

Planar and Minor-Free Metrics Embed into Metrics of Polylogarithmic Treewidth with Expected Multiplicative Distortion Arbitrarily Close to 1 *

Vincent Cohen-Addad[†] Hung Le[‡] Marcin Pilipczuk[§] Michał Pilipczuk[¶]

Abstract

We prove that there is a randomized polynomial-time algorithm that given an edge-weighted graph G excluding a fixed-minor Q on n vertices and an accuracy parameter $\varepsilon > 0$, constructs an edge-weighted graph H and an embedding $\eta: V(G) \rightarrow V(H)$ with the following properties:

- For any constant size Q , the treewidth of H is polynomial in ε^{-1} , $\log n$, and the logarithm of the stretch of the distance metric in G .
- The expected multiplicative distortion is $(1 + \varepsilon)$: for every pair of vertices u, v of G , we have $\text{dist}_H(\eta(u), \eta(v)) \geq \text{dist}_G(u, v)$ always and $\mathbb{E}[\text{dist}_H(\eta(u), \eta(v))] \leq (1 + \varepsilon)\text{dist}_G(u, v)$.

Our embedding is the first to achieve polylogarithmic treewidth of the host graph and comes close to the lower bound by Carroll and Goel, who showed that any embedding of a planar graph with $\mathcal{O}(1)$ expected distortion requires the host graph to have treewidth $\Omega(\log n)$. It also provides a unified framework for obtaining randomized quasi-polynomial-time approximation schemes for a variety of problems including network design, clustering or routing problems, in minor-free metrics where the optimization goal is the sum of selected distances. Applications include the capacitated vehicle routing problem, and capacitated clustering problems.

1 Introduction

Tree metrics are among the easiest metrics that an algorithm designer could have to deal with: a vast number of NP-hard problems in tree metrics can be solved in polynomial time, some even in linear time. Unfortunately, most real-world problems take place in much more complicated metrics. Hence, the question of how well one can embed a complicated metric into a simpler one, such as for example a tree metric, has naturally emerged in the 90s. A sequence of works [7, 8, 26] culminated in showing that any n -point metric stochastically embeds into a tree metric such that for any pair of points, their distance is in expectation preserved up to a multiplicative factor, the *distortion*, of $\mathcal{O}(\log n)$; and that such an embedding can be computed in polynomial time. This result has a number of algorithmic applications and became a part of the standard toolbox for algorithm design.

Bartal [8] showed that the $\mathcal{O}(\log n)$ bound on the distortion is the best one can hope for in the worst-case, for arbitrary metric spaces. To improve upon this bound, one would need to focus on input metrics that exhibit special structure, such as for example shortest-paths metrics in planar and minor-free graphs. Planar and minor-free metrics are well-studied and have appeared in many practical applications, in

*This work is a part of projects CUTACOMBS (Ma. P.) and BOBR (Mi. P.) that have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreements No. 714704 and 948057, respectively). Hung Le is supported by the NSF CAREER Award No. CCF-2237288 and an NSF Grant No. CCF-2121952. Ma. P. is also part of BARC, supported by the VILLUM Foundation grant 16582.

[†]Google Research, France (cohenaddad@google.com)

[‡]University of Massachusetts Amherst, USA (hungle@cs.umass.edu)

[§]Institute of Informatics, University of Warsaw, Poland and IT University of Copenhagen, Denmark (marcin.pilipczuk@mimuw.edu.pl)

[¶]Institute of Informatics, University of Warsaw, Poland (michal.pilipczuk@mimuw.edu.pl)

particular network design, routing or clustering applications. In the early 2000s, researchers have thus asked: Could we improve the distortion when the input metric is planar (and the host metrics are trees)? Carroll and Goel [10] showed that unfortunately, the $\Omega(\log n)$ lower bound also holds when the input metric is planar. (A lower bound of $\Omega(\log n)$ was shown earlier for embedding a planar grid into random *spanning trees* [4].)

While tree metrics are topologically simple, the ultimate goal of algorithmic designers is to solve algorithmic problems efficiently. In this respect, metrics induced by graphs of bounded treewidth are almost equally powerful while permitting much more complicated structures. Specifically, one can solve an extremely broad class of problems on graphs of bounded treewidth using dynamic programming on tree decompositions; see [19, Chapter 7]. The next question in line was thus whether one can improve over the $\mathcal{O}(\log n)$ distortion bound for embedding planar metrics into trees if we instead aim to embed planar metrics into bounded treewidth graphs, hence bypassing the lower bound of [10].

Question 1. *Could we embed planar graphs into low-treewidth graphs with small distortion, ideally constant or $(1 + \varepsilon)$ for any fixed $\varepsilon \in (0, 1)$?*

By low treewidth, we mean a constant or even polylogarithmic treewidth – which would be enough to obtain quasi-polynomial approximation schemes for a large number of NP-hard problems. As we will point out later, constant treewidth is impossible if one insists on $o(\log n)$ distortion.

However, research on Question 1 has been riddled with negative results. In particular, any deterministic embedding of planar graphs with a worst-case constant distortion must have treewidth $\Omega(\sqrt{n})$ of the host graph, which is a nearly trivial bound, since every n -vertex planar graph has treewidth $\mathcal{O}(\sqrt{n})$ [10]. Any randomized embedding of planar graphs into *constant treewidth graphs* must have expected distortion $\Omega(\log n)$ [10, 11]. And any randomized embedding that has an expected *constant distortion* c for any given constant $c \geq 1$ must have treewidth $\Omega(\log n/c)$ [10]. Given these lower bounds, it seems hopeless to achieve any significant progress on Question 1. On a positive note, some progress has been made recently on embeddings with *additive distortion*, where the input metrics were further restricted to have stretch 2^ℓ for some $\ell > 0^1$, and the distortion is additive $+\varepsilon 2^\ell$ [24, 31, 15, 29]. Additive distortion is a much weaker guarantee than multiplicative distortion and is not truly comparable to the $\mathcal{O}(\log n)$ multiplicative distortion obtained for general metrics². These weaknesses severely limit the kind of algorithmic applications of embeddings with additive distortion.

On the other hand, the aforementioned lower bounds still leave room for a slim hope: they do not rule out an embedding with expected distortion $(1 + \varepsilon)$ and treewidth $\mathcal{O}_\varepsilon(\log n)$. The lower bound of treewidth $\Omega(\log n/c)$ mentioned earlier, while holding for any constant distortion c , does not exclude the possibility of an embedding with the treewidth of host graph of $\mathcal{O}(\log n/\varepsilon)$ for the expected distortion $(1 + \varepsilon)$. Nevertheless, it has remained unclear how to obtain an embedding with any constant distortion, say, of an n -vertex planar graphs into graphs with treewidth significantly better than the trivial bound $\mathcal{O}(\sqrt{n})$. We note that the discussion so far applies equally to weighted and unweighted planar graphs; in other words, having the same weight on every edge does not seem to make the problem easier. The lower bound of Carroll and Goel [10] holds for the unweighted planar grid.

In this work, we show that an embedding with polylogarithmic treewidth is possible for an expected distortion $(1 + \varepsilon)$. Our result is the first positive progress on Question 1.

Theorem 1.1. *For every fixed graph K there exists a randomized polynomial-time algorithm that, given an edge-weighted K -minor-free graph G and an accuracy parameter $\varepsilon > 0$, constructs a probabilistic metric embedding of G with expected distortion $(1 + \varepsilon)$ into a graph of treedepth³*

$$\mathcal{O}_K \left((\ell + \ln n)^6 / \varepsilon \cdot \ln^2 n \cdot (\ln n + \ln \ell + \ln(1/\varepsilon))^5 \right),$$

¹The *stretch* is the ratio between the maximum pairwise distance and the minimum pairwise distance in a metric.

²If one sets $\varepsilon \approx 2^{-\ell} \log n$, one gets multiplicative distortion $\mathcal{O}(\log n)$, but then the treewidth is proportional to $2^\ell / \log n$, which is way too large in most cases. We are rather concerned with graphs of treewidth polynomial in ℓ .

³The *treedepth* of a graph G is the minimum possible depth (number of vertices on a root-to-leaf path) of an elimination forest of G . The treedepth of a graph is an upper bound on its treewidth.

where n is the vertex count of G and ℓ is the logarithm of the stretch of the metric induced by G .

In particular, for planar graphs — which exclude K_5 as a minor — with edge weights in $[1, n^c]$ for any constant c , our treewidth bound in Theorem 1.1 is $\text{polylog}(n)$, as $\ell = \log(n^c) = \mathcal{O}(\log n)$. In this work, we do not attempt to minimize the degree of the polylogarithmic factor bounding the treewidth.

Applications. Theorem 1.1 is a part of a broader research agenda, aiming to design approximation schemes for optimization problems in structured metrics. Notable examples of structured metrics include Euclidean and doubling metrics. For Euclidean metrics, tremendous progress was immediately made and quasi-polynomial-time approximation schemes (QPTASs) and PTASs for other geometric problems (including k -median) were presented in a series of subsequent works [6, 5]. The work of Arora and Mitchell, which uses the notion of quad-tree decompositions to break the Euclidean metric instances into subinstances of the problem with limited interactions, quickly became a framework that solved a variety of problems. Talwar [51] took one step further by showing that any doubling metric (a generalization of Euclidean metric that does not necessitate an embedding of the points) embeds with $(1 + \varepsilon)$ expected distortion into a graph of treewidth at most $\mathcal{O}_{d,\varepsilon}(\ell)$, where ℓ is the logarithm of the stretch. This immediately led to (1) QPTASs for a variety of problems for which the objective is the sum of the lengths of a multiset of edges, and for which a QPTAS is known for bounded treewidth graphs; and (2) a QPTAS for TSP beyond Euclidean spaces.

Unfortunately, the picture for shortest-path metrics of graphs such as metrics induced by the shortest paths of planar graphs or graphs excluding a fixed minor proved more challenging to address. Indeed, obtaining PTASes, or even QPTASes for TSP, or other network design problems such as Steiner tree or k -MST, or clustering problems has been an important challenge that has been partially resolved during the last 25 years. An illustrating example is that no PTAS for the k -MST problem was known until last year [13] and no QPTAS is known for the vehicle routing problem (unless the capacities are assumed to be absolute constant [15, 31]), nor for non-uniform facility location in minor-free metrics. The non-uniform facility location problem is indeed an illustrative example: The question of generalizing results for planar graphs to more general families of graphs, such as graph excluding a fixed minor, has been an important research agenda. While Grigni [33] gave a QPTAS for TSP for minor-free graphs, it was only recently improved to EPTAS by Borradaile et al. [9]. These were recently improved (and extended to the subset version⁴) by Le [44] and Cohen-Addad et al. [15]. The recent result of Cohen-Addad [13] is a step toward a framework that generalizes planar results on several network design problems (e.g. Steiner forest) to minor-free metrics. Unfortunately, the approach fails for more sophisticated objectives such as vehicle routing or clustering (facility location or k -median).

The gap between planar (and more generally minor-free) metrics on the one hand and Euclidean and doubling metrics on the other hand is mainly due to the embedding to treewidth- $\mathcal{O}_{d,\varepsilon}(\ell)$ graphs of Talwar [51] which provided QPTASs for most metric optimization problems (except for k -means or k -center where the objective is to minimize a sum of squared distances or a maximum distance and for which different techniques were used [16, 14, 32, 31]). Our Theorem 1.1 overcomes the fundamental embedding bottleneck. As corollaries, we obtain QPTASes for several problems where prior techniques were failing.

The first problem is the Capacitated Vehicle Routing Problem (CVRP) in planar metrics. In this problem, we are given: (i) a set of clients represented by a subset of points in the metric, and each client has a demand, (ii) a vehicle with capacity Q , and (iii) a special point r called a depot. A feasible solution of the CVRP is a collection of tours starting from and ending at the depot r where on each tour the vehicle can collect demand from each client such that the total demand collected is at most the capacity Q and over all tours, the demand of every client is collected. The goal is to find a feasible collection of tours with minimum cost. There are three different variants of CVRP: *unit demands* (where the demand of every client is 1), *splittable demands* (where the demand of each client can be collected in more than one tour), and *unsplittable demands* (where the demand of each client can only be collected in one tour).

⁴In the subset version, the problem is to visit given subset of the input vertices.

There has been a tremendous amount of attention recently on this problem (e.g. [15, 31]); Yet all the techniques developed only address the special case where the capacity is $o(\log \log n)$. This stands in stark contrast with the work of Das and Mathieu [20] which gave a QPTAS for the unrestricted problem in Euclidean spaces of fixed dimension in 2010.

Corollary 1.2. *For any $1/2 > \varepsilon > 0$, there exists a randomized $(1 + \varepsilon)$ -approximation scheme for the unit demand Capacitated Vehicle Routing in minor-free metrics with running time $n^{\varepsilon^{-\mathcal{O}(1)} \log^{\mathcal{O}(1)} n}$.*

The proof follows from Theorem 1.1 and the QPTAS of Jayaprakash and Salavatipour [36] for bounded treewidth graphs. We refer the readers to Section 7 for more details.

We also obtain the first QPTAS for the classic Facility Location problem in minor-free metrics. Here, the input consists of a metric space, two sets of points C, F , and a cost function $f: F \mapsto \mathbb{R}$. The goal is to identify a set $S \subseteq F$ so as to minimize $\sum_{c \in C} \min_{q \in S} \text{dist}(c, q) + \sum_{q \in S} f(q)$. While a QPTAS for the problem on Euclidean metrics of fixed dimension has been known since the late 90s [6] and a PTAS later on [40, 14], the first PTAS for planar metrics only dates back to 2019 [18] (see also [16, 17] for the case with uniform costs). Yet, an approximation scheme in minor-free metrics has eluded the community until now.

Corollary 1.3. *For any $1/2 > \varepsilon > 0$, there exists a randomized $(1 + \varepsilon)$ -approximation scheme for the Facility Location problem in minor-free metrics with running time $n^{\varepsilon^{-\mathcal{O}(1)} \log^{\mathcal{O}(1)} n}$.*

In the Capacitated k -Median problem, we have two sets of points C, F and a capacity constraint $\mu: C \mapsto \mathbb{N}$. The goal is to find a set of k centers from F and assignments of the points of C to the centers such that each center $i \in F$ is assigned at most $\mu(i)$ points, and the total distance from each point to its assigned center is minimum. The Capacitated Facility Location problem is similar, but now each point i has an associated opening cost $f(i)$ (for opening a facility at that point) similar to the Facility Location problem above, and the goal is to open a set of facilities F and assignment of other points to each facility so that: (a) each facility in $i \in S$ is assigned at most $\mu(i)$ points, and (b) the total cost $\sum_{i \in S} f(i) + \sum_p d(p, S(p))$ is minimum, where $S(p)$ is the facility to which p is assigned, and $d(\cdot, \cdot)$ is the distance function of the input metric. Again, these problems have been known to admit a QPTAS for Euclidean metrics of fixed dimension [12], but nothing better than the general metric $\mathcal{O}(\log k)$ -approximation algorithm was known for planar (and so minor-free) metrics.

Corollary 1.4. *For any $1/2 > \varepsilon > 0$, there exists a randomized $(1 + \varepsilon)$ -approximation scheme for Capacitated k -Median and Capacitated Facility Location in minor-free metrics with running time $n^{\varepsilon^{-\mathcal{O}(1)} \log^{\mathcal{O}(1)} n}$.*

The algorithms in Corollary 1.4 follows the same line of the algorithm for the CVRP: embed the input metric to a low-treewidth graph, and solve the low-treewidth instances in quasi-polynomial time.

