Robust Contraction Decomposition for Minor-Free Graphs and its Applications

Sayan Bandyapadhyay[1], William Lochet[2], Daniel Lokshtanov[3], Dániel Marx[4], Pranabendu Misra[5], Daniel Neuen[6], Saket Saurabh[7], Prafullkumar Tale[8], and Jie Xue[9]

[1]Portland State University, USA
[2]LIRMM, Université de Montpellier, CNRS, France
[3]University of California, USA
[4]CISPA Helmholtz Center for Information Security, Germany
[5]Chennai Mathematical Institute, India
[6]Max Planck Institute for Informatics, Germany
[7]Institute of Mathematical Sciences, India
[8]Indian Institute of Science Education and Research Bhopal, India
[9]New York University Shanghai, China

September 14, 2025

Abstract

We prove a robust contraction decomposition theorem for H-minor-free graphs, which states that given an H-minor-free graph G and an integer p, one can partition in polynomial time the vertices of G into p sets $Z_1, \ldots, Z_p$ such that $\mathbf{tw}(G/(Z_i \setminus Z')) = O(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$. Here, $\mathbf{tw}(\cdot)$ denotes the treewidth of a graph and $G/(Z_i \setminus Z')$ denotes the graph obtained from G by contracting all edges with both endpoints in $Z_i \setminus Z'$.

Our result generalizes earlier results by Klein [SICOMP 2008] and Demaine et al. [STOC 2011] based on partitioning $E(G)$, and some recent theorems for planar graphs by Marx et al. [SODA 2022], for bounded-genus graphs (more generally, almost-embeddable graphs) by Bandyapadhyay et al. [SODA 2022], and for unit-disk graphs by Bandyapadhyay et al. [SoCG 2022].

The robust contraction decomposition theorem directly results in parameterized algorithms with running time $2^{\widetilde{O}(\sqrt{k})} \cdot n^{O(1)}$ or $n^{O(\sqrt{k})}$ for every vertex/edge deletion problems on H-minor-free graphs that can be formulated as Permutation CSP Deletion or 2-Conn Permutation CSP Deletion. Consequently, we obtain the first subexponential-time parameterized algorithms for Subset Feedback Vertex Set, Subset Odd Cycle Transversal, Subset Group Feedback Vertex Set, 2-Conn Component Order Connectivity on H-minor-free graphs. For other problems which already have subexponential-time parameterized algorithms on H-minor-free graphs (e.g., Odd Cycle Transversal, Vertex Multiway Cut, Vertex Multicut, etc.), our theorem gives much simpler algorithms of the same running time.

1 Introduction

Baker's layering technique is a fundamental tool for a certain type of algorithms on planar graphs. It was first used implicitly by Baker [1] and can be formalized as the following statement: given a planar graph G and an integer p, we can find in polynomial-time a partitioning $Z_1, \ldots, Z_p$ of the vertex set of G such that $G - Z_i$ has treewidth $O(p)$ for every $i \in [p]$. This decomposition can be used for problems where we can argue that there exist an $i \in [p]$ such that Z_i is irrelevant

to the solution (or has negligible contribution to it) and hence Z_i can be deleted from G. Then we are left with an instance $G - Z_i$ having treewidth $O(p)$ and techniques for bounded-treewidth graphs can be used. This approach has been used in the design of polynomial-time approximation schemes (PTAS) [1, 7, 24] and parameterized algorithms [4, 12, 14, 18, 19, 35] for planar graphs. Moreover, this decomposition algorithm can be further generalized to H-minor free graphs [7].

Contraction Decomposition Theorems. There are many natural problems where a set Z_i cannot be removed even if it is irrelevant to the solution: most notably, problems involving connectivity typically have this property. To handle such problems, Klein [26] introduced a dual version of Baker's layering technique, which is based on contractions of edges. This *(Edge) Contraction Decomposition Theorem* was later generalized to H-minor free graphs by Demaine, Hajiaghayi, and Kawarabayashi [8].

Theorem 1.1 (Demaine, Hajiaghayi, and Kawarabayashi [8])**.** *Let H be a fixed graph. Given an H-minor-free graph G and an integer p, one can compute in polynomial time a partition of $E(G)$ into p sets $Z_1, \ldots, Z_p$ such that $\mathbf{tw}(G/Z_i) = O(p)$ for all $i \in [p]$, where the constant hidden in $O(\cdot)$ depends on H.*

Contraction decomposition theorems of this form can be used as a building block in designing approximation schemes for, e.g., the Traveling Salesman Problem (TSP) and Subset TSP [4, 8, 9, 25, 26], and parameterized algorithms for Bisection, k-Way Cut, Odd Cycle Transversal, Subset Feedback Vertex Set and many other problems [3, 22, 29].

Subexponential Parameterized Algorithms (Edge Problems). To illustrate this, let us briefly discuss how Theorem 1.1 can be used to obtain a subexponential-time parameterized algorithm for Edge Multiway Cut in H-minor-free graphs. In Edge Multiway Cut, we are given a graph G, a set $T \subseteq V(G)$ of terminals, and an integer k, and the task is to find a set $S \subseteq E(G)$ of at most k edges such that every component of $G \setminus S$ contains at most one terminal.

(1) A result of Wahlström [37] gives a randomized quasipolynomial kernel for the problem, which allows us to assume that $|V(G)| = k^{\mathrm{polylog}(k)}$.

(2) Given the decomposition $Z_1, \ldots, Z_p$ of Theorem 1.1 where $p := \lceil \sqrt{k} \rceil$, there is an $i \in [p]$ such that $|Z_i \cap S| \leq \sqrt{k}$ for the hypothetical solution S of size k.

(3) Guess this $i \in [p]$ (there are $\sqrt{k}$ possibilities) and the set $Z_i \cap S$ (there are $|V(G)|^{O(\sqrt{k})} = k^{\sqrt{k} \cdot \mathrm{polylog}(k)}$ possibilities)

(4) Contracting $Z_i \setminus S$ does not change the problem at hand, since the solution S if not affected.

(5) Since the graph G/Z_i has treewidth $O(\sqrt{k})$, we get that the graph $G/(Z_i \setminus S)$ has treewidth $O(\sqrt{k} + |Z_i \cap S|) = O(\sqrt{k})$ (contracting an edge can decrease treewidth by at most one). Finally, we solve Edge Multiway Cut on $G/(Z_i \setminus S)$ in time $2^{O(\sqrt{k} \log k)} \cdot |V(G)|^{O(1)}$ using standard bounded-treewidth techniques.

Overall, we obtain a $2^{\sqrt{k} \cdot \mathrm{polylog}(k)} \cdot n^{O(1)}$ time randomized algorithm for Edge Multiway Cut. The same approach also works for many other problems [2, 29].

Subexponential Parameterized Algorithms (Vertex Problems). Let us try to apply a similar technique for the problem Vertex Multiway Cut, where instead of deleting a set of edges, we need to delete now a set of vertices. There are two main difficulties if we try to apply Theorem 1.1. First, it would be convenient to have a partition $Z_1, \ldots, Z_p$ of *vertices* such that for every $i \in [p]$ contracting Z_i leaves a graph of treewidth $O(p)$. Here contracting a vertex set

Z_i amounts to contracting all edges that have both endpoints in the set Z_i. Second, in Step (5) above, we exploited that the decomposition of Theorem 1.1 is *inherently robust*: if Z_i' is obtained from Z_i by omitting a set of s edges, then the treewidth of G/Z_i' is at most s higher than the treewidth of G/Z_i.

The following example shows that an analogous statements for contracting vertex sets is not true. Let G be obtained from a $k \times k$ grid by adding two universal vertices x and y. Let A and B be the two bipartition classes of the $k \times k$ grid, and define $Z_1 := A \cup \{x\}$ and $Z_2 := B \cup \{y\}$. Then G/Z_i has treewidth 2 for both $i \in [2]$. On the other hand, $G/(Z_1 \setminus \{x\}) = G/A = G$ and $G/(Z_2 \setminus \{y\}) = G/B = G$ which means the jump in treewidth can be unbounded even after omitting just a single vertex. Note that G is K_7-minor free, so we cannot hope that a vertex partition version of Theorem 1.1 is inherently robust even on H-minor-free graphs.

Robust Contraction Decomposition Theorem. The above naturally leads to the question of a *robust* vertex contraction decomposition theorem, where omitting vertices from some Z_i increases the treewidth only in a bounded way. Such decomposition theorems were recently introduced independently for planar graphs [29] and for bounded-genus graphs (and more generally, almost-embeddable graphs) [2] by subsets of the authors, who used them to obtain the first subexponential-time (parameterized) algorithms for a number of problems, such as Edge Bipartization, Odd Cycle Transversal, Edge/Vertex Multiway Cut, Edge/Vertex Multicut, and Group Feedback Edge/Vertex Set on the above graph classes. More precisely, [2, 29] design parameterized algorithms with running time $f(k)n^{O(\sqrt{k})}$ or even $2^{O(\sqrt{k})}n^{O(1)}$, where k is the size of the solution (and $\widetilde{O}(\cdot)$ hides $\log k$ factors).

The main result of this paper is a *vertex* contraction decomposition theorem on H-minor-free graphs that is *robust* in the sense described above.

Theorem 1.2. *Let H be a fixed graph. Given an H-minor-free graph G and an integer p, one can compute in polynomial time a partition of $V(G)$ into p sets $Z_1, \ldots, Z_p$ such that $\mathbf{tw}(G/(Z_i \setminus Z')) = O(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$, where the constant hidden in $O(\cdot)$ depends on H.*

Theorem 1.2 generalizes the robust vertex contraction theorem for planar graphs [29] and almost-embeddable graphs [2] to arbitrary H-minor-free graphs. It also generalizes the edge contraction decomposition (Theorem 1.1) for H-minor free graphs by Demaine et al. [8].

Combined with the results from [29], Theorem 1.2 directly yields parameterized algorithms of running time $2^{\widetilde{O}(\sqrt{k})} \cdot n^{O(1)}$ or $n^{O(\sqrt{k})}$ for all vertex/edge deletion problems on H-minor-free graphs that can be formulated as Permutation CSP Deletion or 2-Conn Permutation CSP Deletion, two broad classes of problems defined in [29]. Roughly speaking, a permutation CSP instance is a binary CSP instance satisfying certain nice properties, and thus corresponds to a graph in which the vertices are variables and the edges are constraints. The Permutation CSP Edge/Vertex Deletion problem basically asks whether one can delete at most k vertices (resp., edges) from (the graph of) a permutation CSP instance such that every connected component in the resulting graph has a satisfying assignment. The 2-Conn Permutation CSP Edge/Vertex Deletion problem is defined similarly, but we only care about the satisfiability on every 2-connected component (instead of connected component). We refer to Section 4 for the precise definition of these problems. Combining the results from [29] and Theorem 1.2, we get the following result.

Theorem 1.3. *Let H be a fixed graph. Then* Permutation CSP Edge/Vertex Deletion *and* 2-Conn Permutation CSP Edge/Vertex Deletion *on H-minor-free instances (Γ, w, δ, k) can be solved in $(|\Gamma| + \delta)^{O(\sqrt{k})}$ time.*

The above theorem directly gives us parameterized algorithms of running time exponential in $O(\sqrt{k})$ for many problems on H-minor-free graphs. Specifically, we obtain the first

subexponential-time parameterized algorithms for the following fundamental problems on H-minor-free graphs.

- $n^{O(\sqrt{k})}$ time algorithms for Subset Feedback Vertex Set, Subset Odd Cycle Transversal, Subset Group Feedback Vertex Set, 2-Conn Component Order Connectivity, and the edge-deletion version of these problems.
- $2^{\widetilde{O}(\sqrt{k})} \cdot n^{O(1)}$ time algorithms for Subset Feedback Vertex Set and Subset Feedback Edge Set.

Additionally, the main algorithmic results from [2] (such as subexponential-time parameterized algorithms for Odd Cycle Transversal, Vertex Multiway Cut, Vertex Multicut, etc.), that required a complicated two-layered dynamic programming algorithm over the structural decomposition theorem of Robertson and Seymour [34] for H-minor free graphs, now admit much cleaner and more direct algorithms by exploiting Theorem 1.2 (these problems belong to the Permutation CSP Deletion category in Theorem 1.3).

Other Relevant Literature. The study of subexponential-time parameterized algorithms on planar and H-minor free graphs has been one of the most active sub-areas of parameterized algorithms, which led to exciting results and powerful methods. Examples include Bidimensionality [6], applications of Baker's layering technique [4, 12, 14, 18, 35], bounds on the treewidth of the solution [17, 27, 28, 30], and pattern coverage [18, 32]. The central theme of all these results is that planar graphs exhibit the " *square root phenomenon*": parameterized problems whose fastest parameterized algorithm run in time $f(k)n^{O(k)}$ or $2^{O(k)}$ on general graphs admit $f(k)n^{O(\sqrt{k})}$ or even $2^{O(\sqrt{k})}n^{O(1)}$ time algorithms when input is restricted to planar or H-minor free graphs. However, these techniques do not apply for designing subexponential time parameterized algorithms for cut and cycle hitting problems such as Multiway Cut, Multicut, or Odd Cycle Transversal. Robust vertex contraction decomposition theorems, first obtained in [2, 29], provide the necessary tools to design subexponential-time parameterized algorithms for some of the cut and cycle hitting problems on H-minor free graphs.

On the decomposition front, apart from edge contraction decomposition theorems, there are several other kind of decomposition theorems that have been used to design polynomial-time approximation schemes (PTASs) and FPT algorithms on planar graphs or more generally on H-minor free graphs. Most of these decomposition theorems prove strengthening of the classic Baker's layering technique [1]. This generally yields, in what is known as (Vertex) Edge Decomposition Theorems [1, 7, 10, 13, 15] (see [33] for a detailed introduction).

Finally, let us also point to some recent developments on the structural theory of H-minor-free graphs [21, 36]. Our current proof of Theorem 1.2 mostly relies on the original Robertson-Seymour decomposition for H-minor-free graphs [34], and is independent of [21, 36]. Reformulating some of our arguments in the language of [21, 36] may allow us to obtain a more streamlined proof in the future.

Structure of Paper. The remainder of the paper is structured as follows. In Section 1.1 we give an informal overview of the proof of Theorem 1.2. After giving additional preliminaries in Section 2, the formal proof of Theorem 1.2 is given in Section 3. Finally, we discuss the algorithmic applications in Section 4.

1.1 Overview of Proof of Theorem 1.2

In this section, we give an informal overview of our proof of Theorem 1.2. For convenience, we say $Z_1, \ldots, Z_p$ is a *robust contraction decomposition (RCD)* of a graph G if $Z_1, \ldots, Z_p$ is a partition of $V(G)$ satisfying $\mathbf{tw}(G/(Z_i \setminus Z')) = O(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$. A p-RCD

simply refers to an RCD of size p. Our goal is to compute a p-RCD of an H-minor-free graph G for a given integer $p \geq 1$.

Our starting point is the well-known Robertson-Seymour decomposition for H-minor-free graphs. The Robertson-Seymour decomposition of an H-minor-free graph G is a tree decomposition (T, β) of G in which the *torso* of each node $t \in V(T)$ is *h-almost-embeddable* and the *adhesion* of each node $t \in V(T)$ is of size at most h, for some constant h depending on H. Here, the adhesion of t, denoted by $\sigma(t)$, is the intersection of the bag $\beta(t)$ and the bag of the parent of t. The torso of t, denoted by $\mathsf{tor}(t)$, is a supergraph of $G[\beta(t)]$ (the subgraph of G induced by $\beta(t)$), obtained by adding edges to $G[\beta(t)]$ to make the adhesion of each child of t a clique. Almost-embeddable graphs generalize bounded-genus graphs. Roughly speaking, an h-almost-embeddable graph has its main part embedded in a surface of genus at most h, and the remaining part consists of at most h well-structured subgraphs (called *vortices*) and a set of at most h "bad" vertices (called *apex vertices* or *apices*). In this overview, the reader can feel free to ignore the vortices/apices of an almost-embeddable graph and think about it as a graph embedded in a surface of bounded genus. The formal definitions of tree decompositions and almost-embeddable graphs can be found in Section 2.

Intuitively, the Robertson-Seymour decomposition partitions an H-minor-free graph into almost-embeddable "pieces". Since it is known that almost-embeddable graphs admit RCDs [2], a natural idea comes: Can we exploit Robertson-Seymour decomposition to somehow "lift" the RCDs of the almost-embeddable pieces to the H-minor-free graph? To explore this idea, let us consider an H-minor-free graph G and the Robertson-Seymour decomposition (T, β) of G. Since the torsos of (T, β) are h-almost-embeddable, we can compute a p-RCD $Z_1^{(t)}, \ldots, Z_p^{(t)} \subseteq \beta(t)$ for each torso $\mathsf{tor}(t)$ using the results of [2]. Ideally, if these RCDs are *consistent* in the sense that for every vertex $v \in V(G)$ there exists an index $i \in [p]$ such that $v \in Z_i^{(t)}$ for all nodes $t \in V(T)$ whose bag contains v, then we can combine them to obtain a partition $Z_1, \ldots, Z_p$ of $V(G)$ where $Z_i = \bigcup_{t \in V(T)} Z_i^{(t)}$ and hope it to be an RCD of G. (Of course, as we will see later, such a naïve construction of $Z_1, \ldots, Z_p$ does not work. But it can provide us some useful intuition towards a proof of the theorem.) Obtaining consistent RCDs for the torsos is actually not difficult, by properly using the small adhesion size of (T, β). The basic idea is the following. For each node $t \in V(T)$, instead of computing a p-RCD for $\mathsf{tor}(t)$, we only compute a p-RCD for $\mathsf{tor}(t) - \sigma(t)$, and how the vertices in $\sigma(t)$ belong to the p classes is determined by the RCDs on the ancestors of t. The decompositions constructed in this way are consistent. Furthermore, since $|\sigma(t)| \leq h$, given a p-RCD for $\mathsf{tor}(t) - \sigma(t)$, no matter how we assign the vertices in $\sigma(t)$ to the p classes, the resulting decomposition is always a p-RCD for $\mathsf{tor}(t)$. Thus, we obtain consistent RCDs for the torsos, which give us the aforementioned partition $Z_1, \ldots, Z_p$ of $V(G)$. The remaining question is then whether $Z_1, \ldots, Z_p$ is an RCD of G, i.e., whether $\mathbf{tw}(G/(Z_i \setminus Z')) = O(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$.

