Separator Theorem for Minor-Free Graphs in Linear Time

Édouard Bonnet* Tuukka Korhonen† Hung Le‡ Jason Li§ Tomáš Masařík¶

Abstract

The planar separator theorem by Lipton and Tarjan [FOCS '77, SIAM Journal on Applied Mathematics '79] states that any planar graph with n vertices has a balanced separator of size $O(\sqrt{n})$ that can be found in linear time. This landmark result kicked off decades of research on designing linear or nearly linear-time algorithms on planar graphs. In an attempt to generalize Lipton-Tarjan's theorem to nonplanar graphs, Alon, Seymour, and Thomas [STOC '90, Journal of the AMS '90] showed that any minor-free graph admits a balanced separator of size $O(\sqrt{n})$ that can be found in $O(n^{3/2})$ time. The superlinear running time in their separator theorem is a key bottleneck for generalizing algorithmic results from planar to minor-free graphs. Despite extensive research for more than two decades, finding a balanced separator of size $O(\sqrt{n})$ in (linear) $O(n)$ time for minor-free graphs remains a major open problem. Known algorithms either give a separator of size much larger than $O(\sqrt{n})$ or have superlinear running time, or both.

In this paper, we answer the open problem affirmatively. Our algorithm is very simple: it runs a vertex-weighted variant of breadth-first search (BFS) a constant number of times on the input graph. Our key technical contribution is a weighting scheme on the vertices to guide the search for a balanced separator, offering a new connection between the size of a balanced separator and the existence of a clique-minor model. We believe that our weighting scheme may be of independent interest.

1 Introduction

In the late 70s, Lipton and Tarjan [LT79] introduced a *planar* separator theorem in *linear time*: there is a linear-time algorithm that, given any n-vertex planar graph, returns a balanced separator of size $O(\sqrt{n})$. (Their separator bound improved upon the earlier bound of $O(\sqrt{n}\log^{3/2} n)$ by Ungar [Ung51].) In a follow-up paper [LT80], they gave a plethora of algorithmic applications of their separator theorem, from approximating NP-hard problems to data structures, circuit lower bounds, and the maximum matching problems, to name a few. Their results have unleashed decades of intensive research on planar graph algorithms, continuing to this day. (See the book draft by Klein and Mozes [KM24] for a sample of results in planar graphs.) There is hardly any result in planar graphs that does not use the separator theorem or its variants [Mil86, Tho04]. Clearly, the linear running time is crucial for many algorithmic applications.

The planar separator theorem by Lipton and Tarjan has motivated a long line of research on separator theorems for nonplanar graphs. Note that constant-degree expander graphs only have balanced separators of size $\Omega(n)$, and therefore, one has to impose additional structures besides sparsity on the input graph to guarantee the existence of a separator of sublinear size. A well-studied and perhaps most natural approach is to study graphs excluding a fixed minor. We say that a graph H is a *minor* of G if H can be obtained from G by a sequence of vertex and edge deletions, and edge contractions. We say that G is *H-minor-free* if it does not contain H as a minor. Wagner's theorem [Wag37] characterizes planar graphs in terms of forbidden minors: a graph is planar if and only if it excludes K_5 and $K_{3,3}$ as minors. Therefore, planar graphs belong to a subclass of K_5-minor-free graphs. This point of view naturally raises the following questions:

*CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR 5668, 69342 Lyon. edouard.bonnet@ens-lyon.fr
†University of Copenhagen. tuko@di.ku.dk
‡University of Massachusetts, Amherst. hungle@cs.umass.edu
§Carnegie Mellon University. jmli@cs.cmu.edu
¶University of Warsaw. masarik@mimuw.edu.pl

Question 1. *Can we construct a balanced separator of size $O(\sqrt{n})$ for any given K_h-minor-free graph in $O(n)$ time for any fixed h? Can the dependence on h in the separator size and the running time be made polynomial?*

A resolution of the first part in Question 1 casts the separator theorem for planar graphs by Lipton and Tarjan as a special case (when $h = 5$) of a much broader landscape. The second part seeks a stronger guarantee for a more practical purpose: by imposing a polynomial dependence on h, one has to avoid sophisticated approaches based on the Robertson–Seymour structure theorem [RS03], such as in [KR10], which are often considered highly impractical [LR10].

Partial progress on this question was made on graphs embeddable on a surface of genus g; such graphs exclude $K_{O(\sqrt{g})}$ as a minor. Indeed, the paper by Lipton and Tarjan [LT80] constructed a separator of size $O(g\sqrt{n})$. Gilbert, Hutchinson, and Tarjan [GHT84] improved the separator size to $O(\sqrt{gn})$ and gave an algorithm that runs in $O(gn)$ time, *given a genus-g embedding* of the input graph. However, finding such an embedding in linear time is difficult: when g is part of the input, it is NP-hard [Tho89]. For a small genus g, an algorithm with running time $2^{O(\mathrm{poly}(g))}n$ was found decades later [Moh96, KMR08]. To find an $O(\mathrm{poly}(g)\sqrt{n})$-size separator, it suffices to embed a genus-g graph G into a surface with genus $\mathrm{poly}(g)$. There is a polynomial-time algorithm [KS15] (even when g is unbounded) with such a guarantee, but it is unclear if the algorithm can be implemented in $O(\mathrm{poly}(g)n)$ time. In other words, even in the special case of genus-g graphs, Question 1 still does not have a satisfactory answer. Note that a K_5-minor-free graph can have genus $g = \Omega(n)$, and therefore the class of K_h-minor-free graphs for a fixed h is vastly broader than bounded-genus graphs.

In a breakthrough paper, Alon, Seymour, and Thomas [AST90a, AST90b] made significant progress toward Question 1 by showing that any K_h-minor-free graph has a balanced separator of size $O(\mathrm{poly}(h)\sqrt{n})$. Their existential proof can then be turned into a polynomial-time algorithm.

Theorem 1 (Alon, Seymour, and Thomas [AST90a, AST90b]). *For any integer $h \geqslant 1$, any K_h-minor-free graph admits a balanced separator of size $O(h^{3/2}\sqrt{n})$ that can be found in time $O(h^{1/2}\sqrt{n}m)$.*

In concluding their paper, Alon, Seymour, and Thomas [AST90b] wrote: "*It may be that by using more sophisticated, dynamic data structures, the algorithm can be implemented more efficiently, but at the moment we do not see how to do so.*" Indeed, subsequent efforts to obtain linear (or almost linear) time algorithms are by designing completely different algorithms. Reed and Wood [RW09] gave an algorithm that finds a balanced separator of size $2^{O(h^2)}n^{2/3}$ in time $2^{O(h^2)}n$; these bounds are part of a more general tradeoff between separator size and running time. Ignoring the dependence on h, their separator size is $O(n^{2/3})$, which is much larger than the $O(\sqrt{n})$ bound in Theorem 1. Kawarabayashi and Reed [KR10] used Robertson–Seymour's structure theorem of minor-free graphs, giving a quadratic algorithm for finding a separator of size $O(h\sqrt{n})$, and sketching another algorithm in time $O(g(h)n^{1+\epsilon})$ for any fixed constant $\epsilon \in (0,1)$. One of the authors [Kaw11] later clarified that the precise separator bound is $O(h\sqrt{n} + f(h))$. Here $f(h)$ and $g(h)$ are functions of the Robertson–Seymour type. Their algorithm is very complicated, along the lines of Robertson–Seymour decomposition. The full version has not yet appeared after over fifteen years.

Plotkin, Rao, and Smith [PRS94] introduced the notion of *shallow minor*,[1] and derived a polynomial-time algorithm for finding a balanced separator for K_h-minor-free graphs of size

$$O(n/\ell + h^2\ell\log n) \tag{1}$$

where $\ell \in [1,n]$ is a parameter. For example, by setting $\ell = \sqrt{n}/(h\sqrt{\log n})$, we obtain a separator of size $O(h\sqrt{n\log n})$. While the running time of their algorithm [PRS94] is not better than Theorem 1, it is amenable to a fast implementation. Specifically, Wulff-Nilsen [Wul11] designed a faster implementation of the algorithm by Plotkin, Rao, and Smith, obtaining two interesting trade-offs: a balanced separator of size $O(h\sqrt{n\log n})$ in time $O(\mathrm{poly}(h)n^{5/4+\epsilon})$ or of size $O(\mathrm{poly}(h)n^{4/5+\epsilon})$ in linear time $O(\mathrm{poly}(h)n)$. In a follow-up paper, Wulff-Nilsen [Wul14] improved the running time further when $\ell \approx n^{\epsilon}$; however, the improvement is not significant when $\ell \approx \sqrt{n}$. Biswal, Lee, and Rao [BLR10] gave an alternative proof of Theorem 1 based on spectral graph theory, but their proof does not yield a better running time. In summary, despite intensive research for more than two decades, known algorithms either give a separator of size much larger than $O(\mathrm{poly}(h)\sqrt{n})$ or the running time is superlinear, or both. Therefore, Question 1 remained wide open. See Table 1 for a summary.

[1] Interestingly, it turns out that shallow minors play a crucial role in recent theory of sparse graphs [NO12].

Separator size	Running time	References	Notes
$O(h^{3/2}\sqrt{n})$	$O(h^{1/2}\sqrt{n}m)$	[AST90b]	
$2^{O(h^2)}n^{2/3}$	$2^{O(h^2)}n$	[RW09]	$2^{O(h^2)}n^{(2-\epsilon)/3}$-size in $2^{O(h^2)}n^{1+\epsilon}$ time $\forall\epsilon\in[0,1/2]$
$O(h\sqrt{n}+f(h))$	$O(g(h)\,n^{1+\epsilon})$	[KR10]	full proof not available yet
$O(h\sqrt{n\log n})$	$O(h\sqrt{n\log n}m)$	[PRS94]	setting $\ell=\sqrt{n}/(h\sqrt{\log n})$ in Equation (1)
$O(\text{poly}(h)n^{4/5+\epsilon})$	$O(\text{poly}(h)\,n)$	[Wul11]	fast implementation of [PRS94]
$O(h\sqrt{n\log n})$	$O(\text{poly}(h)\,n^{5/4+\epsilon})$	[Wul11]	fast implementation of [PRS94]
$O(\text{poly}(h)\sqrt{n}\log^2(n))$	$O(\text{poly}(h)\,n\,\text{polylog}(n))$	[RST14, Pen16]	balanced cut, cannot output a minor model
$O(\text{poly}(h)\sqrt{n})$	$O(\text{poly}(h)\,n)$	This paper	Theorem 2
$O(\sqrt{n})$	$O(n)$	[LT79]	planar graphs
$O(\sqrt{gn})$	$2^{O(\text{poly}(g))}n$	[GHT84, KMR08]	genus-g graphs
$O(\text{poly}(g)\sqrt{n})$	$O(\text{poly}(g)\,n)$	This paper	genus-g graphs, direct corollary of Theorem 2

Table 1. The top part of the table contains separator theorems for K_h-minor-free graphs. Functions $f(h)$ and $g(h)$ in line 3 are of the Robertson–Seymour type: a tower of several exponentials.

