$  holds, the set  $N_k$  has size at least  $k/(2C)$ . Thus the graph class  $\{H_k[N_k] : k \in \mathbb{N}\}$  is infinite, and since we assumed that every clique in  $\mathcal{H}$  has bounded size, Ramsey's theorem guarantees the existence of independent sets  $I_k \subseteq N_k$  whose sizes are unbounded as  $k$  grows.

Finally, we construct a large induced packing of triangles and paths of length 2 using the following greedy procedure: For each  $v \in I_k$  with  $B_v \neq \emptyset$ , we select an arbitrary edge  $e \in B_v$  to contribute one triangle with  $v$ , and we remove  $A_v \cup B_v$  from  $M_k$ . Similarly, each  $v \in I_k$  with  $B_v = \emptyset$  and  $A_v \neq \emptyset$  contributes one copy of  $P_2$  and we delete  $A_v \cup B_v$  from  $M_k$ . By definition of  $A_v$  and  $B_v$ , the vertex  $v$  is not adjacent to any edge in  $M_k \setminus (A_v \cup B_v)$ ; moreover, it is not adjacent to any vertex in  $I_k \setminus \{v\}$ . Hence the constructed disjoint union of triangles and paths of length 2 is indeed an induced subgraph of  $H_k$ . Since all sets  $A_v$  and  $B_v$  are of size at most  $C$ , the number of components we constructed is at least  $|I_k|/(2C)$ , which is unbounded as  $k$  grows. By our assumption on  $\mathcal{H}$ , at most  $C$  of the components are triangles, and at least  $|I_k|/(2C) - C$  components are copies of  $P_2$ . We conclude that  $\mathcal{H}$  contains arbitrarily large induced  $P_2$ -packings. ■

Since hereditary classes  $\mathcal{H}$  are closed under induced subgraphs, Lemma 38 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 need to prove hardness for each of these six families:

**Lemma 39.** *If  $\mathcal{H}$  is the class of all cliques, the class of all bicliques, the class of all subdivided stars, the class of all windmills, the class of all triangle packings, or the class of all wedge packings, then  $\# \text{EdgInj}(\mathcal{H})$  is  $\#W[1]$ -hard.*



As we show in the following, every edge-injective homomorphism from a clique, a biclique, or a windmill into a graph is, in fact, an embedding. For these three graph families, counting edge-injective homomorphisms is thus equivalent to the corresponding subgraph counting problem. Since the families have unbounded vertex-cover number, the main theorem of Curticapean and Marx [14] implies that the subgraph counting problem for these three graph families is  $\#W[1]$ -hard.

**Lemma 40.** *Let  $G$  be a simple graph and let  $H$  be a clique, biclique, or windmill. Then every edge-injective homomorphism  $\varphi$  from  $H$  to  $G$  is an embedding.*

*Proof.* Let  $\varphi$  be an edge-injective homomorphism from  $H$  to  $G$ . For two distinct vertices  $x$  and  $y$  of  $H$ , we have  $\varphi(x) \neq \varphi(y)$  if  $x$  and  $y$  are joined by an edge of  $H$  or if they have a common neighbor  $z$  in  $H$ . If  $H$  is a clique, then all  $x, y \in V(H)$  with  $x \neq y$  are adjacent in  $H$ . If  $H$  is a biclique or a windmill, then any two distinct vertices  $x$  and  $y$  are either adjacent or have a common neighbor. In either case,  $\varphi$  is an embedding. ■

**Proposition 41.** *The problem  $\#EdgInj(\mathcal{H})$  is  $\#W[1]$ -hard if  $\mathcal{H}$  is the class of all cliques, the class of all bicliques, or the class of all windmills.*

*Proof.* By Lemma 40, the problem  $\#EdgInj(\mathcal{H})$  is equivalent to  $\#Emb(\mathcal{H})$ . Thus, since  $\mathcal{H}$  has unbounded vertex-cover number, the problem is  $\#W[1]$ -hard by the dichotomy for counting embeddings [14, Theorem 1.1]. ■

For the class of triangle packings, we devise a straightforward reduction from the problem of counting  $k$ -matchings in bipartite graphs. Essentially, we add an additional vertex that is adjacent to all other vertices, and since the original graph was bipartite, every triangle of the triangle packing must use the new vertex.