The only reason for why $Z_1, \ldots, Z_p$ could be an RCD of G is clearly the fact that every Z_i satisfies the RCD condition "locally", i.e., when restricted to a torso. Specifically, let $i \in [p]$ and $Z' \subseteq Z_i$. Set $Z = Z_i \setminus Z'$ for convenience. What we want is $\mathbf{tw}(G/Z) = O(p + |Z'|)$. For each $t \in V(T)$, we have $Z \cap \beta(t) = Z_i^{(t)} \setminus (Z' \cap Z_i^{(t)})$, and thus $\mathbf{tw}(\mathsf{tor}(t)/(Z \cap \beta(t))) = O(p + |Z'|)$. How can we go from the locally bounded treewidth of each $\mathsf{tor}(t)/(Z \cap \beta(t))$ to the global treewidth of G/Z? The following observation (given in Lemma 2.3) seems useful.

> If a graph admits a tree decomposition $\mathcal{T}$ in which each torso is of treewidth at most w, then the graph itself is of treewidth $O(w)$. Indeed, we can "glue" the width-w tree decompositions of the torsos along the given tree decomposition $\mathcal{T}$ to obtain a width-$O(w)$ tree decomposition of the graph (mainly because in a torso the adhesions of the children are cliques).

This observation allows us to go from the treewidth of the torsos to the treewidth of the entire graph. At the first glance, it does not directly apply to our situation here, because we are

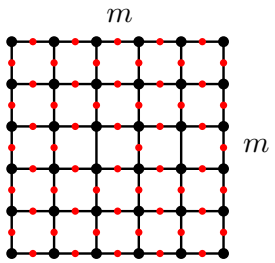

Figure 1: An $m \times m$ grid with subdivided edges. The black vertices are the grid vertices.

working on the *contracted* graphs instead of the original ones: our goal is to lift the treewidth bound from the *contracted* torsos $\mathsf{tor}(t)/(Z \cap \beta(t))$ to the *contracted* graph G/Z. However, it is a well-known fact (given in Fact 2.4) that a tree decomposition of G naturally induces a tree decomposition of the contracted graph G/Z by "contracting" each bag. Specifically, consider the *quotient* map $\pi\colon V(G) \to V(G/Z)$ which maps each vertex of G to the corresponding vertex of G/Z. Set $\beta^*(t) = \pi(\beta(t))$ for all $t \in V(T)$. Then (T, β^*) is a tree decomposition of G/Z. Intuitively, the tree decomposition (T, β^*) should allow us to apply the above observation to lift the treewidth bound from $\mathsf{tor}(t)/(Z \cap \beta(t))$ to G/Z, and eventually show that $Z_1, \ldots, Z_p$ is an RCD of G.

Although the above argument seems promising, a closer inspection reveals that it actually has a fatal issue, which is also the main barrier to proving Theorem 1.2. The issue comes from a somewhat counter-intuitive fact: *the contracted torsos are not identical to the torsos of the contracted graph*! Formally, let $\mathsf{tor}^*(t)$ denote the torso of $t \in V(T)$ in the tree decomposition (T, β^*) of G/Z. Then we have $\mathsf{tor}(t)/(Z \cap \beta(t)) \neq \mathsf{tor}^*(t)$ in general, and even worse, the treewidth of $\mathsf{tor}^*(t)$ can be unbounded even if the treewidth of $\mathsf{tor}(t)/(Z \cap \beta(t))$ is bounded. The main reason for why this happens is the following. The edges of $\mathsf{tor}(t)$ do not necessarily appear in G: some of them are manually added to make the adhesions of the children of t cliques (for convenience, we call them *fake edges*). When we contract G to G/Z, only the edges appearing in G can get contracted. However, when we contract $\mathsf{tor}(t)$ to $\mathsf{tor}(t)/(Z \cap \beta(t))$, all fake edges in $\mathsf{tor}(t)[Z \cap \beta(t)]$ also get contracted. This over-contracting can make $\mathsf{tor}(t)/(Z \cap \beta(t))$ much "smaller" than $\mathsf{tor}^*(t)$. To see an example, suppose G is the graph obtained from an $m \times m$ grid by subdividing each edge into two edges (with an intermediate vertex); see Figure 1. So we have $n = O(m^2)$. Consider a tree decomposition (T, β) of G defined as follows. The tree T consists of a root with some children. The bag of the root contains the m^2 grid vertices. Each child of the root corresponds to an edge e of the grid, whose bag contains the two endpoints of e (two grid vertices) and the intermediate vertex for subdividing e. Now the adhesion of each child of the root consists of the two endpoints of the corresponding grid edge. Let $t \in V(T)$ be the root and $Z \subseteq V(G)$ consist of the grid vertices. Then $\mathsf{tor}(t)$ is an $m \times m$ grid and $\mathsf{tor}(t)/(Z \cap \beta(t)) = \mathsf{tor}(t)/Z$ is a single vertex. However, Z is actually an independent set in G. Thus, $G/Z = G$ and we have $\mathsf{tor}^*(t) = \mathsf{tor}(t)$, which is of treewidth m. In fact, this simple example not only demonstrates that $\mathsf{tor}^*(t)$ can have much higher treewidth than $\mathsf{tor}(t)/(Z \cap \beta(t))$, but also directly shows that a partition $Z_1, \ldots, Z_p$ satisfying the RCD condition locally (i.e., at each torso) may be not a global RCD in general (and thus our previous construction fails). Suppose now we have an RCD $Z_1^{(t)}, \ldots, Z_p^{(t)}$ of $\mathsf{tor}(t)$ for the root $t \in V(T)$. For each child s of t, we arbitrarily assign the only vertex in $\beta(s) \setminus \sigma(s)$ to a class Z_i different from the classes the two vertices in $\sigma(s)$ belong to. Note that, since $|\beta(s)| = 3$, every partition of $\beta(s)$ is actually an RCD of $\mathsf{tor}(s)$. In this way, we obtain a partition $Z_1, \ldots, Z_p$ of $V(G)$ that is an RCD when restricted to every torso. However, each Z_i is an independent set of G, and thus $G/Z_i = G$. So unless $p = \Omega(n)$, $Z_1, \ldots, Z_p$ is not an RCD of G.

The main technical contribution in our proof is to break the above barrier by constructing

$Z_1, \ldots, Z_p$ in a much more sophisticated way. On a high-level, we still follow the idea of constructing RCDs locally for each torso and combine them to obtain $Z_1, \ldots, Z_p$. So in what follows, we always use $Z_1^{(t)}, \ldots, Z_p^{(t)}$ to denote the restriction of $Z_1, \ldots, Z_p$ to $\mathsf{tor}(t)$, i.e., $Z_i^{(t)} := Z_i \cap \beta(t)$. To avoid ambiguity, for a subset $Z \subseteq V(G)$, we use (T, β_Z^*) to denote the tree decomposition of G/Z induced by (T, β), and use $\mathsf{tor}_Z^*(t)$ to denote the torso of a node $t \in V(T)$ in (T, β_Z^*). We have seen that the main reason for why $\mathsf{tor}_Z^*(t)$ can have a much higher treewidth than $\mathsf{tor}(t)/(Z \cap \beta(t))$ is the fake edges in $\mathsf{tor}(t)$. So ideally, if $\mathsf{tor}(t)/(Z \cap \beta(t))$ does not contract any fake edge in $\mathsf{tor}(t)$, we are happy. But this is unlikely, because in the worst case (e.g., the grid example above) every edge in $\mathsf{tor}(t)$ is fake. Fortunately, the fake edges in $\mathsf{tor}(t)$ are not always "uncontractable". Indeed, if a fake edge in $\mathsf{tor}(t)$ has two endpoints lying in the same connected component of $G[Z]$, then we can safely contract it because its two endpoints are also contracted to one vertex in G/Z. Therefore, it is enough to guarantee that $\mathsf{tor}(t)/(Z \cap \beta(t))$ only contracts these safe fake edges, which is equivalent to saying that the two endpoints of every edge of $\mathsf{tor}(t)[Z \cap \beta(t)]$ lie in the same connected component of $G[Z]$. (If this holds, we should be able to somehow relate $\mathbf{tw}(\mathsf{tor}_Z^*(t))$ to $\mathbf{tw}(\mathsf{tor}(t)/(Z \cap \beta(t)))$ and make the previous proof work.) However, this is still too difficult, because we are not dealing with a single set Z. Instead, we have to take care of $Z = Z_i \setminus Z'$ for all $i \in [p]$ and $Z' \subseteq Z_i$, i.e., all subsets of $Z_1, \ldots, Z_p$. Even if the construction of Z_i makes every fake edge uv of $\mathsf{tor}(t)[Z_i \cap \beta(t)]$ have its endpoints u and v in the same connected component of $G[Z_i]$, when a fraction $Z' \subseteq Z_i$ is taken away from Z_i, it can happen that u and v lie in different connected components of $G[Z_i \setminus Z']$ (for example, when Z' is a cut of u and v in $G[Z_i]$) and thus the fake edge uv is no longer safe for contraction when considering $Z = Z_i \setminus Z'$.

The key observation to get rid of this issue is that we do not necessarily need to make *every* fake edge of $\mathsf{tor}(t)[(Z_i \setminus Z') \cap \beta(t)] = \mathsf{tor}(t)[Z_i^{(t)} \setminus Z']$ safe. Indeed, the desired bound for $\mathbf{tw}(\mathsf{tor}_{Z_i \setminus Z'}^*(t))$ and $\mathbf{tw}(G/(Z_i \setminus Z'))$ is $O(p + |Z'|)$, which also depends on $|Z'|$. It turns out that as long as $\mathsf{tor}(t)[Z_i^{(t)} \setminus Z']$ only contains $O(|Z'|)$ "unsafe" fake edges (i.e., those with two endpoints in different connected components of $G[Z_i \setminus Z']$), we can obtain the bound $\mathbf{tw}(\mathsf{tor}_{Z_i \setminus Z'}^*(t)) = O(p + |Z'|)$. To see this, let $Z'' \subseteq (Z_i \setminus Z') \cap \beta(t) = Z_i^{(t)} \setminus Z'$ consist of the endpoints of the $O(|Z'|)$ unsafe fake edges in $\mathsf{tor}(t)[Z_i^{(t)} \setminus Z']$, so we have $|Z''| = O(|Z'|)$. Now every edge in $\mathsf{tor}(t)[Z_i^{(t)} \setminus (Z' \cup Z'')]$ has its two endpoints in the same connected component of $G[Z_i \setminus Z']$, and is thus safe. Since $\mathsf{tor}(t)/(Z_i^{(t)} \setminus (Z' \cup Z''))$ only contracts safe edges, it can be shown that $\mathsf{tor}_{Z_i \setminus Z'}^*(t)$ is (almost[1]) a minor of $\mathsf{tor}(t)/(Z_i^{(t)} \setminus (Z' \cup Z''))$. Thus, we can bound $\mathbf{tw}(\mathsf{tor}_{Z_i \setminus Z'}^*(t))$ using $\mathbf{tw}(\mathsf{tor}(t)/(Z_i^{(t)} \setminus (Z' \cup Z''))$, where the latter is $O(p + |Z' \cup Z''|) = O(p + |Z'|)$, under the assumption that $Z_1^{(t)}, \ldots, Z_p^{(t)}$ is an RCD of $\mathsf{tor}(t)$. To summarize, $Z_1, \ldots, Z_p$ is an RCD of G if the following conditions hold.

(A) $Z_1, \ldots, Z_p$ is an RCD when restricted to every torso, i.e., $Z_1^{(t)}, \ldots, Z_p^{(t)}$ is an RCD of $\mathsf{tor}(t)$.

(B) All but at most $O(|Z'|)$ edges in $\mathsf{tor}(t)[Z_i^{(t)} \setminus Z']$ have both endpoints in the same connected component of $G[Z_i \setminus Z']$, for all $t \in V(T)$, $i \in [p]$, and $Z' \subseteq Z_i$.

Condition (A) naturally holds as long as we insist on constructing $Z_1, \ldots, Z_p$ by combining local RCDs. So the crucial question now is how to satisfy condition (B).

From Global to Local. Condition (B) is not tractable, because it is a global constraint on $Z_1, \ldots, Z_p$. So we need to somehow reduce it to a local constraint on the RCD of each torso. Let us fix a node $t \in V(T)$ and an index $i \in [p]$. To have condition (B), the first thing we need is that (almost) every edge of $\mathsf{tor}(t)[Z_i^{(t)}]$ has its endpoints in the same connected component of

[1] More precisely, $\mathsf{tor}_{Z_i \setminus Z'}^*(t)$ is a minor of a graph obtained from $\mathsf{tor}(t)/(Z_i^{(t)} \setminus (Z' \cup Z''))$ by adding few edges.

$G[Z_i]$, which is the special case $Z' = \emptyset$. But only having this is not enough. We need in addition that loosing k vertices in Z_i only makes $O(k)$ fake edges become "unsafe". To achieve this, our main idea is to require the two endpoints of every edge in $\mathsf{tor}(t)[Z_i^{(t)}]$ to be connected in $G[Z_i]$ by *only* the vertices "strictly below t" in the tree decomposition (T, β), that is, the vertices that only appear in the bags of descendants of t. Formally, for every child s of t, denote by $\gamma(s)$ the union of all bags in T_s, the subtree of T rooted at s. Then our requirement is that for every edge uv of $\mathsf{tor}(t)[Z_i^{(t)}]$, u and v lie in the same connected component of $G[((\gamma(s) \setminus \sigma(s)) \cap Z_i) \cup \{u, v\}]$ for every child s of t such that $u, v \in \sigma(s)$. Note that every non-fake edge trivially satisfies the requirement.

We show that this requirement is sufficient to guarantee condition (B) above for all $Z' \subseteq Z_i$. The definition of tree decompositions guarantees that the sets $\gamma(s) \setminus \sigma(s)$ are disjoint for different children s of t. We say a child s of t is *bad* if Z' contains at least one vertex in $\gamma(s) \setminus \sigma(s)$. By the disjointness of the sets $\gamma(s) \setminus \sigma(s)$, the number of bad children of t is at most $|Z'|$. For convenience, we say a child s of t *witnesses* an edge uv of $\mathsf{tor}(t)[Z_i^{(t)}]$ if $u, v \in \sigma(s)$. Note that each child s of t can witness at most $O(h^2)$ edges, because $|\sigma(s)| \leq h$. Due to our requirement, if an edge uv of $\mathsf{tor}(t)[Z_i^{(t)} \setminus Z']$ violates condition (B) above, i.e., u and v lie in different connected components of $G[Z_i \setminus Z']$, then Z' must contain at least one vertex in $\gamma(s) \setminus \sigma(s)$ for every child s of t satisfying $u, v \in \sigma(s)$, and in particular (u, v) is witnessed by a bad node. Since t has at most $|Z'|$ bad children and each of them can witness $O(h^2)$ edges, the total number of edges of $\mathsf{tor}(t)[Z_i^{(t)} \setminus Z']$ violating condition (B) is bounded by $O(|Z'|)$.

Based on the above argument, it now suffices to construct p-RCDs for each torso of (T, β) such that the partition $Z_1, \ldots, Z_p$ obtained by combining these local RCDs satisfies the aforementioned requirement for all $i \in [p]$ and $t \in V(T)$. However, the current form of our requirement is global, because whether it is satisfied at a node t depends on the construction at not only t but also *all descendants* of t in T. So next, let us transform it to a "local" form which can be checked by looking at the construction at each torso *independently*. We first observe that the following condition implies our previous requirement (which can be shown by a simple induction argument).

> For every edge uv of $\mathsf{tor}(t)[Z_i^{(t)}]$, the endpoints u and v lie in the same connected component of $\mathsf{tor}(s)[(Z_i^{(s)} \setminus \sigma(s)) \cup \{u, v\}]$ for every child s of t such that $u, v \in \sigma(s)$.

The above is actually equivalent to saying that for every child s of t, all $u, v \in \sigma(s) \cap Z_i^{(s)}$ lie in the same connected component of $\mathsf{tor}(s)[(Z_i^{(s)} \setminus \sigma(s)) \cup \{u, v\}]$. Importantly, this condition *only* depends on the construction of $Z_1^{(s)}, \ldots, Z_p^{(s)}$, the RCD of $\mathsf{tor}(s)$. If this condition holds on every node of T, then our previous requirement is also satisfied for every node of T. Therefore, we have our new goal. For every node $t \in V(T)$, we want the following.

(A) $Z_1^{(t)}, \ldots, Z_p^{(t)}$ is an RCD of $\mathsf{tor}(t)$.

(B) For all $i \in [p]$, any two vertices $u, v \in \sigma(t) \cap Z_i^{(t)}$ lie in the same connected component of $\mathsf{tor}(t)[(Z_i^{(t)} \setminus \sigma(t)) \cup \{u, v\}]$.

Now the new conditions above are both local, which allows us to consider each torso individually. We shall construct the RCDs of the torsos in a top-down manner from the root of T to the leaves. Suppose we are at the node $t \in V(T)$ and going to construct the RCD $Z_1^{(t)}, \ldots, Z_p^{(t)}$ of $\mathsf{tor}(t)$, At this point, the RCD at the parent of t has been constructed, so each vertex in $\sigma(t)$ has already been assigned to one of the p classes $Z_1^{(t)}, \ldots, Z_p^{(t)}$. We then compute a p-RCD of $\mathsf{tor}(t) - \sigma(t)$ with an additional property that the i-th class of the RCD "connects" the vertices in $\sigma(t) \cap Z_i^{(t)}$ for all $i \in [p]$. Combining this with the partition of $\sigma(t)$ gives us the desired $Z_1^{(t)}, \ldots, Z_p^{(t)}$ satisfying the above two conditions. As long as such a single step can be achieved, we can eventually construct $Z_1, \ldots, Z_p$. So from now, we can restrict ourselves to one torso.