We note that a balanced separator could be obtained by approximating sparsest cuts, or more precisely, balanced cuts, in graphs. The cut-matching game [KRV09] and its non-stop version [RST14], together with recent developments on approximate maxflow [She13, Pen16], give a polylog(n)-approximation algorithm in $O(m\,\text{polylog}(n))$ time. For minor-free graphs, one may assume $m = O(\text{poly}(h)n)$ [Kos82], and therefore, the running time becomes $O(\text{poly}(h)n\,\text{polylog}(n))$. However, these techniques inherently have multiple log factors overhead in both the separator size and the running time— in particular, the polylog(n) in the running time is at least $\log^{40} n$—and hence do not seem to be the right approach for settling Question 1 completely. Furthermore, they often require sophisticated subroutines, e.g., cut-matching game or graph sparsifiers, and cannot construct a K_h-minor model to certify the failure of obtaining a small separator. The failure to output a K_h-minor model means that these algorithms often offer no insight into why a small separator should exist in K_h-minor-free graphs. Here, a *K_h-minor model* of a graph G is a collection of h vertex-disjoint connected subgraphs $\{C_1, C_2, \ldots, C_h\}$ of G such that for every two subgraphs C_i, C_j where $i \neq j$, there exists an edge $uv \in E(G)$ where $u \in V(C_i)$ and $v \in V(C_j)$.

1.1 Our Contribution

In this paper, we answer Question 1 completely: for a given K_h-minor-free graph with n vertices, we construct a balanced separator of size $O(\sqrt{n})$ in $O(n)$ time for any fixed h, and furthermore, the dependence on h in both the separator size and the running time is polynomial. Specifically, in deterministic $O(\text{poly}(h)n)$ time, our algorithm either constructs a separator of size $O(\text{poly}(h)\sqrt{n})$ or outputs $\perp$ to indicate that the input graph G contains a K_h as a minor. Whenever the algorithm fails to construct a balanced separator, if we are required to output a K_h-minor model, we can do so in *randomized linear time* $O(\text{poly}(h)n)$ with success probability at least $1/2$. As an added bonus, our algorithm is extremely simple: we find a balanced separator by running breadth-first search (BFS) on a suitable graph $O(\text{poly}(h))$ times! See Section 1.2 for more details. Our results are summarized in the following theorem.

Theorem 2. *Let G be any given graph with n vertices, and $h > 0$ be a parameter. There is an algorithm that runs in deterministic $O(\text{poly}(h)n)$ time and outputs either:*

- *a balanced separator of size $O(\text{poly}(h)\sqrt{n})$ or*
- *$\perp$ if G contains a K_h as a minor.*

Furthermore, in the latter case, we can output a K_h-minor model of G with probability at least $1/2$ in an additional randomized $O(\text{poly}(h)n)$ time.

The precise separator size is $O(h^{13}\sqrt{n})$ and the precise running time is $O(h^{13}n)$. In this paper, we do not attempt to minimize the exponent of h to keep a simple presentation. When we are not required to output a K_h-minor model, since a K_h-minor-free graph has $O(h\sqrt{\log h}n)$ edges [Kos82], we simply output $\perp$ when $m > c \cdot h\sqrt{\log h}n$, where $m := |E(G)|$, for a sufficiently large constant c to indicate that G is not K_h-minor-free. If we are required to output a K_h-minor model in this case, then we can only keep the first $c \cdot h\sqrt{\log h}n$ (arbitrary) edges from G, and find a K_h-minor model of the resulting graph. However, existing algorithms [RW09, DHJ+13] have running time $2^{\Omega(h)}n$: [RW09] requires at least $2^{h-3}n$ edges and takes time $2^{O(h)}n$, while [DHJ+13] requires only $\Theta(h\sqrt{\log h})n$ edges but takes time $\mathrm{poly}(h)n + 2^{\tilde{O}(h^2)}$ due to a brute-force step on a $\Theta(h^2 \log h)$-vertex graph. There are two ways we can resolve this issue. We could run our algorithm without the assumption that $m = O(\mathrm{poly}(h)n)$, and the final running time would be $O(\mathrm{poly}(h)m)$. Alternatively, and more efficiently, we provide an algorithm (Lemma 4) that in deterministic $O(\mathrm{poly}(h)n)$ time, produces a K_h-minor model when $m \geqslant 100h^2n$. As a result, we can assume that $m = O(h^2)n$. Note that we do not have a dependence on m in the running time since we only read the first $100h^2n$ edges of G, and ignore other edges.

Our Theorem 2 applied to graphs of genus-g also gives a new result: we can find a separator of size $O(\mathrm{poly}(g)\sqrt{n})$ in $O(\mathrm{poly}(g)n)$ time, avoiding computing the surface embedding of the input graph, which is a difficult task, as remarked above.

1.2 Technical Ideas

For simplicity, we assume that the input graph is K_h-minor-free, and our goal is to construct a separator of size $O(\mathrm{poly}(h)\sqrt{n})$ in deterministic linear time $O(\mathrm{poly}(h)n)$. We say that a subset of vertices $S \subseteq V$ is an *α-balanced separator* of $G = (V, E)$ for some $\alpha \in (0, 1)$ if every connected component of $G \setminus S$ has size at most $\alpha \cdot |V|$. We simply say that S is a *balanced separator* if it is 2/3-balanced. Our algorithm is given in Algorithm 1; it outputs a $(1 - \frac{1}{200h^2})$-balanced separator of size $O(\mathrm{poly}(h)\sqrt{n})$. To get a 2/3-balanced separator, we simply repeat the algorithm $O(h^2)$ times and take the union of all the separators from all the runs. The size of the separator increases by a factor of $O(h^2)$ and hence remains $O(\mathrm{poly}(h)\sqrt{n})$. The highlighted lines in Algorithm 1 are where we apply the vertex-weighted variant of BFS to G. These weights are designed by our algorithm, starting from uniform weights for all vertices. We will clarify the role of the weight function shortly.

To formally define our weighted variant of BFS, we need to introduce some notation. Let $w : V \to \mathbb{Z}_+$ be a positive integer weight function on the vertices of G; we write $\langle G, w\rangle$ to denote G with weight function w attached to it. We define the *length* of a path in $\langle G, w\rangle$ to be the total weight of the vertices on the path. (Note that if the path contains a single vertex v, its length is $w(v)$ instead of 0.) The (vertex-weighted) *distance* between two vertices u and v, denoted by $d_{\langle G,w\rangle}(u, v)$, is the length of the shortest path between u and v. Edges of graphs in our paper are not weighted, so the length and distance terminologies are meant to be vertex-weighted only.

The procedure $\mathrm{BFS}(\langle G, w\rangle, v, r)$ computes a (vertex-weighted) shortest-path tree, say T, rooted at v in $\langle G, w\rangle$ truncated at radius r. That is, $\mathrm{BFS}(\langle G, w\rangle, v, r)$ only contains vertices at distance at most r from v. Let $W = \sum_v w(v)$ be the total vertex weight. By standard reduction, one could reduce computing the vertex-weighted shortest-path tree to computing a BFS tree in a (directed) graph with $\{0, 1\}$-edge weights in time $O(m+W)$; see Lemma 8. In our algorithm, we will guarantee that $W = \mathrm{poly}(h)n$ and hence the running time of $\mathrm{BFS}(\langle G, w\rangle, v, r)$ is $O(m+\mathrm{poly}(h)n)$.

Another key subroutine of our algorithm is the vertex-weighted variant of the *KPR decomposition*, named after Klein, Plotkin, and Rao [KPR93]. The KPR decomposition, given an unweighted[2] K_h-minor-free graph G edges and a parameter $\Delta > 0$, decomposes G into connected components of *weak (hop) diameter* $O(h^2\Delta)$ by removing $O(h \cdot m/\Delta) = (h^2\sqrt{\log h}n/\Delta)$ edges. When $\Delta \approx \sqrt{n}$, the number of edges removed is $O(\mathrm{poly}(h)\sqrt{n})$, coinciding with the separator bound that we are aiming for. The decomposition is obtained by *computing BFS trees in h rounds*. Here in our work, we need (i) a vertex-weighted version of KPR and (ii) to output a K_h-minor model in linear time whenever the algorithm fails to output the largest component of small (weak) hop diameter. Therefore, our guarantees, summarized in Lemma 1 below, are slightly different from the original KPR [KPR93]. The proof will be given in Section A.

[2] KPR could work for edge-weighted graphs as well; in our paper, the intuition came from the unweighted version.

```
FINDSEPARATOR(G = (V, E)):        ⟨⟨Assume that G is K_h-minor-free⟩⟩
  w_1(v) ← 40 for every v ∈ V
  S ← ∅        ⟨⟨The separator⟩⟩
  for t ← 1 to 20h^2
      (C_t^*, S_t) ← KPR(⟨G, w_t⟩, ⌊√n/6h^2⌋, h)        ⟨⟨KPR with Δ = ⌊√n/6h^2⌋, applying BFS h times⟩⟩
      S ← S ∪ S_t
      if |C_t^*| ≤ (1 − 1/(200h^2))n        ⟨⟨The current separator is (1 − 1/(200h^2))-balanced⟩⟩
          return S
      c_t ← an arbitrary vertex in C_t^*
      T_t ← BFS(⟨G, w_t⟩, c_t, √n)        ⟨⟨BFS truncated at radius √n⟩⟩
      for every v ∈ V(T_t)
          w_{t+1}(v) ← w_t(v) + ⌈ |T_t(v)| 20^3 h^6 / √n · w_t(v) ⌉        ⟨⟨Reweighting vertices for next iteration⟩⟩
  return S
```

Algorithm 1. An algorithm for finding a $(1-\frac{1}{200h^2})$-balanced separator with size $O(\text{poly}(h)\sqrt{n})$ of a K_h-minor-free graph G. $T_i(v)$ denotes the vertex set of the subtree of T_i rooted at v.

Lemma 1 (Vertex-Weighted KPR). *Given a graph $\langle G=(V,E),w\rangle$ with integer weights on vertices, m edges and integer parameters $\Delta>0, h>0$. Let $W=\sum_{v\in V} w(v)$. The procedure* KPR$(\langle G,w\rangle,\Delta,h)$ *runs in time $O(h\cdot(m+W))$ and returns either:*

- *a pair (C^*,S) where S is a subset of vertices of size $h\cdot W/\Delta$, and C^* is the connected component of maximum size of $G-S$, which is guaranteed to have (vertex-weighted) weak diameter at most $6\cdot h^2\Delta$, or*
- *a K_h-minor model.*

As we will guarantee that $W=\text{poly}(h)n$, the total running time of KPR$(\langle G,w\rangle,\Delta)$ is $O(\text{poly}(h)m)$. We will set $\Delta=\sqrt{n}/h^2$ (in Algorithm 1) and therefore, the number of vertices in the separator returned by KPR is $O(\text{poly}(h)\sqrt{n})$. Similar to the original KPR [KPR93], our vertex-weighted version of KPR decomposition can be constructed by applying (vertex-weighted) BFS h times. Thus, in total, our algorithm calls vertex-weighted BFS $O(h^3)$ times. Therefore, it is both simple, efficient, and completely different from all other existing algorithms!

Our key technical contribution is a weighting scheme on the vertices, which offers a new insight into the dynamics between the size of the balanced separator and the existence of a K_h-minor model. The scheme serves two different purposes: guiding the search for the small balanced separator, and the search for the minor model when such a separator does not exist. (The second purpose does not show up in Algorithm 1; instead, it will be part of the analysis and in the algorithm for constructing a minor model if required.)

