VC Set Systems in Minor-free (Di)Graphs and Applications

Anonymous

Abstract

A recent line of work on VC set systems in minor-free (undirected) graphs, starting from Li and Parter [LP19], who constructed a new VC set system for planar graphs, has given surprising algorithmic results [LP19, Le23, DHV20, FHMWN20]. In this work, we initialize a more systematic study of VC set systems for minor-free graphs and their applications in both undirected graphs and directed graphs (a.k.a *digraphs*). More precisely:

1. We propose a new variant of Li-Parter set system for *undirected* graphs. Our set system settles two weaknesses of Li-Parter set system: the terminals can be anywhere, and the graph can be K_h-minor-free for any fixed h. We obtain several algorithmic applications, and notably: (i) the first exact distance oracle for unweighted and undirected K_h-minor-free graphs that has truly subquadratic space and constant query time, and (ii) the first truly subquadratic time algorithm for computing Wiener index of K_h-minor-free graphs, resolving an open problem posed by Ducoffe, Habib, and Viennot [DHV20].
2. We extend our set system to K_h-minor-free *digraphs* and show that its VC dimension is $O(h^2)$. We use this result to design the first subquadratic time algorithm for computing (unweighted) diameter and all-vertices eccentricities in K_h-minor-free digraphs.
3. We show that the system of *directed* balls in minor-free digraphs has VC dimension at most $h-1$. We then present a new technique to exploit the VC system of balls, giving the first exact distance oracle for unweighted minor-free digraphs that has truly subquadratic space and logarithmic query time.
4. On the negative side, we show that VC set system constructed from shortest path trees of planar digraphs does not have a bounded VC dimension. This leaves an intriguing open problem: determine a necessary and sufficient condition for a set system derived from a minor-free graph to have a bounded VC-dimension.

The highlight of our work is the results for digraphs, as we are not aware of known algorithmic work on constructing and exploiting VC set systems for digraphs.

Contents

1 Introduction

A pair of seminar papers by Lipton and Tarjan [LT79, LT80] in the 70s initiated a productive line of research on planar graph algorithms. Over the past several decades, numerous algorithmic tools have been developed for planar graphs. We can roughly classify them into two classes: one for coping with NP-hard problems and another for designing fast algorithms for problems in P[1]. The former class aims to provide (efficient) polynomial time approximation schemes or subexponential time (parameterized or exact) algorithms for NP-hard problems. Representative examples are Baker's layering technique [Bak94], contraction decomposition [Kle05a], bidimensionality [DFHT05, DH05], and sphere cut decomposition [DPBF09], to name a few. The latter class aims to design (nearly) linear time, in many cases truly subquadratic time, algorithms for problems in P where no algorithms of the same running time were known for general graphs. A non-exhaustive list of examples includes the separator theorem [LT79, LT80] and r-division [Fed87], shortest path separator [LT79, Tho04], multiple-source shortest paths [Kle05b], Voronoi diagram [Cab18], and VC-dimension [LP19]. (The classification into two classes is not exclusive: there are techniques that can be used for both purposes.)

On the other hand, planarity is fragile: adding a single edge or vertex could make a planar graph become non-planar. Therefore, a major research goal is to extend the aforementioned algorithmic tools beyond planar graphs, specifically graphs that are more robust, such as bounded genus graphs and K_h-minor-free graphs. Bounded genus graphs are robust to edge addition—adding a new edge increases the genus by at most 1—but not to vertex addition as adding a single vertex could increase the genus by $\Omega(n)$. K_h-minor-free graphs are robust to both edge and vertex additions. Also, the class of K_h-minor-free graphs is vastly broader than the classes of planar and bounded genus graphs.

Most algorithmic results mentioned above for planar graphs can be generalized to bounded genus graphs [Epp03, DHT04, CC07, DHM10] using now-standard topological tools. For minor-free graphs, the 20-year graph minor project by Robertson and Seymour provides a deep understanding of their structures [RS83, RS04]. The Robertson-Seymour decomposition [RS03] has been used successfully to transfer almost all algorithmic tools in the first class (for coping with NP-hard problems) from planar graphs to K_h-minor-free graphs. However, the best-known algorithm for constructing the Robertson-Seymour decomposition has quadratic time [KKR12], despite prolonged efforts to simplify the proofs of Robertson and Seymour [KTW18, KTW20]. The quadratic time makes the Robertson-Seymour decomposition inapplicable to transfer results from the second class to minor-free graphs. Furthermore, the dependency on the minor size h is impractically huge even for a very small value of h. As a result, there have been far fewer algorithmic tools for designing truly subquadratic time algorithms in K_h-minor-free graphs. Most focus has been on finding separators, and hence r-divisions, in minor-free graphs in truly subquadratic time [RW09, KR10, WN11, WN14]. This deficiency motivates our work.

Research Goal. *Enriching the algorithmic toolkit for designing truly subquadratic time algorithms in K_h-minor-free graphs.*

Towards realizing our goal, we propose a systematic study of VC set systems (see Section 1.1 for definitions) and their applications in designing truly subquadratic time algorithms. Our work was directly inspired by two recent results; both led to several surprising algorithmic applications.

[1] We are referring to the optimization versions of decision problems in P and NP.

The first is by Li and Parter [LP19], who constructed a VC set system from a set of terminals lying on the outer face of a *planar graph*. However, it remains unclear how to extend their results to K_h-minor-free graphs since the notion of the outer face is not well-defined, and their proof makes heavy use of planarity. The second is by Ducoffe, Habib, and Viennot [DHV20], who designed the first truly subquadratic time algorithms for diameter and related problems in K_h-minor-free graphs via the VC set system of balls studied by Chepoi, Estellon, and Vaxes [CEV07]. However, the set system of balls is very difficult to work with algorithmically; this difficulty also manifests in the construction of Ducoffe, Habib, and Viennot [DHV20], resulting in complicated algorithms. Consequently, the running time of their algorithms degrades exponentially in the size of the minor.