RCD with a Connectivity Constraint. Before discussing how to construct the desired RCD of $\mathsf{tor}(t) - \sigma(t)$, we need another twist to simplify the task in hand. We notice that constructing an RCD of $\mathsf{tor}(t) - \sigma(t)$ satisfying the additional connectivity property is actually *impossible* if we do not add any assumption on how the vertices in $\sigma(t)$ belong to the p classes. To see this, consider the following simple example where $\mathsf{tor}(t)$ is a star of 5 vertices, one central vertex connecting with 4 other vertices. All vertices are contained in $\sigma(t)$ except the central vertex. In $\sigma(t)$, two vertices belong to the class $Z_1^{(t)}$ and the other two vertices belong to $Z_2^{(t)}$. Now in order to connect the two vertices in $\sigma(t) \cap Z_1^{(t)}$, the RCD of $\mathsf{tor}(t) - \sigma(t)$ must assign the central vertex to $Z_1^{(t)}$. But if this is the case, we fail to connect the two vertices in $\sigma(t) \cap Z_2^{(t)}$. Therefore, in this example, it is impossible to construct the desired RCD of $\mathsf{tor}(t) - \sigma(t)$. In order to fix this issue, we have to somehow guarantee the "connectivity task" that $\sigma(t)$ leaves to us is not too complicated.

Fortunately, using a stronger version of Robertson-Seymour decomposition, we are able to reach a nice situation where only *one* class of vertices in $\sigma(t)$ need to be connected by the RCD of $\mathsf{tor}(t) - \sigma(t)$. This version of Robertson-Seymour decomposition, as stated in [7, 10], guarantees that for every node $t \in V(T)$ and every child s of t, the adhesion $\sigma(s)$ contains at most *three* vertices in the embeddable part of $\mathsf{tor}(t)$ (recall that $\mathsf{tor}(t)$ is an h-almost-embeddable graph consisting of an embeddable part, vortices, and apices). To see why this result helps us, let us consider an ideal case where every torso in the Robertson-Seymour decomposition has no vortices and apices. In this case, every adhesion $\sigma(t)$ is of size at most three, because all vertices in the torso $\mathsf{tor}(t')$ of the parent t' of t are in the embeddable part of $\mathsf{tor}(t')$ and thus $\sigma(t)$ can contain at most three of them. Since $|\sigma(t)| \leq 3$, there can be at most one class $Z_i^{(t)}$ that contains at least two vertices in $\sigma(t)$. Therefore, when constructing the RCD of $\mathsf{tor}(t) - \sigma(t)$, we only need the i-th class to connect the vertices in $\sigma(t) \cap Z_i^{(t)}$. In the general case where the torsos have vortices and apices, the situation becomes complicated. But we can still manage to achieve this, which requires us to construct the RCD of each torso more carefully and then use the "three-vertex" property as above.

Now our goal becomes to construct an RCD of $\mathsf{tor}(t) - \sigma(t)$ in which one class is required to connect the corresponding vertices in $\sigma(t)$. Essentially, we achieve this by proving the following.

Lemma 1.4 (simplified version of Corollary 3.15)**.** *Given a connected h-almost-embeddable graph G, a set $\Phi \subseteq V(G)$ of size c, and a number p, one can compute in polynomial time p disjoint sets $Z_1, \ldots, Z_p \subseteq V(G)$ such that the following two conditions hold*[2]*.*

(1) $\mathbf{tw}(G/(Z_i \setminus Z')) = O_{h,c}(p + |Z'|)$ *for all* $i \in [p]$ *and* $Z' \subseteq Z_i$*.*

(2) Φ *is contained in one connected component of* $G - \bigcup_{i=1}^{p} Z_i$*.*

To see why the above lemma achieves our goal, note that $\mathsf{tor}(t) - \sigma(t)$ is h-almost-embeddable. Furthermore, by constructing the Robertson-Seymour decomposition (T, β) carefully, we can always make $\mathsf{tor}(t) - \sigma(t)$ connected and make every vertex in $\sigma(t)$ have a neighbor in $\beta(t) \setminus \sigma(t)$ in the torso $\mathsf{tor}(t)$; see Lemma 3.24 and Observation 3.25. Without loss of generality, suppose we want to connect the vertices in $\sigma(t) \cap Z_k^{(t)}$ using the k-th class of the RCD of $\mathsf{tor}(t) - \sigma(t)$. For each $v \in \sigma(t) \cap Z_i^{(t)}$, we mark a vertex $v' \in \beta(t) \setminus \sigma(t)$ that is neighboring to v in $\mathsf{tor}(t)$. Let $\Phi \subseteq \beta(t) \setminus \sigma(t)$ be the set of marked vertices. Now apply Lemma 1.4 with $G = \mathsf{tor}(t) - \sigma(t)$ and the set Φ. We then get the disjoint sets $Z_1, \ldots, Z_p \subseteq \beta(t) \setminus \sigma(t)$. The vertices in Z_i are assigned to the class $Z_i^{(t)}$ for all $i \in [p]$. Finally, the vertices in $(\beta(t) \setminus \sigma(t)) \setminus (\bigcup_{i=1}^{p} Z_i)$ are assigned to the class $Z_k^{(t)}$. Then condition (1) of Lemma 1.4 implies that $Z_1^{(t)}, \ldots, Z_p^{(t)}$ is an RCD of $\mathsf{tor}(t)$

[2]The actual result, Corollary 3.15, is more complicated, in which the sets $Z_1, \ldots, Z_p$ have some additional properties and also there is an additional assumption on the almost-embeddable graph G. As we are not going to cover these techinical details in this overview, considering this simplified version should be enough.

and condition (2) guarantees that all $u, v \in \sigma(t) \cap Z_k^{(t)}$ lie in the same connected component of $\mathsf{tor}(s)[(Z_k^{(t)} \setminus \sigma(t)) \cup \{u, v\}]$. Next, we summarize our ideas for proving Lemma 1.4.

Proof Sketch of Lemma 1.4. As mentioned at the beginning, the recent work [2] already gives RCDs for almost-embeddable graphs, that is, it proves the special case of Lemma 1.4 without the set Φ and condition (2). Therefore, it is natural to ask whether one can directly generalize (possibly with slight modifications) the proof in [2] to obtain a proof of Lemma 1.4. Unfortunately, this is not the case. A close look at the proof in [2] reveals that the construction of $Z_1, \ldots, Z_p$ in [2] "inherently" prevents us from having condition (2) of Lemma 1.4. To see this, we need to briefly review how [2] constructs the sets $Z_1, \ldots, Z_p$.

For convenience, we discuss the proof of [2] on a surface-embedded graph instead of an almost-embeddable graph. In fact, the most difficult part of the work [2] also lies in decomposing a surface-embedded graph (specifically, the embeddable part of an almost-embeddable graph), and generalizing to almost-embeddable graphs is somehow easy. In [2], the construction of $Z_1, \ldots, Z_p$ itself is simple, while the analysis of the RCD condition is involved. In the first step, it generalizes the *outerplanar layering* for planar graphs to surface-embedded graphs. Recall that the outerplanar layering partitions the vertices of a planar graph G into layers $L_1, \ldots, L_m \subseteq V(G)$, where L_i consists of the vertices on the outer boundary of $G - \bigcup_{j=1}^{i-1} L_j$. Thus, L_1 is outer boundary of G, L_2 is the outer boundary of $G - L_1$, and so forth. If G is a graph embedded in a surface Σ, the same layering procedure still applies. Indeed, we can fix a reference point $x_0 \in \Sigma$ and define the outer boundary of a Σ-embedded graph as the boundary of the face containing x_0 (assuming the image of the embedding is disjoint from x_0). Given this, we can layer G in the same way as above, by defining L_i to be the vertices on the outer boundary of $G - \bigcup_{j=1}^{i-1} L_j$. We call this generalization *radial layering* for surface-embedded graphs. Now let $L_1, \ldots, L_m$ be the radial layering of G. The analysis in [2] implies the following property of the layers (though not stated explicitly in [2]).

Lemma 1.5. *If $Z = L_{i_1} \cup \cdots \cup L_{i_k}$ where $i_1 < \cdots < i_k$ and $|i_j - i_{j-1}| = O(p)$ for all $j \in [k+1]$ (set $i_0 = 0$ and $i_{k+1} = m$ for convenience), then $\mathbf{tw}(G/(Z \setminus Z')) = O(p + |Z'|)$ for all $Z' \subseteq Z$.*

Then [2] simply defines each set Z_i as the union of equidistant layers with distance $O(p)$ apart, that is, $Z_i = L_{q_i} \cup L_{\Delta+q_i} \cup L_{2\Delta+q_i} \cup \cdots$ for some number $q_i \in [\Delta]$ where $\Delta = O(p)$. If the choices of $q_1, \ldots, q_p$ are different, $Z_1, \ldots, Z_p$ are disjoint. By Lemma 1.5, $Z_1, \ldots, Z_p$ satisfy the RCD condition.

Now let us see what is the difficulty to generalize this construction so that it further satisfies condition (2) of Lemma 1.4. Consider the given set Φ in Lemma 1.4, which is of size c. We want Φ to be contained in the complement $G - \bigcup_{i=1}^{p} Z_i$, and more importantly, Φ has to be in one connected component of $G - \bigcup_{i=1}^{p} Z_i$. To make Φ contained in $G - \bigcup_{i=1}^{p} Z_i$ is easy. We can mark the layers L_i intersecting Φ as "bad" layers. There can be at most c bad layers. When constructing $Z_1, \ldots, Z_p$, we do not include these bad layers. According to Lemma 1.5, we can still guarantee that $Z_1, \ldots, Z_p$ satisfy condition (1) of Lemma 1.4 even if we miss all bad layers, because the constant hidden in the bound of condition (1) can depend on c. However, to further require the vertices in Φ lying in the same connected component of $G - \bigcup_{i=1}^{p} Z_i$ is impossible. Indeed, one can easily see that each layer L_i is a separator in G that separates the layers $L_1, \ldots, L_{i-1}$ from the layers $L_{i+1}, \ldots, L_m$. The layers in $Z_1, \ldots, Z_p$ separate the remaining part of G into many "small" pieces where each piece consists of at most $O(p)$ consecutive layers. This highly-disconnected structure of $G - \bigcup_{i=1}^{p} Z_i$ naturally prevents the vertices in Φ from lying in the same connected component, unless the layers containing Φ are all close to each other.

Based on the above discussion, in order to satisfy both conditions in Lemma 1.4, we need to significantly modify the construction of [2] with various new ideas. As we have seen, in the previous construction, the layers in $Z_1, \ldots, Z_p$ serve as "barriers" that prevent the vertices in Φ from being connected in the complement graph. Therefore, a natural idea is to "break" these

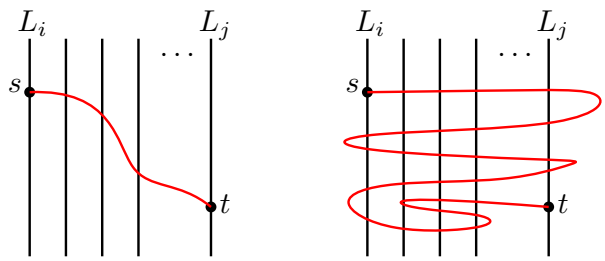

Figure 2: Monotone path vs. back-and-forth path

layers a little bit (i.e., remove some vertices from $Z_1, \ldots, Z_p$) so that the vertices in Φ can go across the barriers to connect with each other. However, by doing this, the layers in each Z_i become broken, and thus the new Z_i might violate the RCD condition, i.e., condition (1) of Lemma 1.4 (as we have fewer vertices to contract). As such, we have to break the layers in some *structured* way such that the vertices in Φ can be connected in the complement graph and *simultaneously* the RCD condition of $Z_1, \ldots, Z_p$ preserves, which is the main challenge in our construction. At this point, it is totally not clear how to achieve this goal. So next, we begin with some simple intuitions.

Let us consider the simplest case where Φ only contains two vertices s and t. In this case, it suffices to properly find an (s,t)-path π in G and then remove the vertices in $Z_1, \ldots, Z_p$ on this path. Since we do not want to lose the RCD condition of $Z_1, \ldots, Z_p$, π should break each layer in $Z_1, \ldots, Z_p$ as "little" as possible. So intuitively, a path that visits the layers $L_1, \ldots, L_m$ *monotonely* (left part of Figure 2) is preferable to a path that goes back and forth across the layers many times (right part of Figure 2). If we do not have any requirement on the embedding of G in Σ, it is not always possible to find a monotone (or mostly monotone) path connecting s and t. Thus, at the beginning (before the radial layering), we need to first modify the embedding of G to a nice one, and do everything with respect to the nice embedding. For surface graphs, a 2-*cell* embedding, in which each face is homeomorphic to a disk, is the nice embedding we need. One can always compute a 2-cell embedding for G in polynomial time as long as the genus of G is a constant. (For almost-embeddable graphs, however, the situation is more involved, as we are not able to arbitrarily change the embedding of the embeddable part because of the vortices. We have to define another type of embedding for which we can safely modify a given embedding of the embeddable part to.) Now if $L_1, \ldots, L_m$ are the radial layers for a 2-cell embedding of G, then one can go from every vertex in a layer L_j to the previous layer L_{j-1} by walking around the boundary of a face in between L_{j-1} and L_j. It turns out that for every vertex $v \in L_j$ and index $i \leq j$, there exists a path from v to (some vertex in) L_i that visits $L_j, L_{j-1}, \ldots, L_i$ monotonely. Suppose $s \in L_i$ and $t \in L_j$ where $i \leq j$. Note that although we can go from t to L_i monotonely, there does not necessarily exist a monotone path from t to s. So the key idea here is to, instead of using one path connecting s and t, construct two monotone paths π_s and π_t, where π_s (resp., π_t) goes from s (resp., t) to L_1. Then we do not include the layer L_1 in any of $Z_1, \ldots, Z_p$ (which is fine according to Lemma 1.5). Because L_1 is the outer boundary of G and the embedding of G is 2-cell, $G[L_1]$ is connected. Therefore, L_1 together with the two paths π_s and π_t connects s and t. The same idea directly genearlizes to the case where Φ has more than two vertices. For each vertex $v \in \Phi$, we try to construct a path π_v which goes monotonely from v to L_1. Now L_1 together with all the paths connect everything in Φ. It then remains to show how to construct these c monotone paths carefully so that they do not break the layers in $Z_1, \ldots, Z_p$ too much.

Since c is treated as a constant in Lemma 1.4, if we are able to construct one path, it is not surprising that the same idea generalizes to constructing c paths. Thus, in what follows, we focus on the case $c = 1$. We have $\Phi = \{v\}$. Since the path π from v to L_1 we are going to construct is monotone, it crosses each layer L_i at most once, that is, after it leaves L_i for

L_{i-1}, it never comes back to L_i. Therefore, we construct each part $L_i \cap \pi$ of π from larger i to smaller i iteratively. When constructing $L_i \cap \pi$, we basically need to consider how to walk in the layer L_i from an arbitrary starting vertex $s \in L_i$ to an "exit" vertex that is adjacent to L_{i-1} so that the walk does not break L_i too much. To this end, we have to figure out what "too much" means. In other words, how much can we break each layer L_i so that $Z_1, \ldots, Z_p$ still satisfy the RCD condition? By checking the proof of Lemma 1.5 in [2], we see that the bound in Lemma 1.5 mainly relies on the following two properties of each layer L_i (where $L_{>i}$ is the union of $L_{i+1}, \ldots, L_m$ and $L_{\leq i}$ is the union of $L_1, \ldots, L_i$).

(I) The neighbors of each connected component of $G[L_{>i}]$ lie in one face of $G[L_{\leq i}]$.

(II) For every $L' \subseteq L_i$, each connected component of $G[L_{>i}]$ is adjacent to $O(|L'|)$ vertices in the graph $G/(L_i \setminus L')$.

Using the above two properties and the fact that the layers in each of $Z_1, \ldots, Z_p$ are only $O(p)$ distance apart, one can finally show that $Z_1, \ldots, Z_p$ satisfies the RCD condition. Now we have the path π that breaks L_i. So we can only include in $Z_1, \ldots, Z_p$ the part $L_i \setminus \pi$. In this case, in order to make the proof work, we need the modified version of condition (II) above: for every $L' \subseteq L_i \setminus \pi$, each connected component of $G[L_{>i}]$ is adjacent to $O(|L'|)$ vertices in the graph $G/((L_i \setminus \pi) \setminus L')$. However, it is easy to see that this is impossible. Consider the simplest case where $L' = \emptyset$. We want each connected component of $G[L_{>i}]$ to have $O(1)$ neighbors in $G/(L_i \setminus \pi)$. But in the worst case, after π reaches L_i, it needs to go inside L_i for a long distance in order to arrive at an exit vertex to leave L_i for L_{i-1}. Therefore, it can happen that π contains a large fraction of L_i in which many vertices can be adjacent to the same connected component C of $G[L_{>i}]$. As these vertices are not contracted in $G/(L_i \setminus \pi)$, the number of neighbors of C in $G/(L_i \setminus \pi)$ can be unbounded.

Going deeper into the proof of [2] shows that the reason for why the above two properties help is basically the following. These two properties allow us to partition G into two parts $G[L_{\leq i}]$ and $G[L_{>i}]$ such that the connection between these two parts becomes "weak" after contracting a large subset of L_i, namely, each connected component of $G[L_{>i}]$ is only adjacent to few vertices in $G[L_{\leq i}]$ which lie in one face of $G[L_{\leq i}]$ (before contraction). Now because the part in $L_i \cap \pi$ cannot be contracted, we are no longer able to have a weak connection between $G[L_{\leq i}]$ and $G[L_{>i}]$. To get rid of this issue, our key idea here is to partition G into two parts in a different way. Specifically, when determining the part of π contained in L_i, we also compute another subset $L_i^+ \subseteq L_i$. Then we partition G into two parts $G[L_{\leq i} \setminus L_i^+]$ and $G[L_{>i} \cup L_i^+]$, that is, we move L_i^+ from the part $G[L_{\leq i}]$ to the part $G[L_{>i}]$. The choice of L_i^+ will guarantee the aforementioned weak connection between the two new parts. Formally, L_i^+ (and π) satisfies the following properties.

