Diameter, eccentricities and distance oracle computations on H-minor free graphs and graphs of bounded (distance) VC-dimension *

Guillaume Ducoffe[1], Michel Habib[2], and Laurent Viennot[3]

[1]University of Bucharest, Faculty of Mathematics and Computer Science, and National Institute for Research and Development in Informatics, Romania
[2]Paris University, and IRIF CNRS, France
[3]Inria, and Paris University, France

Abstract

Under the Strong Exponential-Time Hypothesis, the diameter of general unweighted graphs cannot be computed in truly subquadratic time (in the size $n + m$ of the input), as shown by Roditty and Williams. Nevertheless there are several graph classes for which this can be done such as bounded-treewidth graphs, interval graphs and planar graphs, to name a few. We propose to study unweighted graphs of constant *distance VC-dimension* as a broad generalization of many such classes – where the *distance VC-dimension* of a graph G is defined as the VC-dimension of its ball hypergraph: whose hyperedges are the balls of all possible radii and centers in G. In particular for any fixed H, the class of H-minor free graphs has *distance VC-dimension* at most $|V(H)| - 1$.

- Our first main result is a Monte Carlo algorithm that on graphs of *distance VC-dimension* at most d, for any fixed k, either computes the diameter or concludes that it is larger than k in time $\tilde{\mathcal{O}}(k \cdot mn^{1-\varepsilon_d})$, where $\varepsilon_d \in (0; 1)$ only depends on d [1]. We thus obtain a *truly subquadratic-time parameterized* algorithm for computing the diameter on such graphs.
- Then as a byproduct of our approach, we get a truly subquadratic-time randomized algorithm for *constant* diameter computation on all the *nowhere dense* graph classes. The latter classes include all proper minor-closed graph classes, bounded-degree graphs and graphs of bounded expansion. Before our work, the only known such algorithm was resulting from an application of Courcelle's theorem, see Grohe et al. [47].
- For any graph of constant *distance VC-dimension*, we further prove the existence of an exact distance oracle in truly subquadratic space, that answers distance queries in truly sublinear time (in the number n of vertices). The latter generalizes prior results on proper minor-closed graph classes to a much larger graph class.
- Finally, we show how to remove the dependency on k for *any* graph class that excludes a fixed graph H as a minor. More generally, our techniques apply to any graph with constant

*Results of this paper were partially presented at the SODA'20 conference [33].
[1]The $\tilde{\mathcal{O}}()$ notation suppresses polylogarithmic factors.

distance VC-dimension and *polynomial expansion* (or equivalently having strongly sublinear balanced separators). As a result for all such graphs one obtains a truly subquadratic-time deterministic algorithm for computing all the eccentricities, and thus both the diameter and the radius. Our approach can be generalized to the H-minor free graphs with bounded positive integer weights.

We note that all our algorithms for the diameter problem can be adapted for computing the *radius*, and more generally all the eccentricities. Our approach is based on the work of Chazelle and Welzl who proved the existence of spanning paths with strongly sublinear *stabbing number* for every hypergraph of constant VC-dimension. We show how to compute such paths efficiently by combining known algorithms for the stabbing number problem with a clever use of ε-nets, region decomposition and other partition techniques.

1 Introduction

In this paper we present new results on exact diameter computation within several classes of unweighted (undirected) graphs with a geometric flavor. We recall that the diameter of an unweighted graph is the maximum number of edges on a shortest path. Beyond its many practical applications, this fundamental problem in Graph Theory has attracted a lot of attention in the fine-grained complexity study of polynomial-time solvable problems [1, 4, 9, 19, 22, 28, 31, 40, 67]. More precisely, for every n-vertex m-edge unweighted graph the textbook algorithm for computing its diameter runs in time $\mathcal{O}(nm)$. In a seminal paper [67] this roughly quadratic running-time (in the size $n+m$ of the input) was matched by a quadratic *lower-bound*, assuming the Strong Exponential-Time Hypothesis (SETH). We stress that for graphs with millions of nodes and edges, quadratic time is already prohibitive.

The conditional lower-bound of [67] also holds for sparse graphs *i.e.*, with only $m = \mathcal{O}(n)$ edges [1]. However it does *not* hold for many well-structured graph classes [1, 12, 15, 17, 26, 27, 29, 31, 38, 41, 45, 63]. Our work proposes some new advances on the characterization of graph families for which we can compute the diameter in truly subquadratic time.

1.1 Related work

Before we detail our contributions, we wish to mention a few recent (and not so recent) results that are most related to our approach.

Interval graphs. An early example of linear-time solvable special case for diameter computation is the class of interval graphs [63]. For every interval graph G and for any integer k, if we first compute an interval representation for G in linear-time [48] then we can compute by dynamic programming, for every vertex v, the contiguous segment of all the vertices at a distance $\leq k$ from v in G. It takes almost linear-time and it implies a straightforward quasi linear-time algorithm for diameter computation. More efficient algorithms for diameter computation on interval graphs and related graph classes were proposed in [26]. Nevertheless we will show in what follows that interval orderings are a powerful tool for diameter computation on more general geometric graph classes.

Bounded-treewidth graphs. More recently, quasi linear-time algorithms for diameter computation on bounded-treewidth graphs were presented in [1, 14, 15] with almost optimal dependency on the treewidth parameter. The cornerstone of these algorithms is the use of k-range trees in order to detect the furthest pairs that are disconnected by some small-cardinality separators. This technique was first introduced in [18] in order to compute the sum of all distances on bounded-treewidth graphs. Since then a few other applications of k-range trees and, more generally, orthogonal range searching for diameter computation, have been presented in [31, 32]. In our work we uncover deeper connections between diameter computation and range searching techniques from computational geometry.

Planar graphs. Finally, in a recent breakthrough paper [17], Cabello presented the first truly subquadratic algorithm for diameter computation on planar graphs (see also [44, 45] for improvements on his work). For that he combined r-divisions: a recursive decomposition technique for planar graphs and other hereditary graph classes with sublinear balanced separators, with a clever use of additively weighted Voronoi diagrams. Cabello conjectured that his algorithm could be generalized to bounded-genus graphs. The long version [45] indicates that their techniques could allow such a generalization if computing the diameter of a graph embedded onto a surface of genus g reduces to the planar case with $O(g)$ holes in the regions of some r-division. Although it is known that such a graph can be decomposed into planar subgraphs by removing $2g$ shortest paths [53, 39], such reduction is not clear, and we could not find references formally supporting this. More recently, Li and Parter proposed a distributed algorithm for planar diameter which is based on metric compression [57] and uses a VC-dimension argument to bound the number of distance profiles with respect to a given subset of nodes. Following the basics of planar graphs algorithms, we partly reuse r-divisions within our algorithms. However we replace the intricate use of Voronoi diagrams with a quite different approach that is based on some interval representations of the balls of a given radius in a graph. Our approach is also based on a VC-dimension argument but in a very different way than [57]. By doing so, we can obtain truly subquadratic-time algorithms for diameter computation on bounded genus graphs (and more generally, on any proper minor-closed graph family) while avoiding a great deal of topological complications. Note that, similarly to the Voronoi diagram method, our approach allows us to compute all eccentricities.

We stress that for the aforementioned graph classes, the techniques used for computing their diameter are quite different from each other. Our work is a first step toward unifying all these previous results for unweighted graphs in a single framework. Note that some of the aforementioned results also hold in the directed weighted case. Under some mild conditions (always satisfied by the proper minor-closed graph classes), so does our approach for the undirected graphs of bounded integer edge-weights.

1.2 Our contributions

We study the parameterization of graph diameter by the *VC-dimension* of various hypergraphs. More precisely, a set Y is *shattered* by a hypergraph $\mathcal{H}$ if by intersecting Y with all hyperedges of $\mathcal{H}$ one obtains the power-set of Y. The VC-dimension of $\mathcal{H}$ is then defined as the largest cardinality of a subset shattered by $\mathcal{H}$. This powerful notion was first introduced by Vapnik and Chervonenkis in [70]. Since then it has found applications in sampling complexity and machine learning, among other domains. We refer to [56] for early work on VC-dimension in graphs. In particular, the VC-dimension of a graph G is defined as the VC-dimension of its closed neighbourhood hypergraph: whose hyperedges are the closed neighbourhoods of vertices in G. Graphs of bounded interval number and proper minor-closed graph classes are two examples of graph families with a *constant* upper-bound on their VC-dimension [56, 23, 32].

First example. As an appetizer we first consider an n-vertex split graph with clique-number $\log^{\mathcal{O}(1)} n$, that is a notouriously hard case for diameter computation [9]. Given such a split graph G with stable set S and maximal clique K, we can pre-process G in linear-time so as to partition the vertices of S into *twin classes*: with two vertices in S being called twins if and only if they have the same neighbourhood in K (*e.g.*, see [27]). If the VC-dimension of G is at most d then, by the Sauer-Shelah-Perles Lemma [68, 69] the number of twin classes is in $\mathcal{O}(|K|^d) = \log^{\mathcal{O}(d)} n$. Therefore, after some linear-time preprocessing, we are left with computing the diameter on a graph of *polylogarithmic order.* Unfortunately, such simple brute-force arguments are no longer sufficient for split graphs of arbitrary clique-size.

Overview of our techniques. In order to generalize our approach to any graph of constant VC-dimension, we use the central notion of *spanning paths with low stabbing number.* Chazelle and Welzl [21] defined a spanning path for a hypergraph $\mathcal{H}$ as a total ordering of its vertex-set. The *stabbing number* of such a path is, up to 1, the maximum number of maximal intervals of which a hyperedge in $\mathcal{H}$ can be the union (we refer to Sec. 2 for a formal definition).

Assume for now that we are given a spanning path with stabbing number t for the closed neighbourhood hypergraph of G. Then in linear time, we can compute for every vertex v the ends of the $\mathcal{O}(t)$ intervals of which $N_G[v]$ is the union. We denote this set of intervals by $I(v)$ in what follows. Then, in order to decide whether G has diameter at most two, it is sufficient to check whether for every vertex u we have $\bigcup_{v \in N_G[u]} I(v) = V(G)$. Since we only need to consider the extremities of such intervals, this verification phase takes time $\tilde{\mathcal{O}}(deg_G(u) \cdot t)$ for a vertex of degree $deg_G(u)$, and so, $\tilde{\mathcal{O}}(tm)$ total time. Note that such running-time is always subquadratic if t is sublinear in n. Overall, we reduced the diameter-two problem to the computation of a spanning path with low stabbing number for the closed neighbourhood hypergraph.

Motivated by range searching problems, Chazelle and Welzl proved the existence of spanning paths with *strongly sublinear* stabbing number for every hypergraph of constant VC-dimension [21]! Following this approach, we obtain our first main result in this paper:

Theorem 1. *For every $d > 0$, there exists a constant $\varepsilon_d \in (0;1)$ such that in deterministic time $\tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$ we can decide whether a graph of VC-dimension at most d has diameter two.*

We stress that in contrast to Theorem 1, under the Strong Exponential-Time we cannot decide whether a *general* graph has diameter at most two in truly subquadratic time [67]. Note also that the bound d on the VC-dimension is not needed as part of the input. This is further discussed at the end of this section, and also in Sec. 2.4.

On our way to prove Theorem 1 our main difficulty was to show how to compute for a hypergraph $\mathcal{H}$ a spanning path of low stabbing number. Computing a spanning path of minimum stabbing number is NP-hard [6]. However, there exist approximation algorithms for this problem that run in polynomial time [6, 49]. Their approximation ratio is logarithmic, that is fine for our applications. Unfortunately, the fastest known algorithms require us to solve a linear program. So far, the best known algorithms for this intermediate problem run in superquadratic time [24]. We show how to decrease the running-time of this part, at the price of a slightly increased stabbing number. For that, we carefully apply the deterministic algorithm resulting from [21] to some arbitrary partition of $\mathcal{H}$ in subhypergraphs of sublinear size. This feature might be of independent interest. We thus state the following theorem, where the size of a hypergraph is defined as the sum of its hyperedge cardinalities.

Theorem 2. *For every $d > 0$, there exists a constant $\varepsilon_d \in (0;1)$ such that in $\tilde{\mathcal{O}}(m + n^{2-\varepsilon_d})$ deterministic time, for every n-vertex hypergraph $\mathcal{H}$ of VC-dimension at most d and size m, we can compute a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$. In particular, this algorithm computes for each hyperedge the ends of its corresponding $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ intervals.*

Moreover, $\varepsilon_d = \frac{1}{2^{d+1}(c(d+1)-1)+1}$ for some universal constant $c > 2$.

From VC-dimension to *distance* VC-dimension. In order to go beyond Theorem 1, we need to consider a stronger notion of VC-dimension for graphs. The *distance VC-dimension*[2] of G is equal to the VC-dimension of its *ball hypergraph*: of which the hyperedges are all possible balls in G. Note that a bounded *distance VC-dimension* implies a bounded VC-dimension, but the converse a priori does not hold. Nevertheless, and perhaps surprisingly, there are still many classes of graphs with constant *distance VC-dimension*. These classes include, among others: interval graphs, planar graphs [23] and, more generally, any proper minor-closed graph family (from Remark 3 in [23]), as well as graphs of bounded rank-width [11].

Theorem 3. *There exists a Monte Carlo algorithm such that, for every positive integers d and k, we can decide whether a graph of distance VC-dimension at most d has diameter at most k. The running time is in $\tilde{\mathcal{O}}(k \cdot mn^{1-\varepsilon_d})$, where $\varepsilon_d \in (0;1)$ only depends on d.*

Eppstein proved in [38] that for any *constant* k, we can decide in linear time whether the diameter of a planar graph is at most k. Our result can be seen as a generalization of his to any graph class of constant *distance VC-dimension* – but at the price of a superlinear running-time. Furthermore, our techniques also apply to superconstant diameters, say polylogarithmic in n, or even polynomial in n provided the exponent is in $o(\varepsilon_d)$.

Our main technical contribution in this part is the efficient computation of spanning paths with strongly sublinear stabbing number for some *dense* hypergraphs of constant VC-dimension. More precisely, the *ℓ-neighbourhood hypergraph* of G has for hyperedges the balls of radius ℓ in G. For instance, the 1-neighbourhood hypergraph of G is exactly its closed neighbourhood hypergraph. In order to prove Theorem 3, we reduce the problem of deciding whether a graph has diameter at most k to the computation of a spanning path with low stabbing number for its $(k-1)$-neighbourhood hypergraph. In this sense, the proofs of Theorems 1 and 3 are very similar. However, an additional difficulty here is that we cannot have direct access to this $(k-1)$-neighbourhood hypergraph. Indeed, in the worst case all hyperedges of this hypergraph may have a cardinality in $\Omega(n)$, and then storing the hypergraph itself would already require quadratic space.