We remark that both results [LP19, DHV20] only apply to *undirected graphs*, while our results extend to *directed graphs* as well, which are often much harder to work with. Indeed, we are not aware of any VC set system for directed graphs, let alone using them in algorithmic applications. The pioneering work of Chepoi, Estellon, and Vaxes [CEV07] for planar graphs and of Kranakis et al. [KKR$^+$97] for general graphs do not consider directed graphs.

1.1 VC Set Systems and Dimension

A *set system* is a pair $(U, \mathcal{F})$ where U is a ground set and $\mathcal{F}$ is a collection of subsets of U; we only write $\mathcal{F}$ when the ground set is clear from the context. We say that $Y \subseteq U$ is *shattered* by $\mathcal{F}$ if $\{Y \cap S : S \in \mathcal{F}\} = 2^Y$. That is, the intersections of Y and the sets in $\mathcal{F}$ contain every subset of Y. The *VC-dimension* of a set system $(U, \mathcal{F})$ is the size of the largest subset $Y \subseteq U$ shattered by $\mathcal{F}$. The notion of VC-dimension was introduced by Vapnik and Chervonenkis [VC71]. We say that $(U, \mathcal{F})$ is a *VC set system* if its VC-dimension is bounded by a fixed constant.

Let G be an edge-weighted and *undirected* graphs. For a vertex $v \in V$ and a non-negative real number r, denote by $B(v, r) = \{u : d_G(u, v) \leq r\}$ a ball of radius r centered at v. Let $\mathcal{B}(G) = \{B(v, r) : v \in V, r \in \mathbb{R}^+\}$ be the set of balls of all radii in G. Chepoi, Estellon and Vaxes [CEV07] showed that $\mathcal{B}(G)$ has VC-dimension at most 4 if G is planar and remarked that the same proof should extend to any K_h-minor-free graphs; the proof then was given in detail by Bousquet and Thomassé[BT15].

Theorem 1 (Chepoi, Estellon, and Vaxes [CEV07])**.** *If G is undirected and K_h-minor-free, then $(V, \mathcal{B}(G))$ has VC-dimension at most $h - 1$.*

Theorem 1 had been used exclusively in graph theory and combinatorics [CEV07, BC14, BT15] until very recently, Ducoffe, Habib, and Viennot [DHV20] exploited this result algorithmically. Specifically, they designed the first algorithm for computing the exact diameter, and its variants, of minor-free graphs in truly subquadratic time. They relied on a deep result of Haussler and Welzl [HW87], who showed that any VC set system admits a spanning path with sublinear *stabbing number*. They skillfully combined the low-stabbing spanning path technique with the r-division technique, a standard tool in designing algorithms in minor-free graphs on which almost all truly subquadratic time algorithms rely. Indeed, they had to work very hard to fit both techniques together (Lemma 5.2 in [DHV20]). However, there remain two undesirable aspects of their algorithm.

First, it is difficult to adapt their algorithms to other problems. One specific problem is computing the Wiener index, i.e., the sum of all-pairs distances. They wrote, "we currently do not see any way to extend our approach [...] to also compute their Wiener index in truly subquadratic time." The Wiener index problem was rooted in chemistry [Wie47] and has been studied extensively, e.g. see [CK97, CK09, WN09, Cab18, GKM$^+$21]. As we will see in Section 3.2.2, the Wiener

index problem can be readily handled by our technique. Second, the final running time degrades exponentially in h: $O(n^{2-\varepsilon_h})$ where $\varepsilon_h = 2^{-O(h)}$. (The precise value of ε_h is not given in [DHV20].)

In a completely different context, motivated by the diameter problem in the distributed CONGEST model, Li and Parter [LP19] set up a different VC set system from a fixed set of terminals S. In their paper, they only studied a special case where S contains vertices *on the outer face of a planar graph*, though the definition applies to any S.

Definition 1 (Li-Parter [LP19])**.** *Let $M \subseteq \mathbb{R}$ be a set of real numbers. Let $S = \langle s_0, \ldots, s_{k-1} \rangle$ be a sequence of k vertices in an undirected and edge-weighted graph G. For every $v \in V$, define:*

$$X_v = \{(i, \Delta) : 1 \le i \le k-1, \Delta \in M, d_G(v, s_i) - d_G(v, s_{i-1}) \le \Delta\} \tag{1}$$

Let $\mathcal{LP}_{G,M}(S) = \{X_v : v \in V\}$ be a collection of subsets of the ground set $[k-1] \times M$.

The complicated-looking set X_v intuitively encodes the (approximate) distance from v to each vertex in S: the pair $(i, \Delta) \in X_v$ indicates that $d_G(v, s_i) \le d_G(v, s_{i-1}) + \Delta$. Thus, given $d_G(v, s_0)$ and all the pairs (i, Δ) in X_v, we can iteratively recover an upper bound on $d_G(v, s_i)$ for any $i \in [2, k]$. Depending on the choice of M, we might recover the exact or approximate distance $d_G(v, s_i)$. Li and Parter showed that $\mathcal{LP}_{G,M}(S)$ is a VC set system for a special setting of G and S (Theorem 3.7 in [LP19]):

Theorem 2 (Li-Parter [LP19])**.** *Let G be an edge-weighted, undirected, planar graph. Let S be a set of k vertices ordered clockwise on the outer face of G. For any $M \subseteq \mathbb{R}$, $([k-1] \times M, \mathcal{LP}_{G,M}(S))$ has VC-dimension at most 3.*

As $\mathcal{LP}_{G,M}(S)$ is capable of encoding the graph distances directly into the set system, it is much easier to use than $\mathcal{B}(G)$ in algorithm design. Specifically, it was instrumental in solving several problems in planar graphs: metric compression and distributed approximate diameter computation [LP19], exact distance oracles [FHMWN20], and approximate distance oracles [Le23], despite the restriction on S and G. A natural open problem is: can we remove the restriction on S and G?