**Proposition 42.** *The problem  $\#EdgInj(\mathcal{H})$  is  $\#W[1]$ -hard if  $\mathcal{H}$  is the class of all triangle packings.*

*Proof.* We reduce from the problem of counting  $k$ -matchings in a bipartite graph. Given a simple bipartite graph  $G = (U \cup V, E)$  and a number  $k$ , we construct a graph  $G'$  from  $G$  by adding a single apex  $a$ , that is, a new vertex  $a$  and the edges  $\{a, v\}$  for all  $v \in U \cup V$ . Since  $G$  is bipartite, every triangle in  $G'$  consists of  $a$  and some vertices  $u \in U$  and  $v \in V$ . We denote such a triangle by  $a_{v,u}$ . The output of the reduction is the instance  $(H, G')$ , where  $H$  is the graph  $k \cdot K_3$ . In either case, the edges of  $H$  partition into  $k$  triangles; let us fix this partition and an arbitrary order on the triangles.

Since  $G'$  is a simple graph, the homomorphic image of a triangle is a triangle, and exactly one vertex of each of the  $k$  triangles is mapped to  $a$ . Let  $\varphi$  be an edge-injective homomorphism from  $k \cdot K_3$  to  $G'$ . Let the image of the  $i$ -th triangle be  $\{a, u_i, v_i\}$ . Since  $\varphi$  is edge-injective and  $a$  is an apex, all  $u_i$  and  $v_i$  are mutually distinct. Moreover, the edges  $\{u_i, v_i\}$  form a matching  $M_\varphi$  of size  $k$  in  $G$ .

Finally, we claim that the number  $m_k$  of  $k$ -matchings of  $G$  can be derived from the number of edge-injective homomorphisms. The homomorphism  $\varphi$  can first arbitrarily choose one vertex of each triangle to be mapped to  $a$ , which gives it  $3^k$  choices. For the remaining matching of size  $k$ , the homomorphism  $\varphi$  must map it to a matching in  $G$  by edge-injectivity. Thus it can choose one of the  $2^k \cdot k!$  automorphisms of the  $k$ -matching. Overall, we get  $6^k \cdot k!$  edge-injective homomorphisms  $\varphi$  with  $M_\varphi = M$ . Thus we have that  $|\text{EdgInj}(k \cdot K_3, G')|/(6^k \cdot k!)$ . The reduction takes polynomial time, increases the parameter from  $k$  to  $|H| = O(k)$ , and requires only one query to the oracle. ■

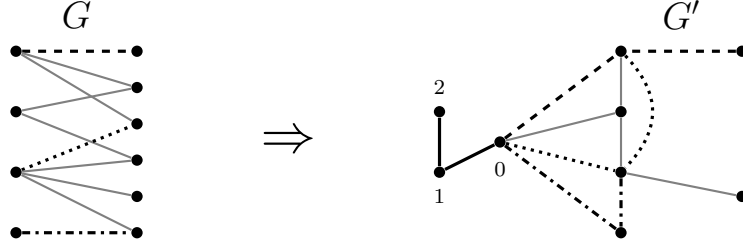
For subdivided stars, we reduce from counting  $k$ -matchings in well-structured bipartite graphs to (essentially) the graph  $G^0$  that was constructed in the proof of Theorem 22. The analysis is much simpler for subdivided stars since we can guarantee easily that the center vertex is mapped to the newly added vertex 0.

**Proposition 43.** *The problem  $\#EdgInj(\mathcal{H})$  is  $\#W[1]$ -hard if  $\mathcal{H}$  is the class of all subdivided stars.*

*Proof.* We reduce from the problem of counting  $k$ -matchings in bipartite graphs where the degree of right-side vertices is at most two and any two distinct left-side vertices have at most one common neighbor. Let  $(G, k)$  be an instance of this problem, and let  $L(G)$  and  $R(G)$  be the left and right vertex sets, respectively. Starting from  $G$ , we construct a new graph  $G'$  (see Figure 7):

1. Insert a vertex 0 that is adjacent to all vertices of  $L(G)$ .
2. For every vertex  $v \in R(G)$  with  $\deg(v) = 2$ , remove  $v$  from the graph and add the set  $N(v)$  as an edge to  $G'$ .
3. Add two special vertices 1 and 2, as well as the edges  $\{0, 1\}$  and  $\{1, 2\}$ .





**Fig. 7** The construction of  $G'$  in the proof of Proposition 43, including the image of a homomorphism from the subdivided star  $SS_4$  such that vertex 2 is contained in the image: One ray of the subdivided star is mapped to the edges  $\{0, 1\}$ ,  $\{1, 2\}$  and functions as an anchor. The other three rays (dashed, dotted, and dash-dotted) correspond to a 3-matching in  $G$ .

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  $G'$  is again a simple graph.