We overcome this issue by computing an *ε-net* [51, 70] in order to partition the vertices of the graph in a small number of groups, with every two vertices in the same group having almost the same ball of radius $k-1$. By selecting only one vertex per group, we so reduce the number of hyperedges (*i.e.*, balls of radius $k-1$) to be considered. Finally, once a spanning path was computed for this smaller hypergraph, for every unselected vertex we compute the symmetric difference between its ball of radius $k-1$ and the one of the unique vertex taken in its group. Our solution in order to do that efficiently is to first compute a spanning path with low stabbing number for the $(k-2)$-*neighbourhood hypergraph*. This is where the dependency on k occurs, as

[2] Our definition of *distance VC-dimension* is slightly weaker than the one proposed in [11].

overall we will need to compute a spanning path for $k-1$ consecutive hypergraphs. Our algorithm is randomized and succeeds with high probability. The use of randomization comes from the ε-net construction. Although deterministic algorithms do exist for that [16], it is not clear whether they can be used as efficiently as the simple sampling technique of the randomized algorithm. We leave open the question of finding a deterministic variant of Theorem 3.

We note that this above technique can be applied under slightly weaker hypothesis than the one we state in Theorem 3. For instance, Nešetřil and Ossona de Mendez proved that for all *nowhere dense* graph classes (i.e., a broad generalization of proper minor-closed graph classes and bounded-degree graphs), for any graph in the class and for any constant k, the VC-dimension of the k-neighbourhood hypergraph is constantly upper-bounded [62]. It allows us to derive the following version of our Theorem 3:

Theorem 4. *Let $\mathcal{G}$ be a class of nowhere dense graphs. There exists a Monte Carlo algorithm such that, for every constant $k = \mathcal{O}(1)$, for any graph in $\mathcal{G}$ we can decide whether its diameter is at most k in $\tilde{\mathcal{O}}(mn^{1-\varepsilon_{\mathcal{G}}(k)})$ time, for some constant $\varepsilon_{\mathcal{G}}(k) \in (0;1)$ that only depends on k.*

We observe that we can express the property of having diameter at most k as a first-order formula of length $\mathcal{O}(k)$. Therefore, it directly follows from [47] that for any class of nowhere dense graphs, there exists an $\mathcal{O}(f(k) \cdot n^{1+o(1)})$-time algorithm for deciding whether the diameter is at most k. However, the function f is (at least) a tower of exponential, since this algorithm results from an application of Courcelle's theorem. Furthermore, let us mention that under SETH, a truly subquadratic algorithm for *constant* diameter computation is the best result that we can hope for nowhere dense graph classes. Indeed, bounded-degree graphs are nowhere dense and, under SETH, we cannot compute their diameter in truly subquadratic time for diameter $\omega(\log n)$ [40].

Application: Distance oracles. The seminal work of Cabello for fast diameter computation within planar graphs also paved the way to the discovery of *exact* distance oracles for this class of graphs, which only require truly subquadratic space and answer distance queries in polylogarithmic time [25, 46]. We derive from our approach the following result:

Theorem 5. *Let $d > 0$ and let ε_d be as defined in Theorem 2. For any graph G of distance VC-dimension at most d, there exists an exact distance oracle in $\tilde{\mathcal{O}}(n^{2-\frac{\varepsilon_d}{2}})$ space, that answers distance queries in $\tilde{\mathcal{O}}(n^{1-\frac{\varepsilon_d}{2}})$ time. Moreover, there is a Monte Carlo algorithm for constructing such an oracle, in $\tilde{\mathcal{O}}(mn^{1-\frac{\varepsilon_d}{2}})$ randomized time. This oracle may fail in reporting a distance correctly with probability at most $1/n^{\mathcal{O}(1)}$.*

In comparison with our Theorem 5, all proper minor-closed graph classes have hub labels of size $\mathcal{O}(\sqrt{n}\log n)$, that follows from the existence of balanced separator of cardinality $\mathcal{O}(\sqrt{n})$ [43]. There also exist *approximate* distance oracles with a better trade-off than those we obtain with Theorem 5 [53]. However, our results apply to a much larger graph class than proper minor-closed graph classes, and interestingly they do not leverage on the existence of small balanced separators.

We conjecture that on every graph family of constant *distance VC-dimension*, we can compute the diameter in truly subquadratic time. Our next main result shows the conjecture to be true for any monotone graph family with strongly sublinear balanced separators, *a.k.a* the graphs of *polynomial expansion* [35].

Theorem 6. *Let $\mathcal{G}$ be a monotone graph class with strongly sublinear balanced separators. Then, for every $d > 0$, for any graph in $\mathcal{G}$ of distance VC-dimension at most d, we can compute all the eccentricities (and so, the diameter) in* deterministic *time $\tilde{\mathcal{O}}(n^{2-\varepsilon_{\mathcal{G}}(d)})$, for some constant $\varepsilon_{\mathcal{G}}(d) \in (0;1)$ that only depends on d.*

Let us recall that H-minor free graphs have a constant *distance VC-dimension* from Remark 3 in [23] (see also [11]), and that they all have strongly sublinear balanced separators [2, 54, 72]. Therefore, as an important consequence of Theorem 6, we get a truly subquadratic-time algorithm for computing all the eccentricities, on all the proper minor-closed graph classes.

It might be tempting, in the above Theorem 6, to drop the assumption that the *distance VC-dimension* must be bounded. Unfortunately, this cannot be done assuming SETH. Indeed, there is also an equivalence between the graphs of strongly sublinear treewidth and those monotone graph classes with strongly sublinear balanced separators [36]; however it follows from [1] that under SETH, we cannot compute the diameter in truly subquadratic time already for n-vertex graphs of treewidth $\omega(\log n)$. Conversely, not all graph classes with constant *distance VC-dimension* have strongly sublinear separators. This can be seen, *e.g.*, with interval graphs.

The speed-up of Theorem 6 follows from a faster computation of spanning paths for the neighbourhood hypergraphs. More precisely, our first main insight is that, in order to decide whether the diameter is at most k, we do not really need to compute a spanning path of low stabbing number for $\mathcal{N}_{k-1}(G)$. In fact, it is sufficient to compute a suboptimal representation for $\mathcal{N}_k(G)$ that minimizes what we call the *total* stabbing number (defined as the sum, over all vertices v, of the number of maximal intervals in the representation whose union is the ball of center v and radius k). Doing so, we avoid using ε-nets, which makes our algorithms fully deterministic. So the problem becomes how to compute efficiently these suboptimal representations?

For that, we use a rather classical divide-and-conquer approach. Frederickson [42] proved that a planar graph can be edge-covered with $\mathcal{O}(n/r)$ subgraphs of order at most r such that at most $\mathcal{O}(\sqrt{r})$ vertices of each subgraph can be contained in another subgraph of this decomposition. His construction directly follows from the planar separator theorem of Lipton and Tarjan [58], and as such it can be easily adapted for any monotone graph family with sublinear balanced separators [52][3]. For illustrating our method, we now focus in this introduction on the planar case. We can first compute, for some well-chosen $r = n^\gamma,\ \gamma \in (0;1)$, a decomposition as described above. For every two vertices in a same subgraph, we can check whether they are at distance at most k *in the subgraph*, by solving All-Pairs Shortest-Paths for the latter; this operation takes $\mathcal{O}(r^2)$ time per subgraph but, assuming r is small enough, this whole phase can be implemented in order to run in truly subquadratic time. Then for every subgraph of the decomposition, we compute a breadth-first search from each of the $\mathcal{O}(\sqrt{r})$ boundary vertices that are also contained in another subgraph. Overall, there can only be $\mathcal{O}(n/\sqrt{r})$ such boundary vertices, and so, it takes truly subquadratic time. Furthermore by doing so, we computed for every subgraph of the decomposition the $\mathcal{O}(r\sqrt{r})$ distances between the boundary vertices and all the others. For any vertex v that is *not* on the boundary, we observe that a vertex u can be at a distance $\leq k$ from v if and only if, `(i)` the unique subgraph of the decomposition that contains v also contains u and a uv-path of length $\leq k$, or `(ii)` $dist_G(u,x) \leq k - dist_G(v,x)$ from some vertex x on the boundary ($\mathcal{O}(\sqrt{r})$ balls to be considered).

[3] Note that Frederickson proposed several refinements of his construction in [42], some of which do use the fact that the input graph is planar. We will use in our proofs an even weaker version of his result than the one presented in this introduction.

Our strategy consists in computing a spanning path with low stabbing number for some "boundary hypergraph" whose hyperedges are the $\mathcal{O}(r\sqrt{r} \times (n/r)) = \mathcal{O}(n\sqrt{r})$ balls that we need to consider. We encounter a similar problem as for Theorem 3 because storing this hypergraph may require superquadratic space. Fortunately, we can encode this hypergraph in a much more compact way by taking advantage of `(i)` the fact that we can only have $\mathcal{O}(n/\sqrt{r})$ different centers for the balls, and `(ii)` that all the balls with a given center have a chain-like inclusion structure.

Note that we can apply the same strategy as above if, instead of deciding whether the diameter is at most k, we are given $(k_v)_{v \in V}$ and, for every vertex v, we want to decide whether its eccentricity is at most k_v. In particular, we can perform n simultaneous binary searches in order to compute all the eccentricities.

Although we keep the focus on computing the diameter, we shall stress in Sec. 2.4 that all our techniques can also be applied to *radius* computation (i.e., see Remark 1). Our algorithms almost need no particular information about the graph structure in order to be applied. In fact, we do not even need to compute the (distance) VC-dimension of the input graph! From the applicative point of view, this observation (further discussed in Sec. 2.4) is quite important. Indeed, computing the VC-dimension is W[1]-hard [30] and LogNP-hard [65]. Results of this paper were partially presented at the SODA'20 conference [33].

1.3 Organization of the paper

In Sec. 2 we formally introduce the concepts of (distance) VC-dimension and stabbing number, along with some of their basic properties. Then, we explain in Sec. 3 how to compute a spanning path with strongly sublinear stabbing number for a hypergraph of constant VC-dimension (Theorem 2). As a direct application, we give a short proof of Theorem 1. Our techniques are generalized in Sec. 4 so as to prove Theorems 3,4,5. Finally, our main technical result (Theorem 6) is proved in Sec. 5. For that, we will need to recall some useful results on the graphs of *polynomial expansion* [35]. We discuss some partial extensions of our results to weighted graphs, and some possible future work, in Sec. 5.3 and 6, respectively.

2 Preliminaries

After recalling a few basic definitions about graphs and hypergraphs (Sec. 2.1 and 2.2) we introduce our framework for computing the diameter of a graph in Sec. 2.3 and 2.4.

2.1 Graphs and diameter

For any undefined graph terminology, see [8]. Throughout all this paper we only consider graphs that are undirected, unweighted and connected. For every graph $G = (V, E)$, let $n := |V|$ be its *order* and $m := |E|$ be its *size*. We denote by $N_G(v)$ and $N_G[v] := N_G(v) \cup \{v\}$ the *open and closed neighbourhoods* of vertex v, respectively. The *degree* of v is equal to $|N_G(v)|$ and is denoted by $deg_G(v)$ in what follows. The *length* of a path is its number of edges, and the *distance* $dist_G(u, v)$ between $u, v \in V$ is equal to the length of a shortest uv-path. For every $v \in V$ and $k \geq 0$, the *k-neighbourhood* of v, also known as the ball of center v and radius k, is defined as $N_G^k[v] = \{u \in V \mid dist_G(u, v) \leq k\}$. For instance, $N_G^1[v]$ is exactly the closed neighbourhood of

v. The *eccentricity* of a vertex v is equal to $ecc_G(v) = \max_{u \in V} dist_G(u,v)$. The *radius* and the *diameter* of G are, respectively, $rad(G) = \min_{v \in V} ecc_G(v)$ and $diam(G) = \max_{v \in V} ecc_G(v)$.

> **Problem 1** (Diameter).
>
> **Input:** *A graph $G = (V, E)$.*
>
> **Output:** *The diameter of G.*

Theorem 7 ([67]). *Under the Strong Exponential-Time Hypothesis, we cannot decide whether a graph has diameter at most two in time $\mathcal{O}(mn^{1-\varepsilon})$, for any $\varepsilon > 0$.*

2.2 Hypergraphs

A *hypergraph* is a pair $\mathcal{H} = (X, R)$ with X being the set of vertices and $R \subseteq 2^X$ being the set of *hyperedges*. See also [5] for any undefined hypergraph terminology. Let $n := |X|$, $m := \sum_{q \in R} |q|$ and $r := |R|$ be the *order*, the *size* and the number of hyperedges of $\mathcal{H}$, respectively. For every vertex $x \in X$, let $R_x := \{q \in R \mid x \in q\}$. The *dual* of $\mathcal{H}$ is the hypergraph $\mathcal{H}^* := (R, X^*)$, where $X^* := \{R_x \mid x \in X\}$. In particular, $\mathcal{H}$ and $\mathcal{H}^{**}$ are isomorphic.

Several hypergraphs can be related to a graph G:

- The *closed neighbourhood hypergraph*, denoted by $\mathcal{N}_1(G)$, has vertex-set $X = V$ and hyperedge-set $R = \{N_G[v] \mid v \in V\}$;
- More generally, for every fixed $\ell \geq 0$, the *ℓ-neighbourhood hypergraph* of G is defined as $\mathcal{N}_\ell(G) = (V, \{N_G^\ell[v] \mid v \in V\})$. We stress that $\mathcal{N}_\ell(G)$ and its dual $\mathcal{N}_\ell^*(G)$ are isomorphic [11].
- Finally, the *ball hypergraph* of G, simply denoted by $\mathcal{B}(G)$, has for hyperedges the balls of all possible centers and radii in G. Equivalently, $\mathcal{B}(G) = \bigcup_{\ell \geq 0} \mathcal{N}_\ell(G)$.

2.3 VC-dimension

Let $\mathcal{H} = (X, R)$ be a fixed hypergraph. A subset $Y \subseteq X$ is *shattered* by $\mathcal{H}$ if, for every $Y' \subseteq Y$, there exists a hyperedge $q \in R$ such that $Y \cap q = Y'$. Then, the Vapnik-Chervonenkis dimension of $\mathcal{H}$ (abbreviated in what follows to *VC-dimension*) is the largest cardinality of a shattered subset. Similarly, the *dual VC-dimension* of $\mathcal{H}$ is the VC-dimension of its dual hypergraph $\mathcal{H}^*$. We will often use the following (easy) properties in our analysis:

Lemma 1 (Sauer-Shelah-Perles, [68, 69]). *Every n-vertex hypergraph of VC-dimension at most d has $\mathcal{O}(n^d)$ hyperedges.*

Lemma 2 ([21]). *Every hypergraph of VC-dimension d has dual VC-dimension at most 2^{d+1}.*

Lemma 3 ([55]). *For every hypergraph $\mathcal{H} = (X, R)$ and $Y \subseteq X$, let $R[Y] = \{q \cap Y \mid q \in R\}$. Then, the VC-dimension of $\mathcal{H}[Y] := (Y, R[Y])$ is at most the VC-dimension of $\mathcal{H}$.*

VC-dimension for graphs. The *VC-dimension* of a graph G is defined as the VC-dimension of its closed neighbourhood hypergraph $\mathcal{N}_1(G)$. For instance, K_h-minor free graphs (and so, H-minor free graphs for any H of order at most h) have VC-dimension at most $h-1$ [3]. Every k-interval graph has VC-dimension in $\mathcal{O}(k \log k)$ [32]. Other classes of constant VC-dimension – at most three – are unit disk graphs, chordal bipartite graphs, C_4-free bipartite graphs, graphs of girth at least five and undirected path graphs [10].