1.2 Our Results and Techniques

We propose several set systems in K_h-minor-free graphs: variants of $\mathcal{LP}_{G,M}(S))$ in both undirected graphs and digraphs, the set system of balls for digraphs, and a set system induced by shortest paths in digraphs. (We refer to directed graphs as *digraphs*.) We obtain both negative and positive results for these systems. We hope for a "unified" view of existing VC set systems to reconcile their differences and guide the development of new ones. Two VC set systems $\mathcal{B}_G$ and $\mathcal{LP}_{G,M}(S)$ differ in three aspects: (i) the ease of application, (ii) the scope of application —one for minor-free while the other for planar graphs —and (iii) the proof techniques. In terms of proof techniques, Chepoi, Estellon, and Vaxes [CEV07] construct a K_5-minor directly assuming (for contradiction) that there is a large set of vertices shattered by $\mathcal{B}_G$ in a planar graph G; the end result is an elegant proof that can be easily extended to K_h-minor-free graphs (as done by Bousquet and Thomassé [BT15])). We call this proof technique *minor-building proof*. The proof of Li and Parter exhaustively considers different crossing patterns of paths between the terminals and hence heavily relies on the assumption that G is planar and S on the outer face to make the number of crossing patterns manageable.

The proofs of our positive results in this work are minor-building, though each VC set system needs its own twist in the proof. Our proofs inherit the simplicity and elegance of the minor-building technique, and are applicable to both undirected graphs and digraphs, as described in

Section 3.1, Section 4.1 and Section 4.2. The minor-building proof technique is also instructive in developing new set systems. Indeed, in an (unsuccessful) attempt to reprove the result by Li and Parter (Theorem 2) using the minor-building technique, we came up with a VC set system slightly different from $\mathcal{LP}_{G,M}(S)$), which retains all the aforementioned strengths of $\mathcal{LP}_{G,M}(S)$) while addressing its two weaknesses: G can be any K_h-minor-free graph, and S could be anywhere in the graph.

Definition 2. *Let $M \subseteq \mathbb{R}, G$, and S as in Definition 1. For every $v \in V$, define:*

$$\widehat{X}_v = \{(i, \Delta) : 1 \le i \le k-1, \Delta \in M, d_G(v, s_i) - d_G(v, s_0) \le \Delta\} \tag{2}$$

Let $\widehat{\mathcal{LP}}_{G,M}(S) = \{\widehat{X}_v : v \in V\}$ be a collection of subsets of the ground set $[k-1] \times M$.

$\widehat{X}_v$ differs X_v (Equation (1)) in the highlighted term: it uses $d_G(v, s_i) - d_G(v, s_0)$ instead of $d_G(v, s_i) - d_G(v, s_{i-1})$. The difference, while superficially small, is technically important for the minor-building proof technique; see Remark 1 for a more formal discussion of why the minor-building proof technique fails for the set system $\mathcal{LP}_{G,M}(S)$. This leads to our first main result:

Theorem 3. *Let S be any set of vertices on an edge weighted, undirected K_h-minor-free graph G. Let $M \subseteq \mathbb{R}$ be any set of real numbers. Then $\widehat{\mathcal{LP}}_{G,M}(S)$ has VC-dimension at most $h-1$.*

Here we sketch key ideas of our proof. In the prior minor-building techniques for the set system of balls, a crucial step is to choose the shattering family of sets, which is the set of balls that shatters a set of vertices of size h. There could be many such choices, and choosing the right tie-breaking scheme for these balls is important: Chepoi, Estellon, and Vaxes [CEV07] broke ties by the sum of distances to be minimum, while Bousquet and Thomassé[BT15] did so by the radii of the balls. However, $\widehat{\mathcal{LP}}_{G,M}(S)$ is very different from a set system of balls, and we have to choose a different tie-breaking scheme for the shattering family of sets. It turns out that by defining $\widehat{X}_v$ as in Definition 2, we could choose a tie-breaking scheme using the Isolation Lemma [VV86]. The Isolation Lemma has been used in breaking ties in different applications, e.g. see [VV86, Eri10, MNNW18, CCE13, BP21], and we expect that this lemma will be used more in future work involving the minor-building technique.

In all applications of the VC set system $\mathcal{LP}_{G,M}(S)$ in planar graphs that we are aware of, including those mentioned in [LP19, FHMWN20, Le23], we can use $\widehat{\mathcal{LP}}_{G,M}(S)$ while obtaining the same, or sometimes stronger, guarantees. For example, we could derive a metric compression scheme with almost the same guarantees obtained by Li and Parter [LP19] but without the assumption that S must be on the outer face and furthermore, G could be any minor-free graphs; see Section 3.2.4.