We first describe how the weighting scheme guides the search for a balanced separator. Initially, every vertex v has the same role and hence is given the same weight $w_1(v)=40$. When we apply KPR to find a separator S_t (at an iteration t), the largest component C_t^* of $G-S_t$ is only guaranteed to have a small diameter (of $O(\sqrt{n})$); it could contain almost all the vertices. If in the next iteration, we simply apply KPR again on the large component, we might not get anything, i.e., the returned separator is empty. By increasing the current weight of a vertex v to:

$$w_{t+1}(v)\leftarrow w_t(v)+\left\lceil\frac{|T_t(v)|20^3h^6}{\sqrt{n}}\cdot w_t(v)\right\rceil\approx w_t(v)\left(1+\frac{|T_t(v)|20^3h^6}{\sqrt{n}}\right), \tag{2}$$

where $T_t(v)$ is the vertex set of the *subtree rooted at v* and $|T_t(v)|$ is its (unweighted) *size*, the (vertex-weighted) diameter of the component in the next iteration increases. The ceiling in Equation (2) is to guarantee that $w_{t+1}(v)$ is an integer. (In Algorithm 1, we apply BFS to G in every iteration, but we should think of this as applying BFS to the largest component from the previous iteration.) Vertices that have many descendants in T_t have more weights, and therefore, are more likely to be added to the separator in the next iteration. Intuitively, vertices with more descendants are responsible for connecting many other vertices, and therefore should be in the separator. A simplistic yet illuminating example is the star graph: the center of the star will be assigned $\Omega(\sqrt{n})$ weight, which is larger than $6h^2\Delta$, and hence will be added to the separator by KPR in the next iteration. (Note that in general, removing a single or $O(1)$ vertices in a graph of diameter $O(\sqrt{n})$ does not produce small components, e.g., the $\sqrt{n} \times \sqrt{n}$ planar grid.) If we only want to take vertices with many descendants to the separator, then we can simply choose a threshold on the number of descendants, say $\sqrt{n}$, and add a vertex to the separator if the number of its descendants crosses the threshold. Simple thresholding does not work since there is no basis for stopping after $O(h^2)$ iterations of KPR. The second purpose of our weighting scheme, described next, provides such a basis, limiting the number of iterations to $20h^2$.

Suppose that we want to certify that our algorithm fails by constructing a K_h-minor model. We construct a (random) function ϕ that maps each vertex $i \in [h]$ of K_h to a (unique) vertex $\phi(i) \in V$ and each edge $(i, j) \in K_h$ to a path, denoted by $\phi(i, j)$, in G. Two paths $\phi(i, j)$ and $\phi(a, b)$ have to be vertex-disjoint whenever $\{i, j\} \cap \{a, b\} = \emptyset$. For each vertex $i \in K_h$, $\phi(i)$ is a vertex in V chosen uniformly at random. To find a path $\phi(i, j)$ for every edge $(i, j) \in K_h$, we rely on the set of trees $\mathcal{T} = \{T_1, T_2, \ldots, T_{h^2}\}$, which is produced by the algorithm[3] in Algorithm 1 after all iterations. Specifically, edge $(i, j) \in K_h$ is assigned a unique tree $T_{ij} \in \mathcal{T}$, and the path $\phi(i, j)$ is the (only) path between[4] $\phi(i)$ and $\phi(j)$ in T_{ij}. To obtain a K_h-minor model from ϕ, it must have the guarantee that whenever $\{i, j\} \cap \{a, b\} = \emptyset$, the probability of $\phi(i, j) \cap \phi(a, b) = \emptyset$ is high, say $1 - \frac{1}{h^2}$, for the union bound to work. Suppose that T_{ij} is added to $\mathcal{T}$ before T_{ab}. If the subtree $T_{ij}(v)$ rooted at v has a large size, say $n/2$ vertices, then with probability $1/4$, $\phi(i, j)$ will contain v since ϕ maps vertices to V uniformly at random. If in T_{ab}, $T_{ab}(v)$ also has a large size, then $\phi(a, b)$ will also contain v with a good probability, say $1/4$. Thus, in this thought experiment, with a probability $1/16$, $\phi(i, j) \cap \phi(a, b) \neq \emptyset$. (Recall we want this probability to be at most $1/h^2$.) By reweighting $w_t(v)$ after the construction of T_{ij} in iteration t, $w_{t+1}(v)$ will be substantially larger than $w_t(v)$. Thus, in future iterations, if v has accumulated too much weight, it will be added to the separator by KPR. Otherwise, v has not accumulated too much weight, then in future trees, v will have very few descendants, which directly translates to a low collision probability between $\phi(i, j)$ and $\phi(a, b)$. We formalize this intuition via the notion of a *stochastic connector* (Definition 1). The precise weight update rule in Equation (2) allows us to construct a stochastic connector, which in turn gives a K_h-minor model, whenever our algorithm fails to return a small separator. Given that balanced separators are deeply connected to sparsest cuts, we believe that the notion of a stochastic connector may be of broader interest.

1.3 Other Related Work

An orthogonal research direction motivated by the work of Alon, Seymour, and Thomas [AST90a, AST90b] is to minimize the dependency on h in the size of the separator. Recall that their separator has size $O(h^{3/2}\sqrt{n})$. The aforementioned result (Equation (1)) by Plotkin, Rao, and Smith [PRS94] implies a separator of size $O(h\sqrt{\log n}\sqrt{n})$, which is an improvement over $O(h^{3/2}\sqrt{n})$ whenever $h \gg \log(n)$. Kawarabayashi and Reed [KR10, Kaw11] claimed to improve the bound to $O(h\sqrt{n} + f(h))$ by following the deep structure decomposition of minor-free graphs in the graph minor theory of Robertson and Seymour. Recently, Spalding-Jamieson [Spa25] gave a separator of size $O(h\,\mathrm{polylog}(h)\sqrt{n})$ based on a reweighted spectral partitioning technique. The algorithm uses very complicated subroutines, e.g., semidefinite programming and padded decomposition [CF25]; it is unclear if one could implement it in truly quadratic time.

2 Stochastic Connector

A central concept in the analysis of our algorithm is that of a *stochastic connector*, which is essentially a collection of rooted trees that have a certain probabilistic guarantee on the random paths sampled from them. We can show that if a stochastic connector of large size (having many trees) exists, then we can construct a clique minor of large size.

Let T be a tree and u, v be two vertices in T. We denote by $T[a, b]$ *the (unique) path between a and b in T*. Suppose that T is rooted at a vertex r. We denote by $\mathcal{R}(T) = \{T[r, v] : v \in V(T)\}$ the set of all rooted paths in T.

[3] The algorithm produces $20h^2$ trees, and the constant 20 is needed for the probabilistic analysis; we assume a constant 1 for simplicity.

[4] T_{ij} might not include all the vertices of G. But it does include at least $(1 - 1/(10h^2))n$ vertices from G, which is the size of the largest component. Therefore, our probabilistic analysis remains feasible, albeit somewhat more complex.

By $P \sim \mathcal{R}(T)$, we denote a path P sampled uniformly at random, or *u.a.r*, from $\mathcal{R}(T)$. The sampling of P can be realized in linear time by sampling a vertex $v \in V(T)$ u.a.r and returning the path $T[r, v]$.

Definition 1 (Stochastic Connector). *A stochastic connector of size k for a given graph $G = (V, E)$ is a set of k trees $\mathcal{T} = \{T_1, T_2, \dots, T_k\}$ such that:*

1. *$|V(T_i)| \geqslant n - n/(10k)$ for every $i \in [k]$.*
2. *For every $i, j \in [k]$ such that $i \neq j$:*

$$\Pr_{P \sim \mathcal{R}(T_i), Q \sim \mathcal{R}(T_j)}[P \cap Q \neq \emptyset] \leqslant \frac{1}{5k^2} \tag{3}$$

The constants 10 and 5 in Definition 1 look somewhat arbitrary. We choose these constants for the union bound to work, and they better fit our algorithm given in Algorithm 1. Here, we note that a tree in $\mathcal{T}$ might not contain all vertices of G.

Let H be a graph. Let $\mathcal{P}(G)$ be the set of all simple paths in G. An *almost-embedding* of H to G is a function ϕ that maps $V(H) \to V(G)$ and $E(H) \to \mathcal{P}(G)$ such that:

1. for every $e = uv \in E(H)$, $\phi(e)$ is a $\phi(u)$-to-$\phi(v)$ path.
2. for any two edges $e_1 \neq e_2$ of H that do not share an endpoint, $\phi(e_1) \cap \phi(e_2) = \emptyset$.

An object similar to an almost-embedding was considered by Korhonen and Lokshtanov [KL24]. Their techniques show that one can construct a K_t-minor model from an almost embedding (into G) of a graph of sufficient size. We capture this in the following lemma, whose proof follows directly from the ideas of [KL24], but for completeness we include a proof in Section B.

Lemma 2 (Corresponds to Lemma 4.3 in [KL24]). *Let G be a connected graph with m edges. For every $t \geqslant 1$, there is a graph H with $|V(H)| + |E(H)| \leqslant 3t^2$ such that an almost-embedding of H can be turned into a K_t-minor model of G in $O(t^2 m)$ time.*

The key result of this section is the following lemma, showing that the existence of a large stochastic connector implies the existence of a large clique minor.

Lemma 3. *Given a stochastic connector $\mathcal{T}$ of size k in a connected graph G with m edges, for any t such that $20t^2 \leqslant k$, we can construct a K_t-minor model of G in $O(km)$ time with probability at least $2/5$.*

Proof: By Lemma 2, it suffices to construct an almost-embedding ϕ of any graph H with $|V(H)| + |E(H)| \leqslant 20t^2$ into G whenever $k \geqslant 20t^2$. Our construction is randomized and as follows:

1. For every vertex $v \in V(H)$, we assign $\phi(v)$ to be a vertex in $V(G)$ chosen u.a.r.
2. For each edge $e = uv \in E(H)$, we associate e with a distinct tree in $\mathcal{T}$, denoted by T_e. This is possible since $k \geqslant |E(H)|$. Then we assign $\phi(e)$ to be the path $T_e[u, v]$ in T_e. (It could be that u or v is not in T_e, and in this case, we set $\phi(e) = \emptyset$.)

Now we bound the probability that ϕ is valid. We say that two edges are *adjacent* if they share an endpoint, and *non-adjacent* otherwise. We say that ϕ is *invalid* when at least one of the following events happens:

- **Event A:** there exists $e = uv \in E(H)$ such that $\{\phi(u), \phi(v)\} \not\subseteq V(T_e)$.
- **Event B:** there exist two non-adjacent e_1, e_2 such that, conditioning on $\bar{A}$, $\phi(e_1) \cap \phi(e_2) \neq \emptyset$.