The *distance VC-dimension* of a graph G is defined as the VC-dimension of its ball hypergraph $\mathcal{B}(G)$. Chepoi, Estellon and Vaxès proved in [23] that planar graphs have *distance VC-dimension* at most 4, and remarked that more generally every K_h-minor free graph has *distance VC-dimension* at most $h-1$. Bousquet and Thomassé proved in [11] that graphs of bounded *distance VC-dimension* also generalize graphs of bounded rankwidth. Indeed, every graph of rankwidth k has *distance VC-dimension* at most $3 \cdot 2^{k+1} + 1$. For purpose of illustration, we next adapt a proof from [10] in order to show that interval graphs have *distance VC-dimension* at most two:

Lemma 4. *Every interval graph has distance VC-dimension at most* 2.

Proof. Let $G = (V, E)$ be an interval graph. We fix an interval model for G. For every $v \in V$, let $I(v) = [a_v, b_v]$ be the corresponding interval in the representation. Suppose now by contradiction that there is a set $S = \{v_1, v_2, v_3\}$ that is shattered by $\mathcal{B}(G)$. W.l.o.g., $a_{v_1} < a_{v_2} < a_{v_3}$. Since S is shattered, there exist some $u \in V$ and $k \geq 0$ such that $N_G^k[u] \cap S = \{v_1, v_3\}$. But then, let $I_{k-1}(u) := \bigcup_{w \in N_G^{k-1}[u]} I(w)$ be the contiguous segment of all the intervals of the vertices at a distance $\leq k-1$ from u. Note that $I_{k-1}(u) \cap I(v_2) = \emptyset$ because we assume that $v_2 \notin N_G^k[u]$. In this situation, either $I_{k-1}(u) \subseteq]-\infty, a_{v_2}[$ or $I_{k-1}(u) \subseteq]b_{v_2}, \infty[$ where $]x, y[= [x, y] \setminus \{x, y\}$ denotes the open interval between x and y. In fact we must have $I_{k-1}(u) \subseteq]b_{v_2}, \infty[$ because otherwise, $I_{k-1}(u) \cap I(v_3) = \emptyset$ would imply $v_3 \notin N_G^k[u]$, a contradiction. Since $I_{k-1}(u) \cap I(v_1) \neq \emptyset$, it implies that $b_{v_1} > b_{v_2}$, and so, $I(v_2) \subseteq I(v_1)$. As a result we have $N_G[v_2] \subseteq N_G[v_1]$. But then, for any $w \in V$ and $\ell \geq 1$, we have $v_2 \in N_G^\ell[w] \Longrightarrow v_1 \in N_G^\ell[w]$. The latter contradicts our hypothesis that S is shattered. □

2.4 Stabbing number and applications to DIAMETER

A spanning tree of $\mathcal{H} = (X, R)$ is a (classical) tree T whose node-set is exactly X. The *stabbing number* of such spanning tree T is the least k such that, for every hyperedge $q \in R$, there exist at most k edges $uv \in E(T)$ such that $|q \cap \{u, v\}| = 1$ (we also say that uv is *stabbed* by q). Given a set $q \subseteq X$, we let $E_T(q) = \{uv \in E(T) \mid u \in q, v \notin q\}$ be the set of all edges stabbed by q. Finally, the stabbing number of $\mathcal{H}$ is the minimum stabbing number over its spanning *paths*. Indeed, as noted in [21], every spanning tree T can be transformed into a spanning path of stabbing number at most twice bigger than for T. Therefore, there is essentially no loss of generality in restricting ourselves to spanning paths.

Lemma 5 ([21])**.** *Every n-vertex hypergraph of dual VC-dimension d has stabbing number* $\tilde{\mathcal{O}}(n^{1-\frac{1}{d}})$.

Overall it follows from Lemmata 2 and 5 that any n-vertex hypergraph of VC-dimension at most d has strongly sublinear stabbing number in $\tilde{\mathcal{O}}(n^{1-\frac{1}{2^{d+1}}})$. We stress that the proof of Lemma 5 is constructive but that it cannot be transformed into a truly subquadratic-time algorithm. Efficient computations of spanning paths with sublinear stabbing number – or related data structures – were proposed for many special cases from computational geometry [20, 59, 71].

Problem 2 (f-Approx Stabbing Number)**.**

Input: *A hypergraph $\mathcal{H} = (X, R)$ of VC-dimension at most d.*

Output: *A spanning path P of stabbing number at most $\tilde{\mathcal{O}}(n^{1-\frac{1}{f(d)}})$ and, for every $q \in R$, the set $E_P(q) = \{uv \in E(P) \mid u \in q, v \notin q\}$ of all edges stabbed by q.*

For simplicity of exposition, we will assume throughout the remainder of this paper that the VC-dimension of all the hypergraphs considered is part of the input. However in practice, we can easily weaken this assumption as follows. Given some "guess" d on the VC-dimension of the input, we can modify our proposed solutions so that they either output a spanning path whose stabbing number is at most $\tilde{\mathcal{O}}(n^{1-\frac{1}{f(d)}})$, for some function f, or conclude that the VC-dimension of the input is larger than d. By dichotomic search, we so can compute some minimum d^* such that, for any $d \geq d^*$, our algorithms always output a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-\frac{1}{f(d)}})$. We stress that d^* is at most the VC-dimension of the input, but that it could be much smaller in practice.

Reduction from diameter computation. We now recall the following simple approach that we use in order to solve Diameter on graphs of constant (distance) VC-dimension.

Lemma 6. *Let G be a graph and $k \geq 2$. If the hypergraph $\mathcal{N}_{k-1}(G)$ has VC-dimension at most d, and we can solve f-Approx Stabbing Number for $\mathcal{N}_{k-1}(G)$ in time $T(n,m)$, then we can decide whether G has diameter at most k in time $\tilde{\mathcal{O}}(T(n,m) + mn^{1-\frac{1}{f(d)}})$.*

Proof. Let us first compute a spanning path P of stabbing number at most $\tilde{\mathcal{O}}(n^{1-\frac{1}{f(d)}})$ for $\mathcal{N}_{k-1}(G)$. By the hypothesis, it takes $\mathcal{O}(T(n,m))$ time. For every $v \in V$ we can compute from $E_P(N_G^{k-1}[v])$ a set $I_{k-1}(v)$ of t_v intervals, where $|E_P(N_G^{k-1}[v])|-1 \leq t_v \leq |E_P(N_G^{k-1}[v])|+1$, such that $\bigcup I_{k-1}(v) = N_G^{k-1}[v]$. Note that the set $E_P(N_G^{k-1}[v])$ defines two possible unions of intervals stabbing its edges. We select the one containing v. This preprocessing phase takes time $\mathcal{O}(|E_P(N_G^{k-1}[v])|) = \tilde{\mathcal{O}}(n^{1-\frac{1}{f(d)}})$, and so, $\tilde{\mathcal{O}}(n^{2-\frac{1}{f(d)}})$ total time. Then in order to decide whether $diam(G) \leq k$, we are left to decide whether for every $u \in V$ we have $\bigcup_{v \in N_G[u]} I_{k-1}(v) = V(G)$. For that, it suffices to collect the $\tilde{\mathcal{O}}(deg_G(u) \cdot n^{1-\frac{1}{f(d)}})$ extremities of the intervals in $\bigcup_{v \in N_G[u]} I_{k-1}(v)$, and then to order them according to the path order. We first check that the first opening extremity occurs at the first node of the path. A linear scan then allows to count for each interval extremity the number of intervals opened before it and not yet closed. It then suffices to check that this number does not reach 0 except for the last closing extremity that should occur at the end of the path. As a result, this last verification phase can be done in total time $\tilde{\mathcal{O}}(mn^{1-\frac{1}{f(d)}})$. ◻

Remark 1. We recall that the *radius* of a graph G is equal to $rad(G) = \min_{v \in V} ecc_G(v)$. Under the Hitting Set conjecture, we cannot compute the radius of a graph in truly subquadratic-time [1]. We here observe that we can easily modify the framework of Lemma 6 in order to decide whether a graph has radius at most k. Indeed, for that it suffices to check whether there exists at least one vertex u such that $\bigcup_{v \in N_G[u]} I_{k-1}(v) = V(G)$.

Our main task in the remainder of this article will be to solve f-Approx Stabbing Number efficiently on ℓ-neighbourhood hypergraphs, for some increasing function f. Then, we can apply Lemma 6 in order to efficiently solve Diameter.

3 Computation of Spanning Paths with low Stabbing Number

We prove in this section our first main result in the paper, whose statement is reminded below:

Theorem 1. *For every $d > 0$, there exists a constant $\varepsilon_d \in (0;1)$ such that in deterministic time $\tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$ we can decide whether a graph of VC-dimension at most d has diameter two.*

We will need the following result in our proofs:

Lemma 7 ([21]). *There is a deterministic polynomial-time algorithm that outputs, for every n-vertex hypergraph $\mathcal{H}$ of VC-dimension at most d, a spanning path of stabbing number $\mathcal{O}(n^{1-1/2^{d+1}} \log n)$.*

This above lemma is a consequence of Theorem 4.3 in [21] and the discussion about the complexity of the algorithm resulting from their proof. We note that their result applies to infinite range spaces too, with the initial step in their proof reducing to the finite case. In order to derive Lemma 7 from [21], we use the bound on the dual VC-dimension resulting from Lemma 2 and the fact that no initial step is required as we start from a *finite* range space. Better randomized algorithms can be obtained through the approximation results in [6, 49]. They are expressed for spanning trees but easily convert to paths as previously noted. The algorithms in [6, 49] use LP relaxation and randomized rounding. It is not immediately clear if they can be derandomized using classical techniques. Indeed, the algorithm from [49] works by phases. During a phase, it needs to solve an ILP relaxation and then to apply some randomized rounding technique. In the worst case, this main phase is repeated $\mathcal{O}(\log n)$ times. We observe that even by using the best known upper-bounds on the time complexity of linear programming [24], this overall process takes super-quadratic time. In what follows, we use the Sauer-Shelah-Perles Lemma (Lemma 1) in order to obtain better trade-offs between the running-time and the quality of the solution.

Theorem 2. *For every $d > 0$, there exists a constant $\varepsilon_d \in (0;1)$ such that in $\tilde{\mathcal{O}}(m + n^{2-\varepsilon_d})$ deterministic time, for every n-vertex hypergraph $\mathcal{H}$ of VC-dimension at most d and size m, we can compute a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$. In particular, this algorithm computes for each hyperedge the ends of its corresponding $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ intervals.*

Moreover, $\varepsilon_d = \frac{1}{2^{d+1}(c(d+1)-1)+1}$ for some universal constant $c > 2$.

We will use several times the following folklore lemma about sorting sets.

Lemma 8. *Given p subsets $S_1, \ldots, S_p$ of $\{1, \ldots, n\}$, it is possible to sort them in lexicographic order in time $\mathcal{O}(n + \sum_{i=1}^{p} |S_i|)$.*

Lemma 8 can be achieved through partition refinement (see [64, 48]): starting from a single part with all sets, we extract the sets containing 1 to obtain two parts, and then similarly split each part by extracting sets containing 2, and so on for remaining elements. In the end each part contains sets that are pairwise equal. When splitting a part according to an element i, we order the sub-part of sets containing i just before the other sub-part. This allows to obtain a lexicographic ordering of the parts in the end.

Proof of Theorem 2. Let $\eta \in (0;1)$ to be fixed later in the proof. We arbitrarily partition the vertex-set X into subsets $X_1, X_2, \ldots, X_p$ such that $p = \Theta(n^{1-\eta})$ and, for every $1 \le i \le p$, $|X_i| = \mathcal{O}(n^{\eta})$. Our aim is to apply Lemma 7 to the induced subhypergraphs $\mathcal{H}[X_1], \mathcal{H}[X_2], \ldots, \mathcal{H}[X_p]$. We stress that all these subhypergraphs can be constructed in total $\mathcal{O}(m)$-time, as follows: we scan all the hyperedges q once in order to compute $(q \cap X_i)_{1 \le i \le p}$; then, for every i, we use the linear-time sorting algorithm of Lemma 8 in order to suppress duplicated values in $\{q \cap X_i \mid q \in R\}$. This is indeed linear-time because, for any hyperedge q the sets in $(q \cap X_i)_{1 \le i \le p}$ are pairwise disjoint, and therefore, $\sum_{i=1}^{p} \sum_{q \in R} |q \cap X_i| = \sum_{q \in R} |q| = m$. We store, for each hyperedge q, what are the non-empty sub-hyperedges $q \cap X_i$, for $1 \le i \le p$.

Claim 1. *Given $\mathcal{H}[X_1], \mathcal{H}[X_2], \ldots, \mathcal{H}[X_p]$, we can compute a spanning path for $\mathcal{H}$ of stabbing number $\tilde{\mathcal{O}}(n^{1-\frac{\eta}{2^{d+1}}})$. Moreover, it takes $\mathcal{O}(n^{1+\eta(c(d+1)-1)})$ time for some universal constant $c > 2$.*

Proof. By Lemma 3, every $\mathcal{H}[X_i]$ has VC-dimension at most d. This implies that $\mathcal{H}[X_i]$ has $\mathcal{O}(n^{\eta d})$ hyperedges (Lemma 1), and so it has size $\mathcal{O}(n^{\eta(d+1)})$. Furthermore by Lemma 7 we can compute deterministically a spanning path of stabbing number $\tilde{\mathcal{O}}\left(n^{\eta\left(1-\frac{1}{2^{d+1}}\right)}\right)$, in time $\mathcal{O}(n^{c\eta(d+1)})$ for some universal constant c.