(I) The neighbors of each connected component of $G[L_{>i} \cup L_i^+]$ lie in one face of $G[L_{\leq i}]$ (and therefore one face of $G[L_{\leq i} \setminus L_i^+]$).

(II) For every $L' \subseteq L_i \setminus \pi$, each connected component of $G[L_{>i} \cup L_i^+]$ is adjacent to $O(|L'|)$ vertices in the graph $G/((L_i \setminus \{\pi \cup L_i^+\}) \setminus L')$.

Note that such a set L_i^+ does not always exist for an arbitrary path π. Therefore, we have to construct $L_i \cap \pi$ and L_i^+ simultaneously, and both of them need to be constructed carefully. This task is achieved by Lemma 3.6. As this step is technical and requires insights for surface-embedded graphs, we are not going to discuss it in detail. Essentially, once we are able to construct each part $L_i \cap \pi$ of π and the corresponding set L_i^+ satisfying the above two conditions, we can combine them to obtain the desired path π from v to L_1 and show that $\mathbf{tw}(G/((Z_i \setminus \pi) \setminus Z')) = O(p + |Z'|)$ for all $Z' \subseteq Z_i \setminus \pi$. The sets L_i^+ are only used in the analysis of the treewidth bounds, and the analysis is built on the argument in [2]. The same idea generalizes to the case where we need to construct c paths (with a bit more work). With this in hand, we are

finally able to prove Lemma 1.4 for surface-embedded graphs. Of course, for almost-embeddable graphs, there are more technical details to be dealt with, but the basic idea remains the same.

Minimal embeddings of almost-embeddable graphs. In the last part of this overview, we discuss an interesting intermediate result achieved in our proof. As aforementioned, we can modify the embedding of a (connected) surface graph to a 2-cell embedding. However, for an almost-embeddable graph, we cannot require its embeddable part is 2-cell embedded. There are two reasons. First, when we change the embedding of the embeddable part, the structure of the vortices might be lost. Second, even if the graph itself is connected, the subgraph excluding the apices is not necessarily connected, and thus does not admit a 2-cell embedding. In order to make our proof work, we define a variant of 2-cell embedding, which we call *minimal embedding*. Basically, an almost-embeddable graph G whose embeddable part is embedded with a minimal embedding satisfies the following condition. Let f be a face of the embeddable part of G, and $\widetilde{\partial} f$ be the subgraph of G consisting of the boundary of f and all vortices contained in f. Then different connected components of $\widetilde{\partial} f$ belong to different connected components of $G - A$ where A is the set of apices of G. If G does not have vortices and apices (i.e., G is a surface graph), then the above condition is equivalent to saying that the boundary of every face of G is connected. We show that given an almost-embeddable graph G, we can compute (in polynomial time) a new almost-embeddable structure for G in which the embeddable part is embedded with a minimal embedding (which is proved in Lemma 3.5). This result is important for our proof, and might be of independent interest.

2 Preliminaries

Basic Notations. Let G be a graph. We denote by $V(G)$ and $E(G)$ the vertex set and the edge set of G, respectively. For $U \subseteq V(G)$, we denote by $G[U]$ the induced subgraph of G on U and denote by $G - U$ the induced subgraph of G on $V(G) \setminus U$. Also, we denote by $N_G(U)$ the set of vertices in $V(G) \setminus U$ that are adjacent to at least one vertex in U, and write $N_G[U] \coloneqq N_G(U) \cup U$. If $U = \{u\}$, we also write $N_G(u)$ and $N_G[u]$ instead of $N_G(\{u\})$ and $N_G[\{u\}]$, respectively.

Tree Decomposition and Treewidth. A *tree decomposition* of a graph G is a pair (T, β) where T is a tree and $\beta\colon V(T) \to 2^{V(G)}$ maps each node $t \in V(T)$ to a set $\beta(t) \subseteq V(G)$, called the *bag* of t, such that

(i) $\bigcup_{t \in V(T)} \beta(t) = V(G)$,

(ii) for every edge $uv \in E(G)$, there exists $t \in V(T)$ with $u, v \in \beta(t)$, and

(iii) for every $v \in V(G)$, the set $\{t \in V(T) \mid v \in \beta(t)\}$ forms a connected subset in T.

The *width* of (T, β) is $\max_{t \in V(T)} |\beta(t)| - 1$. The *treewidth* of a graph G, denoted by $\mathbf{tw}(G)$, is the minimum width of a tree decomposition of G. It is sometimes more convenient to consider *rooted* trees. Throughout this paper, we always view the underlying tree of a tree decomposition as a rooted tree.

Let (T, β) be a tree decomposition of a graph G. The *adhesion* of a non-root node $t \in V(T)$, denoted by $\sigma(t)$, is defined as $\sigma(t) \coloneqq \beta(t) \cap \beta(t')$ where t' is the parent of t. For convenience, we also define the adhesion of the root of T as the empty set. For each node $t \in V(T)$, we define the *γ-set* of t as $\gamma(t) \coloneqq \bigcup_{s \in V(T_t)} \beta(s)$ where T_t is the subtree of T rooted at t. The *torso* of t, denoted by $\mathsf{tor}(t)$, is the graph obtained from $G[\beta(t)]$ by making $\sigma(s)$ a clique for all children s of t, i.e., adding edges between any two vertices $u, v \in \beta(t)$ such that $u, v \in \sigma(s)$ for some child s of t. The following two facts are well-known.

Fact 2.1. *Let (T, β) be a tree decomposition of G. Then $\{\beta(t) \setminus \sigma(t) \mid t \in V(T)\}$ is a partition of $V(G)$.*

Proof. TOPROVE 1 □

Fact 2.2. *Let (T, β) be a tree decomposition of G. If $U \subseteq V(G)$ is a subset of vertices such that $G[U]$ is connected, then for all $t \in V(T)$, either $\mathsf{tor}(t)[U \cap \beta(t)]$ is connected or every connected component of $\mathsf{tor}(t)[U \cap \beta(t)]$ intersects $\sigma(t)$.*

Proof. TOPROVE 2 □

We shall also need the following useful lemma that relates the treewidth of a graph to the treewidth of the torsos in an arbitrary tree decomposition of the graph.

Lemma 2.3. *If a graph G admits a tree decomposition in which the treewidth of every torso is at most k, then $\mathbf{tw}(G) \leq 2k + 1$.*

Proof. TOPROVE 3 □

Edge Contractions. Edge contraction is a basic operation on graphs. When an edge uv of a graph G is contracted, the two endpoints u and v are merged into one vertex whose neighbors are those in $N_G(\{u, v\})$. For $U \subseteq V(G)$ we write G/U to denote the graph obtained from G by contracting all edges in $G[U]$, or equivalently, contracting every connected component of $G[U]$ to a single vertex. If a graph G' is obtained from G by contracting edges, then each vertex of G' corresponds to a subset of vertices of G and these subsets form a partition of $V(G)$. In this case, there is a naturally defined map $\pi\colon V(G) \to V(G')$ which maps each vertex $v \in V(G)$ to the vertex of G' whose corresponding subset of $V(G)$ contains v. We call π the *quotient map* of the contraction of G to G'. We need the following simple facts.

Fact 2.4 ([3])**.** *Let (T, β) be a tree decomposition of G, and G' be a graph obtained from G by edge contraction with the quotient map $\pi\colon V(G) \to V(G')$. Then (T, β^*) is a tree decomposition of G', where $\beta^*(t) = \pi(\beta(t))$ for all $t \in V(T)$.*

Fact 2.5. *Let G be a graph and G' be a graph obtained from G by edge contraction with the quotient map $\pi\colon V(G) \to V(G')$. Also let $U' \subseteq V(G')$. Then $G'[U']$ is connected if and only if $G[\pi^{-1}(U')]$ is connected.*

Proof. TOPROVE 4 □

Graph Minors. A graph H is a *minor* of a graph G (or G contains H as a minor) if H can be obtained from G by deleting vertices, deleting edges, and contracting edges. A graph G is *H-minor-free* if H is not a minor of G. The following fact gives us an alternative criterion for determining whether a graph is a minor of another graph.

Fact 2.6 ([5])**.** *A graph G contains another graph H as a minor if and only if there exists $U \subseteq V(G)$ and a surjective map $\rho\colon U \to V(G')$ satisfying the following condition: for all $U' \subseteq V(G')$ such that $G'[U']$ is connected, the graph $G[\rho^{-1}(U')]$ is connected.*

Almost-Embeddable Graphs. The class of almost-embeddable graphs is a generalization of the class of bounded-genus graphs, and is related to H-minor-free graphs due to the profound work of Robertson and Seymour [34]. A graph G is *h-almost-embeddable* if it admits an h-almost-embeddable structure described below.

Definition 2.7. *An h-*almost-embeddable structure *of a graph G consists of a set $A \subseteq V(G)$ with $|A| \leq h$, a decomposition $G - A = G_0 \cup G_1 \cup \cdots \cup G_r$ for $r \leq h$, an embedding η of G_0 to a surface Σ of (Euler) genus g with $g \leq h$, r disjoint disks $D_1, \ldots, D_r$ each of which is contained in a face (can possibly intersect the boundary of the face) of the Σ-embedded graph (G_0, η), and r pairs $(\tau_1, \mathcal{P}_1), \ldots, (\tau_r, \mathcal{P}_r)$ such that the following conditions hold for all $i \in [r]$:*

- *$G_1, \ldots, G_r$ are mutually disjoint.*
- *$V(G_0) \cap_\eta D_i = V(G_0) \cap V(G_i)$, where $V(G_0) \cap_\eta D_i$ consists of the vertices in $V(G_0)$ whose image under η is contained in the disk D_i. Set $q_i = |V(G_0) \cap_\eta D_i| = |V(G_0) \cap V(G_i)|$.*
- *$\tau_i = (v_{i,1}, \ldots, v_{i,q_i})$ is a permutation of the vertices in $V(G_0) \cap_\eta D_i$ that is compatible with the clockwise or counterclockwise order along the boundary of D_i.*
- *$\mathcal{P}_i = (\pi_i, \beta)$ is a path decomposition*[3] *of G_i of width at most h where the path $\pi_i = (u_{i,1}, \ldots, u_{i,q_i})$ of the decomposition is of length $q_i - 1$ and satisfies $v_{i,j} \in \beta(u_{i,j})$ for all $j \in [q_i]$.*

Conventionally, we call A the apex set*, G_0 the* embeddable part*, $\eta : G_0 \to \Sigma$ the* partial embedding*, $G_1, \ldots, G_r$ the* vortices *attached to the disks $D_1, \ldots, D_r$. Also, we call each pair $(\tau_i, \mathcal{P}_i)$ the* witness pair *of the vortex G_i, for $i \in [r]$. The* vortex vertices *of G refers to the vertices in $\bigcup_{i=1}^r V(G_i)$.*

3 Proof of Theorem 1.2

In this section, we prove our main theorem, which is restated below.

Theorem 1.2. *Let H be a fixed graph. Given an H-minor-free graph G and an integer p, one can compute in polynomial time a partition of $V(G)$ into p sets $Z_1, \ldots, Z_p$ such that $\mathbf{tw}(G/(Z_i \setminus Z')) = O(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$, where the constant hidden in $O(\cdot)$ depends on H.*

3.1 Useful results for surface-embedded graphs

We begin with introducing some basic notions and results about surface-embedded graphs. For the purpose of our proof, sometimes it is more convenient to consider graphs embedded in a surface with a reference point. Let (Σ, x_0) be a pair where Σ is a connected closed surface, and $x_0 \in \Sigma$ is a reference point on Σ. A (Σ, x_0)-embedded graph refers to a Σ-embedded graph whose image on Σ is disjoint from x_0, i.e., a graph embedded on $\Sigma \setminus \{x_0\}$. For example, plane graphs are exactly (Σ, x_0)-embedded graphs for $\Sigma = \mathbb{S}^2$, where the reference point $x_0 \in \mathbb{S}^2$ is the "point at infinity" of the plane. We typically represent a (Σ, x_0)-embedded graph by a pair (G, η), where G is the graph itself and $\eta \colon G \to \Sigma$ is an embedding of G to Σ whose image avoids x_0, which we call a (Σ, x_0)-embedding. Let (G, η) be a (Σ, x_0)-embedded graph. For any subgraph G' of G, η induces an (Σ, x_0)-embedding of G'; for convenience, we usually use the same notation "η" to denote this subgraph embedding. A *face* of (G, η) refers to (the closure of) a connected component of $\Sigma \setminus \eta(G)$, where $\eta(G)$ is the image of G on Σ under the embedding η. We denote by $F_\eta(G)$ the set of faces of (G, η). With the reference point x_0, we can define the *outer face* of (G, η), which is the (unique) face of (G, η) that contains x_0. The other faces of (G, η) are called *inner faces*. The *boundary* of a face $f \in F_\eta(G)$, denoted by ∂f, is the subgraph of G consisting of all vertices and edges that are incident to f (under the embedding η). Note that f itself is a face in its boundary subgraph $(\partial f, \eta)$, i.e., $f \in F_\eta(\partial f)$. We say a face $f \in F_\eta(G)$ is *singular* if its boundary ∂f is not connected. The following lemma gives useful properties of the boundary subgraph of a face.

[3] Path decomposition is a special case of tree decomposition in which the tree is a path.

Lemma 3.1. *Let $o \in F_\eta(G)$ be a face of a (Σ, x_0)-embedded graph (G, η), where g is the genus of Σ. Then $(\partial o, \eta)$ has $O(g)$ singular faces. Furthermore, every face $f \in F_\eta(\partial o) \setminus \{o\}$ satisfies the following properties.*

(1) ∂f has $O(g)$ connected components.

(2) The degree of every vertex of ∂f is $O(g)$.

(3) There are $O(g)$ vertices of ∂f whose degrees are at least 3.

Proof. TOPROVE 5 □

The *vertex-face incidence* (VFI) graph of (G, η) is a bipartite graph with vertex set $V(G) \cup F_\eta(G)$ and edges connecting every pair $(v, f) \in V(G) \times F_\eta(G)$ such that v is incident to f (or equivalently, v is a vertex in ∂f). It is known that the VFI graph of (G, η) is always connected [2]. Let G^* be the VFI graph of (G, η). A *vertex-face alternating* (VFA) path in (G, η) refers to a path in G^*; it is called "vertex-face alternating" because vertices and faces of (G, η) appear alternately on the path. For two vertices $v, v' \in V(G)$, the *vertex-face distance* between v and v' in (G, η) is defined as the shortest-path distance between v and v' in G^*. In the same way, we can define the *vertex-face distance* between a vertex and a face of (G, η), or between two faces of (G, η). The *vertex-face diameter* of (G, η), denoted by $\mathsf{diam}^*(G, \eta)$, is the maximum vertex-face distance between vertices/faces of (G, η), or equivalently, the graph diameter of the VFI graph of (G, η). Following from the classical work [15] of Eppstein, it is known that $\mathbf{tw}(G) = O_g(\mathsf{diam}^*(G, \eta))$. Recently, Bandyapadhyay et al. [2] generalized this result in the following way. Let $w \colon F_\eta(G) \to \mathbb{N}$ be a weight function on the faces of (G, η). Consider a simple path $\pi = (a_0, a_1, \ldots, a_m)$ in G^*, and write $F_\pi = F_\eta(G) \cap \{a_0, a_1, \ldots, a_m\}$. The *cost* of π under the weight function w is defined as $m + \sum_{f \in F_\pi} w(f)$, i.e., the length of π plus the total weights of the faces that π goes through. We define the *w-weighted vertex-face distance* between vertices/faces $a, a' \in V(G) \cup F_\eta(G)$ in (G, η) as the minimum cost of a path connecting a and a' in G^* under the weight function w. Note that w-weighted vertex-face distances satisfy the triangle inequality, although they do not necessarily form a metric because the w-weighted vertex-face distance from a face $f \in F_\eta(G)$ to itself is $w(f)$ rather than 0. The *w-weighted vertex-face diameter* of (G, η), denoted by $\mathsf{diam}^*_w(G, \eta)$, is the maximum w-weighted vertex-face distance between vertices/faces of (G, η). Clearly, when w is the zero function, the w-weighted vertex-face distance/diameter coincides with the "unweighted" vertex-face distance/diameter defined before. Bandyapadhyay et al. [2] generalizes the bound $\mathbf{tw}(G) = O_g(\mathsf{diam}^*(G, \eta))$ by proving the following lemma.

Lemma 3.2 ([2]). *Let (G, η) be a (Σ, x_0)-embedded graph where the genus of the surface Σ is g and $\kappa : F_\eta(G) \to 2^{V(G)}$ be a map satisfying $\kappa(f) \subseteq V(\partial f)$. Then we have $\mathbf{tw}(G^\kappa) = O(\mathsf{diam}^*_{w_\kappa}(G, \eta))$, where G^κ is the graph obtained from G by making $\kappa(f)$ a clique for all $f \in F_\eta(G)$ and $w_\kappa : F_\eta(G) \to \mathbb{N}$ is a weighted function on the faces of (G, η) defined as $w_\kappa(f) = |\kappa(f)|$.*

The *radial layering* of a (Σ, x_0)-embedded graph (G, η) is a partition $L_1, \ldots, L_m$ of $V(G)$ where L_i consists of the vertices which have vertex-face distance $2i-1$ to the outer face of (G, η). Alternatively, L_i can be defined as the vertices incident to the outer face of $(G - \bigcup_{j=1}^{i-1} L_j, \eta)$. The notion of radial layering generalizes the well-known outerplanar layering of plane graphs. We have the following simple observation for the radial layering.

Fact 3.3. *Let $L_1, \ldots, L_m$ be the radial layering of a (Σ, x_0)-embedded graph (G, η). Then, for any $f \in F_\eta(G)$, $V(\partial f) \subseteq L_{i-1} \cup L_i$ for some $i \in [m]$. Also, $N_G[L_i] \subseteq L_{i-1} \cup L_i \cup L_{i+1}$ for all $i \in [m]$.*

Proof. TOPROVE 6 □

For notational convenience, if $L_1, \ldots, L_m$ are the radial layers, then we write $L_{\geq a} = \bigcup_{i=a}^{m} L_i$, $L_{>a} = \bigcup_{i=a+1}^{m} L_i$, $L_{\leq a} = \bigcup_{i=1}^{a} L_i$, $L_{<a} = \bigcup_{i=1}^{a-1} L_i$, and $L_a^b = \bigcup_{i=a}^{b} L_i$. We shall use these notations throughout this section. Also, we shall assume $L_i = \emptyset$ for all indices $i \leq 0$ and $i > m$.