Beyond planar graphs, which is our Research Goal mentioned above, we construct a distance oracle (see Section 2 for the definition) for *unweighted* K_h-minor-free graphs with truly subquadratic space and *constant query time*. This is the first oracle in K_h-minor-free graphs achieving truly subquadratic space-query time product, though many such oracles were known in planar graphs[2] years ago [FR01, MS12, CADWN17, GMWWN18, CGMW19, LP21]. Furthermore, our oracle can also be constructed in truly subquadratic time. ($\tilde{O}$ notation hides a poly-logarithmic factor in n.)

Corollary 1. *Let $G = (V, E)$ be an unweighted K_h-minor-free graph. We can construct an exact distance oracle for G with $\tilde{O}(n^{2-\frac{1}{3h-1}})$ space and $O(1)$ query time. The construction time of our oracle is $\tilde{O}(n^{2-\frac{1}{3h-1}})$.*

[2]It might be possible to extend some distance oracles with truly subquadratic space-query product from planar graphs to bounded genus graphs; however, we are not aware of any prior paper in this direction.

Our oracle in Corollary 1 is obtained by tailoring the construction of Fredslund-Hansen, Mozes, and Wulff-Nilsen to K_h-minor-free graphs and applying Theorem 3 to bound the number of distance patterns; we refer readers to Section 3.2.3 for more details.

Using Theorem 3, we resolve an open problem left by Ducoffe, Habib, and Viennot [DHV20]: computing the Wiener index in any K_h-minor-free graph in truly subquadratic time. We also improve the truly subquadratic time algorithm for computing all-vertices eccentricities and diameter in unweighted K_h-minor-free graphs by Ducoffe, Habib, and Viennot [DHV20] from $n^{2-1/2^{O(h)}}$ to $\tilde{O}(n^{2-\frac{1}{3h-1}})$.

Corollary 2. *Let $G = (V, E)$ be an unweighted K_h-minor-free graph. We can compute the eccentricities of all vertices, the diameter, and the Wiener index of G in $\tilde{O}(n^{2-\frac{1}{3h-1}})$ time.*

We remark that a truly subquadratic running time of the form $2^{o(h)}n^{2-\epsilon}$ for any fixed constant $\epsilon > 0$ for computing diameter in unweighted K_h-minor-free graphs is unlikely due to a conditional lower bound by Abboud, Williams, and Wang [AWW16], which holds even in a special case of graphs of treewidth at most h.

We now describe our results for *digraphs*. Let $d_G(u \to v)$ denotes the distance *from u to v* in a digraph G. It might be that $d_G(u \to v) \neq d_G(v \to u)$. Analogous to Definition 2, we define a set system, denoted by $\overrightarrow{\mathcal{LP}}_{G,M}(S)$.

Definition 3. *Let $M \subseteq \mathbb{R}$ and $S = \{s_0, s_2, \ldots, s_{k-1}\}$ be in Definition 1, but $G = (V, E)$ now is an edge-weighted digraph. For every $v \in V$, let:*

$$\overrightarrow{X}_v = \{(i, \Delta) : 1 \leq i \leq k-1, \Delta \in M, d_G(v \to s_i) - d_G(v \to s_0) \leq \Delta\} \tag{3}$$

We define $\overrightarrow{\mathcal{LP}}_{G,M}(S) = \{\overrightarrow{X}_v : v \in V\}$.

Our second main result is to show that $\overrightarrow{\mathcal{LP}}_{G,M}(S)$ is a VC set system in K_h-minor-free digraphs. (A digraph is K_h-minor-free if its underlying undirected graph is K_h-minor-free.)

Theorem 4. *Let S be any set of vertices on an edge weighted K_h-minor-free digraph G. Let $M \subseteq \mathbb{R}$ be any set of real numbers. Then $\overrightarrow{\mathcal{LP}}_{G,M}(S)$ has VC-dimension at most h^2.*

The VC-dimension bound in Theorem 4 is quadratic instead of linear as in Theorem 3. Our proof of Theorem 4 is also minor-building. However, the main difficulty in the directed case is that two directed shortest paths could intersect an arbitrary number of times (in different directions). In the undirected case, we rely on the fact that two shortest paths intersect at most once, as long as we choose a consistent tie-breaking scheme. The fact that directed paths can intersect in a very complicated way makes the minor construction in digraphs more difficult, and we settle on a looser bound. To construct a minor, we group the vertices into h groups, and loosely speaking, we show that how to choose directed paths between groups so that the paths are vertex disjoint.

We use Theorem 4 to design first truly subquadratic time algorithm for computing diameter and eccentricity for unweighted K_h-minor-free digraphs. Previously, truly subquadratic time algorithms for these problems were only known for planar digraphs [Cab18, GKM+21].

Corollary 3. *Let $G = (V, E)$ be an unweighted K_h-minor-free digraph. We can compute the diameter and all-vertex eccentricities of G in $\tilde{O}(n^{2-1/(3h^2+6)})$ time.*

Designing the truly subquadratic time algorithm for computing diameter and all-vertex eccentricities of digraphs in Corollary 3 is much more difficult than their undirected counterparts in Corollary 2. The algorithm for undirected graphs is based on the notion of *patterns*: each pattern is intuitively a vector of distances from a vertex in the graph to the boundary of a subgraph; the formal definition is given in Equation (12). Two nice properties of patterns in undirected graphs: (i) there is only a polynomial number of them, and (ii) the distance from a vertex u to a vertex v in a connected subgraph of H can be defined in terms of the distance from the pattern of u to v. (We have not defined the notion of distance between a pattern and a vertex; for now it suffices to know that one could define such a notion.) In digraphs, property (i) breaks down completely, and the reason is perhaps unsurprising: the triangle inequality does not hold in digraphs —the asymmetric triangle inequality does not suffice. Instead, we introduce *infinite patterns* where we allow entries with $\pm + \infty$ values. For infinite patterns, we are able to obtain property (i). However, property (ii) fails for infinite patterns. We resolve this by looking at all the distances from the pattern to all vertices of H at once, and we are able to extract the maximum distance from these distances. Thus, we are still able to solve the diameter and all-vertices eccentricities problems in truly subquadratic time. Unfortunately, we are not able to compute the Wiener index in truly subquadratic time using infinite patterns, and we leave this as an open problem for future work.