Let $P_1, P_2, \ldots, P_p$ be the spanning paths that we obtain. We obtain a spanning path P for $\mathcal{H}$ by concatenating all the P_i's. For every $1 \le i \le p$, we recall that the stabbing number of P_i is in $\tilde{\mathcal{O}}\left(n^{\eta\left(1-\frac{1}{2^{d+1}}\right)}\right)$. Therefore by construction, the stabbing number of P is in $\tilde{\mathcal{O}}\left(p \cdot n^{\eta\left(1-\frac{1}{2^{d+1}}\right)} + p - 1\right) = \tilde{\mathcal{O}}\left(n^{1-\frac{\eta}{2^{d+1}}}\right)$. ◇

Let P be the spanning path obtained with Claim 1. Finally, for every $q \in R$, we compute the set $E_P(q)$ of all edges of P stabbed by q, in total $\mathcal{O}(m)$-time. For that, we simply scan once all the edges $uv \in E(P)$. We enumerate all hyperedges in R_u and in R_v. For every $q \in R_u \setminus R_v$ (resp. $s \in R_v \setminus R_u$), we add uv to $E_P(q)$ (resp. $E_P(s)$). Note that for the above, we only need to scan twice all the hyperedges. The total running-time is in $\mathcal{O}(m + p \cdot n^{c\eta(d+1)}) = \mathcal{O}(m + n^{1+\eta(c(d+1)-1)})$. Overall, we achieve a good trade-off between running-time and approximation factor if we have $2 - \frac{\eta}{2^{d+1}} = 1 + \eta\left(c(d+1)-1\right)$. Therefore we set $\eta = \frac{1}{c(d+1)+\frac{1}{2^{d+1}}-1}$, and then $\varepsilon_d = \frac{\eta}{2^{d+1}} = \frac{1}{2^{d+1}(c(d+1)-1)+1}$. □

We observe that our analysis could be easily improved in some particular cases, *e.g.*, for all hypergraphs that are isomorphic to their dual.

We are now ready to prove the main result in this section:

Proof of Theorem 1. We compute the closed neighbourhood hypergraph of G. It can be done in linear time, simply by scanning the adjacency list. Then, we apply Theorem 2 to $\mathcal{N}_1(G)$. The result now follows from Lemma 6 applied to the function $f : d \to 1/\varepsilon_d$. □

4 Bounded Diameter with ε-Nets

For graphs of bounded *distance* VC-dimension we now generalize Theorem 1 from the previous section to larger values for the diameter.

Theorem 3. *There exists a Monte Carlo algorithm such that, for every positive integers d and k, we can decide whether a graph of distance VC-dimension at most d has diameter at most k. The running time is in $\tilde{\mathcal{O}}(k \cdot mn^{1-\varepsilon_d})$, where $\varepsilon_d \in (0;1)$ only depends on d.*

Our proof crucially relies on the concept of ε*-net*. We recall that for a hypergraph $\mathcal{H} = (X, R)$, a subset $Y \subseteq X$ is called an ε-net if, for every $q \in R$ such that $|q| \geq \varepsilon n$, we have $Y \cap q \neq \emptyset$.

Lemma 9 ([51, 70])**.** *For every hypergraph of VC-dimension at most d, any random subset of size* $\Omega\left(\frac{d}{\varepsilon}\log\left(\frac{1}{\varepsilon\delta}\right)\right)$ *is an* ε*-net with probability* $1-\delta$*.*

We will also need the following result:

Lemma 10 ([21])**.** *For every hypergraph* $\mathcal{H} = (X, R)$*, let* $\hat{R} := \{q\Delta q' \mid q, q' \in R\}$ *be the set of symmetric differences between hyperedges. If* $\mathcal{H}$ *has VC-dimension at most d then,* $\hat{\mathcal{H}} := (X, \hat{R})$ *has bounded VC-dimension.*

We observe that no explicit upper bound on the VC-dimension of $\hat{\mathcal{H}}$ was stated in [21]. Nevertheless it can be easily deduced from their proof that it is in $\mathcal{O}(d \log d)$ (see also [37]).

The following partition lemma is the cornerstone of our algorithm.

Lemma 11. *Let* $G = (V, E)$ *be a graph of distance VC-dimension at most d, and let S be any random subset of size* $\tilde{\Theta}(d/\varepsilon)$*. Then w.h.p., for every* $\ell \geq 0$ *and for every* $u, v \in V$ *such that* $N_G^\ell[u] \cap S = N_G^\ell[v] \cap S$*, we have* $\mid N_G^\ell[u]\Delta N_G^\ell[v] \mid \leq \varepsilon n$*.*

Proof. Let $\hat{R} = \{\ N_G^{\ell_1}[x]\Delta N_G^{\ell_2}[y]\ \mid x, y \in V \text{ and } \ell_1, \ell_2 \geq 0\}$ be the set of the symmetric differences between the balls of G. Since G has *distance VC-dimension* at most d then, by Lemma 10, the hypergraph $\hat{\mathcal{H}} = (V, \hat{R})$ has VC-dimension in $\mathcal{O}(d \log d)$. Then by Lemma 9, w.h.p. S is an ε-net for $\hat{\mathcal{H}}$. Therefore, for every $\ell \geq 0$ and for every $u, v \in V$, $\mid N_G^\ell[u]\Delta N_G^\ell[v] \mid > \varepsilon n \Longrightarrow (N_G^\ell[u]\Delta N_G^\ell[v]) \cap S \neq \emptyset$. We stress that $(N_G^\ell[u]\Delta N_G^\ell[v]) \cap S \neq \emptyset \Longrightarrow N_G^\ell[u] \cap S \neq N_G^\ell[v] \cap S$. □

This above partition lemma will be useful in order to group the vertices in a small number of groups, with every two vertices in a group having almost the same ball of radius ℓ. Here there is a trade-off between the number of groups (that we upper-bound by using the Sauer-Shelah-Perles Lemma) and, for every two vertices in the same group, the maximum number of vertices in which their respective balls of radius ℓ can differ.

More precisely, our approach in the next two sections can be summarized as follows:

1. We compute a spanning path P'_k for $\mathcal{N}_k(G)$ of low *total* stabbing number, with the latter being equal to $\sum_{v \in V} |E_{P'_k}(N_G^k[v])|$;

2. Then, we compute an ε-net, for some well-chosen ε, and by doing so we partition the vertex-set into $p(\varepsilon)$ disjoint groups $V_1, V_2, \ldots, V_{p(\varepsilon)}$. For every j we select a unique $v_j \in V_j$. We restrict ourselves to $\mathcal{H}_k := (V, \{N_G^k[v_j] \mid 1 \leq j \leq p(\varepsilon)\})$. We compute a spanning path P_k of low stabbing number for this subhypergraph.

3. We observe that if P_k is a spanning path of stabbing number t for $\mathcal{H}_k$, then it is also a spanning path of stabbing number $t + \mathcal{O}(\varepsilon n)$ for $\mathcal{N}_k(G)$. Finally, for every $1 \leq j \leq p(\varepsilon)$, we consider the unselected vertices $u \in V_j \setminus \{v_j\}$ sequentially. We compute the set of all the edges in $E(P_k)$ that are stabbed by $N_G^k[u]$. For that, it suffices to compute the $\mathcal{O}(\varepsilon n)$ vertices of $N_G^k[u]\Delta N_G^k[v_j]$. We do so efficiently by using the auxiliary spanning path P'_k.

We next give a first application of our approach (our proof of Theorem 6 also follows a quite similar approach).

Proof of Theorem 3. Let ε_d be the constant of Theorem 2. We shall prove the following claim by finite induction:

Claim 2. *For every $1 \leq i \leq k-1$, we can compute a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ for $\mathcal{N}_i(G)$. Moreover, it can be done in time $\tilde{\mathcal{O}}(i \cdot mn^{1-\varepsilon_d})$.*

The result will follow from this claim and Lemma 6 by taking $i = k-1$.

Proof. By Theorem 2, the claim is true for the base case $i = 1$. Assume by our induction hypothesis that the claim holds for $i-1$. We divide the remainder of the proof into two subclaims.

Subclaim 1. *Let P_{i-1} be a spanning path of stabbing number t for $\mathcal{N}_{i-1}(G)$. We can transform P_{i-1} into a spanning path P'_i for $\mathcal{N}_i(G)$, such that $\sum_{v\in V} |E_{P'_i}(N^i_G[v])| = \mathcal{O}(tm)$. Moreover, the transformation takes time $\mathcal{O}(tm)$.*

Proof. Let $u \in V$. We have that $N^i_G[u] = \bigcup_{w\in N_G[u]} N^{i-1}_G[u]$. In particular, the ball $N^i_G[u]$ is the union of all intervals contained in a ball $N^{i-1}_G[w]$, for $w \in N_G[u]$. Then in time $\mathcal{O}(deg_G(u)\cdot t)$, we can collect the edge-sets $E_{P_{i-1}}(N^{i-1}_G[w])$ of all the edges of P_{i-1} that are stabbed by w, for $w \in N_G[u]$. We compute from these edge-sets a (suboptimal) representation of $N^i_G[u]$ into $\mathcal{O}(deg_G(u)\cdot t)$ intervals of P_{i-1}. ◦

Subclaim 2. *Let P'_i be a spanning path for $\mathcal{N}_i(G)$, such that $\sum_{v\in V} |E_{P'_i}(N^i_G[v])| = \mathcal{O}(tm)$. Then, in time $\tilde{\mathcal{O}}((n^{1-\varepsilon_d} + t)\cdot m)$, we can compute a spanning path P_i of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$.*

Proof. Let $\varepsilon := \Theta(n^{-\varepsilon_d})$. We perform a breadth-first search from every vertex in some random subset S of cardinality $\tilde{\mathcal{O}}(d/\varepsilon) = \tilde{\mathcal{O}}(d \cdot n^{\varepsilon_d})$. By doing so we define an equivalence relation $\sim$ on V such that $u \sim v \stackrel{def}{\iff} N^i_G[u] \cap S = N^i_G[v] \cap S$. We so partition V into some groups $V_1, V_2, \ldots, V_p$. Since by the hypothesis G has *distance VC-dimension* at most d then, by Lemma 1 we have $p = \mathcal{O}(|S|^d) = \mathcal{O}(d^d \log^d n \cdot n^{\varepsilon_d d})$. Furthermore by Lemma 11, we have w.h.p. $u \sim v \Longrightarrow |\ N^i_G[u]\Delta N^i_G[v]\ | = \mathcal{O}(\varepsilon n) = \mathcal{O}(n^{1-\varepsilon_d})$.

We first compute $V_1, \ldots, V_p$ thanks to the outputs of breadth-first search from every vertex of S and to Lemma 8. It takes $\mathcal{O}(m|S|) = \tilde{\mathcal{O}}(dmn^{\varepsilon_d})$ time. The algorithm now proceeds as follows:

1. For every $1 \leq j \leq p$, we select a unique $v_j \in V_j$, and then we start a breadth-first search from this vertex. Since $p = \tilde{\mathcal{O}}(d^d \cdot n^{\varepsilon_d d})$ and we have $\varepsilon_d < 1/(d+1)$, this phase can be implemented in time $\mathcal{O}(md^d \log^d n \cdot n^{\varepsilon_d d}) = \mathcal{O}(mn^{1-\varepsilon_d})$, that is truly subquadratic. Note that this time bound also holds for non constant d as long as $d = \mathcal{O}(\frac{\log n}{\log\log n})$.

2. Let $R_i := \{N^i_G[v_j] \mid 1 \leq j \leq p\}$, and let $\mathcal{H}_i := (V, R_i)$. Note that since $\mathcal{H}_i \subseteq \mathcal{B}(G)$, the VC-dimension of $\mathcal{H}_i$ is at most d. Furthermore, the order and size of $\mathcal{H}_i$ are, respectively, n and $m_i := \mathcal{O}(pn) = \tilde{\mathcal{O}}(n^{1+\varepsilon_d d})$. By Theorem 2, we can compute a spanning path P_i for $\mathcal{H}_i$ of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ in time $\tilde{\mathcal{O}}(m_i + n^{2-\varepsilon_d}) = \tilde{\mathcal{O}}(n^{1+\varepsilon_d d} + n^{2-\varepsilon_d}) = \tilde{\mathcal{O}}(n^{1-\varepsilon_d} m)$.

3. We observe that P_i is a spanning path of $\mathcal{N}_i(G)$ of stabbing number:

$$\tilde{\mathcal{O}}(n^{1-\varepsilon_d}) + \max_{1\leq j\leq p} \max_{u\in V_j\setminus\{v_j\}} |\ N^i_G[u]\Delta N^i_G[v_j]\ | = \tilde{\mathcal{O}}(n^{1-\varepsilon_d}).$$

   Indeed, for every $1 \leq j \leq p$, let $u \in V_j \setminus \{v_j\}$ be arbitrary. Let us consider the $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ maximal intervals of which the union equals $N^i_G[v_j]$. Every vertex of $N^i_G[v_j] \setminus N^i_G[u]$ breaks one of these intervals in two sub-intervals, thus increasing by at most one the number of intervals needed for the ball $N^i_G[u]$. Furthermore, every vertex of $N^i_G[u] \setminus N^i_G[v_j]$ (since it is not contained in one of the intervals of which $N_G^i[v_j]$ is the union) may also require one more interval in order to span the ball $N_G^i[u]$. As a result, the ball $N_G^i[u]$ is the union of at most $\tilde{\mathcal{O}}(n^{1-\varepsilon_d}) + \mid N_G^i[u]\Delta N_G^i[v_j] \mid$ intervals of the path P_i.

We are now left with computing, for every $1 \leq j \leq p$ and $u \in V_j \setminus \{v_j\}$, the set $E_{P_i}(N_G^i[u])$ of all the edges stabbed by the ball of radius i centered at u. For that, since we are already given $E_{P_i}(N_G^i[v_j])$, it suffices to compute $N_G^i[u]\Delta N_G^i[v_j]$. We proceed in three steps:

- We use the spanning path P_i' for $\mathcal{N}_i(G)$ and a (suboptimal) representation $I_i(u)$ of $N_G^i[u]$ into $\mathcal{O}(|E_{P_i'}(N_G^i[u])|)$ intervals. We also compute a representation $\overline{I_i(u)}$ of $V \setminus N_G^i[u]$ into $\mathcal{O}(|E_{P_i'}(N_G^i[u])|)$ intervals of P_i'. Overall this step takes total time $\tilde{\mathcal{O}}(tm)$.
- Let $\sigma_i : V \to V(P_i')$ be the permutation that maps every vertex to its position in the spanning path P_i'. For every $1 \leq j \leq p$, we construct two balanced binary search trees whose items are, respectively, $\{\sigma_i(x) \mid x \in N_G^i[v_j]\}$ and $\{\sigma_i(y) \mid y \notin N_G^i[v_j]\}$. Overall, this takes total time $\tilde{\mathcal{O}}(np) = \tilde{\mathcal{O}}(n^{1+\varepsilon_d d}) = \tilde{o}(mn^{1-\varepsilon_d})$.
- Finally, let us again consider some $u \in V_j \setminus \{v_j\}$ for some j. For every interval from $I_i(u)$, we want to enumerate the vertices of $V \setminus N_G^i[v_j]$ that lie on this interval. Since we stored all of $V \setminus N_G^i[v_j]$ into a balanced binary search tree, this can be done in time $\mathcal{O}(\log n)$ plus $\mathcal{O}(1)$ extra time per vertex in the solution. In the same way, for every interval from $\overline{I_i(u)}$, we enumerate the vertices of $N_G^i[v_j]$ that lie on this interval. For a fixed u, the total time for this step is in $\tilde{\mathcal{O}}(\ |I_i(u)| + |\overline{I_i(u)}| + |N_G^i[u]\Delta N_G^i[v_j]|\ ) = \tilde{\mathcal{O}}(|E_{P_i'}(N_G^i[u])| + n^{1-\varepsilon_d})$. Therefore, this last step takes total time $\tilde{\mathcal{O}}(tm + n^{2-\varepsilon_d})$.

