



HAL
open science

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

Guillaume Ducoffe, Michel Habib, Laurent Viennot

► To cite this version:

Guillaume Ducoffe, Michel Habib, Laurent Viennot. Diameter, eccentricities and distance oracle computations on H-minor free graphs and graphs of bounded (distance) VC-dimension. *SIAM Journal on Computing*, 2022, 51 (5), pp.1506-1534. 10.1137/20M136551X . hal-03841015

HAL Id: hal-03841015

<https://hal.science/hal-03841015v1>

Submitted on 6 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

4 Guillaume Ducoffe¹, Michel Habib², and Laurent Viennot³

5 ¹University of Bucharest, Faculty of Mathematics and Computer Science, and National Institute for
6 Research and Development in Informatics, Romania

7 ²Paris University, and IRIF CNRS, France

8 ³Inria, and Paris University, France

9 **Abstract**

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

- 19 • Our first main result is a Monte Carlo algorithm that on graphs of *distance VC-dimension*
20 at most d , for any fixed k , either computes the diameter or concludes that it is larger than
21 k in time $\tilde{O}(k \cdot mn^{1-\varepsilon_d})$, where $\varepsilon_d \in (0; 1)$ only depends on d ¹. We thus obtain a *truly*
22 *subquadratic-time parameterized* algorithm for computing the diameter on such graphs.
- 23 • Then as a byproduct of our approach, we get a truly subquadratic-time randomized algo-
24 rithm for *constant* diameter computation on all the *nowhere dense* graph classes. The latter
25 classes include all proper minor-closed graph classes, bounded-degree graphs and graphs of
26 bounded expansion. Before our work, the only known such algorithm was resulting from
27 an application of Courcelle’s theorem, see Grohe et al. [47].
- 28 • For any graph of constant *distance VC-dimension*, we further prove the existence of an
29 exact distance oracle in truly subquadratic space, that answers distance queries in truly
30 sublinear time (in the number n of vertices). The latter generalizes prior results on proper
31 minor-closed graph classes to a much larger graph class.
- 32 • Finally, we show how to remove the dependency on k for *any* graph class that excludes a
33 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].

¹The $\tilde{O}()$ notation suppresses polylogarithmic factors.

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

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

45 1 Introduction

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

56 The conditional lower-bound of [67] also holds for sparse graphs *i.e.*, with only $m = \mathcal{O}(n)$
57 edges [1]. However it does *not* hold for many well-structured graph classes [1, 12, 15, 17, 26, 27, 29,
58 31, 38, 41, 45, 63]. Our work proposes some new advances on the characterization of graph families
59 for which we can compute the diameter in truly subquadratic time.

60 1.1 Related work

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

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

71 **Bounded-treewidth graphs.** More recently, quasi linear-time algorithms for diameter compu-
72 tation on bounded-treewidth graphs were presented in [1, 14, 15] with almost optimal dependency
73 on the treewidth parameter. The cornerstone of these algorithms is the use of k -range trees in order

74 to detect the furthest pairs that are disconnected by some small-cardinality separators. This tech-
75 nique was first introduced in [18] in order to compute the sum of all distances on bounded-treewidth
76 graphs. Since then a few other applications of k -range trees and, more generally, orthogonal range
77 searching for diameter computation, have been presented in [31, 32]. In our work we uncover deeper
78 connections between diameter computation and range searching techniques from computational ge-
79 ometry.

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

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

106 1.2 Our contributions

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

116 upper-bound on their VC-dimension [56, 23, 32].

117 **First example.** As an appetizer we first consider an n -vertex split graph with clique-number
118 $\log^{\mathcal{O}(1)} n$, that is a notoriously hard case for diameter computation [9]. Given such a split graph
119 G with stable set S and maximal clique K , we can pre-process G in linear-time so as to partition
120 the vertices of S into *twin classes*: with two vertices in S being called twins if and only if they
121 have the same neighbourhood in K (e.g., see [27]). If the VC-dimension of G is at most d then,
122 by the Sauer-Shelah-Perles Lemma [68, 69] the number of twin classes is in $\mathcal{O}(|K|^d) = \log^{\mathcal{O}(d)} n$.
123 Therefore, after some linear-time preprocessing, we are left with computing the diameter on a graph
124 of *polylogarithmic order*. Unfortunately, such simple brute-force arguments are no longer sufficient
125 for split graphs of arbitrary clique-size.

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

131 Assume for now that we are given a spanning path with stabbing number t for the closed
132 neighbourhood hypergraph of G . Then in linear time, we can compute for every vertex v the ends
133 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
134 follows. Then, in order to decide whether G has diameter at most two, it is sufficient to check
135 whether for every vertex u we have $\bigcup_{v \in N_G[u]} I(v) = V(G)$. Since we only need to consider the
136 extremities of such intervals, this verification phase takes time $\tilde{\mathcal{O}}(\deg_G(u) \cdot t)$ for a vertex of degree
137 $\deg_G(u)$, and so, $\tilde{\mathcal{O}}(tm)$ total time. Note that such running-time is always subquadratic if t is
138 sublinear in n . Overall, we reduced the diameter-two problem to the computation of a spanning
139 path with low stabbing number for the closed neighbourhood hypergraph.

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

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

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

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

158 state the following theorem, where the size of a hypergraph is defined as the sum of its hyperedge
 159 cardinalities.

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

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

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

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

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

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

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

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

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

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

210 **Theorem 4.** *Let \mathcal{G} be a class of nowhere dense graphs. There exists a Monte Carlo algorithm such*
 211 *that, for every constant $k = \mathcal{O}(1)$, for any graph in \mathcal{G} we can decide whether its diameter is at most*
 212 *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 .*

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

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

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

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

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

239 **Theorem 6.** *Let \mathcal{G} be a monotone graph class with strongly sublinear balanced separators. Then,*
 240 *for every $d > 0$, for any graph in \mathcal{G} of distance VC-dimension at most d , we can compute all the*
 241 *eccentricities (and so, the diameter) in deterministic time $\tilde{\mathcal{O}}(n^{2-\varepsilon_{\mathcal{G}}(d)})$, for some constant $\varepsilon_{\mathcal{G}}(d) \in$*
 242 *$(0; 1)$ that only depends on d .*

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

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

254 The speed-up of Theorem 6 follows from a faster computation of spanning paths for the neigh-
 255 bourhood hypergraphs. More precisely, our first main insight is that, in order to decide whether the
 256 diameter is at most k , we do not really need to compute a spanning path of low stabbing number for
 257 $\mathcal{N}_{k-1}(G)$. In fact, it is sufficient to compute a suboptimal representation for $\mathcal{N}_k(G)$ that minimizes
 258 what we call the *total* stabbing number (defined as the sum, over all vertices v , of the number of
 259 maximal intervals in the representation whose union is the ball of center v and radius k). Doing
 260 so, we avoid using ε -nets, which makes our algorithms fully deterministic. So the problem becomes
 261 how to compute efficiently these suboptimal representations?