Let  $m_k$  be the number of  $k$ -matchings in  $G$ , let  $H$  be the subdivided star of size  $k + 1$ , and let  $s(G) = |\text{EdgInj}(H, G)|$ . We claim that

$$(k + 1)! \cdot m_k = s(G') - s(G' - \{2\}).$$

Clearly  $s(G') - s(G' - \{2\})$  is exactly the number of edge-injective homomorphisms  $\varphi$  from  $H$  to  $G'$  such that 2 is in the image of  $\varphi$ . The claim is that there is a correspondence between such homomorphisms and the  $k$ -matchings in  $G$ .

Let  $\varphi$  be an edge-injective homomorphism with  $2 = \varphi(z)$  for some  $z \in V(H)$ . Then  $z$  must be a degree-1 vertex in  $H$  since 2 has exactly one neighbor in  $G'$  and  $H$  does not contain isolated vertices. Let  $y$  be the neighbor of  $z$  and let  $x$  be the center vertex of the subdivided star. Then  $\varphi(y) = 1$  and  $\varphi(x) = 0$  holds. Next, let  $y_1, \dots, y_k$  be the other degree-2 vertices of  $H$ , and let  $z_1, \dots, z_k$  be the corresponding degree-1 vertices. Since  $\varphi$  is an edge-injective homomorphism, the vertices  $a_i := \varphi(y_i)$  are mutually distinct and satisfy  $a_i \in L(G)$ .

Note that  $\varphi(z_i) \notin \{0, 1, 2\}$ . We define the matching  $M_\varphi = \{a_1 b_1, \dots, a_k b_k\}$  as follows. If  $\varphi(z_i) \in R(G)$ , then it is a degree-1 vertex of  $R(G)$ , and we set  $b_i = \varphi(z_i)$ . Otherwise we have  $\varphi(z_i) \in L(G)$  and the edge  $\varphi(y_i z_i)$  exists in  $G'$ ; we let  $b_i$  be the unique vertex with  $N_G(b_i) = \{a_i, \varphi(z_i)\}$  that caused this edge to be added to  $G'$  in the construction. The  $b_i$  are mutually distinct due to the edge-injectivity. Hence  $M_\varphi$  is indeed a matching.

For every  $k$ -matching  $M$  of  $G$ , there are  $(k + 1)!$  distinct edge-injective homomorphisms  $\varphi$  with  $M = M_\varphi$  since  $\varphi$  can choose an arbitrary order for the  $k + 1$  rays of the subdivided star. This proves the claim.

Overall, the reduction runs in polynomial time and queries the oracle exactly two times with parameter  $|H| = O(k)$ . ■

Now we have established hardness for all of the minimal cases:

*Proof of Lemma 39.* The  $\#W[1]$ -hardness of counting wedge packings follows from Theorem 22. The remaining five cases were treated in this section. ■

As a consequence, we obtain Theorem 4.

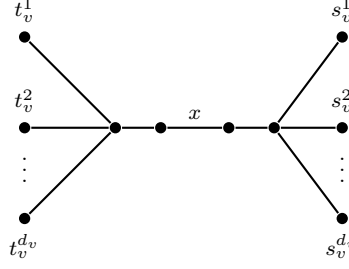
*Proof of Theorem 4.* For classes with bounded weak vertex-cover number, an algorithm with polynomial running time is given in Theorem 32. For every hereditary class of unbounded weak vertex-cover number, Lemma 38 and Lemma 39 together give  $\#W[1]$ -hardness. ■

### 6.3 Hardness for cycles and paths

The dichotomy theorem for  $\#\text{EdgInj}(\mathcal{H})$  with hereditary graph classes  $\mathcal{H}$  leaves open some non-hereditary graph classes of interest. In this final part of the paper, we investigate  $\#\text{EdgInj}(\mathcal{H})$  for the class of cycles and that of paths and prove  $\#W[1]$ -hardness for these problems.

**Theorem 44.** *For the classes  $\mathcal{C}$  and  $\mathcal{P}$  of all cycles and paths, respectively, the problems  $\#\text{EdgInj}(\mathcal{C})$  and  $\#\text{EdgInj}(\mathcal{P})$  are  $\#W[1]$ -hard.*

We point out that the problems of counting edge-injective homomorphisms from  $C_k$  and  $P_k$  are equivalent to the problems of counting edge-disjoint  $k$ -cycles and edge-disjoint  $k$ -paths, respectively. In particular,



**Fig. 8** Gadget  $H_v$  for a vertex  $v$  of degree  $d_v$  as used in the proof of Lemma 45.

for any graph  $G$ , we have that  $\# \text{EdgInj}(C_k, G)$  equals  $2k$  times the number of edge-disjoint  $k$ -cycles in  $G$ , while  $\# \text{EdgInj}(P_k, G)$  equals twice the number of edge-disjoint  $k$ -paths in  $G$ .