$\circ$

Now, by the induction hypothesis we get a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ for $\mathcal{N}_{i-1}(G)$. By Subclaim 1 we transform such spanning path into a spanning path P_i' for $\mathcal{N}_i(G)$, where $\sum_{u\in V} |E_{P_i'}(N_G^i[u])| = \tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$. Finally, by Subclaim 2 we can use P_i' in order to compute, in time $\tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$, a spanning path P_i of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$. The above algorithm achieves proving that our claim holds for i. $\diamond$

Summarizing, by Claim 2 we can compute a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ for the hypergraph $\mathcal{N}_{k-1}(G)$, in time $\tilde{\mathcal{O}}(k \cdot mn^{1-\varepsilon_d})$. By Lemma 6 it implies that we can also decide whether G has diameter at most k, and if so, we compute $diam(G)$ exactly, in time $\tilde{\mathcal{O}}(k \cdot mn^{1-\varepsilon_d})$. $\square$

4.1 Application to nowhere dense graph classes

A closer look at the proof of Theorem 3 shows that it also holds if, instead of having bounded *distance VC-dimension*, there rather exists some constant d such that, for every $1 \leq i \leq k-1$, the VC-dimension of the i-neighbourhood hypergraph is at most d (the latter value is sometimes called the distance-i VC-dimension of the graph [62]). It has algorithmic implications for some special cases of sparse graphs. Namely, H is an *r-shallow minor* of a graph G if it can be obtained from some subgraph of G by the contraction of pairwise disjoint subgraphs of radius at most r [66]; a graph family $\mathcal{G}$ is termed *nowhere dense* if, for any r, there exists a graph H_r which is not an r-shallow minor for any graph in $\mathcal{G}$ [61]. Of interest here is that, for any graph class $\mathcal{G}$ nowhere dense, and for any i, the distance-i VC-dimension of any graph in $\mathcal{G}$ is upper-bounded by some constant d_i [62]. By choosing $d := \max_{1 \le i \le k-1} d_i$, we thus obtain the following weaker version of Theorem 3 for nowhere dense graphs:

Theorem 4. *Let $\mathcal{G}$ be a class of nowhere dense graphs. There exists a Monte Carlo algorithm such that, for every constant $k = \mathcal{O}(1)$, for any graph in $\mathcal{G}$ we can decide whether its diameter is at most k in $\tilde{\mathcal{O}}(mn^{1-\varepsilon_{\mathcal{G}}(k)})$ time, for some constant $\varepsilon_{\mathcal{G}}(k) \in (0;1)$ that only depends on k.*

Note that, for any class of nowhere dense graphs, there also exists an *FPT* algorithm, in time $\mathcal{O}(f(k) \cdot n^{1+o(1)})$, for deciding whether the diameter is at most k [47]. The function f is, at least, a tower of exponentials in k. Our result shows that a better dependency on k is possible at the cost of a higher exponent on n. We leave as open to find FPT algorithms with better trade-offs.

4.2 Exact distance oracles

Before ending this section, we present an interesting by-product of our approach for *exact* distance computations.

Theorem 5. *Let $d > 0$ and let ε_d be as defined in Theorem 2. For any graph G of distance VC-dimension at most d, there exists an exact distance oracle in $\tilde{\mathcal{O}}(n^{2-\frac{\varepsilon_d}{2}})$ space, that answers distance queries in $\tilde{\mathcal{O}}(n^{1-\frac{\varepsilon_d}{2}})$ time. Moreover, there is a Monte Carlo algorithm for constructing such an oracle, in $\tilde{\mathcal{O}}(mn^{1-\frac{\varepsilon_d}{2}})$ randomized time. This oracle may fail in reporting a distance correctly with probability at most $1/n^{\mathcal{O}(1)}$.*

Proof. We start presenting the construction of our distance oracle (pre-processing). Let k be a parameter to be fixed later in our proof.

1. For every $1 \le i \le k$, we construct a spanning path P_i for $\mathcal{N}_i(G)$, of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ — along with the corresponding $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ intervals for $N^i[v]$, for every vertex v.

2. Then, we sample a subset S_k of $\tilde{\mathcal{O}}(n/k)$ vertices, and we compute a shortest-path tree for each such vertex.

Let us analyze the runtime of this above construction. As it was explained in the proof of Theorem 3, the first step (computation of k spanning paths of low stabbing number) can be done in randomized time $\tilde{\mathcal{O}}(kmn^{1-\varepsilon_d})$. For the second step, since we only need to perform a BFS for each vertex of S_k, the runtime is in deterministic time $\tilde{\mathcal{O}}\left(\frac{n}{k}m\right)$. Overall, the total pre-processing time is in $\tilde{\mathcal{O}}\left(\left(kn^{1-\varepsilon_d} + \frac{n}{k}\right)m\right)$.

Furthermore, let us analyze the space of this oracle. Each spanning path P_i requires $\mathcal{O}(n)$ space. Since, for every vertex v, $N^i[v]$ is the union of $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ intervals, we have $\sum_{v \in V} |E_{P_i}(N^i[v])| = \tilde{\mathcal{O}}(n^{2-\varepsilon_d})$. Thus, we need $\tilde{\mathcal{O}}(kn^{2-\varepsilon_d})$ space for the k spanning paths and the corresponding sets E_{P_i}. For the second step, each shortest-path tree requires $\mathcal{O}(n)$ space, and therefore the total space required is in $\tilde{\mathcal{O}}\left(\frac{n^2}{k}\right)$. Overall, the oracle requires $\tilde{\mathcal{O}}\left(kn^{2-\varepsilon_d} + \frac{n^2}{k}\right)$ space.

Finally, given a pair (u, v) of vertices, we compute $dist(u, v)$ as follows:

- We check whether $dist(u, v) \le k$. By using P_k, it can be done in time $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$.

- If $dist(u,v) \leq k$, then we compute the smallest i such that $u \in N^i[v]$, which is precisely $dist(u,v)$. Note that we only need to test $\mathcal{O}(\log k)$ values for that (because we can apply binary search), and that each test takes time $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ by using the corresponding spanning path.

- Otherwise, we output $dist(u,v) = \min_{s \in S_k} dist(u,s) + dist(v,s)$. Since we sampled S_k u.a.r., the probability that $dist(u,v) \neq \min_{s \in S_k} dist(u,s) + dist(v,s)$ for at least one pair (u,v) at distance more than k is at most $1/n^c$, for some arbitrarily large constant c [7].

Overall, the query time is in $\tilde{\mathcal{O}}(n^{1-\varepsilon_d}) + \tilde{\mathcal{O}}\left(\frac{n}{k}\right)$. In order to optimize the space complexity, we set the value of our parameter to $k = \tilde{\mathcal{O}}(n^{\frac{\varepsilon_d}{2}})$. □

5 Diameter Computation in Truly Subquadratic Time

We finally improve the results of Theorem 3 for a more restricted family of graphs of bounded *distance VC-dimension*. Before that, we need to introduce a bit more of graph terminology. A class of graphs is called *monotone* if it is closed by taking subgraphs. For a connected n-vertex graph G, a *separator* is a subset S such that $G \setminus S$ is disconnected. It is called *balanced* if every connected component of $G \setminus S$ has order at most $2n/3$. Finally, a class of graphs has *strongly sublinear* balanced separators if every connected n-vertex graph in the class has a balanced separator of cardinality at most $C \cdot n^{\alpha}$ for some constants C and $\alpha < 1$.

Theorem 6. *Let $\mathcal{G}$ be a monotone graph class with strongly sublinear balanced separators. Then, for every $d > 0$, for any graph in $\mathcal{G}$ of distance VC-dimension at most d, we can compute all the eccentricities (and so, the diameter) in* deterministic *time $\tilde{\mathcal{O}}(n^{2-\varepsilon_{\mathcal{G}}(d)})$, for some constant $\varepsilon_{\mathcal{G}}(d) \in (0;1)$ that only depends on d.*

We postpone the technical proof of this result to Sec. 5.2. Let us emphasize that Theorem 6 cannot be applied to *all* graph classes of bounded *distance VC-dimension*. For instance, we proved in Lemma 4 that the intervals graphs have *distance VC-dimension* at most two. However, there exist intervals graphs with no balanced separators of sublinear size. We give some interesting cases where Theorem 6 *does* apply in Sec. 5.1 (see also Sec. 5.3, where we partially extend our results to weighted graphs).

Finally, we say that a class of graphs $\mathcal{G}$ has *polynomial expansion* if there exists a polynomial p such that, for every r-shallow minor of a graph in $\mathcal{G}$ (cf. Section 4.1), its average degree is at most $p(r)$. We want to stress that there is an equivalence between the monotone classes of graphs $\mathcal{G}$ with strongly sublinear balanced separators and those of polynomial expansion [35]. In particular, the graphs in $\mathcal{G}$ have bounded degeneracy, and so, they are sparse (i.e., with $m = \mathcal{O}(n)$ edges). We will often use this property in what follows.

5.1 Application to H-minor free graphs

Let us now review some interesting classes where Theorem 6 does apply. Since planar graphs have *distance VC-dimension* at most four [11] then, it follows from the planar separator theorem of Lipton and Tarjan [58] that Theorem 6 applies to the class of planar graphs. Therefore, Theorem 6 gives us a new subquadratic-time algorithm for diameter computation on *unweighted* planar graphs, but with a slower running-time than for the algorithms presented in [17, 45]. More generally, the following separator theorem is from Alon et al.:

Lemma 12 ([2])**.** *Every K_h-minor free graph has a balanced separator of cardinality $\mathcal{O}(h^{3/2}\sqrt{n})$. Moreover, such a separator can be found in $\mathcal{O}(n^{3/2})$ time.*

See also [54, 72] for various trade-offs between the size of the separator and the time that is needed in order to find it. We recall that K_h-minor free graphs have *distance VC-dimension* at most $h-1$ [23, 11]. By combining this result with Lemma 12, we so prove the following theorem:

Corollary 1. *For any H-minor free graph, we can compute all the eccentricities in deterministic time $\tilde{\mathcal{O}}(n^{2-\varepsilon_H})$, where $\varepsilon_H \in (0;1)$ is a constant that only depends on H.*

For most values of H this is the first known subquadratic-time algorithm for diameter computation on H-minor free graphs. In particular, this is the first known subquadratic-time algorithm for diameter computation on (unweighted) bounded-genus graphs to the best of our knowledge (see the planar graphs paragraph in the introduction).

5.2 Proof of Theorem 6

The remainder of this section is devoted to the proof of Theorem 6. We start by presenting, in a separate subsection, all the required background on r-divisions.

Algorithmic aspects of r-divisions

Throughout all this section, let $\mathcal{G}_{\alpha,C}$ be the class of all the graphs G such that, for every connected h-vertex subgraph of G, there exists a balanced separator of order at most $C \cdot h^{\alpha}$. The following intermediate result is an almost direct consequence of a previous algorithm from Plotkin et al. [66].

Lemma 13 ([34])**.** *For every n-vertex m-edge graph $G \in \mathcal{G}_{\alpha,C}$, we can find a balanced separator of order $\mathcal{O}(n^{\frac{4+\alpha}{5}})$ in time $\mathcal{O}(mn^{\frac{4+\alpha}{5}}) = \mathcal{O}(n^{2-\frac{1-\alpha}{5}})$.*

We will also use the following simple result:

Lemma 14. *Let G be a graph and S a balanced separator. We can bipartition in linear time the connected components of $G \setminus S$ in two disjoint sets A and B such that* $\max\{|A|,|B|\} \leq 2n/3$.

Proof. Let $C_1, C_2, \ldots, C_k$ be the connected components of $G \setminus S$. They can be computed in linear time. We define $i_0 := \max\{i \mid |\bigcup_{j<i} C_j| \leq 2n/3\}$. This value i_0 can be computed in $\mathcal{O}(n)$ time, simply by scanning the connected components in order until we have scanned more than $2n/3$ vertices. Let $A' := \bigcup_{j<i_0} C_j$ and $B' := \bigcup_{j>i_0} C_j$. If $|B' \cup C_{i_0}| \leq 2n/3$ then we are done by setting $A := A'$, $B := B' \cup C_{i_0}$. Thus, from now on let us assume that $|B' \cup C_{i_0}| > 2n/3$. Note that since S is a balanced separator, it implies that $i_0 < k$. Then, by the very definition of i_0 we also have $|A' \cup C_{i_0}| > 2n/3$. Overall, $|A'| + 2|C_{i_0}| + |B'| > 4n/3$. Since $|A'| + |B'| + |C_{i_0}| < n$, we obtain $|C_{i_0}| > n/3$. We are done by setting $A := A' \cup B'$ and $B := C_{i_0}$. The total runtime is linear. □

Now, set a parameter[4] $\beta := \frac{4+\alpha}{5} < 1$. By Lemma 13, for every n-vertex m-edge graph in $\mathcal{G}_{\alpha,C}$ we can compute a balanced separator of order $\mathcal{O}(n^{\beta})$ in time $\mathcal{O}(n^{1+\beta})$. Following Frederickson [42], we define an *r-division* for an n-vertex graph $G \in \mathcal{G}_{\alpha,C}$ as follows:

[4] More generally, let $\mathcal{G} \subseteq \mathcal{G}_{\alpha,C}$. We may choose any parameter $\beta \in [\alpha; 1)$ such that for all the graphs in $\mathcal{G}$ we can compute a balanced separator of size $\mathcal{O}(n^{\beta})$ in truly subquadratic-time. For instance by Lemma 12, if $\mathcal{G}$ is proper minor-closed then we can set $\beta = \alpha = 1/2$.

- If $n \leq r$ then, we output G;
- Otherwise, let S be a balanced separator of cardinality $\mathcal{O}(n^{\beta})$. Since S is balanced then, by Lemma 14 we can partition the connected components of $G \setminus S$ in two disjoint sets A and B of cardinality $\leq 2n/3$. We end up computing an r-division for the induced subgraphs $G[A \cup S]$ and $G[B \cup S]$ separately. Note that since S is a separator, all edges of G are covered by these two subgraphs.

Therefore by construction, an r-division of a connected graph G is a collection of connected induced subgraphs of order at most r that cover all edges of G. We will use the terminology from [50]. In particular, the subgraphs in an r-division are termed *clusters*. A vertex is *interior* if it is contained in a unique cluster, otherwise it is a *boundary* vertex. Finally, if the sum of the orders of all the clusters is $n + q$ then, we call q the *excess*.

The following result is essentially a reformulation of [50, Lemma 2.2].

Lemma 15 ([50]). *Set $\beta := \frac{4+\alpha}{5}$. There exists a constant r_0 such that, for any n-vertex graph $G \in \mathcal{G}_{\alpha,C}$ and $r \geq r_0$, any r-division of G has an excess in $\mathcal{O}(n/r^{1-\beta})$.*

Note that in our applications, we will choose $r = n^{\gamma}$ for some $\gamma \in (0;1)$ that only depends on β and on the *distance VC-dimension*.