262 For that, we use a rather classical divide-and-conquer approach. Frederickson [42] proved that
 263 a planar graph can be edge-covered with $\mathcal{O}(n/r)$ subgraphs of order at most r such that at most
 264 $\mathcal{O}(\sqrt{r})$ vertices of each subgraph can be contained in another subgraph of this decomposition.
 265 His construction directly follows from the planar separator theorem of Lipton and Tarjan [58],
 266 and as such it can be easily adapted for any monotone graph family with sublinear balanced
 267 separators [52]³. For illustrating our method, we now focus in this introduction on the planar case.
 268 We can first compute, for some well-chosen $r = n^\gamma$, $\gamma \in (0; 1)$, a decomposition as described above.
 269 For every two vertices in a same subgraph, we can check whether they are at distance at most k *in*
 270 *the subgraph*, by solving All-Pairs Shortest-Paths for the latter; this operation takes $\mathcal{O}(r^2)$ time per
 271 subgraph but, assuming r is small enough, this whole phase can be implemented in order to run
 272 in truly subquadratic time. Then for every subgraph of the decomposition, we compute a breadth-
 273 first search from each of the $\mathcal{O}(\sqrt{r})$ boundary vertices that are also contained in another subgraph.
 274 Overall, there can only be $\mathcal{O}(n/\sqrt{r})$ such boundary vertices, and so, it takes truly subquadratic
 275 time. Furthermore by doing so, we computed for every subgraph of the decomposition the $\mathcal{O}(r\sqrt{r})$
 276 distances between the boundary vertices and all the others. For any vertex v that is *not* on the
 277 boundary, we observe that a vertex u can be at a distance $\leq k$ from v if and only if, (i) the unique
 278 subgraph of the decomposition that contains v also contains u and a uv -path of length $\leq k$, or (ii)
 279 $\text{dist}_G(u, x) \leq k - \text{dist}_G(v, x)$ from some vertex x on the boundary ($\mathcal{O}(\sqrt{r})$ balls to be considered).

³Note 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.

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

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

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

297 1.3 Organization of the paper

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

306 2 Preliminaries

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

309 2.1 Graphs and diameter

310 For any undefined graph terminology, see [8]. Throughout all this paper we only consider graphs
 311 that are undirected, unweighted and connected. For every graph $G = (V, E)$, let $n := |V|$ be
 312 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*
 313 *and closed neighbourhoods* of vertex v , respectively. The *degree* of v is equal to $|N_G(v)|$ and is
 314 denoted by $\deg_G(v)$ in what follows. The *length* of a path is its number of edges, and the *distance*
 315 $\text{dist}_G(u, v)$ between $u, v \in V$ is equal to the length of a shortest uv -path. For every $v \in V$ and
 316 $k \geq 0$, the k -*neighbourhood* of v , also known as the ball of center v and radius k , is defined as
 317 $N_G^k[v] = \{u \in V \mid \text{dist}_G(u, v) \leq k\}$. For instance, $N_G^1[v]$ is exactly the closed neighbourhood of

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

Problem 1 (DIAMETER).

Input: A graph $G = (V, E)$.

Output: The diameter of G .

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

323 2.2 Hypergraphs

324 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
 325 *hyperedges*. See also [5] for any undefined hypergraph terminology. Let $n := |X|$, $m := \sum_{q \in R} |q|$
 326 and $r := |R|$ be the *order*, the *size* and the number of hyperedges of \mathcal{H} , respectively. For every
 327 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
 328 $X^* := \{R_x \mid x \in X\}$. In particular, \mathcal{H} and \mathcal{H}^{**} are isomorphic.

329 Several hypergraphs can be related to a graph G :

- 330 • The *closed neighbourhood hypergraph*, denoted by $\mathcal{N}_1(G)$, has vertex-set $X = V$ and hyperedge-
 331 set $R = \{N_G[v] \mid v \in V\}$;
- 332 • More generally, for every fixed $\ell \geq 0$, the ℓ -*neighbourhood hypergraph* of G is defined as
 333 $\mathcal{N}_\ell(G) = (V, \{N_G^\ell[v] \mid v \in V\})$. We stress that $\mathcal{N}_\ell(G)$ and its dual $\mathcal{N}_\ell^*(G)$ are isomorphic [11].
- 334 • Finally, the *ball hypergraph* of G , simply denoted by $\mathcal{B}(G)$, has for hyperedges the balls of all
 335 possible centers and radii in G . Equivalently, $\mathcal{B}(G) = \bigcup_{\ell \geq 0} \mathcal{N}_\ell(G)$.

336 2.3 VC-dimension

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

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

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

345 **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,*
 346 *the VC-dimension of $\mathcal{H}[Y] := (Y, R[Y])$ is at most the VC-dimension of \mathcal{H} .*

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

353 The *distance VC-dimension* of a graph G is defined as the VC-dimension of its ball hypergraph
354 $\mathcal{B}(G)$. Chepoi, Estellon and Vaxès proved in [23] that planar graphs have *distance VC-dimension* at
355 most 4, and remarked that more generally every K_h -minor free graph has *distance VC-dimension* at
356 most $h - 1$. Bousquet and Thomassé proved in [11] that graphs of bounded *distance VC-dimension*
357 also generalize graphs of bounded rankwidth. Indeed, every graph of rankwidth k has *distance*
358 *VC-dimension* at most $3 \cdot 2^{k+1} + 1$. For purpose of illustration, we next adapt a proof from [10] in
359 order to show that interval graphs have *distance VC-dimension* at most two:

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

361 *Proof.* Let $G = (V, E)$ be an interval graph. We fix an interval model for G . For every $v \in V$, let
362 $I(v) = [a_v, b_v]$ be the corresponding interval in the representation. Suppose now by contradiction
363 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
364 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,
365 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
366 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
367 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
368 the open interval between x and y . In fact we must have $I_{k-1}(u) \subseteq]b_{v_2}, \infty[$ because otherwise,
369 $I_{k-1}(u) \cap I(v_3) = \emptyset$ would imply $v_3 \notin N_G^k[u]$, a contradiction. Since $I_{k-1}(u) \cap I(v_1) \neq \emptyset$, it implies
370 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
371 $w \in V$ and $\ell \geq 1$, we have $v_2 \in N_G^\ell[w] \implies v_1 \in N_G^\ell[w]$. The latter contradicts our hypothesis that
372 S is shattered. \square

373 2.4 Stabbing number and applications to DIAMETER

374 A spanning tree of $\mathcal{H} = (X, R)$ is a (classical) tree T whose node-set is exactly X . The *stabbing*
375 *number* of such spanning tree T is the least k such that, for every hyperedge $q \in R$, there exist at
376 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
377 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,
378 the *stabbing number* of \mathcal{H} is the minimum *stabbing number* over its spanning *paths*. Indeed, as
379 noted in [21], every spanning tree T can be transformed into a spanning path of *stabbing number*
380 at most twice bigger than for T . Therefore, there is essentially no loss of generality in restricting
381 ourselves to spanning paths.

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

383 Overall it follows from Lemmata 2 and 5 that any n -vertex hypergraph of VC-dimension at most
384 d has strongly sublinear *stabbing number* in $\tilde{\mathcal{O}}(n^{1-\frac{1}{2d+1}})$. We stress that the proof of Lemma 5 is
385 constructive but that it cannot be transformed into a truly subquadratic-time algorithm. Efficient
386 computations of spanning paths with sublinear *stabbing number* – or related data structures – were
387 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 $\tilde{O}(n^{1-\frac{1}{f(d)}})$ 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 $\tilde{O}(n^{1-\frac{1}{f(d)}})$, 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 $\tilde{O}(n^{1-\frac{1}{f(d)}})$. 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 $\tilde{O}(T(n, m) + mn^{1-\frac{1}{f(d)}})$.*

Proof. Let us first compute a spanning path P of stabbing number at most $\tilde{O}(n^{1-\frac{1}{f(d)}})$ 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 $\mathcal{O}(|E_P(N_G^{k-1}[v])|) = \tilde{O}(n^{1-\frac{1}{f(d)}})$, and so, $\tilde{O}(n^{2-\frac{1}{f(d)}})$ total time. Then in order to decide whether $\text{diam}(G) \leq k$, 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 $\tilde{O}(\text{deg}_G(u) \cdot n^{1-\frac{1}{f(d)}})$ 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 $\tilde{O}(mn^{1-\frac{1}{f(d)}})$. \square