We will first show that  $\# \text{EdgInj}(\mathcal{C})$  is  $\#W[1]$ -hard. To this end, we consider the edge-weighted version of counting edge-injective homomorphisms in an intermediate step. Let  $H$  and  $G$  be graphs and let  $w : E(G) \rightarrow \mathbb{N}$  a weight-function. The number of edge-weighted edge-injective homomorphisms is defined as follows

$$\# \text{EdgInj}(H, G, w) := \sum_{\varphi \in \text{EdgInj}(H, G)} \prod_{e \in E(H)} w(\varphi(e)). \quad (6)$$

Then the problem  $\# \text{WEdgInj}(\mathcal{H})$  asks, given a graph  $H \in \mathcal{H}$  and an arbitrary graph  $G$  with weight-function  $w$ , to compute this quantity. The parameter is  $|V(H)| + \max\{w(e) \mid e \in E(G)\}$ . That is, the edge-weights of  $G$  must be bounded by some function in the size of the pattern graph  $H$ .

**Lemma 45.**  $\# \text{WEdgInj}(\mathcal{C})$  is  $\#W[1]$ -hard.

*Proof.* First we observe that, for all  $k \in \mathbb{N}$ , we have

$$\# \text{EdgInj}(C_k, G, w) = 2k \cdot \sum_{c \in \text{EC}_k(G)} \prod_{e \in c} w(e) \quad (7)$$

where  $\text{EC}_k(G)$  denotes the set of all edge-disjoint cycles of length  $k$  in  $G$ . We show  $\#W[1]$ -hardness by constructing an fpt Turing reduction from the  $\#W[1]$ -hard problem  $\# \text{Sub}(\mathcal{C})$  of counting simple cycles of length  $k$ , see [20, 14] for hardness proofs of this problem. On input a graph  $G$  and  $k \in \mathbb{N}$ , our reduction proceeds as follows: If  $k < 3$ , then return 0. Else consider the graph  $G_x$  obtained from  $G$  by substituting each node  $v \in V$ , of some degree  $d_v$ , by the gadget graph  $H_v$  constructed as follows: We start with a path of length 3 whose intermediate edge has weight  $x$ .<sup>3</sup> Next we add vertices  $s_v^1, \dots, s_v^{d_v}$  and connect each of them with an edge to one endpoint of the path. After that we add vertices  $t_v^1, \dots, t_v^{d_v}$  and connect each of them with an edge to the other endpoint.

Furthermore add edges  $\{s_v^i, t_u^j\}$  and  $\{s_u^j, t_v^i\}$  for every edge  $\{u, v\}$  that is the  $i$ -th edge of  $v$  and the  $j$ -th edge of  $u$ . The resulting graph is shown in Figure 8. Consider  $G_x$  as a weighted graph where every edge has weight 1 except for the edges labeled with  $x$  as above. Now, querying the oracle for  $\# \text{WEdgInj}(\mathcal{C})$  with input  $C_{6k}$  and  $G_x$ , and dividing by  $12k$  yields a polynomial  $p \in \mathbb{Z}[x]$ .

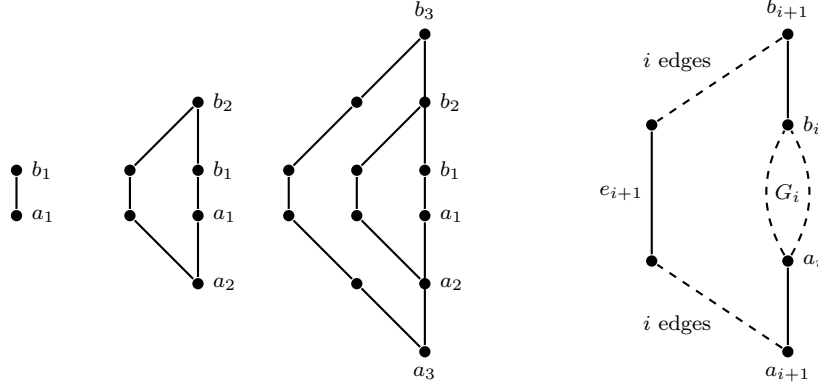
*Claim 46.* The degree of  $p$  is bounded by  $k$ . Furthermore the coefficient of  $x^k$  equals twice the number of simple  $k$ -cycles in  $G$ .