It is easy to prove that an r-division can be computed in polynomial time [50]. Next we use the known connections between strongly sublinear separators and *polynomial expansion* [34] in order to bound the running-time by some truly subquadratic function.

Lemma 16. *Set $\beta := \frac{4+\alpha}{5}$. Then, for any n-vertex m-edge graph $G \in \mathcal{G}_{\alpha,C}$, we can compute an r-division in time $\tilde{\mathcal{O}}(n^{1+\beta})$.*

Proof. We recursively use Lemmata 13 and 14 to split the graph into smaller and smaller clusters. Let us assume that at the initialization step, $n > r$ (otherwise, we are done). We claim that it is sufficient to prove that the total number of edges in the final clusters is in $\mathcal{O}(n)$. Indeed, if this is true for the final clusters then, this is also true for the intermediate clusters at any given step of the decomposition. In particular, every step runs in time $\mathcal{O}(n^{1+\beta})$. Furthermore, since we only consider balanced separators of *sublinear* cardinality, for every n above some constant the two induced subgraphs constructed have sublinear order (say, $\leq 3n/4$). Therefore it takes $\mathcal{O}(\log n)$ steps to decrease the order of all the subgraphs in this collection to less than r. This upper-bound on the number of steps proves, as claimed, that the total running time is in $\tilde{\mathcal{O}}(n^{1+\beta})$.

We are left proving that the total number of edges in the final clusters is indeed in $\mathcal{O}(n)$. For that, let us consider any of the clusters W_i. Since $\mathcal{G}_{\alpha,C}$ is monotone, we have $W_i \in \mathcal{G}_{\alpha,C}$. Furthermore, every graph in $\mathcal{G}_{\alpha,C}$ must be $\mathcal{O}(1)$-degenerate (*e.g.*, see [34, Lemma 2 (b)] where the author proved a stronger result, namely that $\mathcal{G}_{\alpha,C}$ has polynomial expansion). It implies that W_i has size $\mathcal{O}(|V(W_i)|)$. Overall, if the total excess is q then, the total number of edges in the clusters is in $\mathcal{O}(n+q)$. By Lemma 15 we have $q = \mathcal{O}(n)$, and so the total number of edges is also in $\mathcal{O}(n)$. □

Boundary Hypergraphs

Let G be a graph equipped with some r-division, and let $\overrightarrow{\ell} = (\ell_v)_{v \in V}$ be a collection of positive integers that is indexed by the vertex-set of G. Roughly, our objective is to use the r-division in order to compute, for every vertex v, a compact interval representation of its ball of radius ℓ_v. This leads us to the following natural object:

Definition 8. Let Λ_r be an r-division of a graph G, and let $\overrightarrow{\ell} = (\ell_v)_{v \in V}$ be a collection of positive integers that is indexed by the vertex-set of G. The $\overrightarrow{\ell}$*-boundary hypergraph* $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$ has for vertex-set V. Moreover, for every cluster $W_i \in \Lambda_r$ and for every $u, v \in V(W_i)$, if v is a boundary vertex and $dist_G(u,v) < \ell_u$, then the ball $N_G^{\ell_u - dist_G(u,v)}[v]$ is a hyperedge of $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$.

To understand better this above construction, let W_i be a cluster, let $u \in V(W_i)$ be internal and let $z \notin V(W_i)$. Then, since an r-division is also an edge-covering, we have $dist_G(u,z) \leq \ell_u$ if and only if there exists a boundary vertex $v \in V(W_i)$ such that $dist_G(u,v) + dist_G(v,z) \leq \ell_u$. Equivalently, we want to have $z \in N_G^{\ell_u - dist_G(u,v)}[v]$.

Lemma 17. *Let $\beta = \frac{4+\alpha}{5}$. Then, for any n-vertex graph $G \in \mathcal{G}_{\alpha,C}$, and for any r-division Λ_r, the $\overrightarrow{\ell}$-boundary hypergraph $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$ has $\mathcal{O}(nr^\beta)$ hyperedges.*

Proof. For every $W_i \in \Lambda_r$, we create $\mathcal{O}(r \cdot b_i)$ hyperedges, where b_i denotes the number of boundary vertices in the cluster. We observe that the number of boundary nodes is at most the excess and that $\sum_{W_i \in \Lambda_r} b_i$ is at most twice the excess. Then, by Lemma 15 we have $\mathcal{O}(r) \times \mathcal{O}(n/r^{1-\beta}) = \mathcal{O}(nr^\beta)$ hyperedges. □

We stress that by Lemma 17, a boundary hypergraph may have a superlinear number of edges. Therefore, if we restrict ourselves to subquadratic-time computation, we cannot compute this hypergraph explicitly. Fortunately, we show next that this is not needed if one just wants to compute for this hypergraph a spanning path of low stabbing number.

Lemma 18. *Set $\beta := \frac{4+\alpha}{5}$, and let $G \in \mathcal{G}_{\alpha,C}$ have distance VC-dimension at most d. Then, there exists a constant $\varepsilon_d \in (0;1)$ that only depends on d and such that, for any r-division Λ_r, the stabbing number of $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$ is in $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$. Moreover, we can compute a spanning path reaching this upper bound in deterministic time $\tilde{\mathcal{O}}(n^2/r^{1-\beta} + n^{2-\varepsilon_d}r^\beta)$.*

Proof. By construction, $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$ is a subhypergraph of $\mathcal{B}(G)$, the ball hypergraph of G. Therefore, the VC-dimension of $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$ is at most d. Let ε_d be the constant of Theorem 2. In order to prove the result, we are left proving that we can adapt the algorithm of Theorem 2 so that it runs in time $\tilde{\mathcal{O}}(nm/r^{1-\beta} + n^{2-\varepsilon_d}r^\beta)$ when it is given Λ_r and $\overrightarrow{\ell}$ as input. For that, let F be the set of the boundary vertices. We have that $|F|$ is at most the excess, and so, by Lemma 15 we get $|F| = \mathcal{O}(n/r^{1-\beta})$.

1. We start with a breadth-first search from every vertex of F. This pre-processing phase takes time $\mathcal{O}(|F|m) = \mathcal{O}(n^2/r^{1-\beta})$. Furthermore, note that by doing so we can compute all the pairs $(v,t) \in F \times \mathbb{N}$ such that $N_G^t[v]$ is a hyperedge of $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$.

2. Let $\eta = 2^{d+1}\varepsilon_d$. We arbitrarily partition the vertex-set V into subsets $V_1, V_2, \ldots, V_p$ such that $p = \mathcal{O}(n^{1-\eta})$ and, for every $1 \leq i \leq p$, $|V_i| = \mathcal{O}(n^\eta)$. Furthermore, as explained in the proof of Theorem 2 (i.e., Claim 1), we can compute a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ for $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$ if we are given the subhypergraphs $\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_p$ that are induced

by $V_1, V_2, \ldots, V_p$ respectively. It takes time $\tilde{\mathcal{O}}(n^{1+\eta(c(d+1)-1)})$ for some constant c, that is in $\tilde{\mathcal{O}}(n^{2-\varepsilon_d})$.

In order to compute all the subhypergraphs $\mathcal{H}_i$, we could proceed by brute-force, as follows. For every i and for any boundary vertex v, we read the vertices of V_i by non-decreasing distance to v. Furthermore, if $N_G^t[v]$ is a hyperedge of $\mathcal{H}_{\vec{\ell},G}(\Lambda_r)$, then as soon as we exceed distance t all the vertices read so far are exactly $N_G^t[v] \cap V_i$. Overall, for a fixed boundary vertex v we could obtain this way up to $\mathcal{O}(|V_i|)$ different subsets of order $\mathcal{O}(|V_i|)$ each. But unfortunately, that would give us a time complexity in $\mathcal{O}(|F||V_i|^2) = \mathcal{O}(n^{1+2\eta}/r^{1-\beta})$ for a given i, and so a total running time in $\mathcal{O}(n^{2+\eta}/r^{1-\beta})$. In order to lower this running-time, we proceed as follows.

(a) For every $v \in F$, we group all the vertices in V_i at equal distance to v. We totally order this partition by increasing distance of its vertices to v. Doing so we get exactly $n_i := |V_i|$ ordered groups (possibly, by adding some empty groups in the sequence), denoted $V_i^1(v), V_i^2(v), \ldots, V_i^{n_i}(v)$. Overall, this phase takes time $\tilde{\mathcal{O}}(|F||V_i|)$, and so *total* time (for all i) $\tilde{\mathcal{O}}(|F|n) = \tilde{\mathcal{O}}(n^2/r^{1-\beta})$.

(b) Then, we introduce a complex subprocedure in order to gradually remove the duplicates from the sets $N_G^t[v] \cap V_i$, for $v \in F$ and $t \geq 0$. For every $j \in \{0 \ldots n_i\}$, we map every boundary vertex v to $\bigcup_{j' \leq j} V_i^{j'}(v)$. More precisely, we maintain some collection of different subsets of V_i, denoted $\mathcal{P}_j = \left(V_i^{j,1}, V_i^{j,2}, \ldots, V_i^{j,s_i(j)}\right)$ (note that $\mathcal{P}_j$ is a list of lists). For every $v \in F$ we ensure that there is a unique t such that $V_i^{j,t} = \bigcup_{j' \leq j} V_i^{j'}(v)$. Then, there is a pointer from vertex v to this t^{th} subset (equivalently, for every list in $\mathcal{P}_j$, we store an auxiliary list of all the corresponding vertices of F).

We will show next that it is easy to construct $\mathcal{P}_{j+1}$ from $\mathcal{P}_j$, but that the natural method for doing so might generate some duplicates. Roughly, by using in our analysis the Sauer-Shelah-Perles lemma, we prove that it is more efficient to remove duplicates at every single step rather than doing it only once at the end of the subprocedure.

We observe that initially for $j = 0$, there is a unique subset $V_i^{0,1} = \emptyset$. Furthermore if all the subsets $V_i^{j,t}$ have been computed at step j, then we can compute those at step $j+1$, as follows:

- For every $v \in F$, if we have $V_i^{j,t} = \bigcup_{j' \leq j} V_i^{j'}(v)$, then we add a copy of $V_i^{j+1}(v)$ into some buffer $b'_{j+1}(t)$ and a pointer from v to this copy. It takes time $\mathcal{O}(\sum_{v \in F} |V_i^{j+1}(v)|)$.
- Then, for every $1 \leq t \leq s_i(j)$, we remove all the duplicated subsets in the buffer $b'_{j+1}(t)$. The new buffer that we get is denoted $b_{j+1}(t)$. We can compute it by using partition refinement (*e.g.*, see [48] or Lemma 8), that takes time $\mathcal{O}(\sum_{W \in b'_{j+1}(t)} |W|)$ up to some $\mathcal{O}(|V_i|)$-time pre-processing. Overall the removal of all the duplicates, for all t, takes total time $\mathcal{O}(n^\eta + \sum_{v \in F} |V_i^{j+1}(v)|)$. Furthermore on our way to remove the duplicates, we also need to actualize the pointers between the boundary vertices and the buffer contents, that takes additional time $\mathcal{O}(|F|) = \mathcal{O}(n/r^{1-\beta})$.
- For every $1 \leq t \leq s_i(j)$, we can now refine $V_i^{j,t}$ in $|b_{j+1}(t)|$ new subsets. Every such subset has order $\mathcal{O}(n^\eta)$, and so this operation takes total time $\mathcal{O}(n^\eta |b_{j+1}(t)|)$. Overall, we obtain a new collection of $\mathcal{O}(\sum_t |b_{j+1}(t)|)$ subsets. Furthermore, on

our way to construct this collection, we can add a pointer from every boundary vertex v to *one* subset equal to $\bigcup_{j' \leq j+1} V_i^{j'}(v)$ (there may be duplicated subsets). By carefully using the pointers added between the boundary vertices and the buffer contents during the previous phases, this operation takes additional time $\mathcal{O}(|F|) = \mathcal{O}(n/r^{1-\beta})$.

- Finally, since all the subsets in the new collection have order $\mathcal{O}(n^\eta)$, by using again partition refinement we can merge all the duplicated subsets in time

$$\mathcal{O}(|V_i| + n^\eta \cdot \sum_t |b_{j+1}(t)|) = \mathcal{O}(n^\eta \cdot \sum_t |b_{j+1}(t)|).$$

We also need to actualize the pointers between the boundary vertices and the subsets, that takes total time $\mathcal{O}(|F|) = \mathcal{O}(n/r^{1-\beta})$.

Let us upper bound $s_i(j)$. For that we stress that every subset $V_i^{j,t}$ represents a different intersection of V_i with a ball of G, hence of a hyperedge of $\mathcal{B}(G)$. Since $\mathcal{B}(G)$ has VC-dimension at most d, by Lemma 3 so does its subhypergraph $\mathcal{H}_i'$ induced by V_i. In particular, every $V_i^{j,t}$ is a hyperedge of $\mathcal{H}_i'$. By Lemma 1 we get that $s_i(j) = \mathcal{O}(n^{\eta d})$. In the same way, since for a fixed t the $|b_{j+1}(t)|$ new subsets that are obtained by refinement of $V_i^{j,t}$ are pairwise different, we have $|b_{j+1}(t)| \leq s_i(j+1) = \mathcal{O}(n^{\eta d})$. As a result, the passing from step j to step $j+1$ takes time:

$$\mathcal{O}\left(\left[\sum_{v \in F} \left|V_i^{j+1}(t)\right|\right] + n/r^{1-\beta} + n^\eta \cdot n^{\eta d} \cdot n^{\eta d}\right) = \mathcal{O}\left(\left[\sum_{v \in F} \left|V_i^{j+1}(t)\right|\right] + n/r^{1-\beta} + n^{(2d+1)\eta}\right).$$

There are $\mathcal{O}(n^\eta)$ loops, that gives us a total running time of:

$$\begin{aligned}\mathcal{O}\left(\left[\sum_{v \in F} \sum_{j=0}^{n_i - 1} \left|V_i^{j+1}(t)\right|\right] + n^{1+\eta}/r^{1-\beta} + n^{(2d+2)\eta}\right) &= \mathcal{O}\left(\left[\sum_{v \in F} n^\eta\right] + n^{1+\eta}/r^{1-\beta} + n^{2(d+1)\eta}\right) \\ &= \mathcal{O}\left(n^{1+\eta}/r^{1-\beta} + n^{2(d+1)\eta}\right).\end{aligned}$$

(c) Here the key observation is that $\bigcup_j \mathcal{P}_j$ contains the intersection with V_i of *all* the balls whose center is in F. We so computed a *superset* of order $\mathcal{O}(n^{(d+1)\eta})$ (i.e., $\mathcal{O}(n^{\eta d})$ per loop) that contains all possible intersections between a hyperedge of $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$ and V_i. Since every subset in $\bigcup_j \mathcal{P}_j$ represents the intersection of a hyperedge of $\mathcal{B}(G)$ with V_i, and furthermore $\mathcal{B}(G)$ has VC-dimension at most d, then for simplicity we may replace $\mathcal{H}_i$ by the slightly larger hypergraph $\mathcal{H}_i'$ of which these are the hyperedges (i.e., the hyperedges of $\mathcal{H}_i'$ are the intersections of V_i with all the balls whose center is in F). Note that in order to compute $\mathcal{H}_i'$, it is sufficient to eliminate all the duplicated elements in this collection $\bigcup_j \mathcal{P}_j$, that takes total time $\mathcal{O}(n^{(d+2)\eta})$.