An *extension* of a (Σ, x_0)-embedded graph (G, η) is another (Σ, x_0)-embedded graph (G', η') where G' is a supergraph of G and $(G, \eta') = (G, \eta)$, i.e., the restriction of η' to G is the same as η. We say the extension is *outer-preserving* if the images of all vertices in $V(G')\backslash V(G)$ and all edges in $E(G')\backslash E(G)$ under η' are in the inner faces of (G, η). Thus, an outer-preserving extension has the same outer face and the same outer-face boundary as the original graph.

An embedding $\eta\colon G \to \Sigma$ is *minimal* if for every face $o \in F_\eta(G)$, the boundary subgraph $(\partial o, \eta)$ satisfies the following condition: $V(\partial f)$ lies in one connected component of ∂o for every $f \in F_\eta(\partial o)\backslash\{o\}$. We have the following simple observation.

Fact 3.4. *Let $\eta\colon G \to \Sigma$ be a minimal embedding. If G' is an induced subgraph of G that is the disjoint union of some connected components of G, then the induced embedding $\eta\colon G' \to \Sigma$ is also a minimal embedding. Also, if G' is a disjoint union of G and some isolated vertices and $\eta'\colon G' \to \Sigma$ is an extension of η, then η' is a minimal embedding.*

Proof. TOPROVE 7 □

Finally, we need to establish an important lemma which allows us the transform a given almost-embeddable structure of a graph to another almost-embeddable structure in which the partial embedding is minimal.

Lemma 3.5. *Given a graph G with an h-almost-embeddable structure, one can compute in polynomial time a new $O_h(1)$-almost-embeddable structure of G in which the partial embedding is minimal. Furthermore, every apex in the given structure is still an apex in the new structure, while every vortex vertex in the given structure is either a vortex vertex or an apex in the new structure.*

Proof. TOPROVE 8 □

3.2 A technical lemma

In this section, we prove the following key lemma, which serves as a technical core in our proof.

Lemma 3.6. *Let (G, η) be a (Σ, x_0)-embedded graph, and $L_1, \ldots, L_m$ be the radial layering of (G, η). Given any $t \in [m]$ with $t \geq 2$ and a set $\Phi \subseteq L_t$ of size c, if all faces of (G, η) incident to L_t are non-singular, then one can compute in polynomial time a subset $X \subseteq L_t$ containing Φ and another subset $L_t^+ \subseteq L_t\backslash N_G(L_{t-1})$ satisfying the following four conditions (where o_t is the outer face of $(G[L_{\geq t}], \eta)$ and g is the genus of Σ).*

(1) Every connected component of $G[X]$ intersects Φ and is neighboring to L_{t-1}.

(2) For any outer-preserving extension (G', η') of $(\partial o_t, \eta)$, every connected component C of $G' - (L_t\backslash L_t^+)$ satisfies $N_{G'}(C) \subseteq V(\partial f)$ for some $f \in F_\eta(\partial o_t)\backslash\{o_t\}$.

(3) $V(\partial f) \cap (X\backslash L_t^+) = O_{g,c}(1)$ for all $f \in F_\eta(\partial o_t)\backslash\{o_t\}$.

(4) $\partial f - L_t^+$ only has $O_{g,c}(1)$ connected components for all $f \in F_\eta(\partial o_t)\backslash\{o_t\}$.

Let (G, η) be a connected (Σ, x_0)-embedded graph satisfying the properties of the lemma, where Σ is a surface of genus g. Let $L_1, \ldots, L_m$ be the radial layering of (G, η). Fix an index $t \in [m]$ with $t \geq 2$, and assume all faces of (G, η) incident to L_t are non-singular. Denote by o_t the outer face of $(G[L_{\geq t}], \eta)$.

We say a vertex in $L_t = V(\partial o_t)$ is an *exit vertex* if it is adjacent to a vertex in L_{t-1}. In the graph $(\partial o_t, \eta)$, each edge e is incident to two (not necessarily different) faces in $F_\eta(\partial o_t)$, which

correspond to the two "sides" of (the embedding of) e. Note that one of these two faces must be o_t because all edges in $(\partial o_t, \eta)$ are incident to o_t, while the other one can be either o_t or an inner face of $(\partial o_t, \eta)$, which we denote by $f(e)$.

Let $\mathcal{P}$ be the set of all pairs (f, v) where $f \in F_\eta(\partial o_t)\backslash\{o_t\}$ is an inner face of $(\partial o_t, \eta)$ and $v \in V(\partial f)$. In what follows, we classify the pairs in $\mathcal{P}$ into three classes: *singular pairs*, *critical pairs*, and *normal pairs*. A pair $(f, v) \in \mathcal{P}$ is *singular* if (at least) one of the following conditions holds.

(i) There is a path in ∂o_t from v to another vertex of ∂f that does not use any edge of ∂f, i.e., the path only uses the edges in $E(\partial o_t)\backslash E(\partial f)$.

(ii) There is a path in ∂o_t from v to a vertex of $\partial f'$ for some singular face $f' \in F_\eta(\partial o_t)\backslash\{o_t, f\}$ that only uses the edges in $E(\partial o_t)\backslash E(\partial f)$.

A pair $(f, v) \in \mathcal{P}$ is *critical* if it is not singular and there exists a path in ∂o_t from v to an exit vertex that only uses the edges in $E(\partial o_t)\backslash E(\partial f)$. In particular, if (f, v) is not singular and v is an exit, then (f, v) is critical. Finally, a pair in $\mathcal{P}$ is *normal* if it is neither singular nor critical. We first observe that there are only a constant number of singular pairs in $\mathcal{P}$.

Observation 3.7. *For each inner face $f \in F_\eta(\partial o_t)$, there are only $O_g(1)$ vertices $v \in V(\partial f)$ such that the pair $(f, v) \in \mathcal{P}$ is singular.*

Proof. TOPROVE 9 □

Next, we define a notion called *legal paths*. Consider a path $\pi = (v_0, v_1, \ldots, v_r)$ in ∂o_t and write $e_i = v_{i-1}v_i$ for $i \in [r]$. We say π is *legal* if for all $i \in [r-1]$ such that $(f(e_i), v_i)$ is a critical pair in $\mathcal{P}$, we have $f(e_{i+1}) \neq f(e_i)$. We have the following observation.

Observation 3.8. *Let π be a legal path in ∂o_t and denote by $V_\pi \subseteq V(\partial o_t)$ the vertices on π. Then for each face $f \in F_\eta(\partial o_t)\backslash\{o_t\}$, the graph $\partial f[V_\pi \cap V(\partial f)]$ has $O_g(1)$ connected components and there are at most two vertices $v \in V_\pi \cap V(\partial f)$ such that (f, v) is a critical pair.*

Proof. TOPROVE 10 □

Observation 3.9. *For any vertex $v \in L_t$, there exists a legal path in ∂o_t from v to an exit vertex of L_t, which can be computed in polynomial time in $|L_t|$.*

Proof. TOPROVE 11 □

Let $\Phi \subseteq L_t$ be the given set of size c as in Lemma 3.6. Suppose $\Phi = \{v_1, \ldots, v_c\}$. For each $i \in [c]$, we use Observation 3.9 to compute a legal path π_i in ∂o_t from v_i to an exit vertex of L_t. Then we define $X \subseteq L_t$ as the set of all vertices on the paths $\pi_1, \ldots, \pi_c$. Clearly, $v_1, \ldots, v_c \in X$ and every connected component of $G[X]$ is neighboring to L_{t-1} because each of the paths $\pi_1, \ldots, \pi_c$ contains an exit vertex of L_t. In particular, condition (1) of Lemma 3.6 holds. We then define the set L_t^+ as follows. For every pair $(f, v) \in \mathcal{P}$, we define $Y_{f,v} \subseteq L_t$ as the set of vertices that is reachable from v using only the edges in $E(\partial o_t)\backslash E(\partial f)$. Then L_t^+ is defined by the union of all $Y_{f,v}$ where (f, v) is a normal pair and $v \in X$. By definition, if (f, v) is a normal pair, then none of the vertices in $Y_{f,v}$ is an exit vertex, and therefore $L_t^+ \subseteq L_t\backslash N_G(L_{t-1})$. We need to verify that conditions (2)-(4) of Lemma 3.6 are satisfied.

Verifying Condition (2) of Lemma 3.6. To see condition (2), we first observe some basic properties of the sets $Y_{f,v}$ defined above for normal pairs $(f, v) \in \mathcal{P}$.

Observation 3.10. *Let $(f, v) \in \mathcal{P}$ be a normal pair. Then the following properties hold.*

(a) $Y_{f,v}$ does not contain any exit vertex of L_t.

(b) $Y_{f,v} \cap V(\partial f) = \{v\}$.

(c) For all $f' \in F_\eta(o_t)\backslash\{o_t, f\}$*, either* $Y_{f,v} \cap V(\partial f') = \emptyset$ *or* $V(\partial f') \subseteq Y_{f,v}$.

(d) For any outer-preserving extension (G', η') *of* $(\partial o_t, \eta)$*, we have* $N_{G'}(Y_{f,v}) \cap L_t \subseteq V(\partial f)$ *and* $N_{G'}(Y_{f,v}\backslash\{v\}) \cap L_t = \{v\}$.

Proof. TOPROVE 12 □

Let (G', η') be an outer-preserving extension of $(\partial o_t, \eta)$. For every connected component C of $G' - (L_t\backslash L_t^+)$, we have $N_{G'}(C) \subseteq L_t\backslash L_t^+$. We claim that $N_{G'}(C) \subseteq \bigcup_{f\in F_\eta(\partial o_t)\backslash\{o_t\}} V(\partial f)$. As $N_{G'}(C) \subseteq L_t$ and $C \subseteq L_t^+ \cup (V(G')\backslash L_t)$, we have $N_{G'}(C) \subseteq (N_{G'}(L_t^+) \cap L_t) \cup (N_{G'}(V(G')\backslash L_t) \cap L_t)$. Recall that L_t^+ is the union of all $Y_{f,v}$ where (f, v) is a normal pair. Thus, by Observation 3.10(d), we have $N_{G'}(L_t^+) \cap L_t \subseteq \bigcup_{f\in F_\eta(\partial o_t)\backslash\{o_t\}} V(\partial f)$. For any vertex $v \in V(G')\backslash L_t$, the image of v under η' lies in the interior of some face $f \in F_\eta(\partial o_t)\backslash\{o_t\}$, and thus $N_{G'}(\{v\}) \cap L_t \subseteq V(\partial f)$. It then follows that $N_{G'}(C) \subseteq \bigcup_{f\in F_\eta(\partial o_t)\backslash\{o_t\}} V(\partial f)$.

Based on this, we further show that $N_{G'}(C) \subseteq V(\partial f)$ for some $f \in F_\eta(\partial o_t)\backslash\{o_t\}$. If $N_G(C) = \emptyset$, we are done. Also, if $C \cap L_t^+ = \emptyset$, then the images of all vertices in C under η' must lie in one face $f \in F_\eta(\partial o_t)\backslash\{o_t\}$ and thus $N_G(C) \subseteq V(\partial f)$. So suppose $N_G(C) \neq \emptyset$ and $C \cap L_t^+ \neq \emptyset$. We observe the following.

Observation 3.11. *There exists a face* $f^* \in F_\eta(\partial o_t)\backslash\{o_t\}$ *such that* $V(\partial f^*) \nsubseteq C$ *and* $V(\partial f^*) \cap C \neq \emptyset$.

Proof. TOPROVE 13 □

Let $f^* \in F_\eta(\partial o_t)\backslash\{o_t\}$ be the face in the above observation. We show that $N_{G'}(C) \subseteq V(\partial f^*)$, which implies (2) of Lemma 3.6. Define $C^* = V(\partial f^*) \cap C$, which is nonempty by the construction of f^*. Note that $C^* \subseteq L_t^+$, because $V(\partial f^*) \subseteq L_t$ and $C \subseteq V(G')\backslash(L_t\backslash L_t^+)$. We claim that (f^*, u) is normal for all $u \in C^*$. Consider a vertex $u \in C^* \subseteq L_t^+$. Then $u \in Y_{f,v}$ for some normal pair (f, v). If $f \neq f^*$, by Observation 3.10(c), either $Y_{f,v} \cap V(\partial f^*) = \emptyset$ or $V(\partial f^*) \subseteq Y_{f,v}$. The former is false because $u \in Y_{f,v} \cap V(\partial f^*)$. The latter is false because $V(\partial f^*) \nsubseteq C$ (note that $Y_{f,v} \subseteq C$ as $u \in C$ and $G[Y_{f,v}]$ is connected). Thus, $f = f^*$. Then by Observation 3.10(b), $Y_{f,v} \cap V(\partial f^*) = Y_{f,v} \cap V(\partial f) = \{v\}$, which implies $u = v$. It follows that $(f^*, u) = (f, v)$ is a normal pair. Now let C' be the union of $\bigcup_{u\in C^*} Y_{f^*,u}$ and all connected components of $G' - L_t$ that are neighboring to $\bigcup_{u\in C^*} Y_{f^*,u}$ in G'. Clearly, we have $C' \subseteq C$, because $\bigcup_{u\in C^*} Y_{f^*,u} \subseteq C$ and thus any connected component of $G' - L_t$ neighboring to $\bigcup_{u\in C^*} Y_{f^*,u}$ should be contained in C. The following observation implies $N_{G'}(C) \subseteq V(\partial f^*)$.

Observation 3.12. $N_{G'}(C') \subseteq V(\partial f^*)$ *and* $C' = C$.

Proof. TOPROVE 14 □

Verifying Condition (3) of Lemma 3.6. Let $f \in F_\eta(\partial o_t)\backslash\{o_t\}$ be a face. By the construction of L_t^+, we have $v \in L_t^+$ for all vertices $v \in \partial(f) \cap X$ such that (f, v) is normal. Therefore, for every $v \in V(\partial f) \cap (X\backslash L_t^+)$, either (f, v) is singular or (f, v) is critical. By Observation 3.7, there are only $O_g(1)$ vertices $v \in V(\partial f)$ such that (f, v) is singular. So it suffices to bound the number of vertices $v \in V(\partial f) \cap (X\backslash L_t^+)$ such that (f, v) is critical. Recall that X consists of the vertices on the legal paths $\pi_1, \ldots, \pi_c$. By Observation 3.8, each legal path contains at most two vertices $v \in V(\partial f)$ such that (f, v) is critical. So the number of vertices $v \in V(\partial f) \cap (X\backslash L_t^+)$ such that (f, v) is critical is at most $2c$. This further implies $V(\partial f) \cap (X\backslash L_t^+) = O_{g,c}(1)$.

Verifying Condition (4) of Lemma 3.6. Let $f \in F_\eta(\partial o_t)\backslash\{o_t\}$ be a face. We first claim that either $V(\partial f) \subseteq L_t^+$ or $V(\partial f) \cap L_t^+ \subseteq X$. Indeed, if there exists a normal pair (f', v') with $f' \neq f$ and $v' \in X$ such that $V(\partial f) \subseteq Y_{f',v'}$, then $V(\partial f) \subseteq L_t^+$. Otherwise, by Observation 3.10(c), $Y_{f',v'} \cap V(\partial f) = \emptyset$ for all normal pair (f', v') with $f' \neq f$ and $v' \in X$. This implies that any vertex $u \in V(\partial f) \cap L_t^+$ must belong to $Y_{f,v}$ for some normal pair (f, v) with $v \in X$. But by Observation 3.10(b), we have $Y_{f,v} \cap V(\partial f) = \{v\}$ and thus $u = v \in X$.

If $V(\partial f) \subseteq L_t^+$, then $V(\partial f)\backslash L_t^+ = \emptyset$ and we are done. If $V(\partial f) \cap L_t^+ \subseteq X$, then $V(\partial f)\backslash L_t^+ = (V(\partial f)\backslash X) \cup (V(\partial f) \cap (X\backslash L_t^+))$. As shown above, $|V(\partial f) \cap (X\backslash L_t^+)| = O_{g,c}(1)$. Thus, to show (4) of Lemma 3.6, we only need to show $\partial f - X$ has $O_{g,c}(1)$ connected components. Since X consists of the paths $\pi_1, \ldots, \pi_c$ and by Observation 3.8 the intersection of each π_i and $V(\partial f)$ has $O_g(1)$ connected components (in ∂f), we know that $\partial f[V(\partial f) \cap X]$ has $O_{g,c}(1)$ connected components. By (2) and (3) of Lemma 3.1, the maximum degree of ∂f is $O(g)$ and there are $O(g)$ vertices in ∂f of degree at least 3. The following fact then implies that $\partial f - (V(\partial f) \cap X)$ has $O_{g,c}(1)$ connected components (setting $G = \partial f$ and $V = V(\partial f) \cap X$).

Fact 3.13. *Let G be a graph of maximum degree α in which there are β vertices of degree at least 3. Then for every $V \subseteq V(G)$, the graph $G - V$ has at most $O_{\alpha,\beta,\gamma}(1)$ more connected components than G, where γ is the number of connected components of $G[V]$. In particular, deleting γ vertices from G can increase the number of connected components by at most $O_{\alpha,\beta,\gamma}(1)$.*

Proof. TOPROVE 15 □

3.3 Decomposing almost-embeddable graphs

In this section, based on Lemma 3.6, we prove a result on almost-embeddable graphs (Corollary 3.15 below), which roughly states that one can decompose a connected almost-embeddable graph into $p+1$ parts in which the first p parts satisfy the (robust) "bounded-treewidth contraction" property and the last part connects a small set of given vertices. We begin with apex-free almost-embeddable graphs, and establish the following decomposition lemma.