By the first property of $\mathcal{T}$ in Definition 1, for any tree $T \in \mathcal{T}$ and any vertex $v \in H$, $\Pr[\phi(v) \notin V(T)] \leqslant 1/(10k)$. Thus, for every edge uv, by the union bound, $\Pr[\{\phi(u), \phi(v)\} \not\subseteq V(T_e)] \leqslant 2/(10k) = 1/(5k)$. Then by union bound over at most $20t^2 \leqslant k$ edges of H, we have:

$$\Pr[A] \leqslant k \cdot 1/(5k) \leqslant 1/5 \tag{4}$$

To bound $\Pr[B]$, for every edge $e = uv$, let $\hat{\phi}(e) = T_e[u, r_e] \cup T_e[r_e, v]$ where r_e is the root of T_e. Then for any two non-adjacent edges $e_1 = (u_1, v_1)$ and $e_2 = (u_2, v_2)$, we have:

$$\begin{aligned}
\Pr[\phi(e_1) \cap \phi(e_2) \neq \emptyset] &\leqslant \Pr[\hat{\phi}(e_1) \cap \hat{\phi}(e_2) \neq \emptyset] \\
&\leqslant \sum_{x \in \{u_1, v_1\}} \sum_{y \in \{u_2, v_2\}} \Pr[T_{e_1}[x, r_{e_1}] \cap T_{e_2}[y, r_{e_2}] \neq \emptyset] \\
&\leqslant \frac{4}{5k^2} \qquad \text{(by union bound and Equation (3))}
\end{aligned} \tag{5}$$

Thus, by taking the union bound over all non-adjacent pairs, we have:

$$\Pr[B] \leqslant \frac{k^2}{2} \cdot \frac{4}{5k^2} \leqslant 2/5 \tag{6}$$

By union bound, and Equation (4) and Equation (6), we get:

$$\Pr[\phi \text{ is invalid}] \leqslant \Pr[A] + \Pr[B] \leqslant 3/5$$

Thus, with probability at least $2/5$, ϕ is a valid almost-embedding. Since finding a path in each tree in $\mathcal{T}$ takes $O(kn)$ time, by Lemma 2, we can construct a K_t-minor model in time $O(k(n+m)) = O(km)$. □

3 Constructing Balanced Separators

In this section, we prove Theorem 2, which we restate below for convenience.

Theorem 2. *Let G be any given graph with n vertices, and $h > 0$ be a parameter. There is an algorithm that runs in deterministic $O(\mathrm{poly}(h)n)$ time and outputs either:*

- *a balanced separator of size $O(\mathrm{poly}(h)\sqrt{n})$ or*
- *$\perp$ if G contains a K_h as a minor.*

Furthermore, in the latter case, we can output a K_h-minor model of G with probability at least $1/2$ in an additional randomized $O(\mathrm{poly}(h)n)$ time.

The algorithm for constructing a balanced separator in Algorithm 1 assumes that the input graph is K_h-minor-free. Here in Theorem 2, we do not make this assumption; instead, we want to output a K_h-minor when we fail to find a good balanced separator. We do so by making two simple changes to Algorithm 1. First, the KPR decomposition in every iteration t returns a tuple (C_t^*, S_t, M_t) where M_t is a K_h-minor model whenever it fails to find a small separator S_t such that the largest component C_t^* has small weak diameter. By Lemma 1, either $M_t = \emptyset$—in this case, the KPR returns the pair (C_t^*, S_t)—or $M_t \neq \emptyset$, and in this case M_t is a K_h-minor model. Second, instead of returning a separator at the end of the algorithm[5] as in Algorithm 1, we run the algorithm $\textsc{FindMinor}(G, \{T_1, T_2, \ldots, T_k\})$ with $k = 20h^2$ to construct a K_h-minor by Lemma 3. Note that Lemma 3 requires the set of trees $\{T_1, T_2, \ldots, T_k\}$ to be a stochastic connector. Our analysis in this section shows that this indeed is the case. Finally, as noted in Section 1.1, to reduce the running time from $O(\mathrm{poly}(h)m)$ to $O(\mathrm{poly}(h)n)$, we construct a K_h-minor model directly as formally stated in the following lemma, whose proof will be given in Section 4.

Lemma 4. *There is a deterministic algorithm that, given any n-vertex graph G with at least $100h^2n$ edges, reads the first $100h^2n$ edges of G and outputs a K_h-minor model of G in $O(h^6n)$ time.*

Now we have all the ideas needed for proving Theorem 2.

[5] If G is K_h-minor-free, then the analysis in this section shows that the algorithm will never reach this line.

FindSeparator$(G=(V,E))$:
    if $|E| \geqslant 100h^2n$ ⟨⟨G is dense⟩⟩
        return K_h-minor model from Lemma 4
    $w_1(v) \leftarrow 40$ for every $v \in V$
    $S \leftarrow \emptyset$ ⟨⟨The separator⟩⟩
    $k \leftarrow 20h^2$
    for $t \leftarrow 1$ to k ⟨⟨$O(h^2)$ iterations⟩⟩
        $(C_t^*, S_t, M_t) \leftarrow \text{KPR}(\langle G, w_t\rangle, \lfloor \sqrt{n}/6h^2 \rfloor, h)$ ⟨⟨KPR with $\Delta = \lfloor \sqrt{n}/6h^2 \rfloor$, applying BFS h times⟩⟩
        if $M_t \neq \emptyset$ ⟨⟨K_h-minor model found⟩⟩
            return M_t
        $S \leftarrow S \cup S_t$
        if $|C_t^*| \leqslant (1 - \frac{1}{10k})n$ ⟨⟨The current separator is $(1-\frac{1}{10k})$-balanced⟩⟩
            return S
        $c_t \leftarrow$ an arbitrary vertex in C_t^*
        $T_t \leftarrow \text{BFS}(\langle G, w_t\rangle, c_t, \sqrt{n})$ ⟨⟨BFS truncated at radius $\sqrt{n}$⟩⟩
        for every $v \in V(T_t)$
            $w_{t+1}(v) \leftarrow w_t(v) + \lceil \frac{|T_t(v)|k^3}{\sqrt{n}} \cdot w_t(v) \rceil$ ⟨⟨Reweighting vertices for next iteration⟩⟩
    return FindMinor$(G, \{T_1, T_2, \ldots, T_k\})$ ⟨⟨Find K_h-minor model by Lemma 3⟩⟩

Algorithm 2. An algorithm that finds a $(1 - \frac{1}{200h^2})$-balanced separator with size $O(\text{poly}(h)\sqrt{n})$ of G if exists, or outputs a K_h-minor model otherwise.

Proof of Theorem 2: By Lemma 4 we can assume that $m = O(h^2 n)$ since we can find a K_h-minor model in $O(h^6 n)$ time whenever $m \geqslant 100h^2 n$.

If the algorithm returns a separator S, in Lemma 6 of Section 3.1, we will show that $|S| = O(h^{11}\sqrt{n})$. Otherwise, we will show in Lemma 7 of Section 3.1 that the set of trees $\{T_1, T_2, \ldots, T_k\}$ found by the algorithm is a stochastic connector. Thus, the algorithm will return a K_h-minor model either in one of the iterations or by calling FindMinor$(G, \{T_1, T_2, \ldots, T_k\})$.

If we are not required to output a minor, then in the last line of the algorithm, we return $\perp$ instead of calling FindMinor$(G, \{T_1, T_2, \ldots, T_k\})$. The running time, as we will show in Lemma 9, is $O(h^3 m + h^{11} n) = O(h^{11} n)$.

If we are required to output a minor, then by Lemma 3, the total running time is:

$$O(h^3 m + h^{11} n) + O(h^2 m) = O(h^3 m + h^{11} n) = O(h^{11} n).$$

Also by Lemma 3, the algorithm is randomized and succeeds with a probability of at least $2/5$.

Since the separator output by Algorithm 2 is only $(1 - \frac{1}{200h^2})$-balanced, we need to run the algorithm $O(h^2)$ times to get a $2/3$-balanced separator. The total separator size and running time are $|S| = O(h^{13}\sqrt{n})$ and $O(h^{13} n)$, respectively. □

3.1 Correctness

Here, we only consider weight functions on vertices, not on edges. If $\langle G, w\rangle$ is a graph with a weight function w, we say that $X \subseteq V(G)$ has weak diameter at most D if $d_{\langle G,w\rangle}(u,v) \leqslant D$ for every $u, v \in X$.

Let $E_{t,v}$ be the event that a random rooted path $P \sim \mathcal{R}(T_t)$ contains v. (T_t is the tree at iteration t.) We first show that the algorithm maintains five invariants.

Lemma 5. *Algorithm 2 maintains the following invariants at every iteration t if it does not return a K_h-minor model or a separator S:*

- ***Invariant 1:*** $\Pr[E_{i,v}] \leqslant (20k^3\sqrt{n})^{-1} w_{t+1}(v)$ *for every $i \leqslant t$ and every v.*
- ***Invariant 2:*** $\Pr[E_{i,v} \cap E_{j,v}] \leqslant (10k^6 n)^{-1} w_{t+1}(v)$ *for every $i \neq j \leqslant t$ and every v.*
- ***Invariant 3:*** $\sum_{v \in V} w_t(v) \leqslant (40 + (k^3+1)(t-1))n$.

- ***Invariant 4:*** $|V(T_t)| \geqslant (1 - \frac{1}{10k})n$.
- ***Invariant 5:*** $w_t(v) \geqslant 40$ *for every* v.

Proof: Initially, $t = 1$ and the algorithm sets $w_1(v) = 40$ for every v. Since $w_t(v)$ only increases as t increases, Invariant 5 holds. Herein, we focus on four other invariants.

For Invariant 4, we first note that since the algorithm does not return a K_h-minor model or a separator S at iteration t, it must be that:

- The largest connected component C_t^* of $G - S_t$ has size at least $(1 - 1/10k)n$.
- C_t^* has weak (vertex-weighted) diameter $6h^2\Delta \leqslant \sqrt{n}$ by Lemma 1.

Thus, for any vertex $c_t \in V(C^*)$, every other vertex $v \in V(C_t^*)$, $d_{\langle G, w_t \rangle}(c_t, v) \leqslant \sqrt{n}$. Thus, the BFS tree rooted at c_t truncated at radius $\sqrt{n}$ contains all vertices of C_t^*. Thus, we have:

$$|V(T_t)| \geqslant |V(C_t^*)| \geqslant (1 - \frac{1}{10k})n \,, \tag{7}$$

giving Invariant 4.

To show Invariant 3, let $W_t = \sum_{v \in V} w_t(v)$. By induction, we assume that $W_t \leqslant (40 + (k^3+1)(t-1))n$. We now show that $W_{t+1} \leqslant W_t + (k^3+1)n \leqslant (40 + (k^3+1)t)n$ by a simple charging scheme. Observe that, by the definition of $w_{t+1}(v)$:

$$w_{t+1}(v) \leqslant w_t(v) \left(1 + \frac{|T_t(v)|k^3}{\sqrt{n}} \right) + 1, \tag{8}$$

For each vertex $v \in V(T_t)$, the algorithm add at most $\frac{|T_t(v)|k^3}{\sqrt{n}} \cdot w_t(v) + 1$ to $w_t(v)$. We will charge an amount of $(k^3/\sqrt{n})w_t(v)$ to each vertex $u \in T_t(v)$. Since u is only charged by its ancestors, the total charge of each vertex $u \in V(G)$ at iteration t is:

$$\sum_{v \text{ ancestor of } u \text{ in } T_t} \frac{w_t(v)k^3}{\sqrt{n}} \leqslant \frac{w_t(T_t[u, c_t])k^3}{\sqrt{n}} \leqslant k^3$$

since T_t has (vertex-weighted) radius at most $\sqrt{n}$. Thus, the total weight increase of all vertices at iteration t is at most $n \cdot k^3 + n$ (including the $+1$ for each vertex in Equation (8)), giving Invariant 3.