In Euclidean/doubling metrics, many problems were first known to admit QPTASes and then were subsequently improved to PTASes with additional ideas tailoring for each problem. Following the same pattern, we hope that additional ideas will be developed to improve our QPTASes to PTASes.

Our techniques. For simplicity of exposition, we assume that $\ell = \mathcal{O}(\log n)$. A basic idea in the embedding of Talwar [51] for metrics of doubling dimension is a randomized ball carving: given a graph C of diameter D in a doubling metric, one can stochastically partition C (using balls of random radii) into a set of clusters, say \mathcal{P} , of diameter at most $D/2$ such that the probability of cutting every two points x and y into two different clusters is bounded by $\mathcal{O}(\frac{\text{dist}(u,v)}{D})$. Then we construct an $(\varepsilon D / \log n)$ -net N of C and designated it as *portals*: in the output host graph, we connect every $x \in N$ to every other point of C by an edge of length equal to the distance of the endpoints in G , and recurse on each cluster in \mathcal{P} . Due to the bounded doubling dimension assumption, the size of N is bounded polynomially in $\log n / \varepsilon$, which finally results in polylogarithmic treewidth. For every edge uv in G , each of the $\mathcal{O}(\log n)$ levels of recursion incurs an expected distortion of $\mathcal{O}(\frac{\varepsilon}{\log n}) \text{dist}(u, v)$: the edge uv is cut with

probability $\mathcal{O}(\frac{\text{dist}(u,v)}{D})$ and, thanks to the properties of the chosen net N , the detour in the case of cutting uv is bounded by $\mathcal{O}(\frac{\varepsilon}{\log n} D)$.

One could try to apply Talwar's technique for planar metrics. For the stochastic decomposition into clusters, the decomposition of Klein, Plotkin, and Rao [39] provides the same probability of cutting any two points into two different clusters. However, the step of choosing an $(\varepsilon D / \log n)$ -net N is problematic: we now can no longer bound the size of N . In fact, even in as simple graphs as stars it may happen that $|N| = n$. Thus, obtaining any non-trivial treewidth bound following this approach seems very difficult. On the other hand, stars have already pretty good treewidth.

We thus aim at designing an approach that implicitly distinguishes grid-like planar graphs, to which Talwar's decomposition could be applied, from planar metrics that are already "tree-like shaped". As obtaining a small r -net for $r \ll D$ seems impossible in planar graphs, we aim at lowering the probability that uv is cut in a way that incurs a detour $\Omega(D)$. Consider the following example. Assume that we applied the aforementioned clustering scheme to a graph G and we obtained a partition \mathcal{P} into clusters that are arranged in a "grid-like" fashion: the quotient graph G/\mathcal{P} (defined as a graph with the vertex set \mathcal{P} and two elements $A, B \in \mathcal{P}$ adjacent if G contains an edge between A and B) is a grid. Then, instead of cutting out *all* clusters of \mathcal{P} , we would like to do the following: choose randomly one of the middle ξ columns of the grid G/\mathcal{P} (for a parameter ξ to be chosen later), carve out the clusters on that column as subgraphs for recursion, and also recurse on the remaining left and right parts of the grid. Instead of a net, we designate as portals only one vertex per cluster from the chosen column of the grid. In the recursion, the recursive calls for clusters on the chosen column have multiplicatively smaller diameter, while (with enough columns in the entire grid) the left and right parts have at most $0.75|V(G)|$ vertices; this gives $\mathcal{O}(\ell \log n) = \mathcal{O}(\log^2 n)$ bound on the recursion depth. For the detour bound, note that now an edge uv has probability $\mathcal{O}(\frac{\text{dist}(u,v)}{D})$ of being chosen on a boundary of the cluster *times* the probability $\frac{1}{\xi}$ that this cluster is actually in the chosen column. Since an edge that is actually cut experiences a detour of $\mathcal{O}(D)$, the expected additive distortion due to the detours imposed by the entire process is thus $\mathcal{O}(\log^2 n \cdot \frac{1}{\xi} \cdot \frac{\text{dist}(u,v)}{D} D) \sim \mathcal{O}(\frac{\log^2 n}{\xi} \text{dist}(u,v))$, and choosing ξ to be $\log^2 n / \varepsilon$ (which may lead to $\mathcal{O}(\xi \log^2 n) = \mathcal{O}(\log^4 n / \varepsilon)$ final treewidth bound on the output embedding) gives the desired bound.

The main argument of our approach is a proof that this set of " ξ columns to randomly choose from" can always be found, even if G/\mathcal{P} does not look like a grid. To this end, we need another property of the clustering algorithm: with high probability, the diameter of the quotient graph G/\mathcal{P} is bounded by a polylogarithmic function of n , which we henceforth denote by ρ . Then, the treewidth of G/\mathcal{P} is bounded by $\mathcal{O}(\rho)$, which gives us the first "column": a set \mathcal{A}_1 of $\mathcal{O}(\rho)$ of clusters, whose deletion leaves connected components of multiplicatively smaller size. We henceforth refer to such a set of clusters as a *balanced cut*.

We need more balanced cuts to choose from and we would like the balanced cuts to be mostly disjoint in the sense of not containing the same edge of G on their boundary many times. However, there is a number of star-like examples where this is not possible, i.e., there is a cluster contained in every balanced cut. We proceed as follows: after picking the first balanced cut \mathcal{A}_1 as in the previous paragraph, we refine the partition \mathcal{P} by, for every cluster $C_1 \in \mathcal{A}_1$, replacing C_1 with the result of applying again the clustering scheme to C_1 and diameter $D/2$. Then, we find a balanced cut \mathcal{A}_2 in the obtained partition \mathcal{P}_2 , and continue to \mathcal{P}_3 by again replacing the clusters used by \mathcal{A}_2 with a result of applying the clustering scheme to them. The replacement process stops at single-vertex clusters, where cutting would not cause any detour.

Now, every edge uv is contained in the boundary of clusters larger than single vertices in $\mathcal{O}(\ell) = \mathcal{O}(\log n)$ balanced cuts \mathcal{A}_i . Hence, if we produce $\Omega(\frac{\log^3 n}{\varepsilon})$ balanced cuts, the expected detour analysis will still give the desired result. But there is a problem: as we expand the partitions \mathcal{P}_i , the diameter and hence the bound on the treewidth of G/\mathcal{P}_i grows, so our guarantee on the size of \mathcal{A}_i deteriorates as i increases.

This is the moment where the topological assumption on the input graph finally comes into play. Naively, the difference between the diameters of G/\mathcal{P}_{i+1} and G/\mathcal{P}_i can be as large as $\Omega(|\mathcal{A}_i| \cdot \rho)$, as

every cluster of \mathcal{A}_i is replaced with a set of clusters whose quotient graph has diameter ρ . However, if we assume that the given graph excludes some fixed apex graph⁵ as a minor, then we can prove that after k steps, the diameter of C/\mathcal{P}_k is only $\mathcal{O}(\sqrt{\sum_{i=1}^{k-1} |\mathcal{A}_i|} \cdot \ell\rho)$. The key graph theoretic property of apex-minor-free graphs used here is the following: to cover an apex-minor-free (unweighted) graph of treewidth $\geq t$ with balls of radius d , one has to pick at least $\Omega((t/d)^2)$ such balls. This proves that if one continues the process of finding and unraveling cuts for ξ steps, where ξ is any large polylogarithm one wishes for, all created balanced cuts will have sizes bounded by another polylogarithmic bound τ depending only on ξ . This, in turn, gives the final polylogarithmic bound on the treewidth of the obtained embedding.

This concludes the overview of the proof of Theorem 1.1 under the assumption that the input graph excludes a fixed apex graph as a minor; this in particular includes planar graphs, which exclude K_5 . To extend this to the setting when any fixed graph K is excluded, we start by applying the Robertson-Seymour structure theorem [50] that essentially states that the input graph G admits a tree decomposition where the torso of every bag consists of a graph coming from a fixed apex-minor-free graph class, with a constant number of vertices added to it. We then carefully apply our technique to the apex-minor-free part of the centroid bag of that decomposition.

A bit more on related works. There is a long line of work on embedding planar and minor-free metrics into Euclidean and normed spaces. Rao [49] constructed an embedding of planar metrics into Euclidean spaces with distortion $\mathcal{O}(\sqrt{\log n})$. For any ℓ_p space with $p \geq 1$, Rao obtained an embedding with $\mathcal{O}((\log n)^{1/p})$ distortion. Rao's results were extended to minor-free metrics [3, 41]; the distortion is $\mathcal{O}(h^{1-1/p}(\log n)^{1/p})$ where h is the size of the excluded minor. Better embedding results were known for graphs of small pathwidth [2, 46]. For the Euclidean case, Newman and Rabinovich [48] provided a matching lower bound for Rao's embedding, showing that distortion $\mathcal{O}(\sqrt{\log n})$ achieved by Rao is the best possible.

Of special interest is the case of ℓ_1 . The famous ℓ_1 embedding conjecture [34] states that planar metrics are embeddable into ℓ_1 with $\mathcal{O}(1)$ distortion. The conjecture is believed to hold for general minor-free metrics [34, 45]. While the conjecture remains open even for planar metrics, progress has been made for various special cases: k -outerplanar graphs [34], graphs with a small face cover [42, 28], or when every demand pair has a face containing both endpoints [43]. We refer readers to [2] and references therein for more embedding results.

Organization. We present our clustering tool in Section 3. Sections 4 and 5 prove Theorem 1.1 for the case of graph classes excluding a fixed apex graph as a minor. Then, in Section 6, we discuss how to extend the proof to graphs excluding an arbitrary fixed minor. Section 7 provides details on the aforementioned applications of Theorem 1.1.

2 Preliminaries

For a real t , we sometimes write $\exp(t) := e^t$, where e is the base of the natural logarithm. We use the Iverson bracket notation: for a boolean assertion ψ , $[\psi]$ is 1 if ψ holds and 0 otherwise.

Graphs. We use standard graph notation. Most of graphs considered in this paper are edge-weighted, which means that with every edge e we associate a positive weight $\text{length}(e)$, interpreted as the length of e . Hence, for an edge-weighted graph G we can consider the *distance metric* induced in G : for vertices u, v , $\text{dist}_G(u, v)$ is the smallest possible length of a path connecting u and v , or $+\infty$ if such a path does not exist, where the *length* of a path P , denoted $\text{length}(P)$, is the sum of the weights of edges of P . The *stretch* of this metric is the ratio between the maximum finite distance and the minimum distance between two distinct vertices in G .

⁵An *apex graph* is a graph that can be made planar by removing one vertex.

The notions of diameter and radius in an edge-weighted graph G are defined with respect to the distance metric induced in G . When speaking about distances in an unweighted graph, or about its diameter or radius, we mean the standard graph-theoretic notions, where every edge is considered to have length 1.

For a graph G and a partition \mathcal{C} of the vertex set of G , we define the *quotient graph* G/\mathcal{C} as the unweighted graph on vertex set \mathcal{C} where parts $C, C' \in \mathcal{C}$ are adjacent if in G there is an edge with one endpoint in C and the other in C' . Note that while G can be edge-weighted, we always consider G/\mathcal{C} to be an unweighted graph.

Decompositions. A *tree decomposition* of a graph G is a tree T together with a function $\text{bag}(\cdot)$ that with every node x of T associates its bag $\text{bag}(x) \subseteq V(G)$ so that (i) for every vertex u of G , the nodes of T whose bags contain u form a connected subtree of T , and (ii) whenever u and v are adjacent in G , there is a node in T whose bag contains both u and v . The *width* of a tree decomposition is the maximum size of a bag minus 1, and the *treewidth* of a graph G is the minimum possible width of a tree decomposition of G .

An *elimination forest* of a graph G is a rooted forest F with the same vertex set as G satisfying the following property: whenever u and v are adjacent in G , either u is an ancestor of v in F or vice versa. The *treedepth* of a graph G is the minimum possible depth (number of vertices on a root-to-leaf path) of an elimination forest of G .

It is easy to see that the treewidth of a graph is never larger than the treedepth minus 1: to construct a tree decomposition (T, bag) of width $d - 1$ from an elimination forest F of depth d , take T to be F with all roots connected into a path, and for every vertex u set $\text{bag}(u)$ to be the set of ancestors of u in F . On the other hand, it is well-known that the treedepth of an n -vertex graph is never larger than $\log_2 n$ times its treewidth plus 1 [47, Corollary 6.1]. Thus, from the point of view of constructing metric embeddings into graphs of polylogarithmic treewidth, we can equivalently consider embeddings into graphs of polylogarithmic treedepth instead.

We use the standard minor order for graphs. A graph K is an *apex graph* if there is a vertex u such that $K - u$ is planar. The following lemma, which follows from the previous literature, is the sole argument in which we exploit the topological simplicity of the considered graphs.

Lemma 2.1. *For every apex graph K there is a positive integer α_K such that the following holds. Suppose G is a K -minor-free (unweighted) graph and Z is a subset of vertices of G such that every vertex of G is at distance at most d from a vertex of Z . Then the treewidth of G is upper bounded by $\alpha_K d \sqrt{|Z|}$.*

Lemma 2.1 was proved explicitly in the case of planar graphs by Demaine et al. [21, Theorem 3.3]. In their proof, they use only two properties of planar graphs: (a) if a planar graph G has treewidth t , then G contains the $s \times s$ grid as a minor for $s \in \Omega(t)$, and (b) if a planar graph G contains the $s \times s$ grid as a minor, then every *distance- d dominating set* in G — a set Z such that every vertex of G is at distance at most d from Z — has size $\Omega((s/d)^2)$ (see [21, Lemma 3.1]). To lift this proof to the case of K -minor-free graphs for a fixed apex graph K , one can use the result of Fomin et al. [30, Theorem 3]: If a K -minor-free graph G has treewidth t , then G contains the *triangulated $s \times s$ grid* Γ_s as a contraction, for $s \in \Omega_K(t)$ (see [30] for a definition of Γ_s). It can be easily seen that the minimum size of a distance- d dominating set in a graph does not increase under edge contractions, and is of the order $\Omega((s/d)^2)$ in Γ_s . Therefore, every distance- d dominating set in a K -minor-free graph of treewidth t must have size $\Omega_K((t/d)^2)$; this readily implies the statement of Lemma 2.1.

Metric embeddings. A *metric embedding* of an edge-weighted graph G consists of the *host graph* — an edge-weighted graph H — and the *embedding* $\eta: V(G) \rightarrow V(H)$ satisfying $\text{dist}_H(\eta(u), \eta(v)) \geq \text{dist}_G(u, v)$ for all $u, v \in V(G)$. The embedding η has (*multiplicative*) *distortion* c if $\text{dist}_H(\eta(u), \eta(v)) \leq c \cdot \text{dist}_G(u, v)$ for all $u, v \in V(G)$. It will be often the case that a metric embedding is constructed by a randomized procedure; then we can say that η has *expected (multiplicative) distortion* c if $\mathbb{E} \text{dist}_H(\eta(u), \eta(v)) \leq c \cdot \text{dist}_G(u, v)$ for all $u, v \in V(G)$.

3 Partitioning tools

In this section we introduce our probabilistic partitioning tools, which are heavily inspired by the work on *padded decompositions* [1, 26]. We start by presenting a procedure that partitions a given graph into clusters of significantly smaller diameter so that for any path Q , the expected number of edges of Q cut by the boundaries of the clusters is proportional to the length of Q . We then use our partitioning procedure in a hierarchical fashion to create a multi-level decomposition of a given graph.