Remark 1. We recall that the *radius* of a graph G is equal to $\text{rad}(G) = \min_{v \in V} \text{ecc}_G(v)$. 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 $\tilde{O}(mn^{1-\varepsilon_d})$ 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 $\tilde{O}(m + n^{2-\varepsilon_d})$ 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 $\tilde{O}(n^{1-\varepsilon_d})$. In particular, this algorithm computes for each hyperedge the ends of its corresponding $\tilde{O}(n^{1-\varepsilon_d})$ 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, \dots, S_p of $\{1, \dots, n\}$, 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, \dots, X_p such that $p = \Theta(n^{1-\eta})$ and, for every $1 \leq i \leq p$, $|X_i| = \mathcal{O}(n^\eta)$. Our aim is to apply Lemma 7 to the induced subhypergraphs $\mathcal{H}[X_1], \mathcal{H}[X_2], \dots, \mathcal{H}[X_p]$. We stress that all these subhypergraphs can be constructed in total $\mathcal{O}(m)$ -time, as follows: we scan all the

462 hyperedges q once in order to compute $(q \cap X_i)_{1 \leq i \leq p}$; then, for every i , we use the linear-time
463 sorting algorithm of Lemma 8 in order to suppress duplicated values in $\{q \cap X_i \mid q \in R\}$. This
464 is indeed linear-time because, for any hyperedge q the sets in $(q \cap X_i)_{1 \leq i \leq p}$ are pairwise disjoint,
465 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
466 non-empty sub-hyperedges $q \cap X_i$, for $1 \leq i \leq p$.

467 **Claim 1.** *Given $\mathcal{H}[X_1], \mathcal{H}[X_2], \dots, \mathcal{H}[X_p]$, we can compute a spanning path for \mathcal{H} of stabbing
468 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 $c > 2$.*

469 *Proof.* By Lemma 3, every $\mathcal{H}[X_i]$ has VC-dimension at most d . This implies that $\mathcal{H}[X_i]$ has $\mathcal{O}(n^{\eta d})$
470 hyperedges (Lemma 1), and so it has size $\mathcal{O}(n^{\eta(d+1)})$. Furthermore by Lemma 7 we can compute
471 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
472 some universal constant c .

473 Let P_1, P_2, \dots, P_p be the spanning paths that we obtain. We obtain a spanning path P
474 for \mathcal{H} by concatenating all the P_i 's. For every $1 \leq i \leq p$, we recall that the stabbing num-
475 ber 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
476 $\tilde{\mathcal{O}}\left(p \cdot n^{\eta\left(1-\frac{1}{2^{d+1}}\right)} + p - 1\right) = \tilde{\mathcal{O}}\left(n^{1-\frac{\eta}{2^{d+1}}}\right)$. ◊

477
478 Let P be the spanning path obtained with Claim 1. Finally, for every $q \in R$, we compute the
479 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
480 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.
481 $s \in R_v \setminus R_u$), we add uv to $E_P(q)$ (resp. $E_P(s)$). Note that for the above, we only need to scan twice
482 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,
483 we achieve a good trade-off between running-time and approximation factor if we have $2 - \frac{\eta}{2^{d+1}} =$
484 $1 + \eta(c(d+1) - 1)$. Therefore we set $\eta = \frac{1}{c(d+1) + \frac{1}{2^{d+1}} - 1}$, and then $\varepsilon_d = \frac{\eta}{2^{d+1}} = \frac{1}{2^{d+1}(c(d+1)-1)+1}$.
485 □

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

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

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

492 4 Bounded DIAMETER with ε -Nets

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

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

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

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

502 We will also need the following result:

503 **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*
 504 *symmetric differences between hyperedges. If \mathcal{H} has VC-dimension at most d then, $\hat{\mathcal{H}} := (X, \hat{R})$*
 505 *has bounded VC-dimension.*

506 We observe that no explicit upper bound on the VC-dimension of $\hat{\mathcal{H}}$ was stated in [21]. Never-
 507 theless it can be easily deduced from their proof that it is in $\mathcal{O}(d \log d)$ (see also [37]).

508 The following partition lemma is the cornerstone of our algorithm.

509 **Lemma 11.** *Let $G = (V, E)$ be a graph of distance VC-dimension at most d , and let S be any*
 510 *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*
 511 *$N_G^\ell[u] \cap S = N_G^\ell[v] \cap S$, we have $|N_G^\ell[u] \Delta N_G^\ell[v]| \leq \varepsilon n$.*

512 *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
 513 between the balls of G . Since G has distance VC-dimension at most d then, by Lemma 10, the
 514 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
 515 ε -net for $\hat{\mathcal{H}}$. Therefore, for every $\ell \geq 0$ and for every $u, v \in V$, $|N_G^\ell[u] \Delta N_G^\ell[v]| > \varepsilon n \implies$
 516 $(N_G^\ell[u] \Delta N_G^\ell[v]) \cap S \neq \emptyset$. We stress that $(N_G^\ell[u] \Delta N_G^\ell[v]) \cap S \neq \emptyset \implies N_G^\ell[u] \cap S \neq N_G^\ell[v] \cap S$. \square

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

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

- 523 1. We compute a spanning path P'_k for $\mathcal{N}_k(G)$ of low *total* stabbing number, with the latter
 524 being equal to $\sum_{v \in V} |E_{P'_k}(N_G^k[v])|$;
- 525 2. Then, we compute an ε -net, for some well-chosen ε , and by doing so we partition the vertex-
 526 set into $p(\varepsilon)$ disjoint groups $V_1, V_2, \dots, V_{p(\varepsilon)}$. For every j we select a unique $v_j \in V_j$. We
 527 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
 528 low stabbing number for this subhypergraph.
- 529 3. We observe that if P_k is a spanning path of stabbing number t for \mathcal{H}_k , then it is also a
 530 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
 531 consider the unselected vertices $u \in V_j \setminus \{v_j\}$ sequentially. We compute the set of all the
 532 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
 533 of $N_G^k[u] \Delta N_G^k[v_j]$. We do so efficiently by using the auxiliary spanning path P'_k .

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

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

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

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

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

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

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

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

553 *Proof.* Let $\varepsilon := \Theta(n^{-\varepsilon_d})$. We perform a breadth-first search from every vertex in some random
554 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
555 on V such that $u \sim v \iff N_G^i[u] \cap S = N_G^i[v] \cap S$. We so partition V into some groups
556 V_1, V_2, \dots, V_p . Since by the hypothesis G has distance VC-dimension at most d then, by Lemma 1
557 we have $p = \mathcal{O}(|S|^d) = \mathcal{O}(d^d \log^d n \cdot n^{\varepsilon_d})$. Furthermore by Lemma 11, we have w.h.p. $u \sim v \implies$
558 $|N_G^i[u] \Delta N_G^i[v]| = \mathcal{O}(\varepsilon n) = \mathcal{O}(n^{1-\varepsilon_d})$.