In undirected graphs, we can use VC dimension bound on $\widehat{\mathcal{LP}}_{G,M}(S)$ to construct an exact distance oracle with truly subquadratic space and constant query time (Corollary 1). However, we are unable to use the VC dimension bound on $\overrightarrow{\mathcal{LP}}_{G,M}(S)$ to obtain an analogous result for digraphs. This is because the notion of patterns does not work, and the infinite patterns we introduce are not useful in decoding distances. We work around the problem in our third main result. Specifically, let $\overrightarrow{B}(v,r) = \{u : d_G(v \to u) \leq r\}$, and:

$$\overrightarrow{\mathcal{B}}(G) = \{\overrightarrow{B}(v,r) : v \in V\} \tag{4}$$

Theorem 5. *If G is a K_h-minor-free digraph, then $\overrightarrow{B}(G)$ has VC-dimension at most $h-1$.*

We then develop a new technique to exploit the VC set system of directed balls. Our technique fits naturally with the r-division of K_h-minor-free digraphs. Specifically for each cluster in the r-division, we look at all the restrictions of balls in the cluster; the balls are centered at vertices outside the cluster. We exploit Theorem 5 in showing that there are only a polynomial number of different restrictions. Thus, we could keep all of them, along with side information, in a table. Our technique gives the first exact distance oracle for digraphs with truly subquadratic space-query product. We remark that it is unclear how to combine the low-stabbing spanning path technique by Ducoffe, Habib, and Viennot [DHV20] with r-division to construct an exact distance with the same guarantee (even in undirected graphs).

Corollary 4. *Let $G = (V, E)$ be an unweighted K_h-minor-free digraph. We can construct an exact distance oracle for G with $\tilde{O}(n^{2-\frac{1}{2(h-2)}})$ space and $O(\log(n))$ query time.*

We now turn to a negative result. We study set systems whose ground set is the set of edges in digraphs. While there could be many ways to define a set system of edges [KKR$^+$97], the system of shortest path trees is of special interest to us: such a set system, if has bounded VC-dimension, could be used to compute the Wiener index in truly subquadratic time—resolving the problem we pose above—speed up exact diameter computation, construct exact distance oracles for digraphs

with $O(1)$ query time, and potentially has many more applications. Unfortunately, we show that the set system does not have bounded VC dimension. More formally, given a digraph $G = (V, E)$, let τ_v be the shortest path tree rooted at v. In the construction of shortest path trees in G, ties are broken consistently. (If ties are not broken consistently, it is fairly easy to show that the set system of edges introduced below will not have bounded VC dimension.) We think of τ_v as a subset of the E, and define:

$$\overrightarrow{\mathcal{SP}}(G) = \{\tau_v : v \in V\} \tag{5}$$

As our fourth main result, we show that the set system $(E, \overrightarrow{\mathcal{SP}}(G))$ does not have bounded VC dimension even in unweighted planar digraphs.

Theorem 6. *For any constant integer $r \geq 1$, there exists an unweighted planar digraph $G = (V, E)$ and a subset $X \subseteq E$ of size r such that X is shattered by $\overrightarrow{\mathcal{SP}}(G)$.*

Lastly, we briefly mention two other directions which we do not explore in this paper as they are out of scope. The first direction is to explore the applications of our VC dimension results in solving graph-theoretic problems. There have been several works on applying the prior VC dimension results by Chepoi, Estellon, and Vaxes (Theorem 1), for example [BT15, BC14, BBE+21], and by Li and Parter (Theorem 2), for example [JR23], to understand structures of planar and minor-free graphs. We believe that our results will also be applicable in this direction. The second direction is to consider graphs beyond minor-free, such as graphs with polynomial expansion or nowhere dense graphs, as studied in the work by Ducoffe, Habib, and Viennot [DHV20]. As far as we can see, our results could also be extensible to graphs with polynomial expansion and get algorithmic applications along the line of Ducoffe, Habib, and Viennot [DHV20]. However, it seems to us that one has to work harder to be able to extend our results to nowhere dense graphs.

2 Preliminaries

We use graphs to refer to *undirected graphs*, while directed graphs will be called *digraphs*. We reserve V and E for the vertex set and edge set of G, respectively. For any other graph H, we denote it vertex set by $V(H)$ and edge set by $E(H)$. We denote by $\pi(u, v, G)$ a shortest path between u and v in a graph G. If G is a digraph, then we denote by $\pi(u \to v, G)$ the directed shortest path from u to v. If the graph is clear from the context, we simply denote the shortest paths by $\pi(u, v)$ and $\pi(u \to v)$, respectively.

The *eccentricity* of a vertex u, denoted by $\mathsf{ecc}(u)$ in a graph G is $\mathsf{ecc}(u) = \max_{v \in V} d_G(u, v)$. The diameter of G is the maximum eccentricity: $\max_{u \in V} \mathsf{ecc}(u)$. The Wiener index of a graph G is defined to be the sum of all pairwise distances: $\frac{1}{2} \sum_{u \in V} \sum_{v \in V} d_G(u, v)$. The Wiener index, eccentricity, and diameter of digraphs are defined similarly, with $d_G(u \to v)$ being used in place of $d_G(u, v)$.