3.1 Probabilistic preliminaries

We need a few standard tools from the probability theory. We will use the standard Chernoff bound, stated below, and its simple corollary, proven for completeness in Appendix A.

Theorem 3.1 (Multiplicative Chernoff bound). *Let X_1, \dots, X_n be independent random variables with values in $\{0, 1\}$. Let $X = \sum_{i=1}^n X_i$, let $\mu = \mathbb{E}[X]$, and let $0 \leq \delta$ be a real. Then,*

$$\mathbb{P}(X \geq \mu(1 + \delta)) \leq \exp(-\delta^2 \mu / (2 + \delta)).$$

Lemma 3.2. *Let $k \geq 1$ be an integer and let X_1, X_2, \dots be independent random variables with values $\{-1, 1\}$ such that for every $i \geq 1$, $\mathbb{P}(X_i = 1) \leq \frac{1}{8}$. Let Z be a random variable defined as the minimum i such that $k + \sum_{j=1}^i X_j < 0$. Then*

$$\mathbb{P}(Z > 2k) \leq \exp\left(-\frac{k}{12}\right).$$

We will also use random variables with exponential distributions. Recall that a real-valued random variable X has exponential distribution with parameter $\lambda > 0$, denoted $X \sim \text{Exp}(\lambda)$, if its distribution has density given $g(t) = \lambda \exp(-\lambda t) \cdot [t \geq 0]$. It is well-known that exponentially-distributed random variables are *memoryless* in the sense explained by the statement below.

Lemma 3.3. *Suppose X is a random variable with an exponential distribution. Then for every pair of nonnegative reals $s \leq t$, we have*

$$\mathbb{P}(X \leq s + t \mid X \geq s) = \mathbb{P}(X \leq t).$$

In one of the more involved proofs, we will use memorylessness in a quite convoluted fashion. In order not to clutter the arguments later, we abstract the probabilistic part as the lemma below. While this statement may be next to trivial for readers more experienced with the probability theory in general or exponential clocks in particular, for the sake of rigour we provide a proof in Appendix A.

Lemma 3.4. *Let n be a positive integer, X_1, X_2, \dots, X_{2n} be independent random variables, each with distribution $\text{Exp}(1)$, let f_1, f_2, \dots, f_{2n} be measurable functions where $f_i: \mathbb{R}^{i-1} \rightarrow \mathbb{R}_{\geq 0} \cup \{+\infty\}$ for $1 \leq i \leq 2n$ and $f_i \equiv 0$ for $n < i \leq 2n$, and let $p \geq 0$ be a real. We say that an index i is good if $X_i \geq f_i(X_1, X_2, \dots, X_{i-1})$ and excellent if additionally $X_i > f_i(X_1, X_2, \dots, X_{i-1}) + p$. Let $N \in \{n, n+1, \dots, 2n\}$ be the number of good indices (note that every $i > n$ is good almost surely) and let $(I_i)_{i=1}^N$ be random variables being consecutive good indices. For $1 \leq i \leq N$, let $Y_i = X_{I_i} - f_{I_i}(X_1, \dots, X_{I_i-1})$. Then, Y_1, Y_2, \dots, Y_N are independent random variables, each with distribution $\text{Exp}(1)$. Consequently, if for $1 \leq i \leq n$ we denote by A_i the event “ I_i is excellent”, then the events $(A_i)_{i=1}^n$ are independent and each happens with probability e^{-p} .*

3.2 A single level

We first define a simple partitioning procedure to create one level of a hierarchical decomposition. The properties of the procedure are asserted in the following lemma.

Lemma 3.5. *There is a randomized polynomial-time algorithm that given a connected n -vertex edge-weighted graph G with diameter D and a real parameter $r > 0$, outputs a partition \mathcal{C} of the vertex set of G into clusters satisfying the following properties:*

- (P1) *For each $C \in \mathcal{C}$, $G[C]$ is connected.*
- (P2) *For every $t \geq 0$, the probability that in \mathcal{C} there is a cluster inducing a subgraph of diameter larger than $2r(t + 1 + \ln n)$ is at most e^{-t} .*
- (P3) *For every $t \geq 0$, the probability that the diameter of G/\mathcal{C} is larger than $120 \cdot \frac{D}{r} \cdot (t + 1 + \ln n)$ is at most e^{-t} .*
- (P4) *For every path Q in G , the expected number of edges on Q whose endpoints belong to different clusters is at most $\frac{\text{length}(Q)}{D} \cdot 4\ell$, where ℓ is the least integer such that the stretch of the distance metric in G is strictly smaller than 2^ℓ .*

The remainder of this section is devoted to proving Lemma 3.5. For tie-breaking, we assume that there is a fixed in advance total order \preceq on the vertices of G . By scaling, we may assume that $D = 2^\ell$ and that the distance between every pair of distinct vertices of G is strictly larger than 1. Further, we may assume that $r < D$, for otherwise we may return a partition consisting of one part: the whole $V(G)$.

The procedure. At the beginning, the procedure initiates \mathcal{C} to be \emptyset , and it will iteratively add clusters (subsets of $V(G)$) to \mathcal{C} one-by-one until \mathcal{C} becomes a partition of $V(G)$. At every step, a vertex is called *free* if it does not belong to any element of \mathcal{C} .

The procedure iteratively, as long as there exists a free vertex, picks a \preceq -minimal free vertex v , samples a radius $r_v := r(1 + X_v)$, where $X_v \sim \text{Exp}(1)$ is an independent random variable with exponential distribution, and creates a new element C_v in \mathcal{C} consisting of all vertices within distance at most r_v from v in the graph $G - \bigcup \mathcal{C}$ (i.e., subgraph of G induced by the free vertices). Vertex v is called the *center* of C_v , set C_v is the *cluster* of v , and r_v is the *radius* of v .

Running time. The procedure clearly takes (randomized) polynomial time and produces a partition \mathcal{C} of $V(G)$ into clusters. Note that every cluster C_v is connected, contains v , and is of radius at most r_v .

Properties. We now verify properties (P1)–(P4). Let k be the number of steps taken by the procedure, and let v_1, \dots, v_k the centers of the consecutive clusters constructed by the procedure. For brevity, we denote $X_i := X_{v_i}$, $C_i := C_{v_i}$, and $r_i := r_{v_i}$, for $i \in \{1, \dots, k\}$. Furthermore, let $G_i = G - \bigcup_{j < i} C_j$; that is, G_i is the subgraph of G induced by the vertices free at the moment of creation of C_i .

Property (P1) follows directly from the construction. Since the diameter of $G[C_i]$ is at most $2r_i$, property (P2) follows directly from the following statement.

Claim 3.1. *For every $t \geq 0$, the probability that there exists $i \in \{1, \dots, k\}$ such that $r_i > r(t + 1 + \ln n)$ is at most e^{-t} .*

Proof. For every $i \in \{1, \dots, k\}$, we have

$$\mathbb{P}(r_i > r(t + 1 + \ln n)) = \mathbb{P}(X_i > t + \ln n) = e^{-(t + \ln n)} = e^{-t}/n.$$

The lemma follows by the union bound, as there are at most n clusters. \square

In the next claim we verify the key property (P3). For brevity, let $H := G/\mathcal{C}$ be the quotient graph. Recall that H is unweighted.

Claim 3.2. *For every $t \geq 0$, the probability that the diameter of H is larger than $120 \cdot \frac{D}{r} \cdot (t + 1 + \ln n)$ is at most e^{-t} .*

Proof. Consider any two vertices u_1 and u_2 . Let Q be any shortest path from u_1 to u_2 in G (with respect to the edge weights). By definition we have $|Q| \leq D$. In the argumentation below, we consider Q to be oriented from u_1 to u_2 .

Let $Z = \{v \in V(H) \mid C_v \cap V(Q) \neq \emptyset\}$ be the set of those centers whose clusters intersect Q . We color the vertices of Q using palette Z as follows. Initially, all vertices are uncolored. We consider the vertices $v \in Z$ in the same order as the clustering procedure. Upon considering vertex v , we find the first and the last vertex of $C_v \cap V(Q)$ that are uncolored so far, say q_v^1 and q_v^2 , and we color with color v *all* vertices lying between q_v^1 and q_v^2 on Q . Note that this may result in recoloring some vertices that were colored in previous iterations. If all vertices of $C_v \cap V(Q)$ were colored before, we do not color any vertices in this iteration, hence there will be no vertex of color v .

Note that at every moment, all vertices colored with any color $v \in Z$ form an interval on Q . Furthermore, if when considering a center v we recolor at least one vertex colored v' to v , then the interval between q_v^1 and q_v^2 on Q contains *all* vertices colored previously with v' , hence *all* vertices colored v' get recolored to v . Finally, note that at the end of the procedure, every vertex is assigned some color.

Let $Z'' \subseteq Z$ be the set consisting of all vertices $v \in Z$ such that when v was considered during the coloring procedure, at least one vertex of Q was colored with color v . Further, let $Z' \subseteq Z''$ be the set consisting of all those vertices $v \in Z$ for which at least one vertex of Q is colored with color v at the end of the coloring procedure. Let \hat{u}_1 and \hat{u}_2 be cluster centers such that $u_1 \in C_{\hat{u}_1}$ and $u_2 \in C_{\hat{u}_2}$. Observe that in H there is a path from \hat{u}_1 to \hat{u}_2 that uses only vertices of Z' , as for every $v \in Z'$, both q_v^1 and q_v^2 are contracted onto v in H . Hence,

$$\text{dist}_H(\hat{u}_1, \hat{u}_2) \leq |Z'| - 1.$$

So to give an upper bound on the probability that $\text{dist}_H(\hat{u}_1, \hat{u}_2)$ is large, it suffices to bound the probability that Z' is large.

Consider the moment at the beginning of the i -th step of the procedure, that is, when clusters C_j for $j < i$ were already constructed. A *segment* is a maximal subpath of Q whose all vertices are uncolored; let \mathcal{S}_i be the family of segments at the beginning of the i th step. Call the cluster C_i *tenoning* if C_i contains all vertices of at least one segment.

Let

$$\lambda := t + 1 + \ln n.$$

For any center v_i , let R_i be the distance in G_i from v_i to the nearest vertex that is uncolored in G_i . We have $v_i \in Z''$ if and only if $r_i \geq R_i$. Let $p := \frac{1}{12\lambda}$; note that thus, $p \leq \frac{1}{12} \leq \frac{1}{8}$. We say that the vertex v_i is *good* if $r_i > R_i + pr$. The crucial observation is the following: if v_i is good, then it is either tenoning, or there is a subpath Q' of Q of length at least pr such that all vertices of Q' were uncolored before the i th step, and all of them, except possibly for the endpoints, receive color v_i in the i th step. In the former case we have

$$|\mathcal{S}_{i+1}| < |\mathcal{S}_i|,$$

and in the latter case we have

$$\sum_{S \in \mathcal{S}_i} \text{length}(S) - \sum_{S \in \mathcal{S}_{i+1}} \text{length}(S) \geq pr.$$

This motivates the following measurement. The *potential* at the beginning of step i is defined as:

$$\pi_i := |\mathcal{S}_i| + 2 \cdot \frac{\sum_{S \in \mathcal{S}_i} \text{length}(S)}{pr}.$$

Note that

$$\pi_1 \leq 1 + \left\lceil \frac{2D}{pr} \right\rceil =: k.$$

Clearly, $|\mathcal{S}_{i+1}| \leq |\mathcal{S}_i| + 1$ and $\bigcup \mathcal{S}_{i+1} \subseteq \bigcup \mathcal{S}_i$, for all relevant i . Hence, $\pi_{i+1} \leq \pi_i + 1$. Furthermore, if v_i is good, then $\pi_{i+1} \leq \pi_i - 1$. Note that $\pi_i > 0$ for $1 \leq i \leq |Z''|$.

Let $I_1, I_2, \dots, I_{|Z''|}$ be the indices of the consecutive elements of Z'' (note that these are random variables, and even the length of this sequence is a random variable). For $j \geq 1$, let $Y_j := (r_{I_j} - R_{I_j})/r$. For $j \geq 1$, let A_j be the event “ v_{I_j} is good”, which is the same as “ $Y_j > p$ ”.

Now, an important observation is that Lemma 3.4 applies: variables Y_j are independent, each has distribution $\text{Exp}(1)$, and events $(A_j)_{j \geq 1}$ are independent and each individually happens with probability e^{-p} . Indeed, to formally apply Lemma 3.4 we artificially extend the sequence $X_1, X_2, X_3, \dots = X_{v_1}, X_{v_2}, X_{v_3}, \dots$ of independent random variables with distribution $\text{Exp}(1)$ to length $2n$ and set $f_i := R_i/r$ whenever i is still an index of a center and $f_i \equiv 0$ otherwise.

Then, as $1 - e^{-p} \leq p \leq \frac{1}{8}$ and $r < D$, by Lemma 3.2 we infer that

$$\mathbb{P}(|Z''| > 2k) \leq \exp\left(-\frac{k}{12}\right) \leq \exp\left(-\frac{D}{6pr}\right) \leq \exp(-2\lambda) = \exp(-2(t+1+\ln n)).$$

On the other hand, using $r < D$,

$$2k = 2 + 2 \left\lceil \frac{2D}{pr} \right\rceil \leq \frac{D}{r} \cdot 10 \cdot 12\lambda = 120 \cdot \frac{D}{r} \cdot (t+1+\ln n).$$

Denote $w := 120 \cdot \frac{D}{r} \cdot (t+1+\ln n)$. This means that

$$\begin{aligned} \mathbb{P}(\text{dist}_H(\hat{u}_1, \hat{u}_2) > w) &\leq \mathbb{P}(|Z'| > w) \leq \mathbb{P}(|Z''| > w) \\ &\leq \mathbb{P}(|Z''| > 2k) \leq \exp(-2(t+1+\ln n)). \end{aligned}$$

By the union bound over all choices of u_1 and u_2 , we obtain

$$\mathbb{P}(\text{diam}(H) > w) \leq \exp(-2(t+1+\ln n) + 2\ln n) \leq e^{-t}.$$

This finishes the proof. \square

Finally, we show property (P4).

Claim 3.3. *Let Q be a path in G . Then the expected number of edges of Q whose endpoints belong to different clusters in \mathcal{C} is bounded by $\frac{\text{length}(Q)}{r} \cdot 4\ell$.*

Proof. We color the edges of Q with colors being cluster centers as follows: for an edge xy on Q , if $x \in C_u$ and $y \in C_v$, then the edge xy is colored with the center among $\{u, v\}$ whose cluster was conceived earlier (so if $u = v$, the edge xy is colored v). A *segment* is a maximal subpath of edges of Q of the same color. An edge of Q with endpoints belonging to different clusters is *charged* to the segment it is contained in. Note that a segment can be charged by 0, 1, or 2 edges; a segment is *positive* if it is charged by at least one edge.

We now make a crucial observation: For every edge e on Q and every real $d > 0$,

$$\mathbb{P}(e \text{ lies in a positive segment of length } \leq d) \leq \frac{d}{r}. \quad (1)$$

Indeed, for an index i , let x_i be the endpoint of e that is closer to v_i in G_i (or $x_i = \perp$ if both endpoints of e are already in clusters) and let R_i be the distance between v_i and x_i in G_i ($R_i = +\infty$ if $x_i = \perp$). Note that R_i is a measurable function of values $(X_j)_{j < i}$. Let I be the index such that the color of e is v_I ; equivalently, I is the earliest index such that $r_I \geq R_I$. (Note that I is a random variable.) Next, let S be the segment of e , and suppose for a moment that S is positive. Let yz be an edge of S that is charged to S , with $y \in C_I$ and $z \notin C_I$. Note that both y and z are free at the moment when C_I is constructed. Since $z \notin C_I$, we have $\text{dist}_{G_I}(v_I, z) > r_I$, which together with $\text{dist}_{G_I}(v_I, x_I) = R_I$ implies that

$$\text{length}(S) \geq \text{dist}_{G_I}(x_I, z) \geq r_I - R_I.$$

We conclude that the event “ e lies in a positive segment of length $\leq d$ ” is contained in the event $R_I \leq r_I \leq R_I + d$. We may now use Lemma 3.4 to conclude that this event has probability at most $1 - e^{-d/r} \leq d/r$; this establishes (1).