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

- 561 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
562 this vertex. Since $p = \tilde{\mathcal{O}}(d^d \cdot n^{\varepsilon_d})$ and we have $\varepsilon_d < 1/(d+1)$, this phase can be implemented
563 in time $\mathcal{O}(md^d \log^d n \cdot n^{\varepsilon_d}) = \mathcal{O}(mn^{1-\varepsilon_d})$, that is truly subquadratic. Note that this time
564 bound also holds for non constant d as long as $d = \mathcal{O}(\frac{\log n}{\log \log n})$.
- 565 2. Let $R_i := \{N_G^i[v_j] \mid 1 \leq j \leq p\}$, and let $\mathcal{H}_i := (V, R_i)$. Note that since $\mathcal{H}_i \subseteq \mathcal{B}(G)$, the
566 VC-dimension of \mathcal{H}_i is at most d . Furthermore, the order and size of \mathcal{H}_i are, respectively, n
567 and $m_i := \mathcal{O}(pn) = \tilde{\mathcal{O}}(n^{1+\varepsilon_d})$. By Theorem 2, we can compute a spanning path P_i for \mathcal{H}_i of
568 stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ in time $\tilde{\mathcal{O}}(m_i + n^{2-\varepsilon_d}) = \tilde{\mathcal{O}}(n^{1+\varepsilon_d} + n^{2-\varepsilon_d}) = \tilde{\mathcal{O}}(n^{1-\varepsilon_d}m)$.
3. We observe that P_i is a spanning path of $\mathcal{N}_i(G)$ of stabbing number:

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

569 Indeed, for every $1 \leq j \leq p$, let $u \in V_j \setminus \{v_j\}$ be arbitrary. Let us consider the $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$
570 maximal intervals of which the union equals $N_G^i[v_j]$. Every vertex of $N_G^i[v_j] \setminus N_G^i[u]$ breaks
571 one of these intervals in two sub-intervals, thus increasing by at most one the number of
572 intervals needed for the ball $N_G^i[u]$. Furthermore, every vertex of $N_G^i[u] \setminus N_G^i[v_j]$ (since it is

573 not contained in one of the intervals of which $N_G^i[v_j]$ is the union) may also require one more
 574 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
 575 $\tilde{\mathcal{O}}(n^{1-\varepsilon_d}) + |N_G^i[u] \Delta N_G^i[v_j]|$ intervals of the path P_i .

576 We are now left with computing, for every $1 \leq j \leq p$ and $u \in V_j \setminus \{v_j\}$, the set $E_{P_i}(N_G^i[u])$
 577 of all the edges stabbed by the ball of radius i centered at u . For that, since we are already
 578 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:

- 579 • We use the spanning path P'_i for $\mathcal{N}_i(G)$ and a (suboptimal) representation $I_i(u)$ of $N_G^i[u]$
 580 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
 581 $\mathcal{O}(|E_{P'_i}(N_G^i[u])|)$ intervals of P'_i . Overall this step takes total time $\tilde{\mathcal{O}}(tm)$.
- 582 • Let $\sigma_i : V \rightarrow V(P'_i)$ be the permutation that maps every vertex to its position in the
 583 spanning path P'_i . For every $1 \leq j \leq p$, we construct two balanced binary search trees
 584 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,
 585 this takes total time $\tilde{\mathcal{O}}(np) = \tilde{\mathcal{O}}(n^{1+\varepsilon_d}) = \tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$.
- 586 • Finally, let us again consider some $u \in V_j \setminus \{v_j\}$ for some j . For every interval from $I_i(u)$,
 587 we want to enumerate the vertices of $V \setminus N_G^i[v_j]$ that lie on this interval. Since we stored
 588 all of $V \setminus N_G^i[v_j]$ into a balanced binary search tree, this can be done in time $\mathcal{O}(\log n)$
 589 plus $\mathcal{O}(1)$ extra time per vertex in the solution. In the same way, for every interval from
 590 $\overline{I_i(u)}$, we enumerate the vertices of $N_G^i[v_j]$ that lie on this interval. For a fixed u , the total
 591 time for this step is in $\tilde{\mathcal{O}}(|I_i(u)| + |\overline{I_i(u)}| + |N_G^i[u] \Delta N_G^i[v_j]|) = \tilde{\mathcal{O}}(|E_{P'_i}(N_G^i[u])| + n^{1-\varepsilon_d})$.
 592 Therefore, this last step takes total time $\tilde{\mathcal{O}}(tm + n^{2-\varepsilon_d})$.

593 ◦

594 Now, by the induction hypothesis we get a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ for
 595 $\mathcal{N}_{i-1}(G)$. By Subclaim 1 we transform such spanning path into a spanning path P'_i for $\mathcal{N}_i(G)$, where
 596 $\sum_{u \in V} |E_{P'_i}(N_G^i[u])| = \tilde{\mathcal{O}}(mn^{1-\varepsilon_d})$. Finally, by Subclaim 2 we can use P'_i in order to compute, in
 597 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
 598 proving that our claim holds for i . ◊

599 Summarizing, by Claim 2 we can compute a spanning path of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ for
 600 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
 601 whether G has diameter at most k , and if so, we compute $\text{diam}(G)$ exactly, in time $\tilde{\mathcal{O}}(k \cdot mn^{1-\varepsilon_d})$.
 602 □

604 4.1 Application to nowhere dense graph classes

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

613 dense, and for any i , the distance- i VC-dimension of any graph in \mathcal{G} is upper-bounded by some
 614 constant d_i [62]. By choosing $d := \max_{1 \leq i \leq k-1} d_i$, we thus obtain the following weaker version of
 615 Theorem 3 for nowhere dense graphs:

616 **Theorem 4.** *Let \mathcal{G} be a class of nowhere dense graphs. There exists a Monte Carlo algorithm such
 617 that, for every constant $k = \mathcal{O}(1)$, for any graph in \mathcal{G} we can decide whether its diameter is at most
 618 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 .*

619 Note that, for any class of nowhere dense graphs, there also exists an FPT algorithm, in time
 620 $\mathcal{O}(f(k) \cdot n^{1+o(1)})$, for deciding whether the diameter is at most k [47]. The function f is, at least, a
 621 tower of exponentials in k . Our result shows that a better dependency on k is possible at the cost
 622 of a higher exponent on n . We leave as open to find FPT algorithms with better trade-offs.

623 4.2 Exact distance oracles

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

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

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

- 633 1. For every $1 \leq i \leq k$, we construct a spanning path P_i for $\mathcal{N}_i(G)$, of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$
 634 — along with the corresponding $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ intervals for $N^i[v]$, for every vertex v .
- 635 2. Then, we sample a subset S_k of $\tilde{\mathcal{O}}(n/k)$ vertices, and we compute a shortest-path tree for
 636 each such vertex.

637 Let us analyze the runtime of this above construction. As it was explained in the proof of Theorem 3,
 638 the first step (computation of k spanning paths of low stabbing number) can be done in randomized
 639 time $\tilde{\mathcal{O}}(kmn^{1-\varepsilon_d})$. For the second step, since we only need to perform a BFS for each vertex of
 640 S_k , the runtime is in deterministic time $\tilde{\mathcal{O}}(\frac{n}{k}m)$. Overall, the total pre-processing time is in
 641 $\tilde{\mathcal{O}}((kn^{1-\varepsilon_d} + \frac{n}{k})m)$.

642 Furthermore, let us analyze the space of this oracle. Each spanning path P_i requires $\mathcal{O}(n)$ space.
 643 Since, for every vertex v , $N^i[v]$ is the union of $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ intervals, we have $\sum_{v \in V} |E_{P_i}(N^i[v])| =$
 644 $\tilde{\mathcal{O}}(n^{2-\varepsilon_d})$. Thus, we need $\tilde{\mathcal{O}}(kn^{2-\varepsilon_d})$ space for the k spanning paths and the corresponding sets
 645 E_{P_i} . For the second step, each shortest-path tree requires $\mathcal{O}(n)$ space, and therefore the total space
 646 required is in $\tilde{\mathcal{O}}(\frac{n^2}{k})$. Overall, the oracle requires $\tilde{\mathcal{O}}(kn^{2-\varepsilon_d} + \frac{n^2}{k})$ space.