For Invariant 1, we note that $|V(T_t)| \geqslant n/2$ by Invariant 4. Furthermore, by the weight update rule, we have:

$$w_{t+1}(v) \geqslant \frac{|T_t(v)|k^3}{\sqrt{n}} w_t(v) \Rightarrow |T_t(v)| \leqslant \frac{\sqrt{n}\, w_{t+1}(v)}{k^3\, w_t(v)} \tag{9}$$

Now we observe that if $v \notin V(T_t)$, then $\Pr[E_{t,v}] = 0$, and otherwise:

$$\begin{aligned} \Pr[E_{t,v}] &= \frac{|T_t(v)|}{|V(T_t)|} \leqslant \frac{2|T_t(v)|}{n} \qquad \text{(since } |V(T_t)| \geqslant n/2) \\ &\leqslant \frac{2\sqrt{n}\, w_{t+1}(v)}{k^3 n\, w_t(v)} \qquad \text{(by Equation (9))} \\ &\leqslant \frac{w_{t+1}(v)}{20k^3\sqrt{n}} \qquad \text{(since } w_t(v) \geqslant 40 \text{ by invariant 5)}, \end{aligned} \tag{10}$$

By induction, for any $i \leqslant t-1$, we have

$$\Pr[E_{i,v}] \leqslant \frac{w_t(v)}{20k^3\sqrt{n}} \leqslant \frac{w_{t+1}(v)}{20k^3\sqrt{n}},$$

since $w_t(v)$ only increases as t increases. Thus, Invariant 1 holds.

It remains to show Invariant 2. Observe that $E_{i,v}$ and $E_{t,v}$ are independent for any $i \leqslant t-1$. Thus, we have:

$$
\begin{aligned}
\Pr[E_{i,v} \cap E_{t,v}] = \Pr[E_{i,v}]\Pr[E_{t,v}] &\leqslant \frac{w_t(v)}{20k^3\sqrt{n}} \frac{|T_t(v)|}{|V(T_t)|} && \text{(by Invariant 1 and } i \leqslant t-1\text{)} \\
&\leqslant \frac{w_t(v)}{20k^3\sqrt{n}} \frac{2\sqrt{n}\, w_{t+1}(v)}{k^3 n\, w_t(v)} && \text{(by Equation (9) and } |V(T_t)| \geqslant n/2\text{)} \\
&\leqslant \frac{w_{t+1}(v)}{10k^6 n},
\end{aligned}
$$

giving Invariant 2. □

Next, we bound the size of the separator if the algorithm returns one.

Lemma 6. *If Algorithm 2 returns a separator S, then $|S| \leqslant O(h^{11}\sqrt{n})$.*

Proof: Let $W_t = \sum_{v \in V} w_t(v)$. By Lemma 1, we have:

$$
\begin{aligned}
|S_t| &= O(hW_t/(\sqrt{n}/h^2)) && \text{(since } \Delta = \lfloor \sqrt{n}/(6h^2) \rfloor\text{)} \\
&= O(h^3k^3tn/\sqrt{n}) && \text{(by Invariant 3 in Lemma 5)} \\
&\leqslant O(h^3k^4n/\sqrt{n}) && \text{(since } t \leqslant k\text{)} \\
&= O(h^{11}\sqrt{n}), && \text{(since } k = O(h^2)\text{)}
\end{aligned} \tag{11}
$$

as claimed. □

Finally, we show that

Lemma 7. *Suppose that the algorithm does not return a separator S after k iterations, then $\mathcal{T} := \{T_1, \ldots, T_k\}$ is a stochastic connector.*

Proof: We verify all the properties of $\mathcal{T}$ in Definition 1. Observe that the size bound follows directly from Invariant 4 in Lemma 5. It remains to bound the collision probability of two random paths by $1/(5k^2)$. Consider two trees T_i and T_t where $i < t$ formed in iteration i and t, respectively. By Invariant 3, we have:

$$
\sum_{v \in V} w_{t+1}(v) \leqslant (40 + (k^3+1)t)n \leqslant 2k^4 n \tag{12}
$$

since $t \leqslant k$ and $k^4 = 20^4 h^8 \geqslant 40 + k$ as $h \geqslant 1$. By taking the union bound from the probability in Invariant 2, we have:

$$
\Pr_{P \sim \mathcal{R}(T_i), Q \sim \mathcal{R}(T_t)}[P \cap Q \neq \emptyset] \leqslant \sum_{v \in V} \frac{w_{t+1}(v)}{10k^6 n} \overset{\text{eq. (12)}}{\leqslant} \frac{2k^4 n}{10k^6 n} = \frac{1}{5k^2},
$$

as desired. □

3.2 Running time

We first bound the running time of (vertex-weighted) BFS.

Lemma 8. *Let $\langle G = (V, E), w \rangle$ be a connected vertex-weighted graph with m edges. Suppose that the vertex weights are integers. Let $W = \sum_{v \in V} w(v)$. In $O(m + W)$ time, we can compute a node-weighted shortest-path tree rooted at any vertex $c \in V$.*

Proof: First, we apply the textbook reduction (for maxflow with vertex capacities) to reduce to computing the shortest-path tree in a directed graph with edge weights in $\{0, 1\}$: split every vertex v into v_{in} and v_{out}, add a directed path from v_{in} to v_{out} of total length $w(v)$ (so each edge has length 1), and for every edge $uv \in E(G)$, we add two arcs $(v_{out} \to u_{in})$ and $(u_{out} \to v_{in})$ of length 0 each. Let $\hat{G}$ be the resulting graph. Observe that $\hat{G}$ has $O(m+W)$ directed edges. Now we can find a shortest-path tree of $\hat{G}$ from c_{in} by doing a breadth-first search in time $O(m+W)$. Then in the same time, we can convert the BFS tree of $\hat{G}$ into a node-weighted shortest-path tree rooted at $c \in V$. □

Now we bound the running time of Algorithm 2.

Lemma 9. *The running time of Algorithm 2, excluding the running time of* FINDMINOR *(the last line), is* $O(h^3 m + h^{11} n)$

Proof: Let $W_t = \sum_t w_t(v)$. By Lemma 1, the running time of the KPR decomposition in iteration t is $O(h(m+W_t))$. The running time to compute the (vertex-weighted) BFS is $O(m+W_t)$ by Lemma 8. Thus, the total running time is:

$$\sum_{t=1}^{k} O(h(m+W_t)) = O(hkm + h\sum_{t=1}^{k} W_t) = O(hkm + hk^5 n) = O(h^3 m + h^{11} n), \tag{13}$$

as desired. □

4 Reduction to Sparse Graphs

The goal of this section is to show Lemma 4, restated below, in order to avoid any dependence on the number m of edges of Theorem 2.

Lemma 4. *There is a deterministic algorithm that, given any n-vertex graph G with at least $100h^2 n$ edges, reads the first $100h^2 n$ edges of G and outputs a K_h-minor model of G in $O(h^6 n)$ time.*

To that end, we could combine the results in [DHJ+13] and [AKS23]. Dujmović et al. [DHJ+13] describe an algorithm that finds in $O(\text{poly}(h)n + 2^{\tilde{O}(h^2)})$ time a K_h-minor model in any n-vertex graph with average degree at least $10h\sqrt{\log h}$. Apart from the final brute-force search on an $O(h^2 \log h)$-vertex graph, their algorithm runs in time $O(\text{poly}(h)n)$. As this graph still has large enough average degree, one can thus substitute this last step with an effective version of the short proof of Alon et al. [AKS23] of the Kostochka–Thomason bound. One can indeed easily derive a randomized polynomial-time algorithm from their proof.

Instead, we will give a simple proof with a worst (quadratic) bound on the average degree. This way our algorithm is deterministic and self-contained. The first step is to efficiently find, in any graph of average degree $2d$, a minor with minimum degree at least d and only $\Theta(d)$ vertices.

We denote the minimum degree of G by $\delta(G)$. Given a graph H with vertex set $\{v_1, \ldots, v_h\}$, an *H-minor model* of a graph G is a collection of h vertex-disjoint connected subgraphs $\{C_1, C_2, \ldots, C_h\}$ of G such that for every two subgraphs C_i, C_j when $(v_i, v_j) \in E(H)$, there exists an edge $uv \in E(G)$ with $u \in V(C_i)$ and $v \in V(C_j)$.

Lemma 10. *For any positive integer d, there is an algorithm that on any n-vertex graph G with $|E(G)| = dn$ outputs an H-minor model of G with $|V(H)| \leqslant 2d$ and $\delta(H) \geqslant d$, and runs in time $O(d^3 n)$.*

Proof: The algorithm goes through a strictly decreasing chain of graphs for the minor relation. In particular, it builds at most $|E(G)| + |V(G)| = (d+1)n$ (distinct) graphs. In addition to the current graph G', it maintains

- an array t of length n such that $t[i]$ contains a doubly linked list of vertices with degree exactly i in G',
- an array s of length n such that for any $v \in V(G)$, $s[v]$ is a pointer to the occurrence of v in t, or $\perp$ if $v \notin V(G')$,
- $\delta := \delta(G')$ the minimum degree of G', and

- an array f of length n such that for any $v \in V(G')$, $f[v]$ contains the list of original vertices of G contracted into v.

From a representation of G by adjacency lists, s, t, f and δ can be initialized (for $G' = G$) in time $O(dn)$. For G', we use a data structure that supports edge deletion and edge addition in $O(1)$ time, and deletion of vertex $v \in V(G')$ in time $O(d_{G'}(v))$. We define the positive measure $\mu(G') := 2|E(G')| + |V(G')| + (n - \delta(G')) \leqslant (2d+2)n$.

We iteratively apply the following steps:

1. Let u be the first vertex in $t[\delta]$.
2. If $\delta < d$, simply delete u from G',
3. else if $\delta > 2d$, delete all but $2d$ edges incident to u (arbitrarily),
4. else if there is an edge $uv \in E(G')$ such that $|N_{G'}(u) \cap N_{G'}(v)| < d$, then contract uv, by deleting u and adding the edges vw for every $w \in N_{G'}(u) \setminus N_{G'}(v)$,
5. else break the loop and output $H := G'[N_{G'}(u)], f_{|N_{G'}(u)}$ (where $f_{|Z}$ is the restriction of f to Z).

If second item holds, it takes $O(d_{G'}(u))$ time to delete u from G' and update s and t, and $O(\delta(G') - \delta(G' - \{u\}) + 1)$ time to update δ. Meanwhile, μ drops by $2d_{G'}(u) + 1 + \delta(G') - \delta(G' - \{u\})$.

If the third item holds, it takes $O(\delta - 2d)$ time to delete the edges from G' and update s and t, and $O(1)$ time to set the new δ. The measure μ decreases by $2(\delta - 2d) - (\delta - 2d) = \delta - 2d$.

Checking if the 4th item holds takes $O(d^2)$ time since after the third step, u has at most $2d$ neighbors in G', whereas in $O(d)$ time the contraction of uv can be emulated by deleting u and adding fewer than d edges incident to v. Note that the minimum degree cannot decrease by more than one, and the new graph has one less vertex and at least one less edge. Therefore, μ drops by at least $2 + 1 - 1 = 2$. Again, we update s, t, f, δ in $O(d)$ time.

The most time invested per one unit drop of μ is $O(d^2)$, in the third item; hence the total running time $O(d^3 n)$. We should now check that the algorithm indeed reaches the fourth item, and that the output graph H has the required properties.