For an edge e of Q , let $S(e)$ be the segment containing e . We bound the number of positive segments as follows.

$$\begin{aligned} \#(\text{positive segments}) &\leq \sum_{e \in E(Q)} \frac{\text{length}(e) \cdot [S(e) \text{ is positive}]}{\text{length}(S(e))} \\ &\leq \sum_{e \in E(Q)} \sum_{i=1}^{\ell} \text{length}(e) \cdot [S(e) \text{ is positive}] \cdot [2^{i-1} \leq \text{length}(S(e)) \leq 2^i] \cdot 2^{-i+1} \\ &\leq 2 \sum_{i=1}^{\ell} \sum_{e \in E(Q)} \text{length}(e) \cdot [S(e) \text{ is positive}] \cdot [\text{length}(S(e)) \leq 2^i] \cdot 2^{-i}. \end{aligned}$$

Hence, by (1), the expected number of positive segments is bounded by

$$2 \sum_{i=1}^{\ell} \sum_{e \in E(Q)} \text{length}(e) \cdot \frac{2^i}{r} \cdot 2^{-i} = \frac{2\ell}{r} \sum_{e \in E(Q)} \text{length}(e) = \frac{\text{length}(Q)}{r} \cdot 2\ell.$$

This finishes the proof, as the total number of edges of Q whose endpoints belong to different clusters is at most twice the number of positive segments. \square

As all the required properties are argued, the proof of Lemma 3.5 is complete.

3.3 Hierarchical decomposition

We will use the decomposition procedure described in the previous section recursively, to construct a hierarchical decomposition of a given graph into smaller and smaller induced subgraphs of the input graph. To formulate the properties of the hierarchical decompositions, we need a few definitions.

A *clustering chain* of a connected graph G (not necessarily edge-weighted) is a sequence C_0, C_1, \dots, C_k of partitions of $V(G)$ such that

- $C_k = \{V(G)\}$;
- C_0 is the discrete partition with every vertex in a separate part; and
- for all $i, j \in \{0, 1, \dots, k\}$ with $i < j$, C_i *refines* C_j in the following sense: every part of C_i is entirely contained in a single part of C_j ;
- for all $i \in \{0, 1, \dots, k\}$ and each $C \in C_i$, the graph $G[C]$ is connected.

For $i \in \{0, 1, \dots, k-1\}$ and a part $C \in C_{i+1}$, we let $C_i[C]$ be the set of parts of C_i that are entirely contained in C ; note that $C_i[C]$ is a partition of C . For an edge e of G , the *level* of e is the largest index $i \in \{1, \dots, k\}$ such that the endpoints of e belong to different parts of C_i .

With these definition in place, we can formulate a statement that summarizes the properties of the hierarchical decomposition.

Lemma 3.6. *There is a randomized polynomial-time algorithm that given a parameter $\delta > 0$ and a connected edge-weighted n -vertex graph G where the distance between any pair of vertices is in the interval $(1, 2^\ell]$, either reports failure or computes a clustering chain C_0, C_1, \dots, C_ℓ of G with the following properties:*

- (Q1) *For every $i \in \{0, 1, \dots, \ell\}$ and every cluster $C \in C_i$, the graph $G[C]$ has diameter at most 2^i .*
- (Q2) *For every $i \in \{0, \dots, \ell-1\}$ and every cluster $C \in C_{i+1}$, the quotient graph $G[C]/C_i[C]$ has diameter at most $480(\ln(2\ell n^2/\delta) + 1)^2$.*
- (Q3) *For every path Q in G and $i \in \{0, \dots, \ell-1\}$, the expected number of edges of Q of level i is upper bounded by $2^{-i} \cdot 8\ell(\ln(2\ell n^2/\delta) + 1) \cdot \text{length}(Q)$.*

The algorithm reports failure with probability at most δ .

The remainder of this section is devoted to the proof of Lemma 3.6.

The algorithm constructing the chain $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_\ell$ is very simple. First, set $\mathcal{C}_\ell := V(G)$. Then, iteratively for every $i = \ell - 1, \dots, 1$, consider the graph $G[C]$ for every part $C \in \mathcal{C}_{i+1}$, and apply the algorithm of Lemma 3.5 to $G[C]$ with parameter

$$r_i := \frac{2^{i-1}}{\ln(2\ell n^2/\delta) + 1}.$$

Thus we obtain a partition $\mathcal{C}_i[C]$ of C , and we set $\mathcal{C}_i := \bigcup_{C \in \mathcal{C}_{i+1}} \mathcal{C}_i[C]$.

Call the resulting sequence of partitions $\mathcal{L} := (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_\ell)$ *good* if the following conditions hold:

- For each $i \in \{0, \dots, \ell\}$ and $C \in \mathcal{C}_i$, the graph $G[C]$ has diameter at most 2^i .
- For each $i \in \{0, \dots, \ell - 1\}$ and each cluster $C \in \mathcal{C}_{i+1}$, the quotient graph $G[C]/\mathcal{C}_i[C]$ has diameter at most $480(\ln(2\ell n^2/\delta) + 1)^2$.

Note that if \mathcal{L} is good, then it is a clustering chain. Indeed, then for every cluster $C \in \mathcal{C}_0$, $G[C]$ has diameter at most 1, so since distances in G are always strictly larger than 1, C cannot have more than one vertex. All other properties of a clustering chain follow directly from the construction.

We now verify that \mathcal{L} is good with high probability.

Claim 3.4. *We have*

$$\mathbb{P}(\mathcal{L} \text{ is not good}) \leq \delta.$$

Proof. Observe that \mathcal{L} is good provided the in each application of the algorithm of Lemma 3.5, the events described is properties (P2) and (P3) do not happen, where in both properties we set

$$t := \ln(2\ell n/\delta).$$

Indeed, when we apply the algorithm to $G[C]$ for some cluster $C \in \mathcal{C}_{i+1}$ ($i \in \{0, 1, \dots, \ell - 1\}$), the two events described in properties (P2) and (P3) can be rephrased as follows:

- Property (P2): There is a cluster $C' \in \mathcal{C}_i[C]$ such that $G[C']$ has diameter larger than

$$2r_i(t + 1 + \ln n) = 2^i.$$

- Property (P3): The quotient graph $G[C]/\mathcal{C}_i[C]$ has diameter larger than

$$120 \cdot \frac{D}{r_i} \cdot (t + 1 + \ln n) = 120D \cdot \frac{(\ln(2\ell n^2/\delta) + 1)^2}{2^{i-1}},$$

where D is the diameter of $G[C]$. Assuming that $D \leq 2^{i+1}$ (which is implied by the event from property (P2) not holding in the run of the algorithm that constructed C), the event above contains the event that $G[C]/\mathcal{C}_i[C]$ has diameter larger than $480(\ln(2\ell n^2/\delta) + 1)^2$.

By Lemma 3.5, each of the above events happens with probability at most $e^{-t} = \frac{\delta}{2\ell n}$. Since there are at most $2\ell n$ events in question, by the union bound we conclude that $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_\ell$ is not good with probability at most δ . \square

Therefore, the algorithm can simply report failure if \mathcal{L} is not good. So from now on we may assume that \mathcal{L} is good, and therefore a clustering chain. Properties (Q1) and (Q2) follow directly from the goodness of \mathcal{L} , so to finish the proof, it remains to show that \mathcal{L} also satisfies property (Q3). We do this in the next claim.

Claim 3.5. *For every path Q in G and $i \in \{0, \dots, \ell - 1\}$, the expected number of edges of Q of level i is upper bounded by $2^{-i} \cdot 8\ell(\ln(2\ell n^2/\delta) + 1) \cdot \text{length}(Q)$.*

Proof. Removing edges of levels larger than i breaks Q into subpaths Q_1, Q_2, \dots, Q_k , where each path Q_j is entirely contained in a single cluster of \mathcal{C}_{i+1} . By property (P4) of Lemma 3.5, the expected number of edges of level i on each path $Q_j, j \in \{1, \dots, k\}$ is upper bounded by

$$\frac{\text{length}(Q_j)}{r_i} \cdot 4\ell = 2^{-i} \cdot 8\ell(\ln(2\ell n^2/\delta) + 1) \cdot \text{length}(Q_j).$$

By the linearity of expectation, the expected total number of edges of level i on Q is upper bounded by

$$2^{-i} \cdot 8\ell(\ln(2\ell n^2/\delta) + 1) \cdot \sum_{j=1}^k \text{length}(Q_j) \leq 2^{-i} \cdot 8\ell(\ln(2\ell n^2/\delta) + 1) \cdot \text{length}(Q). \quad \square$$

This completes the proof of Lemma 3.6.

4 Treewidth tools

In this section we prepare tools related to treewidth. We first give a combinatorial lemma about apex-minor-free graphs that can be regarded as a variation on Lemma 2.1. Let us start with a definition.

For a connected unweighted graph G and integers $a, b, c \geq 0$, an (a, b, c) -contraction sequence consists of

- a sequence of graphs G_0, G_1, \dots, G_b , and
- for every $1 \leq i \leq b$, a collection of pairwise disjoint subgraphs $H_i^1, \dots, H_i^{a_i}$ of G_{i-1} ,

such that the following conditions hold:

- $G_0 = G$ and G_b is a one-vertex graph.
- Each H_i^j is of radius at most c .
- G_i is obtained from G_{i-1} by contracting each of the subgraphs $H_i^j, 1 \leq j \leq a_i$, to a single vertex.
- It holds that $\sum_{i=1}^b a_i \leq a$.

With this definition in place, we can state and prove the promised lemma.

Lemma 4.1. *For every apex graph K there exists a constant α_K such that if a K -minor-free graph G admits an (a, b, c) -contraction sequence for some integers $a, b, c \geq 1$, then the treewidth of G is bounded by $\alpha_K bc\sqrt{a}$.*

Proof. We let α_K be the constant provided by Lemma 2.1.

For $1 \leq i \leq b$ and $1 \leq j \leq a_i$, let v_i^j be a vertex of H_i^j such that all vertices of H_i^j are within distance c from v_i^j in H_i^j . When H_i^j is contracted onto a single vertex in the construction of G_i from G_{i-1} , we imagine that H_i^j is contracted onto v_i^j and we keep the name v_i^j for the obtained vertex. Thus, we have $V(G_i) = \{v_i^j : 1 \leq i \leq a_i\} \cup (V(G_{i-1}) - \bigcup_{j=1}^{a_i} V(H_i^j))$. In particular, $V(G_i) \subseteq V(G)$ for every $0 \leq i \leq b$.

Let $Z = \{v_i^j : 1 \leq i \leq b, 1 \leq j \leq a_i\}$; thus Z is a subset of $V(G)$ of size at most a . The crucial observation is as follows: for every $1 \leq i \leq b$ and $v \in V(G_i)$, we have

$$\text{dist}_{G_i}(v, Z \cap V(G_i)) \geq \text{dist}_{G_{i-1}}(v, Z \cap V(G_{i-1})) - c. \quad (2)$$

Indeed, consider a shortest path Q connecting v with a vertex of $Z \cap V(G_i)$ in G_i . All vertices and edges of Q , except for possibly the last edge and the endpoint $z \in Z \cap V(G_i)$, exist in G_{i-1} . Thus, Q either is entirely contained in G_{i-1} , or it can be extended to a path from v to $Z \cap V(G_{i-1})$ in G_{i-1} by appending a path of length at most c within the graph H_i^j to which z belongs. This proves (2).

As G_b consists of a single vertex, from (2) we obtain that every vertex of $G_0 = G$ is within distance at most bc from a vertex of Z . As $|Z| \leq a$, the lemma follows from Lemma 2.1. \square

Next, we use Lemma 4.1 to prove a statement that is the cornerstone of our approach. Intuitively it says that in a clustering chain of an edge-weighted apex-minor-free graph, one can find many disjoint balanced separators consisting of clusters. Again, we need a few definitions.

For a (possibly edge-weighted) graph G , a *cut* is a family of pairwise disjoint vertex subsets of G , each inducing a connected subgraph of G . For a cut \mathcal{A} , let $F(\mathcal{A})$ be the set of those edges e of G for which there exists $A \in \mathcal{A}$ that contains exactly one endpoint of e . Note that since every part $A \in \mathcal{A}$ induces a connected subgraph of G , $G[A]$ is a connected component of $G - F(\mathcal{A})$. We say that a cut \mathcal{A} is *balanced* if the vertex set of every component of $G - F(\mathcal{A})$ is contained in \mathcal{A} or consists of at most $|V(G)|/2$ vertices. A pair of cuts $\mathcal{A}, \mathcal{A}'$ is *non-conflicting* if every set in $\mathcal{A} \cap \mathcal{A}'$ consists of one vertex.

Let $\mathcal{L} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_k)$ be a clustering chain of a connected graph G . We say that a cut \mathcal{A} *respects* \mathcal{L} if $\mathcal{A} \subseteq \bigcup_{i=0}^k \mathcal{C}_i$; in other words, every part $A \in \mathcal{A}$ is a cluster belonging to one of partitions $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_k$. A *cut packing* in \mathcal{L} is a family of pairwise non-conflicting cuts that respect \mathcal{L} .

Lemma 4.2. *For every apex graph K there exists a constant $\gamma_K > 0$ such that the following holds. Let G be a K -minor-free graph and let $\mathcal{L} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_k)$ be a clustering chain of G such that for every $i \in \{0, 1, \dots, k-1\}$ and $C \in \mathcal{C}_{i+1}$, the quotient graph $G[C]/\mathcal{C}_i[C]$ has radius at most c . Suppose \mathcal{F} is a cut packing in \mathcal{L} such that every cut in \mathcal{F} has size at most τ and*

$$|\mathcal{F}| \leq \gamma_K \cdot \frac{\tau}{k^2 c^2}.$$

Then there exists a balanced cut \mathcal{A} respecting \mathcal{L} and of size at most τ such that $\mathcal{F} \cup \{\mathcal{A}\}$ is also a cut packing in \mathcal{L} . Moreover, given G, \mathcal{L} , and \mathcal{F} , such a balanced cut \mathcal{A} can be found in polynomial time.

Proof. Let $\widehat{\mathcal{C}} := \bigcup_{i=0}^k \mathcal{C}_i$; the elements of $\widehat{\mathcal{C}}$ will be called *clusters*. Call a cluster $D \in \widehat{\mathcal{C}}$ *free* if D consists of one vertex or is not contained in any member of \mathcal{F} . If D is not free, it is called *used*. A free cluster D is *maximal* if there is no cluster $D' \supsetneq D$ that is free. Let \mathcal{D} be the family consisting of all maximal free clusters. Note that \mathcal{D} is a partition of $V(G)$. Let $H = G/\mathcal{D}$. In other words, H is obtained from G by contracting every maximal free cluster into a single vertex.