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

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

- 649 • If $\text{dist}(u, v) \leq k$, then we compute the smallest i such that $u \in N^i[v]$, which is precisely
650 $\text{dist}(u, v)$. Note that we only need to test $\mathcal{O}(\log k)$ values for that (because we can apply
651 binary search), and that each test takes time $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ by using the corresponding spanning
652 path.
 - 653 • Otherwise, we output $\text{dist}(u, v) = \min_{s \in S_k} \text{dist}(u, s) + \text{dist}(v, s)$. Since we sampled S_k u.a.r.,
654 the probability that $\text{dist}(u, v) \neq \min_{s \in S_k} \text{dist}(u, s) + \text{dist}(v, s)$ for at least one pair (u, v) at
655 distance more than k is at most $1/n^c$, for some arbitrarily large constant c [7].
- 656 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
657 the value of our parameter to $k = \tilde{\mathcal{O}}\left(n^{\frac{\varepsilon_d}{2}}\right)$. □

658 5 Diameter Computation in Truly Subquadratic Time

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

666 **Theorem 6.** *Let \mathcal{G} be a monotone graph class with strongly sublinear balanced separators. Then,*
667 *for every $d > 0$, for any graph in \mathcal{G} of distance VC-dimension at most d , we can compute all the*
668 *eccentricities (and so, the diameter) in deterministic time $\tilde{\mathcal{O}}(n^{2-\varepsilon_{\mathcal{G}}(d)})$, for some constant $\varepsilon_{\mathcal{G}}(d) \in$*
669 *$(0; 1)$ that only depends on d .*

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

676 Finally, we say that a class of graphs \mathcal{G} has *polynomial expansion* if there exists a polynomial p
677 such that, for every r -shallow minor of a graph in \mathcal{G} (cf. Section 4.1), its average degree is at most
678 $p(r)$. We want to stress that there is an equivalence between the monotone classes of graphs \mathcal{G} with
679 strongly sublinear balanced separators and those of polynomial expansion [35]. In particular, the
680 graphs in \mathcal{G} have bounded degeneracy, and so, they are sparse (i.e., with $m = \mathcal{O}(n)$ edges). We
681 will often use this property in what follows.

682 5.1 Application to H -minor free graphs

683 Let us now review some interesting classes where Theorem 6 does apply. Since planar graphs have
684 *distance VC-dimension* at most four [11] then, it follows from the planar separator theorem of
685 Lipton and Tarjan [58] that Theorem 6 applies to the class of planar graphs. Therefore, Theorem 6
686 gives us a new subquadratic-time algorithm for diameter computation on *unweighted* planar graphs,

687 but with a slower running-time than for the algorithms presented in [17, 45]. More generally, the
 688 following separator theorem is from Alon et al.:

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

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

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

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

700 5.2 Proof of Theorem 6

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

703 Algorithmic aspects of r -divisions

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

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

709 We will also use the following simple result:

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

712 *Proof.* Let C_1, C_2, \dots, C_k be the connected components of $G \setminus S$. They can be computed in linear
 713 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,
 714 simply by scanning the connected components in order until we have scanned more than $2n/3$
 715 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
 716 $A := A'$, $B := B' \cup C_{i_0}$. Thus, from now on let us assume that $|B' \cup C_{i_0}| > 2n/3$. Note that since
 717 S is a balanced separator, it implies that $i_0 < k$. Then, by the very definition of i_0 we also have
 718 $|A' \cup C_{i_0}| > 2n/3$. Overall, $|A'| + 2|C_{i_0}| + |B'| > 4n/3$. Since $|A'| + |B'| + |C_{i_0}| < n$, we obtain
 719 $|C_{i_0}| > n/3$. We are done by setting $A := A' \cup B'$ and $B := C_{i_0}$. The total runtime is linear. \square

720 Now, set a parameter⁴ $\beta := \frac{4+\alpha}{5} < 1$. By Lemma 13, for every n -vertex m -edge graph in $\mathcal{G}_{\alpha,C}$
 721 we can compute a balanced separator of order $\mathcal{O}(n^\beta)$ in time $\mathcal{O}(n^{1+\beta})$. Following Frederickson [42],
 722 we define an r -division for an n -vertex graph $G \in \mathcal{G}_{\alpha,C}$ as follows:

⁴More 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 $\beta = \alpha = 1/2$.

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

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

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

735 **Lemma 15** ([50]). *Set $\beta := \frac{4+\alpha}{5}$. There exists a constant r_0 such that, for any n -vertex graph
736 $G \in \mathcal{G}_{\alpha,C}$ and $r \geq r_0$, any r -division of G has an excess in $\mathcal{O}(n/r^{1-\beta})$.*

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

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

742 **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
743 r -division in time $\tilde{\mathcal{O}}(n^{1+\beta})$.*

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

753 We are left proving that the total number of edges in the final clusters is indeed in $\mathcal{O}(n)$.
754 For that, let us consider any of the clusters W_i . Since $\mathcal{G}_{\alpha,C}$ is monotone, we have $W_i \in \mathcal{G}_{\alpha,C}$.
755 Furthermore, every graph in $\mathcal{G}_{\alpha,C}$ must be $\mathcal{O}(1)$ -degenerate (*e.g.*, see [34, Lemma 2 (b)] where the
756 author proved a stronger result, namely that $\mathcal{G}_{\alpha,C}$ has polynomial expansion). It implies that W_i
757 has size $\mathcal{O}(|V(W_i)|)$. Overall, if the total excess is q then, the total number of edges in the clusters
758 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
759 $\mathcal{O}(n)$. □

760 Boundary Hypergraphs

761 Let G be a graph equipped with some r -division, and let $\vec{\ell} = (\ell_v)_{v \in V}$ be a collection of positive
762 integers that is indexed by the vertex-set of G . Roughly, our objective is to use the r -division in

763 order to compute, for every vertex v , a compact interval representation of its ball of radius ℓ_v . This
 764 leads us to the following natural object:

765 **Definition 8.** Let Λ_r be an r -division of a graph G , and let $\vec{\ell} = (\ell_v)_{v \in V}$ be a collection of positive
 766 integers that is indexed by the vertex-set of G . The $\vec{\ell}$ -boundary hypergraph $\mathcal{H}_{\vec{\ell}, G}(\Lambda_r)$ has for
 767 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
 768 vertex and $\text{dist}_G(u, v) < \ell_u$, then the ball $N_G^{\ell_u - \text{dist}_G(u, v)}[v]$ is a hyperedge of $\mathcal{H}_{\vec{\ell}, G}(\Lambda_r)$.

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

773 **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
 774 $\vec{\ell}$ -boundary hypergraph $\mathcal{H}_{\vec{\ell}, G}(\Lambda_r)$ has $\mathcal{O}(nr^\beta)$ hyperedges.

775 *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
 776 vertices in the cluster. We observe that the number of boundary nodes is at most the excess and that
 777 $\sum_{W_i \in \Lambda_r} b_i$ is at most twice the excess. Then, by Lemma 15 we have $\mathcal{O}(r) \times \mathcal{O}(n/r^{1-\beta}) = \mathcal{O}(nr^\beta)$
 778 hyperedges. \square

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

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

787 *Proof.* By construction, $\mathcal{H}_{\vec{\ell}, G}(\Lambda_r)$ is a subhypergraph of $\mathcal{B}(G)$, the ball hypergraph of G . Therefore,
 788 the VC-dimension of $\mathcal{H}_{\vec{\ell}, G}(\Lambda_r)$ is at most d . Let ε_d be the constant of Theorem 2. In order to
 789 prove the result, we are left proving that we can adapt the algorithm of Theorem 2 so that it runs
 790 in time $\tilde{\mathcal{O}}(nm/r^{1-\beta} + n^{2-\varepsilon_d}r^\beta)$ when it is given Λ_r and $\vec{\ell}$ as input. For that, let F be the set
 791 of the boundary vertices. We have that $|F|$ is at most the excess, and so, by Lemma 15 we get
 792 $|F| = \mathcal{O}(n/r^{1-\beta})$.