Lemma 3.14. *Given a connected graph G with an apex-free h-almost-embeddable structure in which the partial embedding is minimal, a number p, and a set $\Phi \subseteq V(G)$ of size c, one can compute in polynomial time p disjoint sets $Z_1, \ldots, Z_p \subseteq V(G)\backslash N_G[\mathsf{Vort} \cup \Phi]$ satisfying the following conditions (where $\mathsf{Vort} \subseteq V(G)$ consists of the vortex vertices in G).*

(1) $\mathbf{tw}(G/(Z_i\backslash Z')) = O_{h,c}(p + |Z'|)$ *for all $i \in [p]$ and $Z' \subseteq Z_i$.*

(2) Φ is contained in one connected component of $G - \bigcup_{i=1}^p Z_i$.

Before proving the above lemma, we first observe that it implies a decomposition for (general) almost-embeddable graphs satisfying similar conditions, which is the following corollary.

Corollary 3.15. *Given a connected graph G with an h-almost-embeddable structure in which the partial embedding is minimal, a number p, and a set $\Phi \subseteq V(G)$ of size c, one can compute in polynomial time p disjoint sets $Z_1, \ldots, Z_p \subseteq V(G)\backslash(N_G[\mathsf{Vort}] \cup A)$ satisfying the following conditions (where $\mathsf{Vort} \subseteq V(G)$ consists of the vortex vertices and $A \subseteq V(G)$ is the apex set of G).*

(1) $\mathbf{tw}(G/(Z_i\backslash Z')) = O_{h,c}(p + |Z'|)$ *for all $i \in [p]$ and $Z' \subseteq Z_i$.*

(2) Φ is contained in one connected component of G', where G' denotes the graph obtained from $G - \bigcup_{i=1}^p Z_i$ by deleting all edges (a, v) where $a \in A$ and $v \in (\bigcup_{i=1}^p N_G(Z_i))\backslash A$.

(3) $Z_i \cap N_G[(\Phi\backslash A)] = \emptyset$ for all $i \in [p]$.

Proof. TOPROVE 16 □

The rest of this section is dedicated to proving Lemma 3.14. Let G be the graph as in the lemma. Suppose G_0 is the embeddable part of G and $\eta\colon G_0 \to \Sigma$ is the minimal partial embedding where Σ is a surface of genus $g \le h$. We view (G_0, η) as a (Σ, x_0)-embedded graph by (arbitrarily) picking a reference point $x_0 \in \Sigma$ not in the image of η. Denote by $\mathsf{Vort} \subseteq V(G)$ the set of vortex vertices of G. Also, let p and Φ be as in the lemma. For each face $f \in F_\eta(G_0)$, we denote by $\tilde{\partial} f$ the union of ∂f and the vortices of G contained in f. Clearly, if f is not a vortex face, then $\tilde{\partial} f = \partial f$.

Observation 3.16. *$\tilde{\partial} f$ is connected for all $f \in F_\eta(G_0)$.*

Proof. TOPROVE 17 □

In order to construct the sets $Z_1, \dots, Z_p$, consider the radial layering $L_1, \dots, L_m$ of (G_0, η). We classify these layers as bad and good layers as follows. The first layer L_1 is always bad. For $i \ge 2$, the layer L_i is bad if $N_G[\Phi] \cap L_i \ne \emptyset$ or $N_G[\mathsf{Vort}] \cap L_i \ne \emptyset$, and is good otherwise. By Fact 3.3, if $N_G[\Phi] \cap L_i \ne \emptyset$, then $L_{i-1} \cup L_i \cup L_{i+1}$ contains at least one vertex in Φ. Therefore, there are at most $3|\Phi| = 3c$ indices $i \in [m]$ such that $N_G[\Phi] \cap L_i \ne \emptyset$. Again by Fact 3.3, each vortex face $f \in F_\eta(G_0)$ is incident to at most two consecutive radial layers L_i, L_{i+1}, and hence the vertices in the vortices contained in f are neighboring to at most four radial layers $L_{i-1}, L_i, L_{i+1}, L_{i+2}$. Since the number of vortex faces is at most h, there are at most $4h$ indices $i \in [m]$ such that $N_G[\mathsf{Vort}] \cap L_i \ne \emptyset$. It follows that the number of bad layers is at most $3c + 4h + 1$.

Before defining $Z_1, \dots, Z_p$, we first figure out which vertices of $V(G)$ are not included in any Z_i and guarantee that these vertices connect Φ. To this end, we shall iteratively define a sequence of sets $X_m, \dots, X_1$ of vertices in G, where $X_i \subseteq L_i$. Suppose $X_{t+1}, \dots, X_m$ have already been constructed, and we now construct X_t. If L_t is a bad layer, we simply set $X_t = L_t$. Otherwise, L_t is a good layer and we construct X_t as follows. Let $\mathcal{C}$ denote the set of connected components of $G[\mathsf{Vort} \cup (\bigcup_{i=t+1}^m X_i)]$. For $C \in \mathcal{C}$ with $C \cap \Phi \ne \emptyset$ and $N_G(C) \cap L_t \ne \emptyset$, we pick a vertex $v_C \in N_G(C) \cap L_t$. Set $\Phi_t = \{v_C \mid C \in \mathcal{C}\}$. Observe that $|\Phi_t| \le |\Phi| = c$. Indeed, since L_t is good, $L_t \cap \mathsf{Vort} = \emptyset$ and thus $L_t \cap (\mathsf{Vort} \cup (\bigcup_{j=i+1}^m X_j)) = \emptyset$. Thus, the number of components $C \in \mathcal{C}$ with $C \cap \Phi \ne \emptyset$ is at most $|\Phi \backslash L_t|$, which implies $|\Phi_t| \le |\Phi \backslash L_t| \le |\Phi| = c$. As L_t is a good layer, all faces of (G_0, η) incident to L_t do not contain vortices. Since η is a minimal embedding, by Observation 3.16, this further implies that every face of (G_0, η) incident to L_t has a connected boundary. Thus, we can apply Lemma 3.6 on G_0 with $\Phi = \Phi_t$ to obtain a set $X \subseteq L_t$ containing Φ_t and another set $L_t^+ \subseteq L_t$ satisfying the properties in the lemma. We then set $X_t = X$.

By our construction, it is clear that $\Phi \subseteq \mathsf{Vort} \cup (\bigcup_{i=1}^m X_i)$. We then prove that all vertices in Φ are contained in the same connected component of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^m X_i)]$. This is achieved by two steps. In the first step, we show that every connected component of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^m X_i)]$ which contains at least one vertex in Φ must intersect L_1. Note that L_1 is a bad layer and hence $L_1 = X_1 \subseteq \mathsf{Vort} \cup (\bigcup_{i=1}^m X_i)$. In the second step, we then show that all vertices in L_1 lie in the same connected component of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^m X_i)]$.

To do the first step, consider a connected component C of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^m X_i)]$ which contains at least one vertex in Φ. We want to show $L_1 \cap C \ne \emptyset$. Let t be the smallest index such that $L_t \cap C \ne \emptyset$. Assume $t > 1$ for a contradiction.

Observation 3.17. *$N_G(C) \cap L_{t-1} \ne \emptyset$.*

Proof. TOPROVE 18 □

Based on the observation $N_G(C) \cap L_{t-1} \ne \emptyset$, we deduce a contradiction as follows. If L_{t-1} is a bad layer, then $X_{t-1} = L_{t-1}$ and thus $N_G(C) \cap X_{t-1} \ne \emptyset$, which contradicts with the fact that C is a connected component of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^m X_i)]$. If L_{t-1} is a good layer, recall the set Φ_{t-1} we define when constructing X_{t-1}. As C is disjoint from $L_1, \dots, L_{t-1}$, C is in fact a connected

component of $G[\mathsf{Vort} \cup (\bigcup_{i=t}^{m} X_i)]$. Also, we have $C \cap \Phi \neq \emptyset$ by assumption and $N_G(C) \cap L_{t-1} \neq \emptyset$ as shown above. Therefore, Φ_{t-1} includes a vertex $v_C \in N_G(C) \cap L_{t-1}$. By construction, we have $\Phi_{t-1} \subseteq X_{t-1}$ and thus $v_C \in X_{t-1}$. This implies $N_G(C) \cap X_{t-1} \neq \emptyset$, which contradicts with the fact that C is a connected component of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^{m} X_i)]$. In both cases, we have contradictions, so the assumption $t > 1$ is wrong. It follows that every connected component of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^{m} X_i)]$ containing at least one vertex in Φ intersects L_1.

We now do the second step, where we want to show that all vertices in L_1 lie in the same connected component of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^{m} X_i)]$. We have $L_1 = V(\partial o)$, where o is the outer face of G_0. It follows that $V(\tilde{\partial} o) \subseteq \mathsf{Vort} \cup L_1 = \mathsf{Vort} \cup X_1$. Thus, $\tilde{\partial} o$ is a subgraph of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^{m} X_i)]$, which is connected by Observation 3.16. Combining this with the first step, we can conclude that all vertices in Φ are contained in the same connected component of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^{m} X_i)]$.

Next, we are ready to construct the sets $Z_1, \ldots, Z_p$. Set $\Delta = p + 3c + 4h + 1$. We say a number $q \in [\Delta]$ is *bad* if q is congruent to the index of a bad layer modulo Δ, i.e., $q \equiv i \pmod{\Delta}$ for some $i \in [m]$ such that L_i is a bad layer, and is *good* otherwise. As argued before, there are at most $3c + 4h + 1$ bad layers, and therefore there are at most $3c + 4h + 1$ bad numbers in $[\Delta]$. So we can always find p good numbers $q_1, \ldots, q_p \in [\Delta]$. We then construct the sets $Z_1, \ldots, Z_p$ by simply defining Z_i as the union of $L_q \backslash (X_q \cup L_q^+)$ for all indices $q \in [m]$ congruent to q_i modulo Δ, i.e.,

$$Z_i := \bigcup_{j=0}^{\lfloor (m-q_i)/\Delta \rfloor} (L_{j\Delta+q_i} \backslash (X_{j\Delta+q_i} \cup L^+_{j\Delta+q_i})). \tag{1}$$

Since the layers $L_1, \ldots, L_m$ are disjoint, the sets $Z_1, \ldots, Z_p$ are also disjoint. Also, $Z_1, \ldots, Z_p \subseteq V(G) \backslash N_G[\mathsf{Vort}]$, because $Z_1, \ldots, Z_p$ only contain vertices in good layers. Furthermore, Φ is contained in one connected component of $G - \bigcup_{i=1}^{p} Z_i$, because $\mathsf{Vort} \cup (\bigcup_{i=1}^{m} X_i) \subseteq V(G) \backslash \bigcup_{i=1}^{p} Z_i$ and all vertices in Φ are contained in the same connected component of $G[\mathsf{Vort} \cup (\bigcup_{i=1}^{m} X_i)]$.

Finally, we show that $\mathbf{tw}(G/(Z_i \backslash Z')) = O_{h,c}(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$, which is the most complicated part in this proof. Without loss of generality, it suffices to show $\mathbf{tw}(G/(Z_1 \backslash Z')) = O_{h,c}(p + |Z'|)$ for all $Z' \subseteq Z_1$. Let $G_1, \ldots, G_r$ be the vortices of G attached to disjoint facial disks $D_1, \ldots, D_r$ in (G_0, η) with witness pairs $(\tau_1, \mathcal{P}_1), \ldots, (\tau_r, \mathcal{P}_r)$, where $r \leq h$. We call the faces of (G_0, η) containing $D_1, \ldots, D_r$ *vortex faces*. We first add some "virtual" edges to G_0 as follows. Consider an index $i \in [r]$. Suppose $\tau_i = (v_{i,1}, \ldots, v_{i,\ell_i})$. By definition, $v_{i,1}, \ldots, v_{i,\ell_i}$ are the vertices of (G_0, η) that lie on the boundary of D_i, sorted in clockwise or counterclockwise order. For convenience, we write $v_{i,0} = v_{i,\ell_i}$. We then add the edges $(v_{i,j-1}, v_{i,j})$ for all $j \in [\ell_i]$ to G_0, and call them *virtual* edges. Furthermore, we draw these virtual edges along the boundary of the disk D_i (this is possible because $v_1, \ldots, v_{\ell_i}$ are sorted along the boundary of D_i). The images of these virtual edges then enclose the disk D_i. We do this for all indices $i \in [r]$. Let G_0' denote the resulting graph after adding the virtual edges. Since $D_1, \ldots, D_r$ are disjoint facial disks in (G_0, η), the images of the virtual edges do not cross each other or cross the original edges in (G_0, η). Therefore, the drawing of the virtual edges extends η to an embedding of G_0' to Σ; for simplicity, we still use the notation η to denote this embedding. By construction, it is clear that $D_1, \ldots, D_r$ are faces of (G_0', η). Note that every face of (G_0, η) that does not contain any vortices is preserved in (G_0', η), while every vortex face is subdivided into multiple faces in (G_0', η) by the virtual edges. The next observation states that the part of (G_0', η) inside each vortex face has a constant vertex-face diameter.

Observation 3.18. *Let $f \in F_\eta(G_0)$ be a vortex face and $F \subseteq F_\eta(G_0')$ consist of the faces of (G_0', η) contained in f. Then for any two vertices $v, v' \in V(\partial f)$, there exists a VFA path in (G_0', η) from v to v' of length $O(h)$ that does not visit any face in $F_\eta(G_0') \backslash F$.*

Proof. TOPROVE 19 □

The reason for why we construct G_0' is to relate the treewidth of $G/(Z_1 \backslash Z')$ with that

of $G'_0/(Z_1\backslash Z')$. In this way, we can reduce the task to bounding $\mathbf{tw}(G'_0/(Z_1\backslash Z'))$ without considering the vortices/apices of G. Similar tricks are also used in previous work [2, 6].

Observation 3.19. $\mathbf{tw}(G/(Z_1\backslash Z')) = O_h(\mathbf{tw}(G'_0/(Z_1\backslash Z')))$.

Proof. TOPROVE 20 □

Now it remains to bound $\mathbf{tw}(G'_0/(Z_1\backslash Z'))$. Recall $Z_1 = \bigcup_{j=0}^{\lfloor (m-q)/\Delta \rfloor}(L_{j\Delta+q}\backslash(X_{j\Delta+q}\cup L^+_{j\Delta+q}))$ for some good number $q \in [\Delta]$ (see Equation (1)). For notational simplicity, we define $L_i = X_i = L^+_i = \emptyset$ for all integers $i < 0$ and write $i_j = (j-1)\Delta + q$ for $j \in \mathbb{N}$. Then we have $Z_1 = \bigcup_{j=1}^{m'}(L_{i_j}\backslash(X_{i_j} \cup L^+_{i_j}))$ for $m' = \lfloor (m-q)/\Delta \rfloor + 1$. For any $j \in [m']$, L_{i_j} is a good layer. Hence X_{i_j} and $L^+_{i_j}$ satisfy the conditions in Lemma 3.6. Let o_{i_j} be the outer face of $(G_0[L_{\geq i_j}], \eta)$, which is also the outer face of $(G'_0[L_{\geq i_j}], \eta)$. Note that $(G'_0[L_{\geq i_j}], \eta)$ is an outer-preserving extension of $(G_0[L_{i_j}], \eta)$. So by (2) of Lemma 3.6, for every connected component C of $G'_0[L^+_{i_j} \cup L_{>i_j}]$, $N_{G'_0}(C) \subseteq V(\partial f)$ for some face $f \in F_\eta(\partial o_{i_j})\backslash\{o_{i_j}\}$.

For each $j \in [m']$ and each face $f \in F_\eta(\partial o_{i_j})\backslash\{o_{i_j}\}$, we define a set $\kappa(f) \subseteq V(\partial f)\backslash L^+_{i_j}$ as follows. Denote by $\mathcal{C}_f$ the set of connected components of $\partial f - (L^+_{i_j} \cup X_{i_j} \cup Z')$. Note that $V(\partial f)\backslash(L^+_{i_j} \cup X_{i_j} \cup Z') \subseteq Z_1\backslash Z'$. Therefore, every $C \in \mathcal{C}_f$ is contracted into one vertex in $G'_0/(Z_1\backslash Z')$. In each $C \in \mathcal{C}_f$, we keep a representative vertex $\xi_C \in C$. We then define $\kappa(f) = \{\xi_C : C \in \mathcal{C}_f\} \cup ((V(\partial f)\backslash L^+_{i_j}) \cap (X_{i_j} \cup Z'))$. The following observation bounds the size of $\kappa(f)$.

Observation 3.20. $|\kappa(f)| = O_{g,c}(|Z'|)$.

Proof. TOPROVE 21 □

In order to bound $\mathbf{tw}(G'_0/(Z_1\backslash Z'))$, we shall construct a tree decomposition (T, β) for $G'_0/(Z_1\backslash Z')$ in which each torso is of treewidth $O_{g,c}(p + |Z'|)$. If such a tree decomposition exists, by Lemma 2.3, we have $\mathbf{tw}(G'_0/(Z_1\backslash Z')) = O_{g,c}(p + |Z'|)$. The construction of (T, β) is the following. The depth of the tree T is m'. For all $j \in \{0\} \cup [m']$, the nodes in the j-th level of T are one-to-one correspondence to the connected components of $G'_0[L^+_{i_j} \cup L_{>i_j}]$. Note that $G'_0[L^+_{i_0} \cup L_{>i_0}] = G'_0$ is connected, so there is exactly one node in the 0-th level of T, which is the root of T. The parents of the nodes are defined as follows. Consider a node $t \in V(T)$ in the j-th level for $j \geq 1$, and let C_t be the connected component of $G'_0[L^+_{i_j} \cup L_{>i_j}]$ corresponding to t. Since $G'_0[L^+_{i_j} \cup L_{>i_j}]$ is a subgraph of $G'_0[L^+_{i_{j-1}} \cup L_{>i_{j-1}}]$, C_t is contained in a unique connected component of $G'_0[L^+_{i_{j-1}} \cup L_{>i_{j-1}}]$, which corresponds to a node t' in the $(j-1)$-th level. We then let t' be the parent of t. For each node $t \in V(T)$, we associate a set V_t to t defined as $V_t := C_t\backslash(L^+_{i_{j+1}} \cup L_{>i_{j+1}})$, where $j \in \{0\} \cup [m']$ is the number such that t is in the j-th level of T and C_t is the connected component of $G'_0[L^+_{i_j} \cup L_{>i_j}]$ corresponding to t. One can easily verify that $\{V_t \mid t \in V(T)\}$ is a partition of $V(G'_0)$. In addition, we associate another set U_t to each node $t \in V(T)$ defined as follows. If $j = 0$ (i.e., t is the roof of T), then set $U_t := \emptyset$. Otherwise, let t' be the parent of t. As $(G'_0[L_{\geq i_j}], \eta)$ is an outer-preserving extension of $(\partial o_{i_j}, \eta)$, by (2) of Lemma 3.6, there exists a face $f \in F_\eta(\partial o_{i_j})\backslash\{o_{i_j}\}$ with $N_{G'_0}(C_t) \subseteq V(\partial f)\backslash L^+_{i_j}$. We then let $U_t := V_{t'} \cap \kappa(f)$. Note that $|U_t| = O_{g,c}(|Z'|)$, as $|\kappa(f)| = O_{g,c}(|Z'|)$ by Observation 3.20. Finally, we define $\beta(t) = \pi(V_t \cup U_t)$ as the bag of each node $t \in V(T)$, where $\pi : V(G'_0) \to V(G'_0/(Z_1\backslash Z'))$ is the quotient map for the contraction of G'_0 to $G'_0/(Z_1\backslash Z')$.