Let $\xi := |\mathcal{F}|$. The crucial observation is as the following: H admits a $(\tau\xi, k, c)$ -contraction sequence. Indeed, we can proceed through levels $i = 1, 2, \dots, k$, and upon considering level i , contract all used clusters contained in \mathcal{C}_i that have not been contracted so far. Since there are at most $\tau\xi$ used clusters in total, and at every point we contract a graph of radius at most c (due to the assumption on the quotient graphs appearing in the clustering chain \mathcal{L}), this is a $(\tau\xi, k, c)$ -contraction sequence.

By Lemma 4.1, the treewidth of H is bounded by $\alpha_K \cdot \sqrt{\tau\xi} \cdot kc$ for a constant $\alpha_K > 0$ depending only on K . Using the algorithm of Feige et al. [27, Theorem 6.4], in polynomial time we can compute a tree decomposition (T, bag) of G where every bag has size at most $\beta_K \cdot \sqrt{\tau\xi} \cdot kc$, where $\beta_K > 0$ is again a constant depending only on K .

We now use the following standard statement about tree decompositions, see e.g. [19, Lemma 7.19].

Claim 4.1. *Suppose (T, bag) is a tree decomposition of a graph G and $\omega: V(G) \rightarrow \mathbb{R}_{\geq 0}$ is a nonnegative weight function on the vertices of G . Then there exists a node x of T such that for every connected component J of $G - \text{bag}(x)$, we have $\omega(V(J)) \leq \omega(V(G))/2$.*

Setting $\omega(D) := |D|$ for each $D \in \mathcal{D}$, we find that there exists a node x of T such that every connected component of $H - \text{bag}(x)$ consists of clusters of total size at most $|V(G)|/2$. Note that such a node x can be found in polynomial time by examining the nodes of T one by one. Denote $\mathcal{A} := \text{bag}(x)$.

As $\mathcal{A} \subseteq V(H) = \mathcal{D} \subseteq \widehat{\mathcal{C}}$, \mathcal{A} is a cut that respects \mathcal{L} . Since \mathcal{A} consists only of free clusters, it is also non-conflicting with every member of \mathcal{F} . Therefore, $\mathcal{F} \cup \{\mathcal{A}\}$ is a cut packing in \mathcal{L} . Further, the property of \mathcal{A} described in the previous paragraph ensures that \mathcal{A} is balanced. Finally, observe that

$$|\mathcal{A}| \leq \beta_K \cdot \sqrt{\tau\xi} \cdot kc \leq \frac{\beta_K}{\sqrt{\gamma_K}} \cdot \tau.$$

Hence, if we set $\gamma_K := \beta_K^2$, then we have $|\mathcal{A}| \leq \tau$ and \mathcal{A} satisfies all the required properties. \square

5 Embedding

In this section we prove Theorem 1.1 for the case of apex-minor-free graphs. Let then G be the input edge-weighted graph that is K -minor-free, where K is a fixed apex graph. We may assume that G is connected, for otherwise we treat every connected component of G separately. We may also assume that for every $uv \in E(G)$, we have $\text{length}(uv) = \text{dist}_G(u, v)$. Let n be the vertex count of G and let $\hat{\ell}$ be the least integer such that the ratio of the sum of the weights of all edges of G to the smallest edge weight of G is strictly less than $2^{\hat{\ell}}$. Note that $\hat{\ell} = \mathcal{O}_K(\ell + \ln n)$ where ℓ is the logarithm of the stretch of the metric of G . By rescaling, we assume that every distance in G is larger than one and the total length of all edges of G is at most $2^{\hat{\ell}}$. We have that any simple path in G has length at most $2^{\hat{\ell}}$, hence, in particular, any connected subgraph of G has diameter at most $2^{\hat{\ell}}$.

Let us also fix the accuracy parameter $\varepsilon > 0$ given on input.

Procedure embed. We first describe a recursive embedding procedure `embed`. The input to the procedure is a connected induced subgraph G' of G . The output is a metric embedding (H', η') of G' together with an elimination forest F' of H' . Procedure `embed`(G') proceeds as follows:

- If G' has one vertex, return the trivial embedding: $H' = G'$ and η' is the identity on $V(G')$.
- Otherwise, apply a subprocedure `split`(G'), which we will describe later. This procedure finds a subset of edges $F \subseteq E(G)$ (called *cutedges*) and a subset of vertices $Z \subseteq V(G)$ (called *portals*).
- Let \mathcal{D} be the set of connected components of the graph $G' - F$. For every component $D \in \mathcal{D}$, recurse on D to obtain a metric embedding (H_D, η_D) of D and an elimination forest F_D of H_D .
- Construct an embedding (H', η') of G' as follows. First, H' is obtained by taking the disjoint union of graphs $\{H_D : D \in \mathcal{D}\}$ and adding, for every portal $z \in Z$, a copy z' of z ; this copy is made adjacent to every vertex u of G' by an edge of length $\text{dist}_G(z, u)$. Next, we set $\eta' := \bigcup_{D \in \mathcal{D}} \eta_D$.
- Finally, construct an elimination forest F' of H' by first taking the disjoint union of forests $\{F_D : D \in \mathcal{D}\}$, and then iteratively adding all vertices $\{z' : z \in Z\}$ in any order. Each time z' is added to F' , we make it the new root and attach all former roots as children of z' .

A straightforward induction shows that procedure `embed` indeed outputs a metric embedding of the graph provided on input. Hence, it remains to specify the procedure `split`. This will be done so that the following two conditions are satisfied:

- The depth of the recursion tree of `embed` is bounded by $\hat{\ell} \cdot \lceil \log_2 n \rceil = \mathcal{O}((\ell + \ln n) \ln n)$.
- In every recursion call, `split` returns a portal set Z of size polynomial in $\hat{\ell}$, $1/\varepsilon$, and $\ln n$.

Thus, the depth of the eventually constructed elimination forest of G will be polynomial in $\hat{\ell}$, $1/\varepsilon$, and $\ln n$, which will certify the treedepth of the target graph in the obtained metric embedding of G .

Procedure split. We now describe the procedure `split`. Recall that the input to this procedure is a connected subgraph G' of G , where we may assume that G' has at least two vertices. It follows that D , the diameter of G' , is strictly larger than 1. Let then $\ell' \in \{1, \dots, \hat{\ell}\}$ be such that $2^{\ell'-1} < D \leq 2^{\ell'}$. We call ℓ' the *level* of the subgraph G' . Let

$$\begin{aligned} \delta &:= \frac{\varepsilon}{c \hat{\ell} n \ln^2 n}, & \xi &:= 64 \hat{\ell}^3 \lceil \log_2 n \rceil \left(\ln(2 \hat{\ell} n^2 / \delta) + 1 \right) \cdot (1/\varepsilon), \\ \sigma &:= 480 (\ln(2 \hat{\ell} n^2 / \delta) + 1)^2, & \tau &:= (\xi + 1) \cdot \gamma_K \hat{\ell}^2 \sigma^2, \end{aligned}$$

where $c > 0$ is a constant that will be determined later and γ_K is the constant provided by Lemma 4.2. Note that

$$\tau \in \mathcal{O}_K \left(\hat{\ell}^5 / \varepsilon \cdot \ln n \cdot (\ln n + \ln \hat{\ell} + \ln(1/\varepsilon))^5 \right).$$

We apply the algorithm of Lemma 3.6 to G' with parameter δ , thus obtaining a clustering chain $\mathcal{L} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_{\ell'})$ of G' . It may happen that this algorithm reports failure; this happens with probability at most δ . In this case, we terminate the whole effort, and instead of trying to compute a metric embedding of G ourselves, we apply the algorithm of Fakcharoenphol, Rao, and Talwar [26] that

finds an embedding of G into a tree (thus, a graph of treedepth $\mathcal{O}(\ln n)$) with expected multiplicative distortion $\mathcal{O}(\ln n)$. From now on, we suppose the algorithm of Lemma 3.6 succeeded in finding the clustering chain \mathcal{L} .

Note that by Lemma 3.6, property (Q2), for every $i \in \{0, \dots, \ell' - 1\}$ and cluster $C \in \mathcal{C}_{i+1}$, the diameter of the quotient graph $G[C]/\mathcal{C}_i[C]$ is bounded by σ . So starting with $\mathcal{F} = \emptyset$, we may apply the algorithm of Lemma 4.2 repeatedly to find a cut packing \mathcal{F} in \mathcal{L} consisting of balanced cuts of size at most τ so that

$$|\mathcal{F}| \geq \frac{\tau}{\gamma_K \hat{\ell}^2 \sigma^2} = \xi + 1.$$

Once \mathcal{F} is constructed, we discard from \mathcal{F} the cut $\{V(G')\}$, if present; let \mathcal{F}' be the obtained packing of size at least ξ . Then we pick a cut $\mathcal{A} \in \mathcal{F}'$ uniformly at random and define the output of $\text{split}(G')$ as follows:

- as cutedges we set $F := F(\mathcal{A})$; and
- as portals we set a set Z consisting of one vertex from each cluster $A \in \mathcal{A}$, selected arbitrarily.

This concludes the description of procedure split .

Properties. We first observe that in procedure embed , at every recursion call we observe progress in significantly reducing either the diameter or the vertex count.

Claim 5.1. *Suppose a call $\text{embed}(G')$ invokes a subcall of $\text{embed}(G'')$, for some induced subgraph G'' of G' . Then $|V(G'')| \leq |V(G')|/2$ or the level of G'' is strictly smaller than the level of G' .*

Proof. Let ℓ' be the level of G' . Recall that procedure $\text{embed}(G')$ calls $\text{split}(G')$ to find suitable cutedges F and portals Z to recurse. Let \mathcal{A} be the cut used by $\text{split}(G')$ to find define F and Z . Recall that \mathcal{A} is balanced, which means that every connected component of $G' - F(\mathcal{A})$ that is not a member of \mathcal{A} has at most $|V(G')|/2$ vertices. On the other hand, if $\mathcal{L} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_{\ell'})$ is the hierarchical clustering considered in procedure $\text{split}(G')$, then $\mathcal{A} \subseteq \mathcal{C}_0 \cup \mathcal{C}_1 \cup \dots \cup \mathcal{C}_{\ell'-1}$, because the cut $\mathcal{C}_{\ell'} = \{V(G')\}$ was explicitly excluded from sampling. Since every cluster within partitions $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_{\ell'-1}$ induces a subgraph of has diameter at most $2^{\ell'-1}$ (Lemma 3.6, property (Q1)), it follows that for each $A \in \mathcal{A}$, $G[A]$ has level strictly smaller than ℓ' . The claim follows. \square

An immediate consequence of Claim 5.1 is the following.

Claim 5.2. *The depth of the recursion tree of $\text{embed}(G)$ is bounded by $\hat{\ell} \lceil \log_2 n \rceil$.*

Proof. Consider any root-to-leaf path P in the recursion tree of $\text{embed}(G)$. By Claim 5.1, we may color every edge of P , say with parent call $\text{embed}(G')$ and child call $\text{embed}(G'')$, blue if $|V(G'')| \leq |V(G')|/2$ and red if the level of G'' is strictly smaller than the level of G' . Clearly, P cannot contain more than $\hat{\ell}$ red edges in a row, and there are no more edges on P after the $\lceil \log_2 n \rceil$ th blue edge. The claim follows. \square

Observe that every call of procedure split returns a portal set Z of size at most τ . Hence, from Claim 5.2 it follows that if $\text{embed}(G)$ returns a metric embedding (H, η) and an elimination forest F of H , then the depth of F is bounded by

$$1 + \tau \hat{\ell} \lceil \log_2 n \rceil = \mathcal{O}_K((\ell + \ln n)^6 / \varepsilon \cdot \ln^2 n \cdot (\ln n + \ln \ell + \ln(1/\varepsilon))^5).$$

Consequently, the above expression is also an upper bound on the treedepth of H , as desired.

It remains to bound the expected distortion.

Consider first the corner case when one of the calls of procedure split failed. Then we invoked the algorithm of [26], which computed a metric embedding into a tree with expected multiplicative distortion $\mathcal{O}(\ln n)$. Every invocation of procedure split fails with probability at most δ , and by Claim 5.2, the total number of calls to split is bounded by $n \hat{\ell} \lceil \log_2 n \rceil$. Therefore, the total contribution to the expected multiplicative distortion from this case is

$$\delta \cdot n \hat{\ell} \lceil \log_2 n \rceil \cdot \mathcal{O}(\ln n),$$

which is upper bounded by $\varepsilon/2$ assuming we choose c large enough.

From now on we assume that all calls to `split` succeeded and our algorithm has indeed constructed a metric embedding (H, η) of G . Fix a pair of vertices $u_1, u_2 \in V(G)$ and let Q be a shortest path connecting u_1 and u_2 in G . Note that within the recursion tree of `embed`(G), there exists a unique call `embed`(G') such that the whole path Q is contained in G' , but this cannot be said about any call `embed`(G'') invoked within `embed`(G'). This means that Q contains at least one of the cutedges belonging to $F(\mathcal{A})$, where \mathcal{A} is the cut considered in the call `split`(G'). Equivalently, there exists a cluster $A \in \mathcal{A}$ such that Q contains an edge with exactly one endpoint in A . Let z be the vertex of A chosen to the portal set Z by `split`(G'), and let $z' \in V(H)$ be the copy of z created within `embed`(G'). By triangle inequality, we have

$$\text{dist}_H(\eta(u_1), \eta(u_2)) \leq \text{dist}_H(\eta(u_1), z') + \text{dist}_H(\eta(u_2), z') = \text{dist}_G(u_1, z) + \text{dist}_G(u_2, z).$$

Since Q intersects A , say at vertex q , we have

$$\text{dist}_G(u_1, z) \leq \text{dist}_G(u_1, q) + \text{dist}_G(q, z) \leq \text{dist}_G(u_1, q) + D,$$

where D is the diameter of $G[A]$. Similarly $\text{dist}_G(u_2, z) \leq \text{dist}_G(u_2, q) + D$, implying that

$$\begin{aligned} \text{dist}_H(\eta(u_1), \eta(u_2)) &\leq \text{dist}_G(u_1, z) + \text{dist}_G(u_2, z) \leq \text{dist}_G(u_1, q) + \text{dist}_G(u_2, q) + 2D \\ &= \text{length}(Q) + 2D = \text{dist}_G(u_1, u_2) + 2D. \end{aligned}$$

For an index $h \in \{0, 1, \dots, \hat{\ell} \lceil \log_2 n \rceil - 1\}$, we introduce a random variable X_h defined as follows:

- If at depth h of the recursion tree of `embed`(G) there is no call `embed`(G') such that Q is entirely contained in G' , then $X_h = 0$.
- Otherwise, consider the unique call `embed`(G') at depth h such that Q is entirely contained in G' . Then we let

$$X_h := \sum_{A \in \mathcal{A}} 2\text{diam}(G[A]) \cdot |E(Q) \cap F(A)|, \quad (3)$$

where \mathcal{A} is the cut considered in procedure `split`(G'), $F(A)$ is the set of edges with exactly one endpoint in A , and $\text{diam}(G[A])$ is the diameter of $G[A]$.