- 793 1. We start with a breadth-first search from every vertex of F . This pre-processing phase takes
 794 time $\mathcal{O}(|F|m) = \mathcal{O}(n^2/r^{1-\beta})$. Furthermore, note that by doing so we can compute all the
 795 pairs $(v, t) \in F \times \mathbb{N}$ such that $N_G^t[v]$ is a hyperedge of $\mathcal{H}_{\vec{\ell}, G}(\Lambda_r)$.
- 796 2. Let $\eta = 2^{d+1}\varepsilon_d$. We arbitrarily partition the vertex-set V into subsets V_1, V_2, \dots, V_p such
 797 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
 798 the proof of Theorem 2 (i.e., Claim 1), we can compute a spanning path of stabbing number
 799 $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ for $\mathcal{H}_{\vec{\ell}, G}(\Lambda_r)$ if we are given the subhypergraphs $\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_p$ that are induced

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

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 $V_i^1(v), V_i^2(v), \dots, V_i^{n_i}(v)$. Overall, this phase takes time $\tilde{O}(|F||V_i|)$, and so *total* time (for all i) $\tilde{O}(|F|n) = \tilde{O}(n^2/r^{1-\beta})$.

(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 $j \in \{0 \dots n_i\}$, 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 $\mathcal{P}_j = \left(V_i^{j,1}, V_i^{j,2}, \dots, V_i^{j,s_i(j)} \right)$ (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

841
842
843
844
845

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

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

846
847

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:

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

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

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

848
849
850
851
852
853
854
855
856

- (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 $\mathcal{H}_{\vec{t}, G}(\Lambda_r)$ 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 $\mathcal{O}(n^{(d+2)\eta})$.

857
858
859
860

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 $\tilde{\mathcal{O}}(n^{1+\eta(c(d+1)-1)}) = \tilde{\mathcal{O}}(n^{2-\varepsilon_a})$ for some constant $c > 2$. As a result, the running-time of this part is also in $\tilde{\mathcal{O}}(n^2/r^{1-\beta} + n^{2-\varepsilon_a})$.

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

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 $\vec{\ell} = (\ell_v)_{v \in V}$, we can decide whether $\forall v \in V, \text{ecc}_G(v) \leq \ell_v$ 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 $\mathcal{E}_{\vec{\ell}}(G) := (V, \{N_G^{\ell_v}[v] \mid v \in V\})$. More precisely, we claim that it is sufficient to compute a spanning path $P_{\vec{\ell}}$ for the latter, along with a collection $(I_{P_{\vec{\ell}}}(v))_{v \in V}$ such that, for every vertex v , $I_{P_{\vec{\ell}}}(v)$ is a set of (possibly intersecting and/or overlapping) intervals of $P_{\vec{\ell}}$ whose union equals $N_G^{\ell_v}[v]$, and furthermore $\sum_{v \in V} |I_{P_{\vec{\ell}}}(v)|$ 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 $\tilde{\mathcal{O}}\left(\sum_{v \in V} |I_{P_{\vec{\ell}}}(v)|\right)$. 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 $P_{\vec{\ell}}$ for $\mathcal{E}_{\vec{\ell}}(G)$, such that $\sum_{v \in V} |I_{P_{\vec{\ell}}}(v)| = \tilde{\mathcal{O}}(n(r + n^{1-\varepsilon_d}r^\beta))$.*

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 $v \in N_G^{\ell_u}[u]$ such that at least one uv -path of length $\leq \ell_u$ is fully contained in a cluster.

899 2. Let us now consider the $\vec{\ell}$ -neighbourhood hypergraph $\mathcal{H}_{\vec{\ell},G}(\Lambda_r)$. By Lemma 18 we can
900 compute a spanning path $P_{\vec{\ell}}$ of stabbing number $\tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ for this hypergraph, in time
901 $\tilde{\mathcal{O}}(nm/r^{1-\beta} + n^{2-\varepsilon_d}r^\beta) = \tilde{\mathcal{O}}(n^2/r^{1-\beta} + n^{2-\varepsilon_d}r^\beta)$. Let $u \in V$. There are two cases:

- 902 • *Case u is a boundary vertex.* Since $N_G^{\ell_u}[u]$ is a hyperedge of the boundary hypergraph,
903 we have $|I_{P_{\vec{\ell}}}(u)| = \tilde{\mathcal{O}}(n^{1-\varepsilon_d})$ (already computed).
- 904 • *Case u is an internal vertex.* Let $W \in \Lambda_r$ be the unique cluster containing u , and set
905 initially $I_{P_{\vec{\ell}}}(u) := \emptyset$. For every boundary vertex $v \in V(W)$, if $\text{dist}_G(u, v) < \ell_u$ then, we
906 add all intervals corresponding to $N_G^{\ell_u - \text{dist}_G(u, v)}[v]$ to $I_{P_{\vec{\ell}}}(u)$. Assuming there are b_W
907 boundary vertices in W , we obtain that $|I_{P_{\vec{\ell}}}(u)| = \tilde{\mathcal{O}}(b_W \cdot n^{1-\varepsilon_d})$. Furthermore, this
908 above set of intervals covers exactly the balls $N_G^{\ell_u - \text{dist}_G(u, v)}[v]$, for the boundary vertices
909 $v \in V(W)$. By construction, every vertex that is contained in one of these balls, defined
910 above, is at a distance $\leq \ell_u$ to u ; conversely, since Λ_r is also an edge-covering, every
911 vertex of $N_G^{\ell_u}[u] \setminus N_{G[W]}^{\ell_u}[u]$ must be in one of these balls. As a result, in order to construct
912 $I_{P_{\vec{\ell}}}(u)$, it suffices to update this set using the vertices of $N_{G[W]}^{\ell_u}[u]$ (already computed
913 during the first step). Note that by doing so, we can only modify the cardinality of
914 $I_{P_{\vec{\ell}}}(u)$ by an $\mathcal{O}(|W|) = \mathcal{O}(r)$.

915 Overall, we obtain that $\sum_{u \in V} |I_{P_{\vec{\ell}}}(u)| = \tilde{\mathcal{O}}(nr + n^{1-\varepsilon_d} \cdot \sum_W (b_W \cdot |V(W)|)) = \tilde{\mathcal{O}}(nr + rn^{1-\varepsilon_d} \cdot$
916 $\sum_W b_W)$. Again we observe that $\sum_W b_W$ is at most twice the excess, and so by Lemma 15
917 $\sum_W b_W = \mathcal{O}(n/r^{1-\beta})$. Therefore, $\sum_{u \in V} |I_{P_{\vec{\ell}}}(u)| = \tilde{\mathcal{O}}(n(r + n^{1-\varepsilon_d}r^\beta))$.

918 ◇

919 Overall, the running-time of our algorithm is optimized when we have $n^2/r^{1-\beta} = n^{2-\varepsilon_d}r^\beta$. As
920 a result, a good choice is $r = \Theta(n^{\varepsilon_d})$. Finally, we stress that in this case, the running time is in
921 $\tilde{\mathcal{O}}(n^{2-(1-\beta)\varepsilon_d})$, that is truly subquadratic because $\beta < 1$. □

922 5.3 Extension to weighted graphs

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

930 **Lemma 19.** *Let \mathcal{G} be a class of unweighted graphs that is closed under edge-subdivisions. If every*
931 *graph in \mathcal{G} has distance VC-dimension at most d , then every weighted graph (G, w) such that $G \in \mathcal{G}$*
932 *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:

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

933 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
 934 now on that all the edge-weights are rational numbers. By multiplying all the edge-weights by a
 935 sufficiently large integer, we may further assume that all the edge-weights are positive integers.