Observation 3.21. *(T, β) is a tree decomposition of $G'_0/(Z_1\backslash Z')$. Furthermore, for all $t \in V(T)$, we have $\sigma(t) = \pi(U_t)$ and thus $|\sigma(t)| = O_{g,c}(|Z'|)$.*

Proof. TOPROVE 22 □

As (T, β) is a tree decomposition of $G'_0/(Z_1 \backslash Z')$, it suffices to show $\mathbf{tw}(\mathsf{tor}(t)) = O_{g,c}(p+|Z'|)$ for all $t \in V(T)$. Consider a node $t \in V(T)$ in the j-th level of T. In the above observation, it has been shown that $|\sigma(t)| = O_{g,c}(|Z'|)$. So we only need to show $\mathbf{tw}(\mathsf{tor}(t) - \sigma(t)) = O_{g,c}(p + |Z'|)$.

For notational convenience, set $\phi = i_j$ and $\psi = i_{j+1}$. What we are going to do is to build a graph that contains $\mathsf{tor}(t) - \sigma(t)$ as a minor, and bound the treewidth of that graph. Let (Γ, η) be a subgraph of $(G'_0[L_\phi^\psi], \eta)$ obtained by removing all edges in $E(L_\psi) \backslash E(\partial o_\psi)$. Note that the edges removed are exactly those embedded in the interiors of the faces $f \in F_\eta(\partial o_\psi) \backslash \{o_\psi\}$. This implies that every face in $F_\eta(\partial o_\psi) \backslash \{o_\psi\}$ is also a face of (Γ, η). Recall the set $\kappa(f) \subseteq V(\partial f) \backslash L_\psi^+$ defined for each $f \in F_\eta(\partial o_\psi) \backslash \{o_\psi\}$. For convenience, we set $\kappa(f) = \emptyset$ for all faces f of (Γ, η) other than the ones in $F_\eta(\partial o_\psi) \backslash \{o_\psi\}$. Clearly, $\kappa(f) \subseteq V(\partial f) = V(\Gamma)$ for all $f \in F_\eta(\Gamma)$. We then define Γ^κ as the graph obtained from Γ by making $\kappa(f)$ a clique for all $f \in F_\eta(\Gamma)$.

Observation 3.22. *Γ^κ contains $\mathsf{tor}(t) - \sigma(t)$ as a minor.*

Proof. TOPROVE 23 □

Finally, it suffices to show that $\mathbf{tw}(\Gamma^\kappa) = O_{h,c}(p + |Z'|)$. By Lemma 3.2, it suffices to bound $\mathsf{diam}^*_{w_\kappa}(\Gamma, \eta)$, where $w_\kappa : F_\eta(\Gamma) \to \mathbb{N}$ is the weight function defined as $w_\kappa(f) = |\kappa(f)|$.

Observation 3.23. $\mathsf{diam}^*_{w_\kappa}(\Gamma, \eta) = O_{h,c}(p + |Z'|)$.

Proof. TOPROVE 24 □

Combining the above observation with Lemma 3.2, we have $\mathbf{tw}(\Gamma^\kappa) = O_{h,c}(p + |Z'|)$. By Observation 3.22, we further have $\mathbf{tw}(\mathsf{tor}(t) - \sigma(t)) = O_{h,c}(p + |Z'|)$ and thus $\mathbf{tw}(\mathsf{tor}(t)) = O_{h,c}(p + |Z'|)$. It follows that $\mathbf{tw}(G'_0/(Z_1 \backslash Z')) = O_{h,c}(p + |Z'|)$, according to Lemma 2.3. Finally, using Observation 3.19, we have $\mathbf{tw}(G/(Z_1 \backslash Z')) = O_{h,c}(p + |Z'|)$, which completes the proof of Lemma 3.14.

3.4 Decomposing minor-free graphs

In this section, we complete the proof of our main theorem (Theorem 1.2), based on the result of the previous section. To this end, we need to introduce the famous Robertson-Seymour decomposition of an H-minor-free graph. By simply combining the results of [2, 7] with Lemma 3.5, we can obtain the following strong version of Robertson-Seymour decomposition.

Lemma 3.24 (Robertson-Seymour decomposition)**.** *Let H be a fixed graph. Any connected H-minor-free graph G admits a tree decomposition (T, β) satisfying the following two conditions (where $h > 0$ is a constant only depending on H).*

(RS.1) For all $t \in V(T)$, $\mathsf{tor}(t)$ admits an h-almost-embeddable structure in which the partial embedding is minimal, and for every child s of t in T, all but at most three vertices in $\sigma(s)$ are vortex vertices or apices of $\mathsf{tor}(t)$. Also, $|\sigma(t)| \leq h$ and all vertices in $\sigma(t)$ are apices of $\mathsf{tor}(t)$.

(RS.2) For all $t \in V(T)$, $G[\gamma(t) \backslash \sigma(t)]$ is connected, and $\sigma(t) \subseteq N_G(\gamma(t) \backslash \sigma(t))$.

Furthermore, given an H-minor-free graph G, the tree decomposition (together with the almost-embeddable structures of the torsos) can be computed in polynomial time.

Proof. TOPROVE 25 □

Let G be an H-minor-free graph. Without loss of generality, we can assume G is connected. We first compute the tree decomposition (T, β) of G in Lemma 3.24 as well as the almost-embeddable structures of the torsos. Condition (RS.2) of Lemma 3.24 implies the following property of the tree decomposition (T, β).

Observation 3.25. *For all $t \in V(T)$, $\mathsf{tor}(t) - \sigma(t)$ is connected and $\sigma(t) \subseteq N_{\mathsf{tor}(t)}(\beta(t) \backslash \sigma(t))$.*

Proof. TOPROVE 26 □

In order to construct the sets $Z_1, \ldots, Z_p \subseteq V(G)$, we consider every node $t \in V(T)$, and compute p sets $Z_1^{(t)}, \ldots, Z_p^{(t)} \subseteq \beta(t) \backslash \sigma(t)$ as follows. Let $t \in V(T)$ be a node and A be the apex set of $\mathsf{tor}(t)$. We say an edge aa' of $\mathsf{tor}(t)[A]$ is *redundant* if there exists a vertex $v \in \beta(t) \backslash A$ adjacent to both a and a' in $\mathsf{tor}(t)$. Define G_t as the graph obtained from $\mathsf{tor}(t)$ by removing all redundant edges. The next observation shows that G_t "inherits" the nice properties of $\mathsf{tor}(t)$ in Observation 3.25.

Observation 3.26. *For all $t \in V(T)$, $G_t - \sigma(t)$ is connected and $\sigma(t) \subseteq N_{G_t}(\beta(t) \backslash \sigma(t))$.*

Proof. TOPROVE 27 □

For each $v \in \sigma(t)$, we pick an arbitrary vertex $v' \in \beta(t) \backslash \sigma(t)$ adjacent to v in G_t and call v' the *witness neighbor* of v (by the above observation, such a witness neighbor exists). Define $\Phi^{(t)} \subseteq \beta(t) \backslash \sigma(t)$ as the set of witness neighbors of all vertices in $\sigma(t)$. We have $|\Phi^{(t)}| \leq |\sigma(t)| \leq h$ by (RS.1) of Lemma 3.24. Next, we apply Corollary 3.15 on $G = G_t - \sigma(t)$ with the set $\Phi = \Phi^{(t)}$, and define $Z_1^{(t)}, \ldots, Z_p^{(t)}$ as the sets $Z_1, \ldots, Z_p$ obtained by the corollary. Note that $G_t - \sigma(t)$ is obtained from $\mathsf{tor}(t)$ by deleting some apices and some edges between apices, so it inherits the almost-embeddable structure of $\mathsf{tor}(t)$, in which the partial embedding is minimal. Also, $G_t - \sigma(t)$ is connected by Observation 3.25. Therefore, Corollary 3.15 is applicable on $G_t - \sigma(t)$.

Next, we define a coloring $\mathsf{col}\colon V(T) \to [p]$ which assigns each node of T one of the colors $1, \ldots, p$. The coloring is defined in a top-down manner in T. If t is the root, define $\mathsf{col}(t)$ as an arbitrary number in $[p]$. Otherwise, let t' be the parent of t and suppose $\mathsf{col}(t')$ is already defined.

Observation 3.27. *There exists at most one index $i \in [p]$ such that the clique $\mathsf{tor}(t')[\sigma(t)]$ contains an edge of $G_{t'}[Z_i^{(t')}]$.*

Proof. TOPROVE 28 □

If there exists some $i \in [p]$ such that $\mathsf{tor}(t')[\sigma(t)]$ contains an edge of $G_{t'}[Z_i^{(t')}]$, we then set $\mathsf{col}(t) = i$; by the above observation, such an index i is unique (if it exists). Otherwise, $\mathsf{tor}(t')[\sigma(t)]$ is edge-disjoint from all of $\mathsf{tor}(t')[Z_1^{(t')}], \ldots, \mathsf{tor}(t')[Z_p^{(t')}]$. In this case, we set $\mathsf{col}(t) = \mathsf{col}(t')$.

Finally, we construct the partition $Z_1, \ldots, Z_p$ in Theorem 1.2 as follows. Let $T_i = \mathsf{col}^{-1}(\{i\})$ be the set of nodes of T colored with i. For each node $t \in V(T)$, let $R^{(t)}$ be the complement of $\bigcup_{i=1}^p Z_i^{(t)}$ in $\beta(t) \backslash \sigma(t)$, i.e., $R^{(t)} := (\beta(t) \backslash \sigma(t)) \backslash (\bigcup_{i=1}^p Z_i^{(t)})$. We then define

$$Z_i := \left(\bigcup_{t \in T} Z_i^{(t)} \right) \cup \left(\bigcup_{t \in T_i} R^{(t)} \right) \tag{2}$$

for every $i \in [p]$. From our construction, one can easily verify the following fact.

Observation 3.28. *$Z_1, \ldots, Z_p$ is a partition of $V(G)$.*

Proof. TOPROVE 29 □

Observation 3.29. *Let $i \in [p]$ and $t \in V(T)$. For any edge vv' of $\mathsf{tor}(t)[Z_i^{(t)}]$, v and v' lie in the same connected component of $G[(\bigcup_{s \in S}(\gamma(s) \backslash \sigma(s)) \cap Z_i) \cup \{v, v'\}]$, where S consists of all children s of t in T satisfying $v, v' \in \sigma(s)$.*

Proof. TOPROVE 30 □

Next, we show that $\mathbf{tw}(G/(Z_i \backslash Z')) = O_h(p + |Z'|)$ for all $i \in [p]$ and $Z' \subseteq Z_i$. Without loss of generality, we only need to consider the case $i = 1$. Let $\pi \colon V(G) \to V(G/(Z_1 \backslash Z'))$ be the quotient map for the contraction of G to $G/(Z_1 \backslash Z')$. For each $t \in V(T)$, let $\beta^*(t) = \pi(\beta(t))$. By Fact 2.4, (T, β^*) is a tree decomposition of $G/(Z_1 \backslash Z')$. We use the notations $\sigma^*(t)$, $\gamma^*(t)$, $\mathsf{tor}^*(t)$ to denote the adhesion, γ-set, and torso of a node $t \in V(T)$ in the tree decomposition (T, β^*), in order to distinguish from those for (T, β). We observe the following simple fact.

Observation 3.30. *For all $t \in V(T)$, $\sigma^*(t) = \pi(\sigma(t))$.*

Proof. TOPROVE 31 □

By Lemma 2.3, it suffices to show that $\mathbf{tw}(\mathsf{tor}^*(t)) = O_h(p + |Z'|)$ for all $t \in V(T)$. Consider a node $t \in V(T)$. In order to bound $\mathbf{tw}(\mathsf{tor}^*(t))$, we shall construct a graph of bounded treewidth that contains $\mathsf{tor}^*(t)$ as a minor. Define $\mathsf{TOR}(t)$ as the graph obtained from $\mathsf{tor}(t)$ by making $\sigma(t)$ a clique. Let S_{bad} be the set of children s of t satisfying $(\gamma(s) \backslash \sigma(s)) \cap Z' \neq \emptyset$. Note that $|S_{\mathsf{bad}}| \leq |Z'|$ as the sets $\gamma(s) \backslash \sigma(s)$ are disjoint for the children s of t. Set $Z'' = Z' \cup (\bigcup_{s \in S_{\mathsf{bad}}} \sigma(s))$. Since $|\sigma(s)| \leq h$ for all $s \in S_{\mathsf{bad}}$ and $|S_{\mathsf{bad}}| \leq |Z'|$, by (RS.1) of Lemma 3.24, we have $|Z''| = O_h(|Z'|)$. Consider the graph $G^* = \mathsf{TOR}(t)/(Z_1^{(t)} \backslash (Z'' \cap Z_1^{(t)}))$. Because $Z_1^{(t)} \subseteq \beta(t) \backslash \sigma(t)$, the vertices in $\sigma(t)$ do not get contracted in G^*. So the vertices in $\sigma(t)$ are also vertices in G^*. We observe that $\mathbf{tw}(G^*) = O_h(p + |Z'|)$. Indeed, $G^* - \sigma(t)$ is isomorphic to $(\mathsf{tor}(t) - \sigma(t))/(Z_1^{(t)} \backslash (Z'' \cap Z_1^{(t)}))$, and $(G_t - \sigma(t))/(Z_1^{(t)} \backslash (Z'' \cap Z_1^{(t)}))$ can be obtained from $(\mathsf{tor}(t) - \sigma(t))/(Z_1^{(t)} \backslash (Z'' \cap Z_1^{(t)}))$ by deleting the $O_h(1)$ redundant edges. Therefore, $(G_t - \sigma(t))/(Z_1^{(t)} \backslash (Z'' \cap Z_1^{(t)}))$ can be viewed as a graph obtained from G^* by deleting $O_h(1)$ vertices and edges, since $\sigma(t) \leq h$ by (RS.1) of Lemma 3.24. So the difference between $\mathbf{tw}((G_t - \sigma(t))/(Z_1^{(t)} \backslash (Z'' \cap Z_1^{(t)})))$ and $\mathbf{tw}(G^*)$ is $O_h(1)$. Since $|Z''| = O_h(|Z'|)$, by condition (1) of Corollary 3.15, we have

$$\mathbf{tw}((G_t - \sigma(t))/(Z_1^{(t)} \backslash (Z'' \cap Z_1^{(t)}))) = O_h(p + |Z'|),$$

which implies $\mathbf{tw}(G^*) = O_h(p + |Z'|)$. Finally, we show that G^* contains $\mathsf{tor}^*(t)$ as a minor. To this end, we need the following observation.

Observation 3.31. *For any $V \subseteq \beta^*(t)$, if $\mathsf{tor}^*(t)[V]$ is connected, then $\mathsf{TOR}(t)[\pi_{|\beta(t)}^{-1}(V)]$ is also connected, where $\pi_{|\beta(t)} : \beta(t) \to \beta^*(t)$ is the map π restricted to $\beta(t)$.*

Proof. TOPROVE 32 □

Observation 3.32. *G^* contains $\mathsf{tor}^*(t)$ as a minor and thus $\mathbf{tw}(\mathsf{tor}^*(t)) = O_h(p + |Z'|)$.*

Proof. TOPROVE 33 □

The above observation implies $\mathbf{tw}(G/(Z_1 \backslash Z')) = O_h(p + |Z'|)$ by Lemma 2.3, because (T, β^*) is a tree decomposition of $G/(Z_1 \backslash Z')$. This completes the proof of Theorem 1.2.

4 Applications

In this section, we briefy discuss some applications of Theorem 1.2 in parameterized complexity building on [2, 29]. Indeed, Theorem 1.2 directly results in parameterized algorithms of running time $n^{O(\sqrt{k})}$ or $2^{\widetilde{O}(\sqrt{k})} \cdot n^{O(1)}$ for a broad class of vertex/edge deletion problems on H-minor-free graphs. For example, this includes all problems that can be formulated as Permutation CSP Deletion or 2-Conn Permutation CSP Deletion [29].