The reasoning of the previous paragraph together with Claim 5.2 proves that

$$\text{dist}_H(\eta(u_1), \eta(u_2)) \leq \text{dist}_G(u_1, u_2) + \sum_{h=0}^{\hat{\ell} \lceil \log_2 n \rceil - 1} X_h.$$

Consider then a fixed $h \in \{0, 1, \dots, \hat{\ell} \lceil \log_2 n \rceil - 1\}$ such that there is a call `embed`(G') at depth h such that Q is entirely contained in G' . Recall that within `split`(G') we constructed a clustering chain $\mathcal{L} = (\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_{\ell'})$ and a cut packing \mathcal{F}' , and we sampled a balanced cut \mathcal{A} from \mathcal{F}' . Since members of \mathcal{F}' are pairwise non-conflicting, for every fixed cluster $C \in \bigcup_{i=1}^{\ell'-1} \mathcal{C}_i$ of positive diameter, the probability that C belongs to \mathcal{A} is upper bounded by $\frac{1}{|\mathcal{F}'|} \leq \frac{1}{\xi}$. Moreover, if $C \in \mathcal{C}_i$, then the diameter of $G[A]$ is upper bounded by 2^i (Lemma 3.6, property (Q1)). By property (Q3) of Lemma 3.6,

we infer the following:

$$\begin{aligned}
\mathbb{E}X_h &\leq \mathbb{E} \left(\sum_{i=1}^{\ell'-1} \sum_{C \in \mathcal{C}_i} |E(Q) \cap F(C)| \cdot [C \in \mathcal{A}] \cdot 2^{i+1} \right) \\
&\leq \frac{1}{\xi} \cdot \sum_{i=1}^{\ell'-1} 2^{i+2} \cdot \mathbb{E} \left(\sum_{e \in E(Q)} [e \text{ is of level } i \text{ in } \mathcal{L}] \right) \\
&\leq \frac{1}{\xi} \cdot \sum_{i=1}^{\ell'-1} 2^{i+2} \cdot 2^{-i} \cdot 8\hat{\ell}(\ln(2\hat{\ell}n^2/\delta) + 1) \cdot \text{length}(Q) \\
&\leq \frac{1}{\xi} \cdot 32\ell^2(\ln(2\hat{\ell}n^2/\delta) + 1) \cdot \text{length}(Q) \\
&= \frac{\varepsilon}{2\hat{\ell}\lceil \log_2 n \rceil} \cdot \text{length}(Q).
\end{aligned}$$

By (3), we conclude that

$$\mathbb{E} \text{dist}_H(\eta(u_1), \eta(u_2)) \leq (1 + \varepsilon/2) \text{dist}_G(u_1, u_2).$$

Taking into account the additional summand of $(\varepsilon/2) \cdot \text{dist}_G(u_1, u_2)$ from the case when one of the calls to `split` reports a failure, we conclude that the expected multiplicative distortion of the embedding is $(1 + \varepsilon)$, as desired. This concludes the proof of Theorem 1.1.

6 Extension to proper minor-closed graph classes

In this section we extend the main result to K -minor-free graphs for a fixed graph K that is not necessarily an apex graph. The arguments of Section 4 crucially rely on the graph class excluding an apex graph. Here, we design a workaround using the structure theorem of Robertson and Seymour [50].

Tools from the graph minors theory. We need a bit more notation on tree decompositions. Let (T, bag) be a tree decomposition of a graph G . For an edge $st \in E(T)$, the *adhesion* of st , denoted $\text{adh}(st)$, equals $\text{bag}(s) \cap \text{bag}(t)$. For a vertex $t \in V(T)$, the *torso* of t , denoted $\text{torso}(t)$, is the graph obtained from $G[\text{bag}(t)]$ by turning the adhesion $\text{adh}(st)$ into a clique, for every $s \in N_T(t)$. If G is an edge-weighted graph, every new edge uv in $\text{adh}(st)$ is assigned length equal to $\text{dist}_G(u, v)$.

We need the following corollary of the structure theorem for graphs excluding a fixed minor. Unfortunately, we did not find a similar form in the literature; we discuss how to obtain this corollary from existing results in Appendix B.

Theorem 6.1. *For every graph K_0 there exists a constant c_{apices} , an apex graph K , and a polynomial-time algorithm that, given a K_0 -minor-free graph G , computes a tree decomposition (T, bag) of G and for every $t \in V(T)$ a set $\text{apices}(t) \subseteq \text{bag}(t)$ of size at most c_{apices} so that for every $t \in V(T)$, the graph $\text{torso}(t) - \text{apices}(t)$ does not contain K as a minor.*

Note that in the decomposition (T, bag) returned by Theorem 6.1, every adhesion needs to be of size strictly smaller than $c_{\text{apices}} + |V(K)|$: an adhesion is turned into a clique in the torso while the clique number of $\text{torso}(t) - \text{apices}(t)$ is strictly less than $|V(K)|$ as this graph is K -minor-free.

We also need the following observation (proved formally in Appendix B).

Lemma 6.2. *For every apex graph K there exists an apex graph K' such that if G is K -minor-free and G' is created from G by adding, for every nonempty clique A in G , a new vertex v_A with neighborhood $N_{G'}(v_A) = A$, then G' is K' -minor-free.*

Initial steps. Armed with Theorem 6.1, we now describe modifications to the `split` procedure. Recall that the input is an induced subgraph G' of the input graph G that is connected and has at least two vertices. We denote by D the diameter of G' and by $\ell' \in \{1, \dots, \hat{\ell}\}$ the least integer such that $D \leq 2^{\ell'}$. Recall also that in G , we assume that for every $uv \in E(G)$, the length of uv equals $\text{dist}_G(u, v)$.

Instead of applying Lemma 3.6 directly to G' , we proceed as follows. We apply the algorithm of Theorem 6.1 to G' , obtaining a tree decomposition (T, bag) and subsets $\text{apices}(t)$ for all $t \in V(T)$. We find a node $t \in V(T)$ such that every connected component of $G' - \text{bag}(t)$ has at most $|V(G')|/2$ vertices (see Claim 4.1). We denote $G^1 := G' - \text{apices}(t)$ and $G^2 := \text{torso}(t) - \text{apices}(t)$. We designate $\text{apices}(t)$ as portals and delete them from G' ; thus, we will focus on G^1 and G^2 in what follows.

Modifying the clustering step. We apply the clustering step (Lemma 3.6) to G^1 with the following modification: in the tie-breaking order \preceq , we put all vertices of $V(G^2)$ before the remaining vertices of $V(G^1) - V(G^2)$. In other words, when the algorithm chooses the next center of a cluster, we prefer a center lying in $\text{bag}(t) - \text{apices}(t)$. We obtain a clustering chain $\mathcal{L}^1 = (\mathcal{C}_0^1, \mathcal{C}_1^1, \dots, \mathcal{C}_{\ell'}^1)$.

Recall that in $\text{torso}(t)$, each new edge put within an adhesion has length equal to the distance of the endpoints in G' . Hence, both in G^1 and G^2 we have the property that the length of an edge equals the distance between its endpoints. This implies that, for every cluster C in \mathcal{L} , if C intersects $V(G^2)$, then $G^2[C \cap V(G^2)]$ is a connected subgraph of G^2 of diameter not larger than the diameter of $G^1[C]$. Hence, we can obtain from \mathcal{L}^1 a clustering chain $\mathcal{L}^2 = (\mathcal{C}_0^2, \dots, \mathcal{C}_{\ell'}^2)$ of G^2 by restricting every cluster C to $C \cap V(G^2)$ and deleting clusters disjoint with $V(G^2)$. Observe that, because adhesions are turned into cliques in the torso, for every $1 \leq i < \ell'$ and $C \in \mathcal{C}_{i+1}^1$ with $C \cap V(G^2) \neq \emptyset$, the diameter of $G^2[C \cap V(G^2)]/\mathcal{C}_i^2[C \cap V(G^2)]$ is not larger than the diameter of $G^1[C]/\mathcal{C}_i^1[C]$.

Furthermore, we make the following crucial observation: the obtained distribution of clustering chain \mathcal{L}^2 of G^2 is the same as if we just applied the algorithm of Lemma 3.6 directly to G^2 without any modifications (and with the same order \preceq , but restricted to the vertices of G^2).

Obtaining a cut packing. Having obtained \mathcal{L}^1 and \mathcal{L}^2 , we proceed as follows. Let G^3 be an unweighted graph obtained from G^2 by adding, for every $s \in N_T(t)$, a vertex v_s adjacent to all vertices of $\text{adh}(st)$. Theorem 6.1 implies that G^2 is K -minor-free for an apex graph K that depends only on K_0 . Now, Lemma 6.2 implies that G^3 is K' -minor-free for another apex graph K' that depends only on K . We extend \mathcal{L}^3 to a clustering chain of G^3 by adding at every level singleton clusters $(\{v_s\})_{s \in N_T(t)}$, except for the top level where we now have $V(G^3)$ as the only cluster. Note that the diameter of every quotient graph discussed in property (Q2) increased by at most 2. This increase is negligible and will be promptly ignored in the argumentation to follow.

We use the graph G^3 and the clustering chain \mathcal{L}^3 in the repeated application of Lemma 4.2 to obtain a large cut packing \mathcal{F} . We do one modification to the process: in the definition of a balanced separator, we put weight 1 on every vertex of $V(G^2)$ but, for every $s \in N_T(t)$, the weight of v_s is defined as $|V_s|$, where V_s is the set of vertices of $V(G^1) - V(G^2)$ that belong to any bag of the connected component of $T - \{st\}$ containing s . Hence, the total weight of all vertices of G^3 is $|V(G^1)|$ and every $\mathcal{A} \in \mathcal{F}$ has the property that every connected component of $G^3 - F(\mathcal{A})$ is either an element of \mathcal{A} or has weight at most $|V(G^1)|/2$.

To obtain \mathcal{F}' from \mathcal{F} , we not only discard the cut $\{V(G^3)\}$ if present, but also, for every $\mathcal{A} \in \mathcal{F}$, whenever \mathcal{A} contains a cluster $\{v_s\}$ for some $s \in N_T(t)$, we replace it within \mathcal{A} with single-vertex clusters $\{\{v\} : v \in \text{adh}(st)\}$. Since we replace $\{v_s\}$ with single-vertex clusters, the members of the cut packing \mathcal{F}' are still pairwise non-conflicting. Since every adhesion is of size less than $c_{\text{apices}} + |V(K)|$, we increase the size of every element of \mathcal{F}' by at most a constant multiplicative factor. Since the weight of every v_s is at most $|V(G^1)|/2$, every cut in \mathcal{F}' is still balanced. We obtain that every cut $\mathcal{A} \in \mathcal{F}'$ contains only clusters of \mathcal{L}^2 .

Cutting. The cut packing \mathcal{F}' in clustering chain \mathcal{L}^2 projects to a cut packing \mathcal{F}^1 in clustering chain \mathcal{L}^1 as follows: for every $\mathcal{A} \in \mathcal{F}'$ and $C \in \mathcal{A}$, we put into \mathcal{A}^1 the unique inclusion-wise minimal cluster C' appearing in \mathcal{L}^1 such that $C' \cap V(G^2) = C$, and \mathcal{F}^1 consists of all cuts \mathcal{A}^1 constructed in this manner. Note that thus, if C is a single-vertex cluster, then $C' = C$.

We make two observations. First, it follows directly from the construction above and the fact that the members of \mathcal{F}' are pairwise non-conflicting that the members of \mathcal{F}^1 are also non-conflicting. Second, thanks to the weights used in G^3 , the cuts in \mathcal{F}^1 are balanced in the usual way in the graph G^1 : for every $\mathcal{A}^1 \in \mathcal{F}^1$, every component of $G^1 - F(\mathcal{A}^1)$ is either a member of \mathcal{A}^1 or contains at most $|V(G^1)|/2$ vertices. To see this, note that for every component D^1 of $G^1 - F(\mathcal{A}^1)$ that is not member of \mathcal{A}^1 , if \mathcal{A} is the member of \mathcal{F}' corresponding to \mathcal{A}^1 , then there is a component D' of $G^3 - F(\mathcal{A})$ that is not a member of \mathcal{A} , $D' \cap V(G^2) = D^1 \cap V(G^2)$, and for every $s \in N_T(t)$ we have that $D^1 \cap V_s \neq \emptyset$ implies $v_s \in D'$. Then, $|D^1|$ is bounded by the weight of D' , which in turn is bounded by $|V(G^1)|/2$ as every cut in \mathcal{F}' is balanced.

Hence, we can proceed with G^1 and \mathcal{F}^1 as in the original algorithm: choose $\mathcal{A}^1 \in \mathcal{F}^1$ at random and cut along it.

Since we obtained a family of pairwise non-conflicting balanced cuts in G^1 in the end, the analysis of the depth of the recursion and the expected distortion is the same. The depth of the obtained elimination forest of the host graph grew by a constant factor: we add a constant number of vertices of $\text{apices}(t)$ in the beginning and then, in the process of constructing \mathcal{F}' from \mathcal{F} , we may have increased the size of cuts by a constant factor. This finishes the proof of the extension to K -minor-free graphs, for an arbitrary fixed graph K .

7 Algorithmic applications

In this section, we present some algorithmic applications of our embedding in Theorem 1.1, specifically, in solving the CVRP and capacitated clustering/facility location problems.

The algorithms for all of these problems follow the same framework. Suppose that we are given an instance Π_G of a problem Π to solve on an input planar graph G , and OPT_G be the optimal solution of Π_G . We use $|\text{OPT}_G|$ to denote the value of OPT_G . Our algorithm for solving Π_G has three steps.

- **Step 1.** Normalize the edge weights of G so that $w(e) \in [1, n^{\mathcal{O}(1)}]$ for every edge $e \in E(G)$. Then embed G into a graph H of treewidth $k_H = \text{polylog}(n)$ by Theorem 1.1 via embedding η .
- **Step 2.** Let Π_H be the instance of problem Π to solve on graph H . We solve Π_H in H to get an optimal solution OPT_H of Π_H .
- **Step 3.** Lift OPT_H to a solution $\widehat{\text{OPT}}_G$ of the problem Π_G in the input graph G .

We note that if we skip the normalization in Step 1, our algorithm will have a factor of $2^{\text{poly}(\ell)}$ in the running time. We show in all of the problems we consider below, we can do the normalization step, and the framework gives a $(1 + \varepsilon)$ approximation algorithm.

Capacitated vehicle routing problem. Here we show the implementation of each step for the CVRP. In Step 1, we first compute a constant approximation solution, denoted by A , for the CVRP problem in polynomial time [35]. (Indeed, even any $\text{poly}(n)$ -approximation suffices for our purpose.). Then we round every edge of weight at most $\varepsilon|A|/(\alpha n^3)$ to $\varepsilon|A|/(\alpha n^3)$ for some big constant α , and remove every edge of weight more than $|A|$ out of the graph. Then scale every edge so that the minimum weight edge is 1. The maximum weight edge will be $\mathcal{O}(n^3/\varepsilon)$. In Step 2, we use the algorithm for small treewidth graphs by Jayaprakash and Salavatipour [37]:

Theorem 7.1 (Jayaprakash and Salavatipour [37]). *Given a parameter $\varepsilon > 0$ and graph G of treewidth at most k , one can find a $(1 + \varepsilon)$ -approximate solution for :*

- the CVRP with unit demands in time $n^{\mathcal{O}(k^2 \log^3 n/\varepsilon)}$
- the CVRP with splittable/unsplittable demands in time $n^{\mathcal{O}(k^2 \log^{2c+3} n/\varepsilon^2)}$ when the capacity of the vehicle is $Q = n^{\mathcal{O}(\log^c n)}$ for some constant $c > 0$.

Let OPT_H be the $(1 + \varepsilon)$ -approximate solution returned by the algorithm in Theorem 7.1. Then in Step 3, we simply look at every two consecutive vertices \hat{u} and \hat{v} on the same tour that correspond to two vertices u and v in G , respectively, and replace the portion of the tour between \hat{u} and \hat{v} by the shortest path from u to v in G . Let $\widehat{\text{OPT}}_G$ be the obtained tour. We now show that:

$$E[\widehat{\text{OPT}}_G] \leq (1 + 4\varepsilon)|\text{OPT}_G| \quad (4)$$

and by scaling ε , we get a $(1 + \varepsilon)$ approximation.