936 Under this above assumption, we may replace every edge $e \in E(G)$ by a path of length w_e .
 937 Doing so, we get an unweighted graph G_w such that $V(G) \subseteq V(G_w)$ and, for every $u, v \in$
 938 $V(G)$, $dist_{G,w}(u, v) = dist_{G_w}(u, v)$. By the hypothesis, $G_w \in \mathcal{G}$, and therefore it has *distance*
 939 *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
 940 ball hypergraph of G_w , as it was defined in Sec. 2.2). \square

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

- 950 • If all the edge-weights are positive integers bounded by M , then the eccentricities must be
 951 between 1 and Mn . We compute the exact value of the eccentricities with n simultaneous
 952 binary searches, that induces an $\mathcal{O}(\log(nM))$ overhead in the total running time.
- 953 • If now all the edge-weights are positive real numbers, then the range of possible eccentricities
 954 for each vertex is too large and we cannot perform a binary search directly. We compute
 955 the eccentricity of an arbitrary vertex, which we denote by ℓ_0 . By the triangular inequality,
 956 every vertex has its eccentricity between $\ell_0/2$ and $2\ell_0$. Then, we only consider in this interval
 957 $[\ell_0/2; 2\ell_0]$ the powers of $1 + \varepsilon$, where $\varepsilon > 0$ is an arbitrarily small precision parameter. We
 958 observe that the number of distinct powers of $1 + \varepsilon$ between these two values is in $\mathcal{O}(\varepsilon^{-1})$.
 959 As a result, we can compute an $(1 + \varepsilon)$ -approximation of all the eccentricities by using binary
 960 search, that induces an $\mathcal{O}(\log(1/\varepsilon))$ overhead in the running time.

961 Summarizing, we get:

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

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

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

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 $\tilde{O}(n^{2-\varepsilon_H} \log M)$, 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 $\tilde{O}(n^{2-\varepsilon_H} \log(1/\varepsilon))$.

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.

1008 Acknowledgements

1009 This work was supported by Inria Gang project-team, by Irif laboratory from CNRS and Paris
1010 University, and by the ANR projects DISTANCIA (ANR-17-CE40-0015) and Multimod (ANR-
1011 17-CE22-0016). This work was also supported by project PN 19 37 04 01 “New solutions for
1012 complex problems in current ICT research fields based on modelling and optimization”, funded
1013 by the Romanian Core Program of the Ministry of Research and Innovation (MCI) 2019-2022,
1014 and by a grant of Romanian Ministry of Research and Innovation CCCDI-UEFISCDI, project no.
1015 17PCCDI/2018.

1016 References

- 1017 [1] A. Abboud, V. Vassilevska Williams, and J. Wang. Approximation and fixed parameter sub-
1018 quadratic algorithms for radius and diameter in sparse graphs. In *SODA*, pages 377–391.
1019 SIAM, 2016.
- 1020 [2] N. Alon, P. Seymour, and R. Thomas. A separator theorem for nonplanar graphs. *Journal of*
1021 *the American Mathematical Society*, 3(4):801–808, 1990.
- 1022 [3] M. Anthony, G. Brightwell, and C. Cooper. The Vapnik-Chervonenkis dimension of a random
1023 graph. *Discrete Mathematics*, 138(1-3):43–56, 1995.
- 1024 [4] A. Backurs, L. Roditty, G. Segal, V. Vassilevska Williams, and N. Wein. Towards tight
1025 approximation bounds for graph diameter and eccentricities. In *Proceedings of the 50th Annual*
1026 *ACM SIGACT Symposium on Theory of Computing (STOC 2018)*, pages 267–280. ACM, 2018.
- 1027 [5] C. Berge. *Graphs and hypergraphs*. North-Holland Pub. Co., 1973.
- 1028 [6] V. Bilo, V. Goyal, R. Ravi, and M. Singh. On the crossing spanning tree problem. In *Approx-*
1029 *imation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages
1030 51–60. Springer, 2004.
- 1031 [7] B. Bollobás, D. Coppersmith, and M. Elkin. Sparse distance preservers and additive spanners.
1032 *SIAM Journal on Discrete Mathematics*, 19(4):1029–1055, 2005.
- 1033 [8] J. A. Bondy and U. S. R. Murty. *Graph theory*. Springer, 2008.
- 1034 [9] M. Borassi, P. Crescenzi, and M. Habib. Into the square: On the complexity of some quadratic-
1035 time solvable problems. *Electronic Notes in Theoretical Computer Science*, 322:51–67, 2016.
- 1036 [10] N. Bousquet, A. Lagoutte, Z. Li, A. Parreau, and S. Thomassé. Identifying codes in hereditary
1037 classes of graphs and VC-dimension. *SIAM Journal on Discrete Mathematics*, 29(4):2047–2064,
1038 2015.
- 1039 [11] N. Bousquet and S. Thomassé. VC-dimension and Erdős–Pósa property. *Discrete Mathematics*,
1040 338(12):2302–2317, 2015.
- 1041 [12] A. Brandstädt, V. Chepoi, and F. Dragan. The algorithmic use of hypertree structure and
1042 maximum neighbourhood orderings. *Discrete Applied Mathematics*, 82(1-3):43–77, 1998.

- 1043 [13] A. Brandstädt, F. Dragan, V. Chepoi, and V. Voloshin. Dually chordal graphs. *SIAM Journal*
1044 *on Discrete Mathematics*, 11(3):437–455, 1998.
- 1045 [14] K. Bringmann, T. Husfeldt, and M. Magnusson. Multivariate analysis of orthogonal range
1046 searching and graph distances parameterized by treewidth. In *IPEC*, 2018.
- 1047 [15] K. Bringmann, T. Husfeldt, and M. Magnusson. Multivariate analysis of orthogonal range
1048 searching and graph distances. *Algorithmica*, 82(8):2292–2315, 2020.
- 1049 [16] H. Brönnimann, B. Chazelle, and J. Matoušek. Product range spaces, sensitive sampling,
1050 and derandomization. *SIAM J. Comput.*, 28(5):1552–1575, 1999.
- 1051 [17] S. Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in
1052 planar graphs. *ACM Transactions on Algorithms (TALG)*, 15(2):21, 2018.
- 1053 [18] S. Cabello and C. Knauer. Algorithms for graphs of bounded treewidth via orthogonal range
1054 searching. *Computational Geometry*, 42(9):815–824, 2009.
- 1055 [19] M. Cairo, R. Grossi, and R. Rizzi. New bounds for approximating extremal distances in
1056 undirected graphs. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on*
1057 *Discrete algorithms (SODA 2016)*, pages 363–376. SIAM, 2016.
- 1058 [20] T. Chan. Optimal partition trees. *Discrete & Computational Geometry*, 47(4):661–690, 2012.
- 1059 [21] B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension.
1060 *Discrete & Computational Geometry*, 4(5):467–489, 1989.
- 1061 [22] S. Chechik, D. Larkin, L. Roditty, G. Schoenebeck, R. Tarjan, and V. Vassilevska Williams.
1062 Better approximation algorithms for the graph diameter. In *Proceedings of the twenty-fifth*
1063 *annual ACM-SIAM symposium on Discrete algorithms (SODA 2014)*, pages 1041–1052. SIAM,
1064 2014.
- 1065 [23] V. Chepoi, B. Estellon, and Y. Vaxès. Covering planar graphs with a fixed number of balls.
1066 *Discrete & Computational Geometry*, 37(2):237–244, 2007.
- 1067 [24] M. Cohen, Y. Lee, and Z. Song. Solving linear programs in the current matrix multiplication
1068 time. In *STOC*, pages 938–942. ACM, 2019.
- 1069 [25] V. Cohen-Addad, S. Dahlgaard, and C. Wulff-Nilsen. Fast and compact exact distance oracle
1070 for planar graphs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science*
1071 *(FOCS)*, pages 962–973. IEEE, 2017.
- 1072 [26] D. Corneil, F. Dragan, M. Habib, and C. Paul. Diameter determination on restricted graph
1073 families. *Discrete Applied Mathematics*, 113(2-3):143–166, 2001.
- 1074 [27] D. Coudert, G. Ducoffe, and A. Popa. Fully polynomial FPT algorithms for some classes of
1075 bounded clique-width graphs. In *SODA’18*, pages 2765–2784. SIAM, 2018.
- 1076 [28] S. Dahlgaard. On the hardness of partially dynamic graph problems and connections to diam-
1077 eter. In *43rd International Colloquium on Automata, Languages, and Programming, (ICALP*
1078 *2016)*, pages 48:1–48:14, 2016.