*Proof.* The shortest edge-disjoint path between any pair of two different edges with weight  $x$  is at least 5 (excluding the two edges with weight  $x$  from the length.) As we search for edge-disjoint cycles of length  $6k$ , the weight  $x$  can occur at most  $\frac{6k}{5+1} = k$  times in one cycle. Therefore the degree of  $p$  is bounded by  $k$ . Furthermore the distance is equal to 5 if and only if the two edges belong to gadgets  $H_v$  and  $H_u$  such that  $\{v, u\} \in E(G)$ . In particular, this path either leaves  $H_v$  through  $s_v^i$  and enters  $H_u$  through  $t_u^j$  for some  $i$  and  $j$  or it leaves  $H_v$  through  $t_v^{i'}$  and enters  $H_u$  through  $s_u^{j'}$  for some  $i'$  and  $j'$ .

Now consider an edge-disjoint cycle  $c$  of length  $6k$  that includes  $k$  edges with weight  $x$ . It follows that  $c = (e_1, P_1, \dots, e_k, P_k, e_1)$  where each  $e_i$  has weight  $x$  and each  $P_i$  is a path consisting of 5 edges with weight 1.

Next let  $H_{v_i}$  be the gadget containing  $e_i$  and consider  $H_{v_1}$ . It holds that  $P_1$  either passes through  $s_{v_1}^i$  and  $t_{v_2}^j$  for some  $i$  and  $j$  or through  $t_{v_1}^{i'}$  and  $s_{v_2}^{j'}$  for some  $i'$  and  $j'$ . However, if we fix one of these two options,

<sup>3</sup> Here,  $x$  is an indeterminate, so the quantity (7) is a polynomial in  $x$ .



**Fig. 9** Graphs  $G_1$ ,  $G_2$  and  $G_3$ , as well as the inductive construction of graph  $G_{i+1}$  as used in the proof of Lemma 47.

only one possibility remains for all other  $P_2, \dots, P_k$  as we cannot turn around in a gadget if we consider edge-disjoint cycles. Therefore there are exactly two edge-disjoint cycles  $c_1 = (e_1, P_1, \dots, e_k, P_k, e_1)$  and  $c_2 = (e_1, P'_1, \dots, e_k, P'_k, e_1)$  that correspond to the cycle  $c = (v_1, \dots, v_k, v_1)$  in  $G$  and vice versa. Furthermore  $c$  is simple as  $c_1, c_2$  are edge-disjoint, that is, the  $e_i$ 's and therefore the  $v_i$ 's are pairwise different.  $\square$

To conclude the proof of Lemma 45, we compute the coefficient of  $x^k$  in the degree- $k$  polynomial  $p$  by means of polynomial interpolation from the evaluations  $p(0), \dots, p(k)$ . These evaluations are obtained by oracle calls to  $\#\text{WEdgInj}(\mathcal{C})$  with input  $C_{6k}$  and  $G_b$  for  $b = 0, \dots, k$  (and dividing by  $12k$ ). As the edge-weights of every graph  $G_b$  are bounded by  $k$ , the overall parameter  $|V(C_{6k})| + \max\{w(e) \mid e \in G_b\}$  is bounded by  $7k$ , proving that this reduction is indeed an fpt Turing-reduction.  $\blacksquare$

We show hardness of the unweighted version by reduction from the weighted version; this requires us to devise a strategy for removing weights.

**Lemma 47.** *There is an fpt Turing reduction from  $\#\text{WEdgInj}(\mathcal{C})$  to  $\#\text{EdgInj}(\mathcal{C})$ .*

*Proof.* The input for the reduction is a number  $k \in \mathbb{N}$  and an edge-weighted graph  $G$  whose edge weights are bounded by  $k$ . We assume  $k \geq 4$ , as we can otherwise solve the problem in polynomial time by brute-force. The following gadgets will be used in the reduction:

- $G_1$  is simply one undirected edge  $e_1 := \{a_1, b_1\}$
- $G_{i+1}$  is constructed from  $G_i$  as follows: We add vertices  $a_{i+1}$  and  $b_{i+1}$  and edges  $\{a_{i+1}, a_i\}$  and  $\{b_{i+1}, b_i\}$ . Furthermore we add a path of length  $2i + 1$  between  $a_{i+1}$  and  $b_{i+1}$  and denote the  $i + 1$ -th edge of this path  $e_{i+1}$ .  $G_{i+1}$  is depicted in Figure 9.