First, observe that since each edge belongs to at most n tours, and it could appear at most n times in a tour, the normalization in Step 1 introduces a total error of $|E(G)|n^2\varepsilon/A/(\alpha n^3) = \varepsilon|\text{OPT}_G|$ for a sufficiently large constant α .

We abuse notation by using OPT_G to denote an optimal solution of the CVRP (with unit/splittable/unsplittable demands) in G after the normalization step. Let $\widehat{\text{OPT}}_H$ be the solution of CVRP in H obtained by replacing every edge $(x, y) \in \text{OPT}_G$ by a shortest path between $\eta(x)$ and $\eta(y)$ in H . Let $\alpha_{x,y}$ be the number of times edge (x, y) appears in all tours in OPT_G . We have:

$$\begin{aligned} \mathbb{E}[\widehat{\text{OPT}}_H] &= \sum_{(x,y) \in \text{OPT}_G} \alpha_{x,y} \mathbb{E}[d_H(\eta(x), \eta(y))] \\ &\leq (1 + \varepsilon) \sum_{(x,y) \in \text{OPT}_G} \alpha_{x,y} d_G(x, y) = (1 + \varepsilon)|\text{OPT}_G| \end{aligned} \quad (5)$$

Since OPT_H is a $(1 + \varepsilon)$ -approximate solution of CVRP in H , $|\text{OPT}_H| \leq (1 + \varepsilon)|\widehat{\text{OPT}}_H|$. Thus, Equation (5) gives $\mathbb{E}[|\text{OPT}_H|] \leq (1 + \varepsilon)^2|\text{OPT}_G| \leq (1 + 3\varepsilon)|\text{OPT}_G|$ when $\varepsilon \leq 1$. Then Equation (4) follows from that $|\widehat{\text{OPT}}_G| \leq |\text{OPT}_H|$ and that the normalization step introduce an error of $\varepsilon|\text{OPT}_G|$.

To complete the proof of Corollary 1.2, it remains to analyze the running time. The embedding step (Step 1) and the lifting step (Step 3) can both be done in polynomial time. The most expensive step is the second step. The running time in Corollary 1.2 follows directly from Theorem 7.1 as the treewidth $k = \text{poly}(\log n, 1/\varepsilon)$ by Theorem 1.1.

We remark that for the unsplittable/splittable versions of the CVRP, we also obtain QPTASes. However, the capacity of the vehicle is now restricted to be $Q = n^{\log^{\mathcal{O}(1)}(n)}$ due to the restriction in Theorem 7.1. Removing this restriction is an interesting open problem for future work.

Capacitated k -median and capacitated/uncapacitated facility location. The algorithms for these problems use the same framework we set up above. For the normalization step, we also use a simple $\mathcal{O}(\log n)$ approximation obtained via tree embedding [26]. For the dynamic programming step, we can solve both problems in time $n^{\mathcal{O}(k)}$ where k is the treewidth. One could improve this running time to $(\log n/\varepsilon)^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$ by discretizing the distances to be a power of $(1 + \varepsilon)$ but this does not change the final running time of our algorithm. The final running time of the algorithm is dominated by the running time of the dynamic programming step, which is $2^{\text{poly}(1/\varepsilon, \log(n))}$. The approximation factor analysis is the same as the analysis of CVRP, implying Corollary 1.3 and Corollary 1.4.

Acknowledgements

Over the last few years, we have conducted various discussions on embeddings of planar graphs into low treewidth metrics. We thank Christian Wulff-Nilsen for participating in some of them in the last few months and asking tricky questions. Marcin and Michał would like to acknowledge discussions on the topic with François Dross, when he was a postdoc in Warsaw around 2020, when in particular we have understood the limitations of the current lower bound techniques and of the Talwar's scheme.

References

- [1] Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. In Jon M. Kleinberg, editor, *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 271–286. ACM, 2006. Full text at <https://www.cs.bgu.ac.il/~neimano/j37.pdf>. URL: <https://doi.org/10.1145/1132516.1132557>, doi:10.1145/1132516.1132557.
- [2] Ittai Abraham, Arnold Filtser, Anupam Gupta, and Ofer Neiman. Metric embedding via shortest path decompositions. *SIAM Journal on Computing*, 51(2):290–314, 2022. URL: <https://doi.org/10.1137/19m1296021>, doi:10.1137/19m1296021.
- [3] Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. *SIAM Journal on Computing*, 48(3):1120–1145, January 2019. URL: <https://doi.org/10.1137/17m1112406>, doi:10.1137/17m1112406.
- [4] Noga Alon, Richard M. Karp, David Peleg, and Douglas West. A graph-theoretic game and its application to the k -server problem. *SIAM Journal on Computing*, 24(1):78–100, 1995. doi:10.1137/s0097539792224474.
- [5] Sanjeev Arora. Nearly linear time approximation schemes for euclidean TSP and other geometric problems. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 554–563. IEEE Computer Society, 1997. URL: <https://doi.org/10.1109/SFCS.1997.646145>, doi:10.1109/SFCS.1997.646145.
- [6] Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for euclidean k -medians and related problems. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 106–113. ACM, 1998. URL: <https://doi.org/10.1145/276698.276718>, doi:10.1145/276698.276718.
- [7] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193, 1996. URL: <https://doi.org/10.1109/SFCS.1996.548477>.
- [8] Yair Bartal. On approximating arbitrary metrics by tree metrics. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 161–168, 1998. doi:10.1145/276698.276725.
- [9] G. Borradaile, H. Le, and C. Wulff-Nilsen. Minor-free graphs have light spanners. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science, FOCS '17*, pages 767–778, 2017. doi:10.1109/FOCS.2017.76.
- [10] Douglas E. Carroll and Ashish Goel. Lower bounds for embedding into distributions over excluded minor graph families. In *Algorithms - ESA 2004, 12th Annual European Symposium, Bergen, Norway, September 14-17, 2004, Proceedings*, pages 146–156, 2004. URL: https://doi.org/10.1007/978-3-540-30140-0_15, doi:10.1007/978-3-540-30140-0_15.
- [11] Amit Chakrabarti, Alexander Jaffe, James R. Lee, and Justin Vincent. Embeddings of topological graphs: Lossy invariants, linearization, and 2-sums. In *The Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS' 08.*, 2008. URL: <https://doi.org/10.1109/focs.2008.79>, doi:10.1109/focs.2008.79.

- [12] Vincent Cohen-Addad. Approximation schemes for capacitated clustering in doubling metrics. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2241–2259. SIAM, 2020. URL: <https://doi.org/10.1137/1.9781611975994.138>, doi:10.1137/1.9781611975994.138.
- [13] Vincent Cohen-Addad. Bypassing the surface embedding: approximation schemes for network design in minor-free graphs. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 343–356. ACM, 2022. URL: <https://doi.org/10.1145/3519935.3520049>, doi:10.1145/3519935.3520049.
- [14] Vincent Cohen-Addad, Andreas Emil Feldmann, and David Saulpic. Near-linear time approximation schemes for clustering in doubling metrics. *J. ACM*, 68(6):44:1–44:34, 2021. URL: <https://doi.org/10.1145/3477541>, doi:10.1145/3477541.
- [15] Vincent Cohen-Addad, Arnold Filtser, Philip N. Klein, and Hung Le. On light spanners, low-treewidth embeddings and efficient traversing in minor-free graphs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 589–600. IEEE, 2020. URL: <https://doi.org/10.1109/FOCS46700.2020.00061>, doi:10.1109/FOCS46700.2020.00061.
- [16] Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k -means and k -median in euclidean and minor-free metrics. *SIAM J. Comput.*, 48(2):644–667, 2019. URL: <https://doi.org/10.1137/17M112717X>, doi:10.1137/17M112717X.
- [17] Vincent Cohen-Addad, Marcin Pilipczuk, and Michal Pilipczuk. Efficient approximation schemes for uniform-cost clustering problems in planar graphs. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPIcs*, pages 33:1–33:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. URL: <https://doi.org/10.4230/LIPIcs.ESA.2019.33>, doi:10.4230/LIPIcs.ESA.2019.33.
- [18] Vincent Cohen-Addad, Michal Pilipczuk, and Marcin Pilipczuk. A polynomial-time approximation scheme for facility location on planar graphs. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 560–581. IEEE Computer Society, 2019. URL: <https://doi.org/10.1109/FOCS.2019.00042>, doi:10.1109/FOCS.2019.00042.
- [19] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. URL: <https://doi.org/10.1007/978-3-319-21275-3>, doi:10.1007/978-3-319-21275-3.
- [20] Aparna Das and Claire Mathieu. A quasi-polynomial time approximation scheme for euclidean capacitated vehicle routing. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*, pages 390–403. SIAM, 2010. URL: <https://doi.org/10.1137/1.9781611973075.33>, doi:10.1137/1.9781611973075.33.
- [21] Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Fixed-parameter algorithms for (k, r) -center in planar graphs and map graphs. *ACM Trans. Algorithms*, 1(1):33–47, 2005. URL: <https://doi.org/10.1145/1077464.1077468>, doi:10.1145/1077464.1077468.

- [22] 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. URL: <https://doi.org/10.1109/SFCS.2005.14>, doi:10.1109/SFCS.2005.14.
- [23] Reinhard Diestel, Ken-ichi Kawarabayashi, Theodor Müller, and Paul Wollan. On the excluded minor structure theorem for graphs of large tree-width. *J. Comb. Theory, Ser. B*, 102(6):1189–1210, 2012. URL: <https://doi.org/10.1016/j.jctb.2012.07.001>, doi:10.1016/j.jctb.2012.07.001.
- [24] David Eisenstat, Philip N. Klein, and Claire Mathieu. Approximating k -center in planar graphs. In *Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA’ 05*. Society for Industrial and Applied Mathematics, 2004. URL: <https://doi.org/10.1137/1.9781611973402.47>, doi:10.1137/1.9781611973402.47.
- [25] David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000. URL: <https://doi.org/10.1007/s004530010020>, doi:10.1007/s004530010020.
- [26] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004. URL: <https://doi.org/10.1016/j.jcss.2004.04.011>, doi:10.1016/j.jcss.2004.04.011.
- [27] Uriel Feige, MohammadTaghi Hajiaghayi, and James R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM J. Comput.*, 38(2):629–657, 2008. URL: <https://doi.org/10.1137/05064299X>, doi:10.1137/05064299X.
- [28] Arnold Filtser. A face cover perspective to ℓ_1 embeddings of planar graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1945–1954. 2020. URL: <https://doi.org/10.1137/1.9781611975994.120>, doi:10.1137/1.9781611975994.120.
- [29] Arnold Filtser and Hung Le. Low treewidth embeddings of planar and minor-free metrics. In *Proceedings of the 63rd Annual Symposium on Foundations of Computer Science, FOCS’ 22*, 2022. URL: <https://doi.org/10.1109/focs54457.2022.00105>, doi:10.1109/focs54457.2022.00105.
- [30] Fedor V. Fomin, Petr A. Golovach, and Dimitrios M. Thilikos. Contraction obstructions for treewidth. *J. Comb. Theory, Ser. B*, 101(5):302–314, 2011. URL: <https://doi.org/10.1016/j.jctb.2011.02.008>, doi:10.1016/j.jctb.2011.02.008.
- [31] E. Fox-Epstein, P. N. Klein, and A. Schild. Embedding planar graphs into low-treewidth graphs with applications to efficient approximation schemes for metric problems. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA’ 19*, page 1069–1088, 2019. doi:10.1137/1.9781611975482.66.
- [32] Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a PTAS for k -means in doubling metrics. *SIAM J. Comput.*, 48(2):452–480, 2019. URL: <https://doi.org/10.1137/17M1127181>, doi:10.1137/17M1127181.
- [33] Michelangelo Grigni. Approximate TSP in graphs with forbidden minors. In *Automata, Languages and Programming, 27th International Colloquium, ICALP 2000, Geneva, Switzerland, July 9-15, 2000, Proceedings*, pages 869–877, 2000. URL: https://doi.org/10.1007/3-540-45022-X_73, doi:10.1007/3-540-45022-X_73.

- [34] Anupam Gupta, Ilan Newman, Yuri Rabinovich, and Alistair Sinclair. Cuts, trees and ℓ_1 -embeddings of graphs. *Combinatorica*, 24(2):233–269, 2004. URL: <https://doi.org/10.1007/s00493-004-0015-x>, doi:10.1007/s00493-004-0015-x.
- [35] M. Haimovich and A. H. G. Rinnooy Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542, 1985. URL: <https://doi.org/10.1287/moor.10.4.527>, doi:10.1287/moor.10.4.527.
- [36] Aditya Jayaprakash and Mohammad R. Salavatipour. Approximation schemes for capacitated vehicle routing on graphs of bounded treewidth, bounded doubling, or highway dimension. *ACM Trans. Algorithms*, 19(2), mar 2023. URL: <https://doi.org/10.1145/3582500>, doi:10.1145/3582500.
- [37] Aditya Jayaprakash and Mohammad R. Salavatipour. Approximation schemes for capacitated vehicle routing on graphs of bounded treewidth, bounded doubling, or highway dimension. *ACM Transactions on Algorithms*, 19(2):1–36, 2023. URL: <https://doi.org/10.1145/3582500>, doi:10.1145/3582500.
- [38] 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. URL: <https://doi.org/10.1145/1993636.1993697>, doi:10.1145/1993636.1993697.
- [39] Philip Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proceedings of the 25th annual ACM symposium on Theory of Computing, STOC’93*, 1993. URL: <https://doi.org/10.1145/167088.167261>, doi:10.1145/167088.167261.
- [40] Stavros G. Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the euclidean k-median problem. *SIAM Journal on Computing*, 37(3):757–782, 2007. URL: <https://doi.org/10.1137/S0097539702404055>, arXiv:<https://doi.org/10.1137/S0097539702404055>, doi:10.1137/S0097539702404055.
- [41] Robert Krauthgamer, James R. Lee, Manor Mendel, and Assaf Naor. Measured descent: a new embedding method for finite metrics. *Geometric and Functional Analysis*, 15(4):839–858, 2005. URL: <http://dx.doi.org/10.1007/s00039-005-0527-6>, doi:10.1007/s00039-005-0527-6.
- [42] Robert Krauthgamer, James R. Lee, and Havana (Inbal) Rika. Flow-cut gaps and face covers in planar graphs. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 525–534. 2019. URL: <https://doi.org/10.1137/1.9781611975482.33>, doi:10.1137/1.9781611975482.33.
- [43] Nikhil Kumar. An approximate generalization of the okamura-seymour theorem. In *Proceedings of the 63rd Annual Symposium on Foundations of Computer Science, FOCS’22*, October 2022. URL: <https://doi.org/10.1109/focs54457.2022.00106>, doi:10.1109/focs54457.2022.00106.
- [44] Hung Le. A PTAS for subset TSP in minor-free graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2279–2298, 2020. Full version: arxiv:1804.01588. URL: <https://doi.org/10.1137/1.9781611975994.140>, doi:10.1137/1.9781611975994.140.