The running-time is in $\tilde{\mathcal{O}}(n^{1+\eta}/r^{1-\beta} + n^{2(d+1)\eta})$ for any fixed i. Therefore, the total running-time is in $\tilde{\mathcal{O}}(n^2/r^{1-\beta} + n^{1+[2(d+1)-1]\eta})$. Recall (see Theorem 2 and its proof) that we have $\tilde{\mathcal{O}}(n^{1+\eta(c(d+1)-1)}) = \tilde{\mathcal{O}}(n^{2-\varepsilon_d})$ for some constant $c > 2$. As a result, the running-time of this part is also in $\tilde{\mathcal{O}}(n^2/r^{1-\beta} + n^{2-\varepsilon_d})$.

3. By continuing the algorithm of Theorem 2 with the hypergraphs $\mathcal{H}'_1, \mathcal{H}'_2, \ldots, \mathcal{H}'_p$, we get a spanning path of $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$ whose stabbing number is in $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$. It remains to compute, for every hyperedge of $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$, the set of the stabbed edges. For that, let $v \in F$ be fixed. We add all the radii t such that $N^t_G[v]$ is a hyperedge of $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$ in a balanced binary research tree T_v. Then, we scan all the edges xy of the spanning path. By symmetry let us assume that $dist_G(v,x) \leq dist_G(v,y)$. The edge xy is stabbed by all the hyperedges $N^t_G[v]$ such that $dist_G(v,x) \leq t < dist_G(v,y)$. Then by using T_v, after some pre-computation in time $\mathcal{O}(\log n)$ every value t in the range $[dist_G(v,x); dist_G(v,y))$ can be enumerated in constant-time.

   Overall, by Lemma 17 there are $\mathcal{O}(nr^\beta)$ hyperedges, and so the construction of all the balanced binary research trees takes time $\tilde{\mathcal{O}}(nr^\beta)$. Scanning all the edges, for every boundary vertex, takes total time $\tilde{\mathcal{O}}(n^2/r^{1-\beta})$. Any other operation corresponds to an edge of the spanning path that is stabbed by a hyperedge of $\mathcal{H}_{\ell,G}(\Lambda_r)$, and as a result there can only be $\tilde{\mathcal{O}}(n^{1-\varepsilon_d}) \times \mathcal{O}(nr^\beta) = \tilde{\mathcal{O}}(n^{2-\varepsilon_d} r^\beta)$ such operations.

Altogether combined, the running time of the algorithm is in $\tilde{\mathcal{O}}(n^2/r^{1-\beta} + n^{2-\varepsilon_d} r^\beta)$. ◻

The algorithm

We are now ready to prove the main result of this section.

Proof of Theorem 6. By a classical dichotomic argument it is sufficient to prove that for any $\overrightarrow{\ell} = (\ell_v)_{v \in V}$, we can decide whether $\forall v \in V,\ ecc_G(v) \leq \ell_v$ in truly subquadratic time (*i.e.*, we perform n simultaneous binary searches in order to compute all the eccentricities). Furthermore, we claim that in order to solve this decision problem, we are left with computing a spanning path of strongly subquadratic *total* stabbing number for the hypergraph $\mathcal{E}_{\overrightarrow{\ell}}(G) := (V, \{N^{\ell_v}_G[v] \mid v \in V\})$. More precisely, we claim that it is sufficient to compute a spanning path $P_{\overrightarrow{\ell}}$ for the latter, along with a collection $(I_{P_{\overrightarrow{\ell}}}(v))_{v \in V}$ such that, for every vertex v, $I_{P_{\overrightarrow{\ell}}}(v)$ is a set of (possibly intersecting and/or overlapping) intervals of $P_{\overrightarrow{\ell}}$ whose union equals $N^{\ell_v}[v]$, and furthermore $\sum_{v \in V} |I_{P_{\overrightarrow{\ell}}}(v)|$ is strongly subquadratic in n. Indeed, with essentially the same proof as for Lemma 6, then we can solve our decision problem, for every vertex separately, in time $\tilde{\mathcal{O}}\left(\sum_{v \in V} |I_{P_{\overrightarrow{\ell}}}(v)|\right)$. Let C and $\alpha < 1$ be such that $\mathcal{G} \subseteq \mathcal{G}_{\alpha,C}$ and set $\beta := \frac{4+\alpha}{5} < 1$. We first prove the following intermediate result for any value $r > 0$.

Claim 3. *In $\tilde{\mathcal{O}}(nr + n^2/r^{1-\beta})$ time, we can compute a spanning path $P_{\overrightarrow{\ell}}$ for $\mathcal{E}_{\overrightarrow{\ell}}(G)$, such that $\sum_{v \in V} |I_{P_{\overrightarrow{\ell}}}(v)| = \tilde{\mathcal{O}}(n(r + n^{1-\varepsilon_d} r^\beta))$.*

Proof. By Lemma 16 we can compute an r-division, denoted Λ_r, in time $\tilde{\mathcal{O}}(n^{1+\beta}) = \tilde{\mathcal{O}}(n^2/r^{1-\beta})$. Then, we proceed as follows.

1. We first consider all the clusters $W \in \Lambda_r$ sequentially. For every $x \in W$, we compute a breadth-first-search from x in $G[W]$, the subgraph induced by W. It takes time $\mathcal{O}(r)$ per vertex. Furthermore by Lemma 15 we have $\sum_{W \in \Lambda_r} |W| = \Theta(n)$, and so this step takes time $\sum_{W \in \Lambda_r} \mathcal{O}(|W|^2) = \mathcal{O}(r) \times \sum_{W \in \Lambda_r} |W| = \mathcal{O}(rn)$. Overall for every $u \in V$, we computed all the vertices $v \in N^{\ell_u}_G[u]$ such that at least one uv-path of length $\leq \ell_u$ is fully contained in a cluster.

2. Let us now consider the $\overrightarrow{\ell}$-neighbourhood hypergraph $\mathcal{H}_{\overrightarrow{\ell},G}(\Lambda_r)$. By Lemma 18 we can compute a spanning path $P_{\overrightarrow{\ell}}$ of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ for this hypergraph, in time $\tilde{\mathcal{O}}(nm/r^{1-\beta} + n^{2-\varepsilon_d}r^{\beta}) = \tilde{\mathcal{O}}(n^2/r^{1-\beta} + n^{2-\varepsilon_d}r^{\beta})$. Let $u \in V$. There are two cases:

   - *Case u is a boundary vertex.* Since $N_G^{\ell_u}[u]$ is a hyperedge of the boundary hypergraph, we have $|I_{P_{\overrightarrow{\ell}}}(u)| = \tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ (already computed).
   - *Case u is an internal vertex.* Let $W \in \Lambda_r$ be the unique cluster containing u, and set initially $I_{P_{\overrightarrow{\ell}}}(u) := \emptyset$. For every boundary vertex $v \in V(W)$, if $dist_G(u,v) < \ell_u$ then, we add all intervals corresponding to $N_G^{\ell_u - dist_G(u,v)}[v]$ to $I_{P_{\overrightarrow{\ell}}}(u)$. Assuming there are b_W boundary vertices in W, we obtain that $|I_{P_{\overrightarrow{\ell}}}(u)| = \tilde{\mathcal{O}}(b_W \cdot n^{1-\varepsilon_d})$. Furthermore, this above set of intervals covers exactly the balls $N_G^{\ell_u - dist_G(u,v)}[v]$, for the boundary vertices $v \in V(W)$. By construction, every vertex that is contained in one of these balls, defined above, is at a distance $\leq \ell_u$ to u; conversely, since Λ_r is also an edge-covering, every vertex of $N_G^{\ell_u}[u] \backslash N_{G[W]}^{\ell_u}[u]$ must be in one of these balls. As a result, in order to construct $I_{P_{\overrightarrow{\ell}}}(u)$, it suffices to update this set using the vertices of $N_{G[W]}^{\ell_u}[u]$ (already computed during the first step). Note that by doing so, we can only modify the cardinality of $I_{P_{\overrightarrow{\ell}}}(u)$ by an $\mathcal{O}(|W|) = \mathcal{O}(r)$.

   Overall, we obtain that $\sum_{u \in V} |I_{P_{\overrightarrow{\ell}}}(u)| = \tilde{\mathcal{O}}(nr + n^{1-\varepsilon_d} \cdot \sum_W (b_W \cdot |V(W)|)) = \tilde{\mathcal{O}}(nr + rn^{1-\varepsilon_d} \cdot \sum_W b_W)$. Again we observe that $\sum_W b_W$ is at most twice the excess, and so by Lemma 15 $\sum_W b_W = \mathcal{O}(n/r^{1-\beta})$. Therefore, $\sum_{u \in V} |I_{P_{\overrightarrow{\ell}}}(u)| = \tilde{\mathcal{O}}(n(r + n^{1-\varepsilon_d} r^{\beta}))$.

◇

Overall, the running-time of our algorithm is optimized when we have $n^2/r^{1-\beta} = n^{2-\varepsilon_d}r^{\beta}$. As a result, a good choice is $r = \Theta(n^{\varepsilon_d})$. Finally, we stress that in this case, the running time is in $\tilde{\mathcal{O}}(n^{2-(1-\beta)\cdot\varepsilon_d})$, that is truly subquadratic because $\beta < 1$. ☐

5.3 Extension to weighted graphs

Finally, we study whether some of our results can also be applied to weighted graphs. By a weighted graph, here we mean a pair (G, w) where $G = (V, E)$ is an unweighted graph and $w : E \to \mathbb{R}^+$ assigns a positive weight to every edge. The distance $dist_{G,w}(u,v)$ between two vertices u, v is the least weight of a uv-path in G. For any non-negative real r and any vertex v, we define similarly as before $N_{G,w}^r[v] = \{u \in V \mid dist_{G,w}(u,v) \leq r\}$. Finally, let $\mathcal{B}_w(G) = \{N_{G,w}^r[v] \mid v \in V, r \geq 0\}$ be the ball hypergraph of (G,w). Note that $\mathcal{B}_w(G)$ is finite since G also is. We define the *distance VC-dimension* of (G,w) as the VC-dimension of $\mathcal{B}_w(G)$.

Lemma 19. *Let $\mathcal{G}$ be a class of* unweighted *graphs that is closed under edge-subdivisions. If every graph in $\mathcal{G}$ has distance VC-dimension at most d, then every weighted graph (G,w) such that $G \in \mathcal{G}$ also has distance VC-dimension at most d.*

Proof. We first transform all the edge-weights in rational numbers, then in integers. Specifically, fix $\varepsilon > 0$, and for every edge e, replace w_e by a rational number w'_e such that $|w_e - w'_e| < \frac{\varepsilon}{n}$.

Note that doing so, we have for every pair u, v of vertices: $|dist_{G,w}(u,v) - dist_{G,w'}(u,v)| < \varepsilon$. In particular, for a small enough ε, we will have:

$$\forall u,v,x,y \in V(G) \begin{cases} |dist_{G,w'}(u,v) - dist_{G,w'}(x,y)| < 2\varepsilon \text{ if } dist_{G,w}(u,v) = dist_{G,w}(x,y) \\ |dist_{G,w'}(u,v) - dist_{G,w'}(x,y)| \geq 2\varepsilon \text{ otherwise.} \end{cases}$$

As a result, every ball in $\mathcal{B}_w(G)$ is a ball in $\mathcal{B}_{w'}(G)$, *i.e.*, $\mathcal{B}_w(G) \subseteq \mathcal{B}_{w'}(G)$. So, we assume from now on that all the edge-weights are rational numbers. By multiplying all the edge-weights by a sufficiently large integer, we may further assume that all the edge-weights are positive integers. Under this above assumption, we may replace every edge $e \in E(G)$ by a path of length w_e. Doing so, we get an unweighted graph G_w such that $V(G) \subseteq V(G_w)$ and, for every $u,v \in V(G)$, $dist_{G,w}(u,v) = dist_{G_w}(u,v)$. By the hypothesis, $G_w \in \mathcal{G}$, and therefore it has *distance VC-dimension* at most d. We are done as $\mathcal{B}_w(G)$ is a partial sub-hypergraph of $\mathcal{B}(G_w)$ (*i.e.*, the ball hypergraph of G_w, as it was defined in Sec. 2.2). □

Now, let us consider the framework introduced in Theorem 6. Given $\overrightarrow{\ell} = (\ell_v)_{v \in V}$, we want to decide whether $\forall v \in V,\ ecc(v) \leq \ell_v$. For that, in the algorithm that we proposed for Theorem 6, we need to compute an appropriate r-division. We also need to compute shortest-path trees from different source vertices, which for weighted graphs can be done in quasi linear time by using Dijkstra's algorithm. Correctness of this algorithm only follows from the boundedness of the VC-dimension for the ball hypergraph. So, in particular, under this same condition, we may apply our algorithm to *weighted* graphs, and for an arbitrary collection of positive *real numbers*. However, in order to compute the exact value of the eccentricities, we need to apply this algorithm for different values of $\overrightarrow{\ell}$. More precisely:

- If all the edge-weights are positive integers bounded by M, then the eccentricities must be between 1 and Mn. We compute the exact value of the eccentricities with n simultaneous binary searches, that induces an $\mathcal{O}(\log{(nM)})$ overhead in the total running time.

- If now all the edge-weights are positive real numbers, then the range of possible eccentricities for each vertex is too large and we cannot perform a binary search directly. We compute the eccentricity of an arbitrary vertex, which we denote by ℓ_0. By the triangular inequality, every vertex has its eccentricity between $\ell_0/2$ and $2\ell_0$. Then, we only consider in this interval $[\ell_0/2; 2\ell_0]$ the powers of $1+\varepsilon$, where $\varepsilon > 0$ is an arbitrarily small precision parameter. We observe that the number of distinct powers of $1+\varepsilon$ between these two values is in $\mathcal{O}(\varepsilon^{-1})$. As a result, we can compute an $(1+\varepsilon)$-approximation of all the eccentricities by using binary search, that induces an $\mathcal{O}(\log{(1/\varepsilon)})$ overhead in the running time.