It is easy to see that  $|V(G_k)| \leq O(k^2)$ .

*Claim 48.* For every  $k \geq 1$ , there are exactly  $k$  possibilities to construct an edge-disjoint walk from  $a_k$  to  $b_k$  in  $G_k$ , each of which has length  $2k - 1$ . Furthermore, for every  $j \in [k]$ , the edge  $e_j$  is contained in exactly one of this walks.

*Proof.* We prove the claim by induction on  $k$ ; it is obvious for  $k = 1$ . For the induction step, consider  $G_{k+1}$ : An edge-disjoint walk from  $a_{k+1}$  to  $b_{k+1}$  either takes the "left" way and therefore contains  $e_{k+1}$  or takes a way through  $G_k$ :

- The "left" way has length  $2k + 1 = 2(k + 1) - 1$ .
- Every way through  $G_k$  corresponds one-to-one to a closed walk from  $a_k$  to  $b_k$  in  $G_k$ . Applying the induction hypothesis we obtain that there are exactly  $k$  edge-disjoint walks from  $a_k$  to  $b_k$  in  $G_k$ , one for every  $e_j$  for  $j \in [k]$ . Furthermore each of this walks has length  $2k - 1$ . It follows that there are exactly  $k$  edge-disjoint walks from  $a_{k+1}$  to  $b_{k+1}$ , each of length  $2 + 2k - 1 = 2(k + 1) - 1$ .  $e_j$  is contained in exactly one of this walks for every  $j \in [k]$ .

We conclude that the claim is fulfilled for  $G_{k+1}$ .  $\square$

It follows that the longest edge-disjoint cycle in  $G_k$  has length  $2 \cdot (2k + 1) = 4k + 2$ . Let  $W$  be the maximum weight of an edge, where  $W \leq k$ . Now let  $H_i$  be the gadget constructed from  $G_W$  by removing edges  $e_W, \dots, e_{i+1}$ . We have  $H_W = G_W$ . Applying Claim 48, we obtain that there are exactly  $i$  edge-disjoint walks from  $a_W$  to  $b_W$  in  $H_i$ . Furthermore each of this walks has length  $2W - 1$ . Finally, we construct  $G'$  from  $G$  by substituting each edge  $e = \{a, b\}$  with  $H_{w(e)}$  and edges  $\{a, a_W\}$  and  $\{b, b_W\}$ .

*Claim 49.* The number of edge-disjoint cycles of length  $(2Wk + k)$  in  $G'$  equals

$$\sum_{c \in \text{EC}_k(G)} \prod_{e \in c} w(e).$$

*Proof.* Consider an edge-disjoint cycle  $c$  of length  $(2Wk + k)$  in  $G'$ . Assuming  $c$  does contain an edge  $\{a, a_W\}$  (that is, it is not entirely contained in one gadget), it follows that  $c$  can cross every  $a_W$  and  $b_W$  at most once. To see this, observe that every time when such a node is reached we can consider the cycle coming from "outside the gadget" (e.g. by choosing a fitting orientation of  $c$ ). Since  $c$  is an edge-disjoint cycle, we have to continue by an edge-disjoint walk through the end of the gadget. This walk has length  $2W - 1$  by Claim 48. Now we cannot turn around inside the gadget again and complete the cycle afterwards since otherwise we would have constructed a longer edge-disjoint walk from one endpoint of the gadget to the other, which contradicts Claim 48. It follows that every edge-disjoint cycle of length  $(2Wk + k)$  that is not entirely contained in one gadget consists of  $2W - 1$  walks through gadgets. Now, taking an edge  $e = \{a, b\}$  with weight  $w(e)$  in  $G$  corresponds to taking one of the  $w(e)$  edge-disjoint walks  $(a, a_W, \dots, b_W, b)$  of length  $2W - 1 + 2$  through  $H_{w(e)}$  in  $G'$ . As  $k \cdot (2W - 1 + 2) = (2Wk + k)$  it follows that an edge-disjoint cycle of length  $k$  in  $G$  corresponds to the edge-disjoint cycles of length  $(2Wk + k)$  in  $G'$  that cross the gadgets corresponding to the weighted edges in  $G$ , but only if no edge-disjoint cycle of length  $(2Wk + k)$  entirely fits in one gadget. However, the latter cannot be the case since the longest edge-disjoint cycle in  $G_W$  has length  $4W + 2$  and for every  $k > 2$  it holds that

$$4W + 2 < 4W \leq 2Wk < 2Wk + k.$$

□

Using Claim 49, Equation 7 and the fact that for every graph  $G$  and  $k \in \mathbb{N}$ , we have that  $\#\text{EdgInj}(C_k \rightarrow G)$  equals  $2k$  times the number of edge-disjoint  $k$ -cycles in  $G$ , we obtain that

$$\begin{aligned} \#\text{EdgInj}(C_{2Wk+k}, G') &= 2(2Wk + k) \sum_{c \in \text{EC}_k(G)} \prod_{e \in c} w(e) \\ &= \frac{2(2Wk + k)}{2k} \cdot \#\text{EdgInj}(C_k, G, w) \\ &= (2W + 1) \cdot \#\text{EdgInj}(C_k, G, w). \end{aligned}$$