- 1079 [29] P. Damaschke. Computing giant graph diameters. In *IWOCA*, pages 373–384. Springer, 2016.
- 1080 [30] R. Downey, P. Evans, and M. Fellows. Parameterized learning complexity. In *COLT*, pages
1081 51–57. ACM, 1993.
- 1082 [31] G. Ducoffe. A New Application of Orthogonal Range Searching for Computing Giant Graph
1083 Diameters. In *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*, 2019.
- 1084 [32] G. Ducoffe, M. Habib, and L. Viennot. Fast diameter computation within split graphs. In
1085 *International Conference on Combinatorial Optimization and Applications (COCOA)*, pages
1086 155–167. Springer, 2019.
- 1087 [33] G. Ducoffe, M. Habib, and L. Viennot. Diameter computation on H -minor free graphs and
1088 graphs of bounded (distance) VC-dimension. In *Proceedings of the Fourteenth Annual ACM-
1089 SIAM Symposium on Discrete Algorithms (SODA)*, pages 1905–1922. SIAM, 2020.
- 1090 [34] Z. Dvořák. On classes of graphs with strongly sublinear separators. *European Journal of
1091 Combinatorics*, 71:1–11, 2018.
- 1092 [35] Z. Dvorak and S. Norin. Strongly sublinear separators and polynomial expansion. *SIAM
1093 Journal on Discrete Mathematics*, 30(2):1095–1101, 2016.
- 1094 [36] Z. Dvořák and S. Norin. Treewidth of graphs with balanced separations. *Journal of Combi-
1095 natorial Theory, Series B*, 137:137–144, 2019.
- 1096 [37] D. Eisenstat and D. Angluin. The VC dimension of k -fold union. *Information Processing
1097 Letters*, 101(5):181–184, 2007.
- 1098 [38] D. Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3-
1099 4):275–291, 2000.
- 1100 [39] D. Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the
1101 Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003,
1102 Baltimore, Maryland, USA.*, pages 599–608. ACM/SIAM, 2003.
- 1103 [40] J. Evald and S. Dahlgaard. Tight Hardness Results for Distance and Centrality Problems in
1104 Constant Degree Graphs. Technical Report arXiv:1609.08403, ArXiv, 2016.
- 1105 [41] A. Farley and A. Proskurowski. Computation of the center and diameter of outerplanar graphs.
1106 *Discrete Applied Mathematics*, 2(3):185–191, 1980.
- 1107 [42] G. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM
1108 Journal on Computing*, 16(6):1004–1022, 1987.
- 1109 [43] C. Gavoille, D. Peleg, S. Pérennes, and R. Raz. Distance labeling in graphs. *Journal of
1110 Algorithms*, 53(1):85–112, 2004.
- 1111 [44] P. Gawrychowski, H. Kaplan, S. Mozes, M. Sharir, and O. Weimann. Voronoi diagrams on
1112 planar graphs, and computing the diameter in deterministic $\tilde{O}(n^{5/3})$ time. In *Proceedings
1113 of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2018)*,
1114 pages 495–514. SIAM, 2018.

- 1115 [45] P. Gawrychowski, H. Kaplan, S. Mozes, M. Sharir, and O. Weimann. Voronoi diagrams on
 1116 planar graphs, and computing the diameter in deterministic $\tilde{O}(n^{5/3})$ time. *SIAM J. Comput.*,
 1117 50(2):509–554, 2021.
- 1118 [46] P. Gawrychowski, S. Mozes, O. Weimann, and C. Wulff-Nilsen. Better tradeoffs for exact
 1119 distance oracles in planar graphs. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM*
 1120 *Symposium on Discrete Algorithms (SODA)*, pages 515–529. SIAM, 2018.
- 1121 [47] M. Grohe, S. Kreutzer, and S. Siebertz. Deciding first-order properties of nowhere dense
 1122 graphs. *Journal of the ACM (JACM)*, 64(3):1–32, 2017.
- 1123 [48] M. Habib, R. McConnell, C. Paul, and L. Viennot. Lex-BFS and partition refinement, with
 1124 applications to transitive orientation, interval graph recognition and consecutive ones testing.
 1125 *Theoretical Computer Science*, 234(1-2):59–84, 2000.
- 1126 [49] S. Har-Peled. Approximating Spanning Trees with Low Crossing Number. Technical Report
 1127 abs/0907.1131, arXiv, 2009.
- 1128 [50] S. Har-Peled and K. Quanrud. Approximation algorithms for polynomial-expansion and low-
 1129 density graphs. *SIAM Journal on Computing*, 46(6):1712–1744, 2017.
- 1130 [51] D. Haussler and E. Welzl. ε -nets and simplex range queries. *Discrete & Computational Ge-*
 1131 *ometry*, 2(2):127–151, 1987.
- 1132 [52] M. Henzinger, P. Klein, S. Rao, and S. Subramanian. Faster shortest-path algorithms for
 1133 planar graphs. *Journal of computer and system sciences*, 55(1):3–23, 1997.
- 1134 [53] K. Kawarabayashi, P. N. Klein, and C. Sommer. Linear-space approximate distance oracles
 1135 for planar, bounded-genus and minor-free graphs. In L. Aceto, M. Henzinger, and J. Sgall, ed-
 1136 itors, *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011,*
 1137 *Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*, volume 6755 of *Lecture Notes in*
 1138 *Computer Science*, pages 135–146. Springer, 2011.
- 1139 [54] K. Kawarabayashi and B. Reed. A separator theorem in minor-closed classes. In *2010 IEEE*
 1140 *51st Annual Symposium on Foundations of Computer Science*, pages 153–162. IEEE, 2010.
- 1141 [55] J. Kleinberg. Detecting a network failure. *Internet Mathematics*, 1(1):37–55, 2004.
- 1142 [56] E. Kranakis, D. Krizanc, B. Ruf, J. Urrutia, and G. Woeginger. The VC-dimension of set
 1143 systems defined by graphs. *Discrete Applied Mathematics*, 77(3):237–257, 1997.
- 1144 [57] J. Li and M. Parter. Planar diameter via metric compression. In *STOC*, pages 152–163. ACM,
 1145 2019.
- 1146 [58] R. Lipton and R. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied*
 1147 *Mathematics*, 36(2):177–189, 1979.
- 1148 [59] J. Matoušek. Spanning trees with low crossing number. *RAIRO-Theoretical Informatics and*
 1149 *Applications-Informatique Théorique et Applications*, 25(2):103–123, 1991.

- 1150 [60] J. Matoušek. Bounded VC-dimension implies a fractional Helly theorem. *Discrete & Computational Geometry*, 31(2):251–255, 2004.
1151
- 1152 [61] J. Nešetřil and P. Ossona de Mendez. *Sparsity: graphs, structures, and algorithms*, volume 28. Springer Science & Business Media, 2012.
1153
- 1154 [62] J. Nešetřil and P. Ossona de Mendez. Structural sparsity. *Russian Mathematical Surveys*, 71(1):79, 2016.
1155
- 1156 [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.
1157
- 1158 [64] R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM J. Comput.*, 16(6):973–989, 1987.
1159
- 1160 [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.
1161
- 1162 [66] S. Plotkin, S. Rao, and W. Smith. Shallow Excluded Minors and Improved Graph Decompositions. In *SODA*, volume 90, pages 462–470, 1994.
1163
- 1164 [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.
1165
- 1166 [68] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.
1167
- 1168 [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.
1169
- 1170 [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.
1171
- 1172 [71] E. Welzl. On spanning trees with low crossing numbers. In *Data structures and efficient algorithms*, pages 233–249. Springer, 1992.
1173
- 1174 [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.
1175
1176