Note that initially (in G) the average degree is $2d$, and each operation of the first three items cannot produce a graph with average degree less than d from a graph with average degree at least d. In particular, as $|E(G')| + |V(G')|$ strictly decreases and G' cannot have less than $d+1$ vertices (as otherwise its average degree would be smaller than d), the algorithm eventually returns $H := G'[N_{G'}(u)]$ with $d \leqslant d_{G'}(u) \leqslant 2d$ and $|N_{G'}(u) \cap N_{G'}(v)| \geqslant d$ for every $v \in N_{G'}(u)$. Therefore, H has at most $2d$ vertices and minimum degree at least d. ◻

We now greedily build, in the minor given by the previous lemma, a ($\leqslant 2$)-subdivision of an appropriately large clique. If this fails, we obtain a subgraph on $\Theta(d)$ vertices in which every pair of vertices shares $\Theta(d)$ neighbors.

Lemma 11. *There is an algorithm that given any graph H with at most $2d$ vertices and minimum degree at least d, outputs:*

- *a $K_{\lfloor \sqrt{d}/10 \rfloor}$-minor model of H or*
- *a subgraph H' of H with at most $1.02d$ vertices and minimum degree at least $0.94d$,*

and runs in $O(d^3)$ time.

Proof: Let $X \subseteq V(H)$ be an arbitrary subset of size $s := \lfloor \sqrt{d}/10 \rfloor$. We denote by $\tilde{H}$ the current graph, initialized by $\tilde{H} := H$. We in fact try to build K_s as a topological minor whose branching vertices are the vertices of X, and every edge is subdivided at most twice. For each pair $x \neq y \in X$, if there is an x–y path P on at most 3 edges in $\tilde{H} - (X \setminus \{x, y\})$, add the 0, 1, or 2 internal vertices of P to the topological minor and remove them from $\tilde{H}$. Else, output $H' := \tilde{H}[(N_{\tilde{H}}(x) \setminus X) \cup \{x\}]$ and terminate.

First, note that if none of the $\binom{s}{2}$ iterations terminates by outputting a subgraph H', then we have eventually built a K_s-minor model. Checking if there is such a path of length at most 3 between x and y, and if so, removing its internal vertices, can be done in $O(d^2)$ time. We repeat this at most $\binom{s}{2} = O(d)$ times, hence the total running time $O(d^3)$.

We now show that if the algorithm outputs a subgraph H', it has the required properties. Overall, as we remove at most $2\binom{s}{2}$ vertices from H, the minimum degree in every built subgraph $\tilde{H}$ is at least $d-2\binom{s}{2} \geqslant 0.99d$. Let $x \neq y \in X$ be the (first) pair such that the algorithm outputs a subgraph H'. Let $A := (N_{\tilde{H}}(x) \setminus X) \cup \{x\}$ and $B := (N_{\tilde{H}}(y) \setminus X) \cup \{y\}$. In particular, $|A| > 0.99d - s > 0.98d$ and $|B| > 0.99d - s > 0.98d$. As x and y are by assumption at distance greater than 3, A and B are disjoint and no vertex of A has a neighbor in B. Therefore, every vertex of A has at least $0.98d - 0.04d = 0.94d$ neighbors in A. Indeed, $|V(\tilde{H})| \leqslant 2d$ so there are at most $2d - 2 \cdot 0.98d = 0.04d$ vertices in $V(\tilde{H}) \setminus (A \cup B)$. Finally, as $A \cap B = \emptyset$ and $|B| > 0.98d$, it holds that $|A| = |V(H')| \leqslant 1.02d$. ☐

In the very dense graphs given by the previous lemma, it is easy to find the 1-subdivision of a large clique as a subgraph.

Lemma 12. *Given a graph H' with at most $1.02d$ vertices and minimum degree at least $0.94d$, a $K_{\lfloor \sqrt{d}/10 \rfloor}$-minor model of H' can be found in time $O(d^2)$.*

Proof: Take an arbitrary set $X \subseteq V(H')$ of size $s := \lfloor \sqrt{d}/10 \rfloor$. We find the 1-subdivision of K_s as a subgraph of H' with branching vertices in X. For each pair $x \neq y \in X$, pick a common neighbor of x and y outside X, add it to the topological minor, and remove it from the graph.

This can be done in time $O(d^2)$. We just need to argue that every pair has a common neighbor outside X in the remaining graph. In total, we remove $\binom{s}{2}$ vertices. Thus, at any point, the minimum degree is at least $0.94d - \binom{s}{2} > 0.93d$. Therefore, x and y have at least $0.93d - s > 0.92d$ neighbors outside X. As the total number of vertices is at most $1.02d$, this implies the existence of a shared neighbor outside X. ☐

We can now prove the main lemma of the section.

Proof of Lemma 4: We apply Lemma 10 to the spanning subgraph of the input graph G made by its first $100h^2n$ edges, and $d := 100h^2$. This yields an H-minor model of G such that H satisfies the preconditions of Lemma 11. The latter lemma either returns a K_h-minor model (and we are done), or a subgraph H' of H satisfying the preconditions of Lemma 12. In turn, this lemma yields a K_h-minor model. Unpacking the K_h-minor model in G is straightforward via the contraction array $f_{|V(H)}$. The overall running time is $O(d^3n + d^3 + d^2) = O(h^6n)$. ☐

5 Conclusion

In this paper, we gave the first algorithm that finds a balanced separator of size $O(\sqrt{n})$ in $O(n)$ time in any graph class excluding a minor. If the smallest excluded minors have h vertices (equivalently, K_h is excluded), the multiplicative dependence on h in both the separator size and the running time is polynomial, and more precisely, h^{13}. There are a few places where we can further optimize the exponent of this polynomial. For example, in Lemma 3, we could apply the Lovász local lemma instead of the union bound to reduce the value of k in Algorithm 2. If randomness is allowed, we could use the probabilistic construction of [AKS23], which has a smaller dependence on h. However, these optimizations would complicate our construction.

These optimizations, if implemented, are unlikely to give us running time, say, $O(h \operatorname{polylog}(h)\, n)$. For bounded-degree graphs, we can trade running time for separator size: we can construct a separator of size $O(h^{26}\sqrt{n})$ in $O(n)$ time. The idea is rather simple: we contract connected subgraphs of size $\Theta(h^{13})$ to get a new graph H of size $O(n/h^{13})$, apply our separator algorithm in Theorem 2 to get a balanced separator S_H, and then uncontract vertices in S_H to get a balanced separator of G. The details are given in Section C.

Ideally, we would want an algorithm with running time $O(h \operatorname{polylog}(h)\, n)$ producing a separator of size $O(h\sqrt{n})$. Such an algorithm seems to require substantially more (complicated) ideas. We leave it as an open problem for future work.

Acknowledgement. This research has been initiated/conducted at the AlgUW workshop (Bedlewo 09.2025), supported by the Excellence Initiative – Research University (IDUB) funds of the University of Warsaw. H.L. is supported by NSF grant CCF-2517033 and NSF CAREER Award CCF-2237288. É.B. has been supported by the French National Research Agency through the project TWIN-WIDTH with reference number ANR-21-CE48-0014. T.K. is supported by the European Union under Marie Skłodowska-Curie Actions (MSCA), project no. 101206430 Funded by the European Union, and by the VILLUM Foundation, Grant Number 54451, Basic Algorithms Research Copenhagen (BARC). T.M. was supported by the Polish National Science Centre SONATA-17 grant number 2021/43/D/ST6/03312.

References

[AKS23] Noga Alon, Michael Krivelevich, and Benny Sudakov. Complete minors and average degree: A short proof. *Journal of Graph Theory*, 103(3):599–602, 2023. doi:10.1002/JGT.22937.

[AST90a] Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for graphs with an excluded minor and its applications. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, STOC '90, pages 293–299, 1990. Conference version of [AST90b]. doi:10.1145/100216.100254.

[AST90b] Noga Alon, Paul Seymour, and Robin Thomas. A separator theorem for nonplanar graphs. *Journal of the American Mathematical Society*, 3(4):801–808, 1990. Announced at STOC '90. doi:10.1090/s0894-0347-1990-1065053-0.

[BLR10] Punyashloka Biswal, James R. Lee, and Satish Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *J. ACM*, 57(3), 2010. Announced at FOCS '08. doi:10.1145/1706591.1706593.

[CF25] Jonathan Conroy and Arnold Filtser. How to protect yourself from threatening skeletons: Optimal padded decompositions for minor-free graphs. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, STOC '25, pages 2281–2292, 2025. doi:10.1145/3717823.3718252.

[DHJ+13] Vida Dujmović, Daniel J. Harvey, Gwenaël Joret, Bruce Reed, and David R. Wood. A linear-time algorithm for finding a complete graph minor in a dense graph. *SIAM Journal on Discrete Mathematics*, 27(4):1770–1774, 2013. doi:10.1137/120866725.

[GHT84] John R Gilbert, Joan P Hutchinson, and Robert Endre Tarjan. A separator theorem for graphs of bounded genus. *Journal of Algorithms*, 5(3):391–407, 1984. doi:10.1016/0196-6774(84)90019-1.

[Kaw11] Kenichi Kawarabayashi. A separator theorem in minor-closed class of graphs, 2011. Talk at KAIST Graph Theory Day. Accessed: 2025-11-03. URL: https://youtu.be/xTEwQ6KgBdU?si=ZFII4GZWAIC_ywfj&t=1470.

[KL24] Tuukka Korhonen and Daniel Lokshtanov. Induced-minor-free graphs: Separator theorem, subexponential algorithms, and improved hardness of recognition. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '24, pages 5249–5275, 2024. doi:10.1137/1.9781611977912.188.

[KM24] Philip N Klein and Shay Mozes. Optimization algorithms for planar graphs. 2024. Book draft, available at https://planarity.org/.

[KMR08] Ken-ichi Kawarabayashi, Bojan Mohar, and Bruce Reed. A simpler linear time algorithm for embedding graphs into an arbitrary surface and the genus of graphs of bounded tree-width. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '08, pages 771–780. IEEE, 2008. doi:10.1109/FOCS.2008.53.

[Kos82] Alexandr Kostochka. A lower bound for the hadwiger number of a graph as a function of the average degree of its vertices. *Diskret. Analiz. Novosibirsk*, 38:37–58, 1982.

[KPR93] Philip Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '93, pages 682–690, 1993. doi:10.1145/167088.167261.

[KR10] Ken-ichi Kawarabayashi and Bruce Reed. A separator theorem in minor-closed classes. In *Proceedings of the 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 153–162. IEEE, 2010. doi:10.1109/FOCS.2010.22.

[KRV09] Rohit Khandekar, Satish Rao, and Umesh Vazirani. Graph partitioning using single commodity flows. *J. ACM*, 56(4), 2009. Announced at STOC '06. doi:10.1145/1538902.1538903.

[KS15] Ken-ichi Kawarabayashi and Anastasios Sidiropoulos. Beyond the euler characteristic: Approximating the genus of general graphs. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, pages 675–682, 2015. doi:10.1145/2746539.2746583.

[LR10] Richard Lipton and Ken Regan. Galactic algorithms, 2010. Accessed: 2025-10-26. URL: https://rjlipton.com/2010/10/23/galactic-algorithms/.

[LT79] Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979. doi:10.1137/0136016.

[LT80] Richard J. Lipton and Robert Endre Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980. Announced at FOCS '77. doi:10.1137/0209046.

[Mil86] Gary L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *Journal of Computer and System Sciences*, (3):265–279, 1986. Announced at STOC '84'. doi:10.1016/0022-0000(86)90030-9.