We say that a subgraph H of G is *induced* if every edge in G between two vertices in H also appears in H. We will use the *r-division* of minor-free graphs in our algorithms. A *cluster* is a *connected, induced subgraph* of G. Let C be a cluster of G. We say that a vertex $v \in C$ is a *boundary vertex* if v is adjacent to a vertex $u \in V \setminus V(C)$. We use ∂C to denote the set of all boundary vertices of C. An *r-division* of G is a collection $\mathcal{R}$ of clusters G such that every cluster $R \in \mathcal{R}$ has at most r vertices.

Our definition of r-division is somewhat non-standard in the sense that we do not have $O(\sqrt{|V(R)|})$ bound on the number of boundary vertices of each cluster R. It is called *r-clustering* in the paper of Wulff-Nilsen [WN11]. Here we still call it an r-division as most of the intuition in the use of r-clustering comes from r-division.

Wulff-Nilsen [WN11] showed that one can construct an r-division of any K_h-minor-free graphs such that the total number of boundary vertices, counted with multiplicity, is small. We note that in our applications, it is important that each cluster $R \in \mathcal{R}$ is a connected subgraph of G.

Lemma 1 (Wulff-Nilsen, Lemma 2 [WN11])**.** *Let G be a K_h-minor-free graphs with n vertices, and $r \in [Ch^2 \log n, n]$ for a sufficiently large constant C. For any fixed constant $\epsilon > 0$, we can construct in time $O(n^{1+\epsilon}\sqrt{r})$ an r-division, say $\mathcal{R}$, of G such that (a) $\sum_{R\in\mathcal{R}} |\partial R| = \tilde{O}(nh/\sqrt{r})$, and (b) every cluster $R \in \mathcal{R}$ has $|V(R)| \le r$ and $|\partial R| = \tilde{O}(h\sqrt{r})$. Furthermore, the number of clusters in $\mathcal{R}$ is at most $\tilde{O}(hn/\sqrt{r})$.*

One could obtain an r-division with a number of clusters being $\tilde{O}(hn/r)$ with a larger running time. For us, the weaker bound in Lemma 1 suffices.

In many of our results, we will use the following well-known Sauer–Shelah Lemma, which gives a polynomial upper bound on the size of a VC set system.

Lemma 2 (Sauer–Shelah Lemma)**.** *Let $\mathcal{F}$ be a family of subsets of a ground set with n elements. If VC-dimension of $\mathcal{F}$ is at most k, then $|\mathcal{F}| = O(n^k)$.*

A distance oracle for a graph G is a compact data structure that given any two vertices u and v, returns $d_G(u, v)$ quickly. The query time is the maximum time it takes to answer a query over all pairs of vertices. There is often a trade-off between the space of the oracle and the query time.

3 VC Dimension of Undirected Graphs and Applications

3.1 VC dimension of $\widehat{\mathcal{LP}}_{G,M}$

In this section, we fix $G = (V, E)$ to be an undirected K_h-minor-free graph. We first prove Theorem 3, which we restate below.

Theorem 3. *Let S be any set of vertices on an edge weighted, undirected K_h-minor-free graph G. Let $M \subseteq \mathbb{R}$ be any set of real numbers. Then $\widehat{\mathcal{LP}}_{G,M}(S)$ has VC-dimension at most $h - 1$.*

Our proof is by contradiction. Suppose that there is a set Y of size h that is shattered by $\widehat{\mathcal{LP}}_{G,M}$. W.l.o.g., we assume that $Y = \{(s_1, \Delta_1), \ldots, (s_h, \Delta_h)\}$. We first observe that:

Observation 1. *$s_i \neq s_j$ for any $1 \le i \neq j \le h$.*

Proof. Suppose otherwise, that $s_i = s_j$. W.l.o.g, we assume that $\Delta_i < \Delta_j$. This means if $(s_i, \Delta_i) \in \widehat{X}_v$ for some vertex $v \in V$, then $(s_i, \Delta_j) \in \widehat{X}_v$, since $d_G(v, s_i) \le d_G(v, s_0) + \Delta_i$ implies that $d_G(v, s_i) \le d_G(v, s_0) + \Delta_j$. However, since $\widehat{\mathcal{LP}}_{G,M}$ shatters Y, by definition of shattering, there exists a set $\widehat{X}_v$ containing (s_i, Δ_i) but not (s_i, Δ_j), a contradiction. □

For every two elements (s_i, Δ_i) and (s_j, Δ_j) with $i \neq j$ in Y, let v_{ij} be a vertex such that $\{(s_i, \Delta_i), (s_j, \Delta_j)\} = \widehat{X}_{v_{ij}} \cap Y$.

Let $\widehat{G}$ be a graph obtained from G by adding (tiny) perturbed weights to edges of G in such a way that (i) shortest paths in $\widehat{G}$ between vertices are unique and (ii) every shortest path in $\widehat{G}$ is also a shortest path in G. (Some shortest path in G may no longer be a shortest in $\widehat{G}$.) We can think of $\widehat{G}$ as providing a tie-breaking scheme for shortest paths in G. The perturbation exists by the Isolation Lemma [VV86].

Definition 4. *We define vertex t_{ij} to be the vertex in $\pi(v_{ij}, s_i, \widehat{G}) \cup \pi(v_{ij}, s_j, \widehat{G})$ such that:*

(a) $d_G(v_{ij}, t_{ij}) + d_G(t_{ij}, s_i) \leq \Delta_i + d_G(v_{ij}, s_0)$ and $d_G(v_{ij}, t_{ij}) + d_G(t_{ij}, s_j) \leq \Delta_j + d_G(v_{ij}, s_0)$.