The above reduction is indeed an fpt Turing reduction, as  $G'$  can be constructed in time  $O(n^2 \cdot k^2)$  and the value of the new parameter is  $2Wk + k \leq O(k^2)$ . This concludes the proof. ■

**Corollary 50.** *The problem  $\#\text{EdgInj}(C)$  is  $\#\text{W}[1]$ -hard.*

It remains to show hardness for  $\mathcal{P}$ :

**Lemma 51.** *The problem  $\#\text{EdgInj}(\mathcal{P})$  is  $\#\text{W}[1]$ -hard.*

*Proof.* We will reduce from  $\#\text{EdgInj}(C)$ . First, we let  $\text{EC}_k(G, v)$  be the set of all edge-disjoint cycles of length  $k$  in  $G$  that contain  $v \in V(G)$ . Recall, for comparison, that  $\text{EC}_k(G)$  denotes the set of all edge-disjoint cycles of length  $k$  in  $G$ .

*Claim 52.* It holds that

$$\text{EC}_k(G) = \bigcup_{i=1}^{|V(G)|} \text{EC}_k(G - \{v_{i+1}, \dots, v_n\}, v_i),$$

where the union is in fact a pairwise disjoint union.

*Proof.* By induction on  $|V(G)|$ . If  $|V(G)| = 0$ , the union is empty and therefore the claim holds. Otherwise let  $|V(G)| = n + 1$ . It holds that

$$\begin{aligned} \text{EC}_k(G) &= \text{EC}_k(G, v_{n+1}) \dot{\cup} (\text{EC}_k(G) \setminus \text{EC}_k(G, v_{n+1})) \\ &= \text{EC}_k(G, v_{n+1}) \dot{\cup} \text{EC}_k(G - \{v_{n+1}\}) \\ &= \text{EC}_k(G, v_{n+1}) \dot{\cup} \left( \bigcup_{i=1}^n \text{EC}_k((G - \{v_{n+1}\}) - \{v_{i+1}, \dots, v_n\}, v_i) \right) \\ &= \text{EC}_k(G, v_{n+1}) \dot{\cup} \left( \bigcup_{i=1}^n \text{EC}_k(G - \{v_{i+1}, \dots, v_{n+1}\}, v_i) \right) \\ &= \bigcup_{i=1}^{n+1} \text{EC}_k(G - \{v_{i+1}, \dots, v_{n+1}\}, v_i) \end{aligned}$$

Here, the third equality follows from the induction hypothesis. □

It follows that

$$|\text{EC}_k(G)| = \sum_{i=1}^{|V(G)|} |\text{EC}_k(G - \{v_{i+1}, \dots, v_n\}, v_i)|. \quad (8)$$

Now let  $G_i = G - \{v_{i+1}, \dots, v_n\}$ . We show that  $|\text{EC}_k(G_i, v_i)|$  can be computed using an oracle for  $\#\text{EdgInj}(\mathcal{P})$ :

First, we construct the graph  $G'_i$  by adding vertices  $s$  and  $t$  and edges  $\{s, v_i\}$  and  $\{t, v_i\}$ . For  $M \subseteq \{s, t\}$  let  $A_{i,M}$  be the set of edge-disjoint paths of length  $k+2$  that do not pass through a vertex  $u \in M$  and let  $G'_{i,M}$  be the graph obtained from  $G'_i$  by removing every vertex that lives in  $M$ . Note that  $|A_M|$  can be computed by querying the oracle for  $P_{k+1}$  and  $G'_{i,M}$  (and dividing by 2). Now it holds that for all  $i \in \{1, \dots, |V(G)|\}$ :

$$|\text{EC}_k(G_i, v_i)| = |A_{i,\emptyset} \setminus (A_{i,\{s\}} \cup A_{i,\{t\}})| = |A_{i,\emptyset}| - |A_{i,\{s\}}| - |A_{i,\{t\}}| + |A_{i,\{s,t\}}|,$$

where the last equality follows from the principle of inclusion and exclusion. Finally the values of  $|\text{EC}_k(G_i, v_i)|$  for all  $i \in \{1, \dots, |V(G)|\}$  allow us to compute  $|\text{EC}_k(G)|$  (see Equation 8), which equals  $1/2 \cdot \#\text{EdgInj}(C_k, G)$ . ■