[Moh96] Bojan Mohar. Embedding graphs in an arbitrary surface in linear time. In *Proceedings of the 28th annual ACM symposium on Theory of computing*, STOC '96, pages 392–397, 1996. doi:10.1145/237814.237986.

[NO12] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity: Graphs, Structures, and Algorithms*. 2012. doi:10.1007/978-3-642-27875-4.

[Pen16] Richard Peng. Approximate undirected maximum flows in $O(m\text{polylog}(n))$ time. In *Proceedings of the 27th annual ACM-SIAM symposium on Discrete algorithms*, SODA '16, pages 1862–1867, 2016. doi:10.1137/1.9781611974331.ch130.

[PRS94] Serge Plotkin, Satish Rao, and Warren D. Smith. Shallow excluded minors and improved graph decompositions. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '94, pages 462–470, 1994.

[RS03] Neil Robertson and Paul D. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43–76, 2003. doi:10.1016/S0095-8956(03)00042-X.

[RST14] Harald Räcke, Chintan Shah, and Hanjo Täubig. Computing cut-based hierarchical decompositions in almost linear time. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete algorithms*, SODA '14, pages 227–238, 2014. doi:10.1137/1.9781611973402.17.

[RW09] Bruce Reed and David R. Wood. A linear-time algorithm to find a separator in a graph excluding a minor. *ACM Transactions on Algorithms*, 5(4):1–16, 2009. doi:10.1145/1597036.1597043.

[She13] Jonah Sherman. Nearly maximum flows in nearly linear time. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, pages 263–269, 2013. doi:10.1109/FOCS.2013.36.

[Spa25] Jack Spalding-Jamieson. Reweighted spectral partitioning works: Bounds for special graph classes. *arXiv preprint arXiv:2506.01228*, 2025. URL: https://arxiv.org/abs/2506.01228.

[Tar72] Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972. doi:10.1137/0201010.

[Tho89] Carsten Thomassen. The graph genus problem is np-complete. *Journal of Algorithms*, 10(4):568–576, 1989. doi:10.1016/0196-6774(89)90006-0.

[Tho04] Mikkel Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *J. ACM*, 51(6):993–1024, November 2004. doi:10.1145/1039488.1039493.

[Ung51] Peter Ungar. A theorem on planar graphs. *Journal of the London Mathematical Society*, s1-26(4):256–262, 1951. doi:10.1112/jlms/s1-26.4.256.

[Wag37] Klaus Wagner. Über eine eigenschaft der ebenen komplexe. *Mathematische Annalen*, 114(1):570–590, 1937. doi:10.1007/BF01594196.

[Wul11] Christian Wulff-Nilsen. Separator theorems for minor-free and shallow minor-free graphs with applications. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, pages 37–46. IEEE, 2011. doi:10.1109/focs.2011.15.

[Wul14] Christian Wulff-Nilsen. Faster separators for shallow minor-free graphs via dynamic approximate distance oracles. In *International Colloquium on Automata, Languages, and Programming*, ICALP '14, pages 1063–1074. Springer, 2014. doi:10.1007/978-3-662-43948-7_88.

A Proof of Lemma 1

We restate Lemma 1 here for convenience.

Lemma 1 (Vertex-Weighted KPR). *Given a graph $\langle G = (V, E), w\rangle$ with integer weights on vertices, m edges and integer parameters $\Delta > 0, h > 0$. Let $W = \sum_{v \in V} w(v)$. The procedure* KPR($\langle G, w\rangle, \Delta, h$) *runs in time $O(h \cdot (m + W))$ and returns either:*

- *a pair (C^*, S) where S is a subset of vertices of size $h \cdot W/\Delta$, and C^* is the connected component of maximum size of $G - S$, which is guaranteed to have (vertex-weighted) weak diameter at most $6 \cdot h^2\Delta$, or*
- *a K_h-minor model.*

In this section, we will work with $\langle G, w\rangle$ and we set $W := \sum_{v \in V(G)} w(v)$. We first slightly modify the procedure BFS($\langle G, w\rangle, v, r$) defined in the Section 1.2. For simplicity, we define *w-BFS* as a shortcut for BFS($\langle G, w\rangle, v, r$) with $r = W$ and v being an arbitrary vertex unless explicitly specified otherwise. We also refer to *levels* of the w-BFS, based on the distance to the root: vertex v is in $\lfloor w(v) \rfloor$ levels from the floor of its distance down to the successor integer to its distance minus $w(v)$. Observe that Lemma 8 still applies in this setting. Moreover, if a graph consists of multiple components, we can run Lemma 8 component-wise, which yields the very same bound on the whole graph. Similarly, it is easy to observe that we can run w-BFS from a set of vertices, still with the same running time guarantees.

We first define the core objects of the KPR approach. Given integers $\Delta, d, \ell > 0$, we define a *(Δ, d, ℓ)-KPR-separator* $\mathcal{S}$ in a vertex-weighted graph $\langle G, w\rangle$ as a collection $\mathcal{S} = \{S_1, \ldots, S_\ell\}$ of at most ℓ vertex separators such that:

- for all $v \in V(G) \setminus S_1$, $w(v) \leqslant \Delta$,
- for all $i \in [\ell]$, $|S_i| \leqslant \frac{W}{\Delta}$,
- the weak diameter of the largest connected component C^* of $G - \bigcup \mathcal{S}$ is at most d.

It is readily seen that $(\Delta, h^2\Delta, h)$-KPR-separator matches the first outcome of Lemma 1 with $S = \bigcup \mathcal{S}$. We now define a complementary object that will serve as a certificate for the second outcome. Given integers $\Delta, d, \ell > 0$, we define a *(Δ, d, ℓ)-KPR-decomposition* in a vertex-weighted graph $\langle G, w\rangle$ as a collection $\mathcal{S} = \{S_1, \ldots, S_\ell\}$ of ℓ separators and ℓ rooted trees $\mathcal{T} := \{T_1, \ldots, T_\ell\}$ such that

(P1) the largest connected component C^* of $G - \bigcup \mathcal{S}$ has weak diameter greater than d,

(P2) for all $v \in V(G) \setminus S_1$, $w(v) \leqslant \Delta$,

(P3) for all $i \in [\ell]$, T_i is a spanning tree of a connected component in graph $G - \cup_{j=1}^{i-1} S_j$ containing C^*,

(P4) all connected components of $G - \cup_{j=1}^{i} S_j$ are within Δ consecutive layers of T_i, and

(P5) for all $i \in [\ell]$, the distance from any vertex of C^* to the root of T_i is at least $(h+1)\Delta + 1$.

We remind the reader that the distances, diameter, etc., are considered in vertex-weighted graphs, where two vertices u, v are at distance d if the minimum of the sum of weights on any u–v path is equal to d. The following lemma states that given a KPR-decomposition with sufficiently large d, we will be able to extract a $K_{h,h}$ minor, hence a K_h minor, which corresponds to the second outcome of Lemma 1.

Lemma 13. *Let $h \geqslant 3$ and $\Delta > 0$ be integers. If a graph admits a $(\Delta, 6h^2\Delta, h)$-KPR-decomposition $(\mathcal{S}, \mathcal{T})$ then it contains a $K_{h,h}$ minor. Moreover, the minor model can be constructed in time $O(h(m + W))$.*

Proof: Let C^* be the largest connected component, and by assumption (P1), it has weak diameter at least $6h^2\Delta + 1$. We show that there are at least h vertices in $V(C^*)$ at weak distance at least $2(h+2)\Delta + 2$ from each other. We find two vertices $a_1, a_h \in V(C^*)$ by running w-BFS from an arbitrary vertex (denote it as a_1) of C^*, and we pick a_h as the furthest vertex within C^* in the w-BFS tree. As the weak diameter is at least $6h^2\Delta + 1$, then the weak distance of the a_1–a_h path is at least $3h^2\Delta$. As the weak diameter of C^* is at most its diameter and it is connected, there is an a_1–a_h path P in C^* of length at least $3h^2\Delta$. In particular, any a_1–a_h path within C^* has this property, and hence P can be found by a w-BFS on the component C^* rooted at a_1.

Inductively, for all $1 < i < h$, we set a_i to be a vertex on P such that its weak distance to any of $a_1, \ldots, a_{i-1}$ is at least $2(h+2)\Delta + 2$ and its weak distance to one of $a_1, \ldots, a_{i-1}$ is at most $2(h+2)\Delta + 2 + \Delta$. Such a vertex exists as by (P2), the weight of each vertex in $G - S_1$ is at most Δ. Moreover, such a vertex can be found by a w-BFS rooted at $a_1, \ldots, a_{i-1}$. To avoid a contradiction with a_1, a_h having weak distance at least $3h^2\Delta$ we know that a_i has a weak distance to a_h at least

$$3h^2\Delta - (i-1)(2h\Delta + 3\Delta). \tag{14}$$

Computing Equation (14) for $i = h-1$ checks that the weak distance between a_{h-1} and a_h is at least:

$$3h^2\Delta - (i-1)(2h\Delta + 3\Delta) \geqslant 2(h+2)\Delta + 2.$$

Hence, we found distinct vertices far away from each other: $a_1, \ldots, a_h$. For $i, j \in [h]$, we define branch sets A_i^j such that $A_i^h = \{a_i\}$ and $A_i^j \subseteq A_i^{j-1}$. We denote $A_i \coloneqq A_i^0$ and those represent the branch sets of one partite of $K_{h,h}$ minor. For $j \in [h]$, we create branch sets B_j representing the vertices in the other partite.

Now, we proceed with an inductive argument for $1 \leqslant j \leqslant h$, starting with the step $j = h$. Suppose that $B_{j+1}, \ldots B_h$ are already created as well as $A_i^{j+1}, \ldots A_i^{j+1}$ for all $i \in [h]$. We take T_j and set its root r_j as the center of B_j. For each $i \in [h]$, we consider path from a_i to r_j in T_j. We denote by b_i^j the vertex whose distance in tree T_j from a_i is at least $(h+1)\Delta + 1$ but all closer vertices to a_i have distance strictly smaller. Observe that b_i^j has to exist by (P5). We denote a_i, b_i^j path in T_j as P_i^j. Observe that b_i^j does not belong to A_i^j which could contain path of length at most $\Delta + (h-j)\Delta \leqslant (h+1)\Delta$ starting at a_i by Claim 1 (for $j = h$, A_i^h contains one vertex of weight at most Δ by (P4).). Observe that for all i, k, P_i^j and P_k^j do not collide, as this would mean that a_i and a_k have weak diameter at most $2[(h+2)\Delta + 1] - 1$, but they have weak diameter at least $2(h+2)\Delta + 2$. Hence, we will define B_j as a branch set consisting of a subtree of B_j rooted at r_j with leaves being b_i^j for every $i \in [h]$. We now extend A_i^{h-1} to contain vertices of $P_i^h \cap V(G - \cup_{k=1}^{j-1} S_k)$. By (P2) and (P4), we have:

Claim 1. *For all $i \in [h]$ and for all $h \geqslant j > 1$, the maximum length of a path from a_i in A_i^{j-1} is at most that value in A_i^j plus Δ.*

Indeed, by (P2) $P_i^h \cap V(G - \cup_{k=1}^{j-1} S_k)$ never spans more than Δ consecutive layers when the vertices of S_j are returned back for $j > 1$. Observe that each iteration j describing a branch sets construction in one step only consists of a single pass on a single tree T_j.