(b) the sum of distance $d_{\widehat{G}}(t_{ij}, s_i) + d_{\widehat{G}}(t_{ij}, s_j)$ is minimum.

Note that the distances in Item (a) of Definition 4 are w.r.t. graph G while the distances in Item (b) are w.r.t. $\widehat{G}$. By the definition of $\widehat{G}$, $\pi(s_i, v_{ij}, \widehat{G})$ is also a shortest path in G; sometimes we abuse notation by using $\pi(s_i, v_{ij}, \widehat{G})$ to refer to its corresponding shortest path in G. We remark that t_{ij} exists since v_{ij} is a possible choice for t_{ij} satisfying (a). A good, but not accurate, interpretation of t_{ij} to keep in mind is that when the two shortest paths $\pi(s_i, v_{ij}, \widehat{G})$ and $\pi(s_i, v_{ij}, \widehat{G})$ shares the same vertex other than v_{ij}, then t_{ij} is the common vertex furthest from v_{ij}; this would be the case if we restrict t_{ij} to be in $\pi(v_{ij}, s_i, \widehat{G}) \cap \pi(v_{ij}, s_j, \widehat{G})$ instead of $\pi(v_{ij}, s_i, \widehat{G}) \cup \pi(v_{ij}, s_j, \widehat{G})$ as in Definition 4. Indeed, the role of t_{ij} in the proof is subtler than just being the furthest common vertex.

Claim 1. *$\pi(s_i, t_{ij}, G)$ and $\pi(s_j, t_{ij}, G)$ are internally disjoint.*

Proof. Suppose otherwise; there would be a vertex $x_{ij} \in (\pi(v_i, t_{ij}, G) \cap \pi(v_j, t_{ij}, G)) \setminus \{t_{ij}\}$; see Figure 1(a). Then, we have:

$$\begin{aligned} d_G(v_{ij}, x_{ij}) + d_G(x_{ij}, s_i) &\leq d_G(v_{ij}, t_{ij}) + d_G(t_{ij}, x_{ij}) + d_G(x_{ij}, s_i) \\ &= d_G(v_{ij}, t_{ij}) + d_G(t_{ij}, s_i) \\ &\leq \Delta_i + d_G(v_{ij}, s_0) \qquad \text{(by Item (a) in Definition 4)} \end{aligned}$$

By the same argument, we have:

$$d_G(v_{ij}, x_{ij}) + d_G(x_{ij}, s_j) \leq \Delta_j + d_G(v_{ij}, s_0).$$

which means x_{ij} satisfies Item (a) in Definition 4. Furthermore, as $x_{ij} \in (\pi(v_i, t_{ij}, G) \cap \pi(v_j, t_{ij}, G)) \setminus \{t_{ij}\}$, we have:

$$\begin{aligned} d_{\widehat{G}}(x_{ij}, s_i) + d_{\widehat{G}}(x_{ij}, s_j) &= d_{\widehat{G}}(t_{ij}, s_i) - d_{\widehat{G}}(t_{ij}, x_{ij}) + d_{\widehat{G}}(t_{ij}, s_j) - d_{\widehat{G}}(t_{ij}, x_{ij}) \\ &< d_{\widehat{G}}(t_{ij}, s_i) + d_{\widehat{G}}(t_{ij}, s_j) \end{aligned}$$

contradicting the minimality of t_{ij} by Item (b) in Definition 4. □

We define the *bunch of each vertex s_i* (see Figure 1(b)):

$$\mathsf{Bunch}(s_i) = \cup_{j \neq i} \pi(s_i, t_{ij}, \widehat{G}) \tag{6}$$

We say that t_{ij} is an *endpoint* of $\mathsf{Bunch}(s_i)$ if it has degree 1 in the subgraph $\mathsf{Bunch}(s_i)$. Otherwise, we say that t_{ij} is an internal vertex of $\mathsf{Bunch}(s_i)$. (One case where t_{ij} is not an endpoint is when the path $\pi(s_i, t_{ij}, \widehat{G})$ is a subpath of $\pi(s_i, t_{ij'}, \widehat{G})$ of another vertex $t_{ij'}$.)