*Proof of Theorem 44.* Follows from Corollary 50 and Lemma 50. ■

*Acknowledgments.* The authors thank Cornelius Brand and Markus Bläser for interesting discussions, and Johannes Schmitt for pointing out a proof of Lemma 20 and allowing us to use it in this paper.

## References

1. 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>.
2. 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.
3. Jin-yi Cai and Pinyan Lu. Holographic algorithms: From art to science. *J. Comput. Syst. Sci.*, 77(1):41–61, 2011. doi:10.1016/j.jcss.2010.06.005.
4. Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms by Fibonacci gates and holographic reductions for hardness. In *Proceedings of the 49th Annual Symposium on Foundations of Computer Science, FOCS*, pages 644–653, 2008. doi:10.1109/FOCS.2008.34.
5. 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.
6. Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Holographic algorithms with matchgates capture precisely tractable planar  $\#\text{CSP}$ . *SIAM J. Comput.*, 46(3):853–889, 2017. doi:10.1137/16M1073984.
7. 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.
8. 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.
9. Maria Chudnovsky, Neil Robertson, Paul Seymour, and Robin Thomas. The strong perfect graph theorem. *Annals of Mathematics*, 164:51–229, 2006. doi:10.4007/annals.2006.164.51.
10. Nadia Creignou and Miki Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125(1):1–12, 1996. doi:10.1006/inco.1996.0016.
11. Radu Curticapean. Counting matchings of size  $k$  is  $\#W[1]$ -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, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In *Proceedings of the 49th ACM Symposium on Theory of Computing, STOC*, pages 210–223, 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 *Proceedings of the 55th Annual Symposium on Foundations of Computer Science, FOCS*, pages 130–139. IEEE, 2014. doi:10.1109/FOCS.2014.22.
15. 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.
16. 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.
17. 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.
18. 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.
19. Jörg Flum and Martin Grohe. The parameterized complexity of counting problems. *SIAM Journal of Computing*, 33(4):892–922, 2004. doi:10.1137/S0097539703427203.

20. Jörg Flum and Martin Grohe. *Parameterized complexity theory*. Springer, 2006.
21. 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.
22. 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.
23. 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.
24. 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.
25. Frank Harary. *Graph theory*. Addison-Wesley, Reading, 1969.
26. Mark Jerrum. Two-dimensional monomer-dimer systems are computationally intractable. *Journal of Statistical Physics*, 48(1-2):121–134, 1987. doi:10.1007/BF01010403.
27. 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.
28. Pieter W. Kasteleyn. Graph theory and crystal physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.
29. 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.
30. László Lovász. *Large Networks and Graph Limits*, volume 60 of *Colloquium Publications*. American Mathematical Society, 2012. URL: <http://www.ams.org/bookstore-getitem/item=COLL-60>.
31. 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.
32. 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.
33. Marc Roth. Counting restricted homomorphisms via Möbius inversion over matroid lattices. In *25th Annual European Symposium on Algorithms, ESA 2017, September 4-6, 2017, Vienna, Austria*, pages 63:1–63:14, 2017. doi:10.4230/LIPIcs.ESA.2017.63.
34. 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.
35. Harold N. V. Temperley and Michael E. Fisher. Dimer problem in statistical mechanics - an exact result. *Philosophical Magazine*, 6(68):1478–1485, 1961. doi:10.1080/14786436108243366.
36. 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.
37. Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979. doi:10.1016/0304-3975(79)90044-6.
38. Leslie G. Valiant. Holographic algorithms. *SIAM Journal of Computing*, 37(5):1565–1594, 2008. doi:10.1137/070682575.
39. Ľubomír Šoltés. Forbidden induced subgraphs for line graphs. *Discrete Mathematics*, 132(1):391–394, 1994. doi:10.1016/0012-365X(92)00577-E.
40. Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM Journal on Computing*, 42(3):831–854, 2013. doi:10.1137/09076619X.
41. 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.
42. Xiao-Dong Zhang and Stanislaw Bylka. Disjoint triangles of a cubic line graph. *Graphs and Combinatorics*, 20(2):275–280, 2004. doi:10.1007/s00373-004-0551-6.