- [45] James R. Lee and Anastasios Sidiropoulos. On the geometry of graphs with a forbidden minor. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC'09*, 2009. URL: <https://doi.org/10.1145/1536414.1536450>, doi:10.1145/1536414.1536450.
- [46] James R. Lee and Anastasios Sidiropoulos. Pathwidth, trees, and random embeddings. *Combinatorica*, 33(3):349–374, 2013. URL: <https://doi.org/10.1007/s00493-013-2685-8>, doi:10.1007/s00493-013-2685-8.
- [47] Jaroslav Nešetřil and Patrice Ossona de Mendez. *Sparsity — Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012.
- [48] Ilan Newman and Yuri Rabinovich. A lower bound on the distortion of embedding planar metrics into euclidean space. *Discrete and Computational Geometry*, 29(1):77–81, 2002. URL: <https://doi.org/10.1007/s00454-002-2813-5>, doi:10.1007/s00454-002-2813-5.
- [49] Satish Rao. Small distortion and volume preserving embeddings for planar and euclidean metrics. In *Proceedings of the 15th Annual Symposium on Computational Geometry*, 1999. URL: <https://doi.org/10.1145/304893.304983>, doi:10.1145/304893.304983.
- [50] Neil Robertson and Paul D. Seymour. Graph Minors XVI. Excluding a non-planar graph. *J. Comb. Theory, Ser. B*, 89(1):43–76, 2003.
- [51] Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 281–290, 2004. URL: <https://doi.org/10.1145/1007352.1007399>, doi:10.1145/1007352.1007399.

A Omitted proofs of tools from probability theory

A.1 Proof of Lemma 3.2

For every X_i , define a random variable $X'_i \in \{1, -1\}$ such that $\mathbb{P}(X'_i = 1) = \frac{1}{8}$, $X'_i \geq X_i$ almost surely, and X'_1, X'_2, \dots are independent. Then, for every i , $\sum_{j=1}^i X_j \leq \sum_{j=1}^i X'_j$ so, for a variable Z' defined analogously for X'_1, X'_2, \dots , we have $Z' \geq Z$ almost surely. Thus, it suffices to prove the lemma for the sequence X'_1, X'_2, \dots and the variable Z' . By slightly abusing the notation, we rename X'_i to X_i and Z' to Z , that is, we assume $\mathbb{P}(X_i = 1) = \frac{1}{8}$ for every $i \geq 1$.

Let $Y = \sum_{j=1}^{2k} X_j$. Then the condition $k + Y \geq 0$ is equivalent to the following condition: for at least $\frac{k}{2}$ indices $i \in \{1, \dots, 2k\}$ we have $X_i = 1$. Since the expected number of such indices is $\frac{k}{4}$, by Theorem 3.1 applied for $\delta = 1$, we have

$$\mathbb{P}(k + Y \geq 0) \leq \exp\left(-\frac{k}{12}\right).$$

As the event “ $k + Y \geq 0$ ” contains the event “ $Z > 2k$ ”, the proof is finished.

A.2 Proof of Lemma 3.4

Fix measurable sets $S_1, S_2, \dots, S_n \subseteq \mathbb{R}_{\geq 0}$ and for $1 \leq j \leq n$, let p_j be the probability that a random variable with distribution $\text{Exp}(1)$ has its value in S_j . To prove the lemma, it suffices to show that

$$\mathbb{P}(Y_1 \in S_1 \wedge \dots \wedge Y_n \in S_n) = \prod_{j=1}^n p_j. \quad (6)$$

We will need the following computation. Fix $1 \leq a \leq 2n$, $1 \leq j \leq n$, and $x_1, x_2, \dots, x_{a-1} \in \mathbb{R}_{\geq 0}$.

$$\begin{aligned}
& \sum_{b=a+1}^{2n} \int_0^{f_{a+1}(x_1, \dots, x_a)} e^{-x_{a+1}} dx_{a+1} \dots \int_0^{f_{b-1}(x_1, \dots, x_{b-2})} e^{-x_{b-1}} dx_{b-1} \\
& \quad \int_{f_b(x_1, \dots, x_{b-1})}^{+\infty} e^{-x_b} [x_b - f_b(x_1, \dots, x_{b-1}) \in S_j] dx_b \\
&= \sum_{b=a+1}^{2n} \int_0^{f_{a+1}(x_1, \dots, x_a)} e^{-x_{a+1}} dx_{a+1} \dots \int_0^{f_{b-1}(x_1, \dots, x_{b-2})} e^{-x_{b-1}} dx_{b-1} \\
& \quad e^{-f_b(x_1, \dots, x_{b-1})} \int_0^{+\infty} e^{-y} [y \in S_j] dy \\
&= \sum_{b=a+1}^{2n} \int_0^{f_{a+1}(x_1, \dots, x_a)} e^{-x_{a+1}} dx_{a+1} \dots \int_0^{f_{b-1}(x_1, \dots, x_{b-2})} e^{-x_{b-1}} dx_{b-1} \\
& \quad e^{-f_b(x_1, \dots, x_{b-1})} p_j \\
&= p_j \left(e^{-f_{a+1}(x_1, \dots, x_a)} + \int_0^{f_{a+1}(x_1, \dots, x_a)} e^{-x_{a+1}} dx_{a+1} \right. \\
& \quad \left(e^{-f_{a+2}(x_1, \dots, x_{a+1})} + \int_0^{f_{a+2}(x_1, \dots, x_{a+1})} e^{-x_{a+2}} dx_{a+2} \right. \\
& \quad \dots \\
& \quad \left. \left(e^{-f_{2n-1}(x_1, \dots, x_{2n-2})} + \int_0^{f_{2n-1}(x_1, \dots, x_{2n-2})} e^{-x_{2n-1}} dx_{2n-1} \right. \right. \\
& \quad \left. \left. e^{-f_{2n}(x_1, \dots, x_{2n-1})} \right) \dots \right) \\
&= p_j \left(e^{-f_{a+1}(x_1, \dots, x_a)} + \int_0^{f_{a+1}(x_1, \dots, x_a)} e^{-x_{a+1}} dx_{a+1} \right. \\
& \quad \left(e^{-f_{a+2}(x_1, \dots, x_{a+1})} + \int_0^{f_{a+2}(x_1, \dots, x_{a+1})} e^{-x_{a+2}} dx_{a+2} \right. \\
& \quad \dots \\
& \quad \left. \left(e^{-f_{2n-1}(x_1, \dots, x_{2n-2})} + \int_0^{f_{2n-1}(x_1, \dots, x_{2n-2})} e^{-x_{2n-1}} dx_{2n-1} \right) \dots \right) \\
&= p_j \left(e^{-f_{a+1}(x_1, \dots, x_a)} + \int_0^{f_{a+1}(x_1, \dots, x_a)} e^{-x_{a+1}} dx_{a+1} \right. \\
& \quad \left(e^{-f_{a+2}(x_1, \dots, x_{a+1})} + \int_0^{f_{a+2}(x_1, \dots, x_{a+1})} e^{-x_{a+2}} dx_{a+2} \right. \\
& \quad \dots \\
& \quad \left. \left(e^{-f_{2n-2}(x_1, \dots, x_{2n-3})} + \int_0^{f_{2n-2}(x_1, \dots, x_{2n-3})} e^{-x_{2n-2}} dx_{2n-2} \right) \dots \right) \\
&= \dots = p_j.
\end{aligned} \tag{7}$$

Here, we first used that $f_{2n} \equiv 0$ so

$$e^{-f_{2n}(x_1, \dots, x_{2n-1})} = 1.$$

Then, for $i = 2n-1, 2n-2, \dots$ we used

$$e^{-f_i(x_1, \dots, x_{i-1})} + \int_0^{f_i(x_1, \dots, x_{i-1})} e^{-x_i} dx_i = e^{-f_i(x_1, \dots, x_{i-1})} + 1 - e^{-f_i(x_1, \dots, x_{i-1})} = 1.$$

We expand the left hand side of (6), conditioning on the values of I_1, \dots, I_n , and then use repeatedly (7) for $a = i_{n-1}, i_{n-2}, \dots, i_1, 1$.

$$\begin{aligned}
\mathbb{P}(\forall_{j=1}^n Y_j \in S_j) &= \sum_{1 \leq i_1 < i_2 < \dots < i_n \leq 2n} \mathbb{P}(\forall_{j=1}^n I_j = i_j \wedge X_{i_j} - f_{i_j}(X_1, \dots) \in S_j) \\
&= \sum_{i_1=1}^{2n} \int_0^{f_1} e^{-x_1} dx_1 \int_0^{f_2(x_1)} e^{-x_2} dx_2 \dots \int_0^{f_{i_1-1}(x_1, \dots, x_{i_1-2})} e^{-x_{i_1-1}} dx_{i_1-1} \\
&\quad \int_{f_{i_1}(x_1, \dots, x_{i_1-1})}^{+\infty} e^{-x_{i_1}} [x_{i_1} - f_{i_1}(x_1, \dots, x_{i_1-1}) \in S_1] dx_{i_1} \\
&\quad \sum_{i_2=i_1+1}^{2n} \int_0^{f_{i_1+1}(x_1, \dots, x_{i_1})} e^{-x_{i_1+1}} dx_{i_1+1} \dots \int_0^{f_{i_2-1}(x_1, \dots, x_{i_2-2})} e^{-x_{i_2-1}} dx_{i_2-1} \\
&\quad \int_{f_{i_2}(x_1, \dots, x_{i_2-1})}^{+\infty} e^{-x_{i_2}} [x_{i_2} - f_{i_2}(x_1, \dots, x_{i_2-1}) \in S_2] dx_{i_2} \\
&\quad \dots \\
&\quad \sum_{i_n=i_{n-1}+1}^{2n} \int_0^{f_{i_{n-1}+1}(x_1, \dots, x_{i_{n-1}})} e^{-x_{i_{n-1}+1}} dx_{i_{n-1}+1} \dots \int_0^{f_{i_n-1}(x_1, \dots, x_{i_n-2})} e^{-x_{i_n-1}} dx_{i_n-1} \\
&\quad \int_{f_{i_n}(x_1, \dots, x_{i_n-1})}^{+\infty} e^{-x_{i_n}} [x_{i_n} - f_{i_n}(x_1, \dots, x_{i_n-1}) \in S_n] dx_{i_n} \\
&= p_n \sum_{i_1=1}^{2n} \int_0^{f_1} e^{-x_1} dx_1 \int_0^{f_2(x_1)} e^{-x_2} dx_2 \dots \int_0^{f_{i_1-1}(x_1, \dots, x_{i_1-2})} e^{-x_{i_1-1}} dx_{i_1-1} \\
&\quad \int_{f_{i_1}(x_1, \dots, x_{i_1-1})}^{+\infty} e^{-x_{i_1}} [x_{i_1} - f_{i_1}(x_1, \dots, x_{i_1-1}) \in S_1] dx_{i_1} \\
&\quad \sum_{i_2=i_1+1}^{2n} \int_0^{f_{i_1+1}(x_1, \dots, x_{i_1})} e^{-x_{i_1+1}} dx_{i_1+1} \dots \int_0^{f_{i_2-1}(x_1, \dots, x_{i_2-2})} e^{-x_{i_2-1}} dx_{i_2-1} \\
&\quad \int_{f_{i_2}(x_1, \dots, x_{i_2-1})}^{+\infty} e^{-x_{i_2}} [x_{i_2} - f_{i_2}(x_1, \dots, x_{i_2-1}) \in S_2] dx_{i_2} \\
&\quad \dots \\
&\quad \sum_{i_{n-1}=i_{n-2}+1}^{2n} \int_0^{f_{i_{n-2}+1}(x_1, \dots, x_{i_{n-2}})} e^{-x_{i_{n-2}+1}} dx_{i_{n-2}+1} \dots \int_0^{f_{i_{n-1}-1}(x_1, \dots, x_{i_{n-1}-2})} e^{-x_{i_{n-1}-1}} dx_{i_{n-1}-1} \\
&\quad \int_{f_{i_{n-1}}(x_1, \dots, x_{i_{n-1}-1})}^{+\infty} e^{-x_{i_{n-1}}} [x_{i_{n-1}} - f_{i_{n-1}}(x_1, \dots, x_{i_{n-1}-1}) \in S_{n-1}] dx_{i_{n-1}} \\
&= \dots = \prod_{j=1}^n p_j.
\end{aligned}$$

This finishes the proof.

B Omitted proofs on graphs with an excluded minor

B.1 Proof of Theorem 6.1

In this section we sketch how to derive Theorem 6.1 from known results. We assume that the reader is familiar with the notions of vortices and of near-embeddings and the structure theorem of Robertson and Seymour [50]. In what follows, we will use the notation of Diestel et al. [23].

The crux lies in the following observation.

Lemma B.1. *For every surface Σ and constants a, b there exists an apex graph K such that every graph nearly-embeddable on Σ with at most a vortices of depth at most b each (and with no apices) does not contain the graph K as a minor.*

Proof. For an integer t , let W_t be the $t \times t$ grid and let \widehat{W}_t be the graph W_t with an additional vertex added and made adjacent to all vertices of the grid. We will prove that for some t , depending on Σ , a , and b only, \widehat{W}_t cannot be a minor of a graph G that is nearly-embedded as in the lemma statement.

Let G be a graph with a near embedding as in the lemma statement and assume there is a minor model of \widehat{W}_t in G for large t . We focus on the grid W_t part of \widehat{W}_t for a moment. It follows from [23, Lemma 22] that there is a large subgrid $W_{t'}$ of W_t whose minor model is disjoint with vortices in the following sense: for every vertex of the subgrid $W_{t'}$, its branch set in the minor model of $W_{t'}$ is a subgraph of its branch set in the minor model of W_t and is vertex-disjoint with vortices of the near-embedding. By possibly enlarging the branch set of the apex vertex of \widehat{W}_t , we obtain a minor model of $\widehat{W}_{t'}$ in G such that the branch sets of the grid part are vertex-disjoint with vortices.

If we now contract every vortex of G into a single vertex, we obtain a graph embedded in Σ (in the classic sense). The minor model of $\widehat{W}_{t'}$ in G projects to a minor model of $\widehat{W}_{t'}$ in the obtained graph. However, as the Euler genus of \widehat{W}_s grows with s to infinity, this implies that t' is bounded by a function of Σ , which in turn implies that t is bounded as a function of Σ , a , and b . This finishes the proof. \square

Theorem 6.1 now follows from the structure theorem of Robertson and Seymour [50] (with its algorithmic versions of [22, 38]) where in every bag we designate the apices of the near-embedding as the set apices(t).

B.2 Proof of Lemma 6.2

Let \mathcal{G} be the class of K -minor-free graphs. Let \mathcal{G}' be the class of graphs that can be constructed from a member $G \in \mathcal{G}$ by adding, for some cliques in G , a new vertex adjacent to the clique. We observe that \mathcal{G}' is contraction-closed. Hence, the class \mathcal{G}'' defined as all subgraphs of graphs in \mathcal{G}' is minor-closed.

Recall that a graph class \mathcal{H} has *locally bounded treewidth* if there exists a function f such that for every $H \in \mathcal{H}$, $v \in V(H)$, and integer $r \geq 0$, the treewidth of the subgraph induced by the radius- r ball around v in H is bounded by $f(r)$. Eppstein [25] showed that for minor-closed graph classes, the property of having locally bounded treewidth is equivalent to excluding some apex graph as a minor.

Hence, \mathcal{G} has locally bounded treewidth, say with a function f . Then, \mathcal{G}' has locally bounded treewidth with the function $f'(r) := f(r) + 1$. Note that for every $G' \in \mathcal{G}'$, $r \geq 0$, subgraph G'' of G' , and a vertex $v \in V(G'')$, the ball of radius r centered at v in G'' is a subgraph of the ball of radius r centered at v in G' . Hence, \mathcal{G}'' has locally bounded treewidth with the same function f' . Consequently, as \mathcal{G}'' is minor-closed, it excludes some apex graph by [25]. This concludes the proof.