Summarizing, we get:

Theorem 9. *Let $\mathcal{G}$ be a monotone graph class with strongly sublinear balanced separators, that is stable under edge-subdivisions and such that all the graphs in $\mathcal{G}$ have distance VC-dimension at most d. Then for some constant $\varepsilon_{\mathcal{G}}(d) \in (0;1)$ that only depends on d, for any weighted graph (G,w) such that $G \in \mathcal{G}$ we can compute:*

- *the exact value of all the eccentricities, in deterministic time $\tilde{\mathcal{O}}(n^{2-\varepsilon_{\mathcal{G}}(d)} \log M)$, if all the edge-weights are integers bounded by M;*

- *or, for any $\varepsilon > 0$, an $(1+\varepsilon)$-approximation of all the eccentricities, in deterministic time $\tilde{\mathcal{O}}(n^{2-\varepsilon_{\mathcal{G}}(d)} \log(1/\varepsilon))$.*

Finally, we observe that all the required conditions for the graph class $\mathcal{G}$ in Theorem 9 hold for the proper minor-closed graph classes:

Corollary 2. *For every weighted H-minor free graph, for some constant $\varepsilon_H \in (0;1)$ that only depends on H, we can compute:*

- *the exact value of all the eccentricities, in deterministic time $\tilde{\mathcal{O}}(n^{2-\varepsilon_H} \log M)$, if all the edge-weights are integers bounded by M;*
- *or, for any $\varepsilon > 0$, an $(1+\varepsilon)$-approximation of all the eccentricities, in deterministic time $\tilde{\mathcal{O}}(n^{2-\varepsilon_H} \log(1/\varepsilon))$.*

6 Open Problems

We left open whether we can compute the diameter of all the graphs of constant *distance VC-dimension* in truly subquadratic time. In order to prove that it is the case, we stress that by our Theorem 3 we only need to consider the graphs of large diameter, i.e., above some polynomial.

Furthermore, we observe that there exist graph families of *unbounded* (distance) VC-dimension for which we can compute the diameter very efficiently. For instance, recall that a vertex is universal if its closed neighbourhood contains all vertices. If we add a universal vertex to a graph G, thus getting a supergraph G' with one more (universal) vertex, then any subset shattered by $\mathcal{N}_1(G)$ is also shattered by $\mathcal{N}_1(G')$, and therefore the VC-dimension of G' is at least the one of G. It implies that the class of all the graphs with a universal vertex has unbounded VC-dimension. Clearly, we can compute the diameter of any graph with a universal vertex in linear time. Even more strongly, such graphs are a particular case of *dually chordal* graphs, for which we also know how to compute the diameter in linear time [13]. We observe that the ball hypergraphs of dually chordal graphs also admit some nice characterizations. Thus, it would be very interesting to study whether a truly subquadratic algorithm for computing the diameter could be derived from some common property of dually chordal graphs and graphs of constant *distance VC-dimension* (say, a bounded fractional Helly number [60]).

Finally, it would be interesting to study whether we can solve other distance problems using our techniques in this paper. For instance, the *Wiener index* of a graph is the sum of all its distances. In [17], Cabello also presented the first truly subquadratic algorithm for computing the Wiener index on planar graphs, using the same techniques based on Voronoi diagrams as for diameter computation. For the graphs of constant *distance VC-dimension* and constant diameter, we can slightly modify the proof of Theorem 3 in order to also compute their Wiener index in truly subquadratic time. Indeed, this is because we compute a spanning path of low stabbing number for *every* k-neighbourhood hypergraph (from $k = 1$ up to the diameter value). Doing so, we can compute the distance distribution of the graph (*i.e.*, the number of pairs of vertices at distance i, for any i), and therefore, also the Wiener index. However, for the proper minor-closed graph classes, we currently do not see any way to extend our approach in Theorem 6 in order to also compute their Wiener index in truly subquadratic time. The fine-grained complexity of computing the Wiener index within proper minor-closed graph classes is left as an intriguing open question.

Acknowledgements

This work was supported by Inria Gang project-team, by Irif laboratory from CNRS and Paris University, and by the ANR projects DISTANCIA (ANR-17-CE40-0015) and Multimod (ANR-17-CE22-0016). This work was also supported by project PN 19 37 04 01 "New solutions for complex problems in current ICT research fields based on modelling and optimization", funded by the Romanian Core Program of the Ministry of Research and Innovation (MCI) 2019-2022, and by a grant of Romanian Ministry of Research and Innovation CCCDI-UEFISCDI, project no. 17PCCDI/2018.

References

[1] A. Abboud, V. Vassilevska Williams, and J. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *SODA*, pages 377–391. SIAM, 2016.

[2] N. Alon, P. Seymour, and R. Thomas. A separator theorem for nonplanar graphs. *Journal of the American Mathematical Society*, 3(4):801–808, 1990.

[3] M. Anthony, G. Brightwell, and C. Cooper. The Vapnik-Chervonenkis dimension of a random graph. *Discrete Mathematics*, 138(1-3):43–56, 1995.

[4] A. Backurs, L. Roditty, G. Segal, V. Vassilevska Williams, and N. Wein. Towards tight approximation bounds for graph diameter and eccentricities. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC 2018)*, pages 267–280. ACM, 2018.

[5] C. Berge. *Graphs and hypergraphs.* North-Holland Pub. Co., 1973.

[6] V. Bilo, V. Goyal, R. Ravi, and M. Singh. On the crossing spanning tree problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 51–60. Springer, 2004.

[7] B. Bollobás, D. Coppersmith, and M. Elkin. Sparse distance preservers and additive spanners. *SIAM Journal on Discrete Mathematics*, 19(4):1029–1055, 2005.

[8] J. A. Bondy and U. S. R. Murty. *Graph theory.* Springer, 2008.

[9] M. Borassi, P. Crescenzi, and M. Habib. Into the square: On the complexity of some quadratic-time solvable problems. *Electronic Notes in Theoretical Computer Science*, 322:51–67, 2016.

[10] N. Bousquet, A. Lagoutte, Z. Li, A. Parreau, and S. Thomassé. Identifying codes in hereditary classes of graphs and VC-dimension. *SIAM Journal on Discrete Mathematics*, 29(4):2047–2064, 2015.

[11] N. Bousquet and S. Thomassé. VC-dimension and Erdős–Pósa property. *Discrete Mathematics*, 338(12):2302–2317, 2015.

[12] A. Brandstädt, V. Chepoi, and F. Dragan. The algorithmic use of hypertree structure and maximum neighbourhood orderings. *Discrete Applied Mathematics*, 82(1-3):43–77, 1998.

[13] A. Brandstädt, F. Dragan, V. Chepoi, and V. Voloshin. Dually chordal graphs. *SIAM Journal on Discrete Mathematics*, 11(3):437–455, 1998.

[14] K. Bringmann, T. Husfeldt, and M. Magnusson. Multivariate analysis of orthogonal range searching and graph distances parameterized by treewidth. In *IPEC*, 2018.

[15] K. Bringmann, T. Husfeldt, and M. Magnusson. Multivariate analysis of orthogonal range searching and graph distances. *Algorithmica*, 82(8):2292–2315, 2020.

[16] H. Brönnimann, B. Chazelle, and J. Matouvsek. Product range spaces, sensitive sampling, and derandomization. *SIAM J. Comput.*, 28(5):1552–1575, 1999.

[17] S. Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. *ACM Transactions on Algorithms (TALG)*, 15(2):21, 2018.

[18] S. Cabello and C. Knauer. Algorithms for graphs of bounded treewidth via orthogonal range searching. *Computational Geometry*, 42(9):815–824, 2009.

[19] M. Cairo, R. Grossi, and R. Rizzi. New bounds for approximating extremal distances in undirected graphs. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms (SODA 2016)*, pages 363–376. SIAM, 2016.

[20] T. Chan. Optimal partition trees. *Discrete & Computational Geometry*, 47(4):661–690, 2012.

[21] B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete & Computational Geometry*, 4(5):467–489, 1989.

[22] S. Chechik, D. Larkin, L. Roditty, G. Schoenebeck, R. Tarjan, and V. Vassilevska Williams. Better approximation algorithms for the graph diameter. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms (SODA 2014)*, pages 1041–1052. SIAM, 2014.

[23] V. Chepoi, B. Estellon, and Y. Vaxès. Covering planar graphs with a fixed number of balls. *Discrete & Computational Geometry*, 37(2):237–244, 2007.

[24] M. Cohen, Y. Lee, and Z. Song. Solving linear programs in the current matrix multiplication time. In *STOC*, pages 938–942. ACM, 2019.

[25] V. Cohen-Addad, S. Dahlgaard, and C. Wulff-Nilsen. Fast and compact exact distance oracle for planar graphs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 962–973. IEEE, 2017.

[26] D. Corneil, F. Dragan, M. Habib, and C. Paul. Diameter determination on restricted graph families. *Discrete Applied Mathematics*, 113(2-3):143–166, 2001.

[27] D. Coudert, G. Ducoffe, and A. Popa. Fully polynomial FPT algorithms for some classes of bounded clique-width graphs. In *SODA'18*, pages 2765–2784. SIAM, 2018.

[28] S. Dahlgaard. On the hardness of partially dynamic graph problems and connections to diameter. In *43rd International Colloquium on Automata, Languages, and Programming, (ICALP 2016)*, pages 48:1–48:14, 2016.

[29] P. Damaschke. Computing giant graph diameters. In *IWOCA*, pages 373–384. Springer, 2016.

[30] R. Downey, P. Evans, and M. Fellows. Parameterized learning complexity. In *COLT*, pages 51–57. ACM, 1993.

[31] G. Ducoffe. A New Application of Orthogonal Range Searching for Computing Giant Graph Diameters. In *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*, 2019.

[32] G. Ducoffe, M. Habib, and L. Viennot. Fast diameter computation within split graphs. In *International Conference on Combinatorial Optimization and Applications (COCOA)*, pages 155–167. Springer, 2019.

[33] G. Ducoffe, M. Habib, and L. Viennot. Diameter computation on H-minor free graphs and graphs of bounded (distance) VC-dimension. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1905–1922. SIAM, 2020.

[34] Z. Dvořák. On classes of graphs with strongly sublinear separators. *European Journal of Combinatorics*, 71:1–11, 2018.

[35] Z. Dvorak and S. Norin. Strongly sublinear separators and polynomial expansion. *SIAM Journal on Discrete Mathematics*, 30(2):1095–1101, 2016.

[36] Z. Dvořák and S. Norin. Treewidth of graphs with balanced separations. *Journal of Combinatorial Theory, Series B*, 137:137–144, 2019.

[37] D. Eisenstat and D. Angluin. The VC dimension of k-fold union. *Information Processing Letters*, 101(5):181–184, 2007.

[38] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3-4):275–291, 2000.

[39] D. Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003, Baltimore, Maryland, USA.*, pages 599–608. ACM/SIAM, 2003.

[40] J. Evald and S. Dahlgaard. Tight Hardness Results for Distance and Centrality Problems in Constant Degree Graphs. Technical Report arXiv:1609.08403, ArXiv, 2016.

[41] A. Farley and A. Proskurowski. Computation of the center and diameter of outerplanar graphs. *Discrete Applied Mathematics*, 2(3):185–191, 1980.

[42] G. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM Journal on Computing*, 16(6):1004–1022, 1987.

[43] C. Gavoille, D. Peleg, S. Pérennes, and R. Raz. Distance labeling in graphs. *Journal of Algorithms*, 53(1):85–112, 2004.

[44] P. Gawrychowski, H. Kaplan, S. Mozes, M. Sharir, and O. Weimann. Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{O}(n^{5/3})$ time. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*, pages 495–514. SIAM, 2018.

[45] P. Gawrychowski, H. Kaplan, S. Mozes, M. Sharir, and O. Weimann. Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{o}(n^{5/3})$ time. *SIAM J. Comput.*, 50(2):509–554, 2021.

[46] P. Gawrychowski, S. Mozes, O. Weimann, and C. Wulff-Nilsen. Better tradeoffs for exact distance oracles in planar graphs. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 515–529. SIAM, 2018.

[47] M. Grohe, S. Kreutzer, and S. Siebertz. Deciding first-order properties of nowhere dense graphs. *Journal of the ACM (JACM)*, 64(3):1–32, 2017.

[48] M. Habib, R. McConnell, C. Paul, and L. Viennot. Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theoretical Computer Science*, 234(1-2):59–84, 2000.

[49] S. Har-Peled. Approximating Spanning Trees with Low Crossing Number. Technical Report abs/0907.1131, arXiv, 2009.

[50] S. Har-Peled and K. Quanrud. Approximation algorithms for polynomial-expansion and low-density graphs. *SIAM Journal on Computing*, 46(6):1712–1744, 2017.

[51] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete & Computational Geometry*, 2(2):127–151, 1987.

[52] M. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for planar graphs. *Journal of computer and system sciences*, 55(1):3–23, 1997.

[53] K. Kawarabayashi, P. N. Klein, and C. Sommer. Linear-space approximate distance oracles for planar, bounded-genus and minor-free graphs. In L. Aceto, M. Henzinger, and J. Sgall, editors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, volume 6755 of *Lecture Notes in Computer Science*, pages 135–146. Springer, 2011.

[54] K. Kawarabayashi and B. Reed. A separator theorem in minor-closed classes. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 153–162. IEEE, 2010.

[55] J. Kleinberg. Detecting a network failure. *Internet Mathematics*, 1(1):37–55, 2004.

[56] E. Kranakis, D. Krizanc, B. Ruf, J. Urrutia, and G. Woeginger. The VC-dimension of set systems defined by graphs. *Discrete Applied Mathematics*, 77(3):237–257, 1997.

[57] J. Li and M. Parter. Planar diameter via metric compression. In *STOC*, pages 152–163. ACM, 2019.

[58] R. Lipton and R. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

[59] J. Matoušek. Spanning trees with low crossing number. *RAIRO-Theoretical Informatics and Applications-Informatique Théorique et Applications*, 25(2):103–123, 1991.

[60] J. Matoušek. Bounded VC-dimension implies a fractional Helly theorem. *Discrete & Computational Geometry*, 31(2):251–255, 2004.

[61] J. Nešetřil and P. Ossona de Mendez. *Sparsity: graphs, structures, and algorithms*, volume 28. Springer Science & Business Media, 2012.

[62] J. Nešetřil and P. Ossona de Mendez. Structural sparsity. *Russian Mathematical Surveys*, 71(1):79, 2016.

[63] S. Olariu. A simple linear-time algorithm for computing the center of an interval graph. *International Journal of Computer Mathematics*, 34(3-4):121–128, 1990.

[64] R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987.

[65] C. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the VC dimension. *Journal of Computer and System Sciences*, 53(2):161–170, 1996.

[66] S. Plotkin, S. Rao, and W. Smith. Shallow Excluded Minors and Improved Graph Decompositions. In *SODA*, volume 90, pages 462–470, 1994.

[67] L. Roditty and V. Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *STOC*, pages 515–524. ACM, 2013.

[68] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

[69] S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

[70] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

[71] E. Welzl. On spanning trees with low crossing numbers. In *Data structures and efficient algorithms*, pages 233–249. Springer, 1992.

[72] C. Wulff-Nilsen. Separator theorems for minor-free and shallow minor-free graphs with applications. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pages 37–46. IEEE, 2011.