In this way, we obtain $K_{h,h}$-minor by contracting the branch sets above as well as maintaining $P_i^j - A_i$ as their mutually disjoint connectors. Hence, it remains to argue that for all $i, j \in [h]$, A_i^j and B_j are mutually disjoint and $P_i^j - A_i$ are non-empty. It is easy to observe that B_j are mutually disjoint for all $j \in [h]$ by a combination of (P4) and (P5). Recall that the maximum length of a path from a_i in A_i is $(h+1)\Delta$ by h applications of Claim 1. Therefore, it cannot reach B_j, which is at distance at least $(h+1)\Delta + 1$ from a_i. The same argument is true for A_i and A_k being disjoint, as the weak distance between their centers a_i and a_k is at least $2(h+1)\Delta$. It remains to show that for all $i, j \in [h]$, $P_i^j - A_i$ are disjoint and non-empty. The disjointness is guaranteed by the fact that the only overlap of T_j happens in the last Δ layers by (P4), and this part is gradually added to form A_i. Hence, it remains to show the non-emptiness of $P_i^j - A_i$. It follows from the length of P_i^j, which is at least $(h+1)\Delta + 1$, which is larger than $(h+1)\Delta$.

The running time is $O(h)$ runs of the w-BFS algorithm (Lemma 8). □

Proof of Lemma 1: If $h < 3$, the problem is trivial. By Lemma 8, we construct a w-BFS tree starting from an arbitrary node v of G and obtain a partition into layers $\mathcal{L}_1$. If there are at most $\frac{\Delta h^2}{2}$ layers in $\mathcal{L}_1$, we stop the procedure as the whole graph has diameter at most Δh^2. In that case, $S \coloneqq \emptyset$ and C^* is the single connected component found during the run of w-BFS tree. Otherwise, we show how to construct in linear time a $(\Delta, 6h^2\Delta, h-1)$-KPR-separator $\mathcal{S}$

or a $(\Delta, 6h^2\Delta, h-1)$-KPR-decomposition $(\mathcal{S}, \mathcal{T})$. In case the outcome is a $(\Delta, 6h^2\Delta, h)$-KPR-decomposition by Lemma 13, we obtain $K_{h,h}$-minor which could be contracted to K_{h+1} by contracting a matching of size $h-1$. In case the outcome is a KPR-cut $\mathcal{S}$, we return $S := \bigcup \mathcal{S}$ such that C^* is of the weak diameter at most $6h^2\Delta$.

Recall that there is more than $\frac{\Delta h^2}{2}$ layers in $\mathcal{L}_1$. Let L_0 contain the root of $\mathcal{L}_1$. We define $S_1 \in \mathcal{S}$ as the cheapest out of the following possible separators $L_{1+i}, L_{1+i+\Delta}, L_{1+i+2\Delta} \dots$, where $i \in [\Delta]$. Observe that as the total vertex weight is W, which corresponds to the total number of occurrences of the vertices in layers, the cheapest out of the possible proposed Δ separators has the size at most $\lfloor \frac{W}{\Delta} \rfloor$. Moreover, if a vertex $v \in V(G)$ has $w(v) > \Delta$ then it is part of all the separators defined above and will never be part of $G - S_1$. As a result of this step, we are left with a set of connected components of $G - S_1$ denoted as $\mathcal{C}_1$. Now, we repeat the same procedure for each component in $\mathcal{C}_1$ separately and recursively. That means in step $j \leqslant h+1$ we consider all connected components of graph $G - \bigcup_{k=1}^{j-1} S_k$. We run w-BFS for each separately. If such a connected component has at most $\frac{6h^2\Delta}{2}$ levels, we do nothing as their weak diameter is even smaller. If all connected components belong to this category, then we return $S := S_1 \cup \ldots \cup S_{j-1}$ as $(6h^2\Delta, h)$-KPR-separator. If a connected component has more levels, we apply the cutting procedure described above, i.e., choosing the smallest cut out of possible options $i \in [\Delta]$. Then we set S_j as the union of the smallest size cuts over all connected components considered in that step. The analysis still holds as we take at most the average out of Δ disjoint options.

If we reach the step $j = h-1$, we have a set of connected components $\mathcal{C}_{h-1}$. We check which component has the largest weighted size, and we denote it as C^*. This can be easily checked by a component-wise w-BFS check in total time $m + W$. If C^* has weak diameter at most $6h^2\Delta$, we have the right KPR-separator and we return $\mathcal{S}$. This can be checked by one iteration of w-BFS on the whole graph. Hence, we are left with C^* of weak diameter greater than $6h^2\Delta$. We now verify that it is $(\Delta, 6h^2\Delta, h)$-KPR-decomposition. Properties (P1) and (P2) are straightforward. For the other properties, we take the w-BFS trees that were built during the procedure for the connected component containing C^* and we set those as $\mathcal{T}$. Then Properties (P3) and (P4) follow by the construction of $\mathcal{T}$. To check Property (P5), we recall that any $T \in \mathcal{T}$ has the depth at least $\frac{6h^2\Delta}{2}$ out of which at most Δ consecutive levels can be occupied by vertices in C^*. Those levels of T containing C^* could not be as close to the root as $(h+1)\Delta + 1$ as it would imply the whole component C^* having weak diameter at most $2(h+1)\Delta + 2 < 6h^2\Delta$. Hence, Property (P5) is verified.

The running time matches the statement as we invoked at most h times a w-BFS algorithm component-wise (Lemma 8), each running in total time $O(m + W)$. In addition to it, we run two more w-BFS on the whole graph when determining C^*. In case of returning the K_h-minor model, we additionally run Lemma 13 in time $O(h \cdot (m + W))$. ◻

B Proof of Lemma 2

We restate Lemma 2 and then prove it.

Lemma 2 (Corresponds to Lemma 4.3 in [KL24]). *Let G be a connected graph with m edges. For every $t \geqslant 1$, there is a graph H with $|V(H)| + |E(H)| \leqslant 3t^2$ such that an almost-embedding of H can be turned into a K_t-minor model of G in $O(t^2 m)$ time.*

Proof: Let H be the graph obtained by subdividing every edge of K_t twice, i.e., replacing the edges of K_t by paths of 3 edges. We have that $|V(H)| + |E(H)| = t + 5 \cdot \binom{t}{2} \leqslant 3t^2$. We claim that an almost-embedding of H can be turned into a K_t-minor model $\{C_1, C_2, \ldots, C_t\}$ in $O(t^2 m)$ time.

Denote the almost-embedding by ϕ, and the vertices of H corresponding to the vertices of K_t by $v_1, \ldots, v_t$. First, we set $C'_i = \bigcup_{u \in N(v_i)} V(\phi(v_i u))$, i.e., it is the union of the paths of ϕ corresponding to the edges incident to v_i. The properties of ϕ guarantee that (1) each $G[C'_i]$ is connected, and (2) the sets C'_i are pairwise disjoint.

Then, consider a pair i, j with $1 \leqslant i < j \leqslant t$. Let $v_i, a_{i,j}, b_{i,j}, v_j$ be the path of three edges in H between v_i and v_j. We have that $\phi(a_{i,j} b_{i,j})$ is a path that intersects both C'_i and C'_j. We let $C'_{i,j}$ be the internal vertices of a minimal subpath of $\phi(a_{i,j} b_{i,j})$ that intersects both C'_i and C'_j. In particular, $C'_{i,j}$ can be empty if C'_i and C'_j are adjacent. It holds that (1) each $G[C'_{i,j}]$ is connected or empty, (2) each $C'_{i,j}$ either is adjacent to both C'_i and C'_j, or is empty, in which case C'_i and C'_j are adjacent, and (3) the sets $C'_{i,j}$ are pairwise disjoint and disjoint from the sets C'_k for all k.

Now, taking $C_i = C'_i \cup \bigcup_{j>i} C'_{i,j}$ results in $\{C_1, C_2, \ldots, C_t\}$ being a K_t-minor model. This construction can clearly be implemented in $O(t^2 m)$ time. □

C Balanced Separators of Size $O(\text{poly}(h)\sqrt{n})$ in Time $O(n)$ in Bounded-Degree K_t-Minor-Free Graphs

Lemma 14. *There is an algorithm that, given any connected graph G of maximum degree Δ and positive integer $p \leqslant |V(G)|$, builds a partition $\mathcal{P} = \{P_1, \ldots, P_s\}$ of $V(G)$ such that for every $i \in [s]$*

- *$G[P_i]$ is connected, and*
- *$p \leqslant |P_i| \leqslant p(\Delta + 1)$,*

and runs in time $O(n\Delta)$.

Proof: In $O(n\Delta)$ time run a depth-first search (DFS) on G [Tar72], from an arbitrary vertex u. Let T be the resulting rooted spanning tree. In $O(n)$ time, mark each node v of T whenever the subtree of T rooted at v has fewer than p nodes. This can be done in a bottom-up way by computing the size of each subtree (and capping at p). Let Q be the leftmost root-to-leaf branch of T.

We build the first part of $\mathcal{P}$ as follows. Initialize $x := u$ and $X := \{u\} \cup Y$, where Y is the union of the vertex sets of the subtrees of T rooted at a marked child of x. While $|X| < p$, set the new x as the first node appearing after (the old) x in the preorder traversal of T that is not yet in X, add it to X as well as every subtree of T rooted at a marked child of x. When exiting the while loop, the size of X is between p and $p\Delta$. We set $P_1 := X$. It is indeed connected, and $p \leqslant |P_1| \leqslant p(\Delta + 1)$.

We then proceed inductively in each connected component of $T - P_1$ (observe that several such components may well be a single component in $G - P_1$) and same marking. Note that each such component has at least p vertices, so the while loop of each call terminates. Computing a new part P of $\mathcal{P}$ takes $O(|P|)$ time. The partition $\mathcal{P}$ is thus computed in $O(n\Delta)$ total time. □

Theorem 3. *Let G be a given graph with n vertices and maximum degree $\Delta = O(1)$, and $h > 0$ be a parameter. There is an algorithm that runs in deterministic $O(n)$ time and outputs either:*

- *a balanced separator of size $O(h^{26}\sqrt{n})$ or*
- *$\perp$ if G contains a K_h as a minor.*

Furthermore, in the latter case, we can output a K_h-minor model of G with probability at least $1/2$ in an additional randomized $O(n)$ time.

Proof: Apply Lemma 14 to $G, p := h^{13}, \Delta$, and get, in $O(n)$ time, a partition $\mathcal{P}$ satisfying the lemma. In further $O(n)$ time, compute the graph G' obtained by contracting each part of $\mathcal{P}$ into a single vertex. As each part of $\mathcal{P}$ is connected, G' is a minor of G. Note that G' has at most n/h^{13} vertices.

Now run Theorem 2 on G'. This takes time $O(h^{13} n/h^{13}) = O(n)$. If this yields a K_t-minor model, the partition $\mathcal{P}$ lifts it to a K_t-minor model in G. If, instead, we get a balanced separator of size $O(h^{13}\sqrt{n})$ in G', this gives a $(1 - \frac{1}{2\Delta+3})$-balanced separator of size $O(h^{26}\sqrt{n})$ in G. With a constant number of iterations (recall $\Delta = O(1)$), we get a K_h-minor model in G or a balanced separator of size $O(h^{26}\sqrt{n})$. □