An instance of the (binary) CSP problem is specified by a triple $\Gamma = (X, D, \mathcal{C})$ where X is a finite set of variables, D is a finite domain, and $\mathcal{C}$ is a set of constraints each of which is of the form $c = (x, y, R)$ where $x, y \in X$ and $R \subseteq D^2$. We say Γ is a *permutation CSP*

instance if for every constraint $c = (x, y, R) \in \mathcal{C}$ it holds that $|\{b \in D \mid (a, b) \in R\}| \leq 1$ and $|\{b \in D \mid (b, a) \in R\}| \leq 1$ for all $a \in D$. We write $|\Gamma| = |X| + |D|$ as the *size* of Γ. An *assignment* of Γ is a function $\alpha : Y \to D$ on a subset $Y \subseteq X$, and we say α is *satisfying* if for all $c = (x, y, R) \in \mathcal{C}$ such that $x, y \in Y$, we have $(\alpha(x), \alpha(y)) \in R$. A permutation CSP instance $\Gamma = (X, D, \mathcal{C})$ naturally induces a graph G_Γ with vertex set X in which two variables $x, y \in X$ are connected by an edge if there exists $c \in \mathcal{C}$ such that $c = (x, y, R)$ for some $R \subseteq D^2$. We say the permutation CSP instance Γ is *H-minor-free* if its underlying graph G_Γ is H-minor-free. For a subset $\mathcal{C}' \subseteq \mathcal{C}$ of constraints, we write $\Gamma - \mathcal{C}' = (X, D, \mathcal{C} \backslash \mathcal{C}')$. For a subset $X' \subseteq X$ of variables, we write $\Gamma - X' = (X \backslash X', D, \mathcal{C} \backslash \mathcal{C}_{X'})$, where $\mathcal{C}_{X'} \subseteq \mathcal{C}$ consists of all constraints that involve at least one variable from X'.

Let $\Gamma = (X, D, \mathcal{C})$ be a permutation CSP instance. A *size constraint* for Γ is a pair (w, δ) where $w \colon X \times D \to \mathbb{N}$ is a weight function and $\delta \in \mathbb{N}$ is a threshold. We say an assignment $\alpha \colon Y \to D$ of Γ *respects* the size constraint (w, δ) if $\sum_{y \in Y} w(y, \alpha(y)) \leq \delta$. We say Γ is *satisfiable* subject to (w, δ) on a subset $Y \subseteq X$ if there is a satisfying assignment $\alpha \colon Y \to D$ of Γ that respects (w, δ).

4.1 Permutation CSP Deletion Problems

In [29] a subset of the authors define the problems Permutation CSP Vertex Deletion and Permutation CSP Edge Deletion. These problems essentially ask whether one can remove from a given permutation CSP instance a small number of variables (resp., constraints), or equivalently vertices (resp., edges) in the underlying graph, such that the resulting instance is satisfiable on every connected component of the underlying graph. The formal definitions are the following[4].

The Permutation CSP Vertex Deletion problem is a parameterized problem that takes as input a tuple (Γ, w, δ, k), where $\Gamma = (X, D, \mathcal{C})$ is a permutation CSP instance, (w, δ) is a size constraint for Γ, and k is the solution size parameter. The goal is to find a subset $X' \subseteq X$ with $|X'| \leq k$ such that $\Gamma - X'$ is satisfiable subject to (w, δ)[5] on every subset $Y \subseteq X \backslash X'$ satisfying that $G_{\Gamma - X'}[Y]$ is a connected component of $G_{\Gamma - X'}$.

Similarly, the Permutation CSP Edge Deletion problem takes the same input, but the goal is to find a subset $\mathcal{C}' \subseteq \mathcal{C}$ with $|\mathcal{C}'| \leq k$ such that $\Gamma - \mathcal{C}'$ is satisfiable subject to (w, δ) on every subset $Y \subseteq X$ satisfying that $G_{\Gamma - \mathcal{C}'}[Y]$ is a connected component of $G_{\Gamma - \mathcal{C}'}$.

Theorem 4.1. *Let H be a fixed graph. Then the* Permutation CSP Vertex Deletion *(or the* Permutation CSP Edge Deletion*) problem on H-minor-free instances (Γ, w, δ, k) can be solved in $(|\Gamma| + \delta)^{O(\sqrt{k})}$ time.*

Proof. TOPROVE 34 □

Many natural problems can be formulated as Permutation CSP Vertex/Edge Deletion problems including the following examples (see [29] for more details):

- Odd Cycle Transversal: The input consists of a graph G and a parameter k. The goal is to find a subset $S \subseteq V(G)$ of at most k vertices such that $G - S$ contains no odd cycle.
- Vertex Multiway Cut: The input consists of a graph G, a terminal set $T \subseteq V(G)$, and a parameter k. The goal is to find a subset $S \subseteq V(G)$ of at most k vertices such that no two vertices in T lie in the same connected component of $G - S$.

[4] We remark that our definition of (2-connected) permutation CSP deletion is a simplified version of the original definition in [29], which is already sufficient for our applications. The original definition is more complicated and allows multiple size constraints of a more general form.

[5] Strictly speaking, (w, δ) is a size constraint for Γ instead of $\Gamma - X'$. But it naturally induces a size constraint for $\Gamma - X'$ by restricting w to $(X \backslash X') \times D$. For convenience, we still denote it by (w, δ) here.

- Group Feedback Vertex Set: The input consists of a graph G with a map $\lambda : V(G) \times V(G) \to \Gamma$ where Γ is a group and $\lambda(u,v) = (\lambda(v,u))^{-1}$ and a parameter k. The goal is to find a subset $S \subseteq V(G)$ of at most k vertices such that $G - S$ contains no *non-null* cycle where a cycle $(v_0, v_1, \ldots, v_r = v_0)$ is non-null if $\prod_{i=1}^{r} \lambda(v_{i-1}, v_i) = \mathbf{1}$.

- Component Order Connectivity: The input consists of a graph G, a threshold δ, and a parameter k. The goal is to find a subset $S \subseteq V(G)$ of at most k vertices such that every component of $G - S$ is of size at most δ.

- The edge-deletion version of the above problems where the goal is to find a subset $S \subseteq E(G)$ of at most k *edges* such that $G - S$ satisfies the corresponding properties.

Corollary 4.2. *There exist algorithms with running time $n^{O(\sqrt{k})}$ for* Odd Cycle Transversal, Vertex Multiway Cut, Vertex Multicut, Group Feedback Vertex Set, Component Order Connectivity, *and the edge-deletion version of these problems on H-minor-free graphs, where n is the number of vertices of the input graph and k is the solution-size parameter.*

Proof. TOPROVE 35 □

Using the minor-preserving (quasi-)polynomial kernels from [29] or the (quasi-)polynomial-size candidate sets given in [2], we can also obtain (randomized) subexponential-time FPT algorithms for these problems. Here the notation $\widetilde{O}(\cdot)$ hides $\log k$ factors.

Corollary 4.3. *There exist randomized algorithms with running time $2^{\widetilde{O}(\sqrt{k})} \cdot n^{O(1)}$ for* Odd Cycle Transversal, Vertex Multiway Cut, Group Feedback Vertex Set *(for a fixed group), and the edge-deletion version of these problems on H-minor-free graphs, where n is the number of vertices of the input graph and k is the solution-size parameter.*

4.2 Two-Connected Permutation CSP Deletion Problems

To extend the scope of problems covered by this approach, [29] also considers the 2-Conn Permutation CSP Deletion problem. The problem is similar to Permutation CSP Deletion defined above, but the satisfiability we care about is on the 2-connected components of the underlying graph (after deletion), instead of connected components.

The 2-Conn Permutation CSP Vertex Deletion problem is a parameterized problem that takes as input a tuple (Γ, w, δ, k), where $\Gamma = (X, D, \mathcal{C})$ is a permutation CSP instance, (w, δ) is a size constraint for Γ, and k is the solution size parameter. The goal is to find a subset $X' \subseteq X$ with $|X'| \leq k$ such that $\Gamma - X'$ is satisfiable subject to (w, δ)[6] on every subset $Y \subseteq X \backslash X'$ satisfying that $G_{\Gamma - X'}[Y]$ is a 2-connected component of $G_{\Gamma - X'}$.

Similarly, the 2-Conn Permutation CSP Edge Deletion problem takes the same input, but the goal is to find a subset $\mathcal{C}' \subseteq \mathcal{C}$ with $|\mathcal{C}'| \leq k$ such that $\Gamma - \mathcal{C}'$ is satisfiable subject to (w, δ) on every subset $Y \subseteq X$ satisfying that $G_{\Gamma - \mathcal{C}'}[Y]$ is a 2-connected component of $G_{\Gamma - \mathcal{C}'}$.

Theorem 4.4. *Let H be a fixed graph. Then the* 2-Conn Permutation CSP Vertex Deletion *(or the* 2-Conn Permutation CSP Edge Deletion*) problem on H-minor-free instances (Γ, w, δ, k) can be solved in $(|\Gamma| + \delta)^{O(\sqrt{k})}$ time.*

Proof. TOPROVE 36 □

The following problems can be formulated as 2-connected permutation CSP deletion problems (see [29] for more details):

[6] Strictly speaking, (w, δ) is a size constraint for Γ instead of $\Gamma - X'$. But it naturally induces a size constraint for $\Gamma - X'$ by restricting w to $(X \backslash X') \times D$. For convenience, we still denote it by (w, δ) here.

- Subset Feedback Vertex Set: The input consists of a graph G, a terminal set $T \subseteq V(G)$, and a parameter k. The goal is to find a subset $S \subseteq V(G)$ of at most k vertices such that every cycle in $G - S$ is disjoint from T.

- Subset Odd Cycle Transversal: The input consists of a graph G, a terminal set $T \subseteq V(G)$, and a parameter k. The goal is to find a subset $S \subseteq V(G)$ of at most k vertices such that every odd cycle in $G - S$ is disjoint from T.

- Subset Group Feedback Vertex Set: The input consists of a graph G with a map $\lambda\colon V(G) \times V(G) \to \Gamma$ where Γ is a group and $\lambda(u, v) = (\lambda(v, u))^{-1}$, a terminal set $T \subseteq V(G)$, and a parameter k. The goal is to find a subset $S \subseteq V(G)$ of at most k vertices such that every non-null cycle in $G - S$ is disjoint from T.

- 2-Conn Component Order Connectivity: The input consists of a graph G, a threshold δ, and a parameter k. The goal is to find a subset $S \subseteq V(G)$ of at most k vertices such that every every 2-connected component of $G - S$ is of size at most δ.

- The edge-deletion version of the above problems where the goal is to find a subset $S \subseteq E(G)$ of at most k *edges* such that $G - S$ satisfies the corresponding properties.

Corollary 4.5. *There exist algorithms with running time $n^{O(\sqrt{k})}$ for* Subset Feedback Vertex Set, Subset Odd Cycle Transversal, Subset Group Feedback Vertex Set, 2-Conn Component Order Connectivity, *and the edge-deletion version of these problems on H-minor-free graphs, where n is the number of vertices of the input graph and k is the solution-size parameter.*

Using the minor-preserving (quasi-)polynomial kernels for Subset Feedback Vertex Set and a reduction from Subset Feedback Edge Set to Subset Feedback Vertex Set given in [29], we can also obtain (randomized) subexponential-time FPT algorithms for these two problems.

Corollary 4.6. *There exist randomized algorithms with running time $2^{\widetilde{O}(\sqrt{k})} \cdot n^{O(1)}$ for* Subset Feedback Vertex Set *and* Subset Feedback Edge Set *on H-minor-free graphs, where n is the number of vertices of the input graph and k is the solution-size parameter.*

References

[1] Brenda S. Baker. Approximation algorithms for np-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994. doi:10.1145/174644.174650. 1, 2, 4

[2] Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue. Subexponential parameterized algorithms for cut and cycle hitting problems on H-minor-free graphs. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2063–2084. SIAM, 2022. doi:10.1137/1.9781611977073.82. 2, 3, 4, 5, 10, 12, 16, 23, 24, 26, 28

[3] Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue. True contraction decomposition and almost eth-tight bipartization for unit-disk graphs. *ACM Trans. Algorithms*, 20(3):20, 2024. doi:10.1145/3656042. 2, 14

[4] Thang Nguyen Bui and Andrew Peck. Partitioning planar graphs. *SIAM J. Comput.*, 21(2):203–215, 1992. doi:10.1137/0221016. 2, 4

[5] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3. 14

[6] Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and H-minor-free graphs. *J. ACM*, 52(6):866–893, 2005. doi:10.1145/1101821.1101823. 4, 23

[7] Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), 23-25 October 2005, Pittsburgh, PA, USA, Proceedings*, pages 637–646. IEEE Computer Society, 2005. doi:10.1109/SFCS.2005.14. 2, 4, 9, 24

[8] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Contraction decomposition in h-minor-free graphs and algorithmic applications. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 441–450. ACM, 2011. doi:10.1145/1993636.1993696. 2, 3

[9] Erik D. Demaine, MohammadTaghi Hajiaghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. *Comb.*, 30(5):533–552, 2010. doi:10.1007/s00493-010-2341-5. 2

[10] Matt DeVos, Guoli Ding, Bogdan Oporowski, Daniel P. Sanders, Bruce A. Reed, Paul D. Seymour, and Dirk Vertigan. Excluding any graph as a minor allows a low tree-width 2-coloring. *J. Comb. Theory B*, 91(1):25–41, 2004. doi:10.1016/j.jctb.2003.09.001. 4, 9

[11] Reinhard Diestel. *Graph Theory*. Springer Berlin, 5 edition, 2017. doi:10.1007/978-3-662-53622-3.

[12] Frederic Dorn, Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Beyond bidimensionality: Parameterized subexponential algorithms on directed graphs. *Inf. Comput.*, 233:60–70, 2013. doi:10.1016/j.ic.2013.11.006. 2, 4

[13] Zdenek Dvorák. Thin graph classes and polynomial-time approximation schemes. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1685–1701. SIAM, 2018. doi:10.1137/1.9781611975031.110. 4

[14] David Eppstein. Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms Appl.*, 3(3):1–27, 1999. doi:10.7155/jgaa.00014. 2, 4

[15] David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000. doi:10.1007/s004530010020. 4, 16

[16] Jeff Erickson, Kyle Fox, and Amir Nayyeri. Global minimum cuts in surface embedded graphs. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1309–1318. SIAM, 2012. doi:10.1137/1.9781611973099.103.

[17] Fedor V. Fomin, Daniel Lokshtanov, Sudeshna Kolay, Fahad Panolan, and Saket Saurabh. Subexponential algorithms for rectilinear Steiner tree and arborescence problems. *ACM Trans. Algorithms*, 16(2):21:1–21:37, 2020. doi:10.1145/3381420. 4

[18] Fedor V. Fomin, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Subexponential parameterized algorithms for planar and apex-minor-free graphs via low treewidth pattern covering. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 515–524. IEEE Computer Society, 2016. doi:10.1109/FOCS.2016.62. 2, 4

[19] Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *Comb.*, 23(4):613–632, 2003. doi:10.1007/s00493-003-0037-9. 2

[20] Martin Grohe, Ken-ichi Kawarabayashi, and Bruce A. Reed. A simple algorithm for the graph minor decomposition - logic meets structural graph theory. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 414–431. SIAM, 2013. doi:10.1137/1.9781611973105.30.

[21] Ken-ichi Kawarabayashi, Robin Thomas, and Paul Wollan. Quickly excluding a non-planar graph. *CoRR*, abs/2010.12397, 2020. URL: https://arxiv.org/abs/2010.12397, arXiv:2010.12397. 4

[22] Ken-ichi Kawarabayashi and Mikkel Thorup. The minimum k-way cut of bounded size is fixed-parameter tractable. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 160–169. IEEE Computer Society, 2011. doi:10.1109/FOCS.2011.53. 2

[23] Ken-ichi Kawarabayashi and Paul Wollan. A simpler algorithm and shorter proof for the graph minor decomposition. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 451–458. ACM, 2011. doi:10.1145/1993636.1993697.

[24] Sanjeev Khanna and Rajeev Motwani. Towards a syntactic characterization of PTAS. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 329–337. ACM, 1996. doi:10.1145/237814.237979. 2

[25] Philip N. Klein. A subset spanner for planar graphs, with application to subset TSP. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 749–756. ACM, 2006. doi:10.1145/1132516.1132620. 2

[26] Philip N. Klein. A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM J. Comput.*, 37(6):1926–1952, 2008. doi:10.1137/060649562. 2

[27] Philip N. Klein and Dániel Marx. Solving planar k -terminal cut in $O(n^{c\sqrt{k}})$ time. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 569–580. Springer, 2012. doi:10.1007/978-3-642-31594-7_48. 4

[28] Philip N. Klein and Dániel Marx. A subexponential parameterized algorithm for subset TSP on planar graphs. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1812–1830. SIAM, 2014. doi:10.1137/1.9781611973402.131. 4

[29] Dániel Marx, Pranabendu Misra, Daniel Neuen, and Prafullkumar Tale. A framework for parameterized subexponential algorithms for generalized cycle hitting problems on planar graphs. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022, Virtual Conference / Alexandria, VA, USA, January 9 - 12, 2022*, pages 2085–2127. SIAM, 2022. doi:10.1137/1.9781611977073.83. 2, 3, 4, 26, 27, 28, 29

[30] Dániel Marx, Marcin Pilipczuk, and Michal Pilipczuk. On subexponential parameterized algorithms for Steiner tree and directed subset TSP on planar graphs. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 474–484. IEEE Computer Society, 2018. doi:10.1109/FOCS.2018.00052. 4

[31] Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins series in the mathematical sciences. Johns Hopkins University Press, 2001. doi:10.56021/9780801866890.

[32] Jesper Nederlof. Detecting and counting small patterns in planar graphs in subexponential parameterized time. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1293–1306. ACM, 2020. doi:10.1145/3357713.3384261. 4

[33] Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Contraction decomposition in unit disk graphs and algorithmic applications in parameterized complexity. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 1035–1054. SIAM, 2019. doi:10.1137/1.9781611975482.64. 4

[34] Neil Robertson and Paul D. Seymour. Graph minors. XVI. excluding a non-planar graph. *J. Comb. Theory B*, 89(1):43–76, 2003. doi:10.1016/S0095-8956(03)00042-X. 4, 14

[35] Siamak Tazari. Faster approximation schemes and parameterized algorithms on (odd-)h-minor-free graphs. *Theor. Comput. Sci.*, 417:95–107, 2012. doi:10.1016/j.tcs.2011.09.014. 2, 4

[36] Dimitrios M. Thilikos and Sebastian Wiederrecht. Excluding surfaces as minors in graphs. *CoRR*, abs/2306.01724, 2023. URL: https://arxiv.org/abs/2306.01724, arXiv:2306.01724. 4

[37] Magnus Wahlström. On quasipolynomial multicut-mimicking networks and kernelization of multiway cut problems. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 101:1–101:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.101. 2