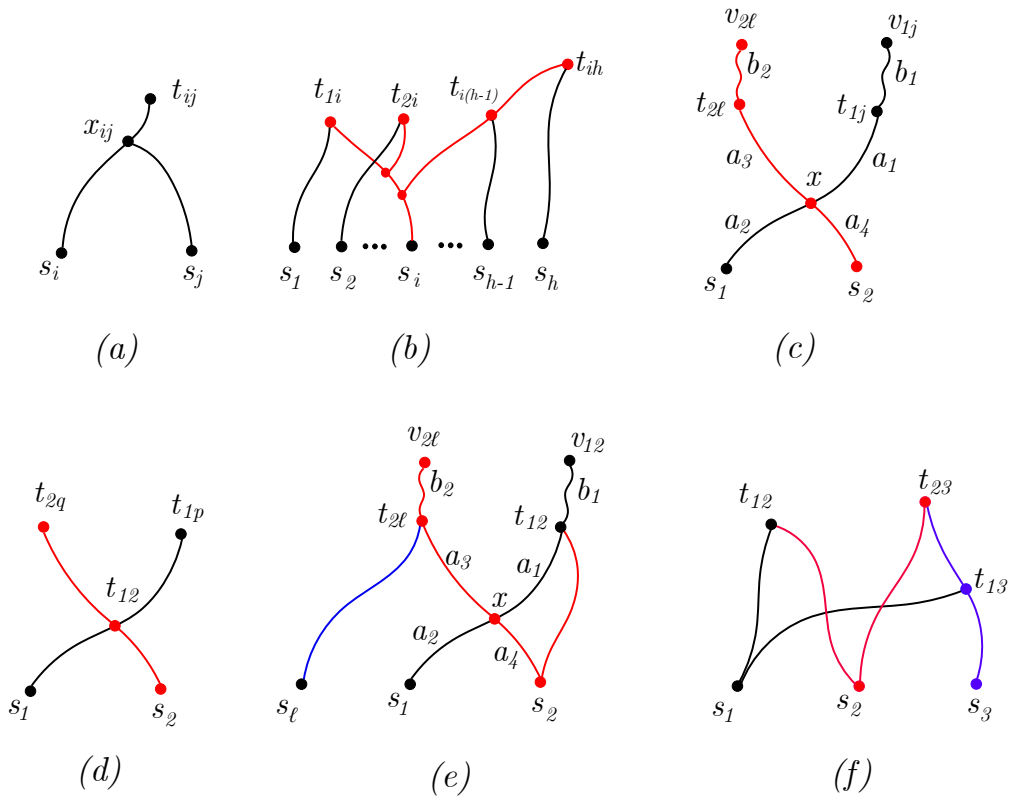

Figure 1: (a) $x_{ij} \in (\pi(s_i, t_{ij}, G) \cap \pi(s_j, t_{ij}, G)) \setminus \{t_{ij}\}$; (b) the bunch of s_i contains all red paths, $t_{i(h-1)}$ is an internal vertex of $\mathsf{Bunch}(s_i)$ but an endpoint of $\mathsf{Bunch}(s_{h-1})$; (c) $x \in \pi(s_1, t_{1j}, G) \cap \pi(s_2, t_{2k}, G)$; (d) Illustration for the proof that t_{12} must be an endpoint; (e) $\{x\} \in \pi(s_1, t_{12}, G) \cap \pi(s_2, t_{2\ell}, G)$; (f) A K_3 minor constructed from three bunches $\mathsf{Bunch}(s_1)$, $\mathsf{Bunch}(s_2)$, and $\mathsf{Bunch}(s_3)$; t_{12} will be contracted to s_1, t_{23} is contracted to s_2, and t_{13} is contracted to s_3.

Lemma 3. *For every $a \neq b$, $\mathsf{Bunch}(s_{k_a}) \cap \mathsf{Bunch}(s_{k_b}) = \{t_{ab}\}$. Furthermore, t_{ab} is either an endpoint of $\mathsf{Bunch}(s_{k_a})$, or an endpoint of $\mathsf{Bunch}(s_{k_b})$, or both.*

Proof. By the symmetry of s_i, we prove the lemma for $a = 1, b = 2$. Let $\pi(s_1, t_{1j}, \widehat{G})$ and $\pi(s_2, t_{2\ell}, \widehat{G})$ be paths in $\mathsf{Bunch}(s_1)$ and $\mathsf{Bunch}(s_2)$, respectively.

Claim 2. *If $\{j, \ell\} \cap \{1, 2\} = \varnothing$, then $\pi(s_1, t_{1j}, \widehat{G}) \cap \pi(s_2, t_{2\ell}, \widehat{G}) = \varnothing$.*

Proof. Suppose otherwise, there exists $x \in \pi(s_1, t_{1j}, \widehat{G}) \cap \pi(s_2, t_{2k}, \widehat{G})$; see Figure 1(c). Let:

$$\begin{aligned} a_1 &= d_G(t_{1j}, x) & a_2 &= d_G(x, s_1) \\ a_3 &= d_G(t_{2\ell}, x) & a_4 &= d_G(x, s_2) \\ b_1 &= d_G(v_{1j}, t_{1j}) & b_2 &= d_G(v_{2\ell}, t_{2\ell}) \end{aligned}$$

By definition of t_{1j} (Item (a) in Definition 4), it holds that $b_1 + a_1 + a_2 \leq \Delta_1 + d_G(v_{1j}, s_0)$. For the same reason, we have $b_2 + a_3 + a_4 \leq \Delta_2 + d_G(v_{2k}, s_0)$. It follows that:

$$a_1 + a_2 + a_3 + a_4 + b_1 + b_2 \leq \Delta_1 + \Delta_2 + d_G(v_{1j}, s_0) + d_G(v_{2\ell}, s_0) \tag{7}$$

On the other hand, $(s_2, \Delta_2) \notin \widehat{X}_{v_{1j}}$. Thus, $d_G(v_{1j}, s_2) > \Delta_2 + d_G(v_{1j}, s_0)$. By the triangle inequality, $b_1 + a_1 + a_4 \geq d_G(v_{1j}, s_2)$, which implies that:

$$b_1 + a_1 + a_4 > \Delta_2 + d_G(v_{1j}, s_0) \tag{8}$$

By the same argument, we have that $b_2 + a_2 + a_3 > \Delta_1 + d_G(v_{2\ell}, s_0)$. Combining with Equation (8), we get:

$$a_1 + a_2 + a_3 + a_4 + b_1 + b_2 > \Delta_1 + \Delta_2 + d_G(v_{1j}, s_0) + d_G(v_{2\ell}, s_0), \tag{9}$$

which contradicts Equation (7). Thus, x does not exist. □

Claim 3. *If $\{j, \ell\} \cap \{1, 2\} \neq \varnothing$, then $\pi(s_1, t_{1j}, \widehat{G}) \cap \pi(s_2, t_{2\ell}, \widehat{G}) \subseteq \{t_{12}\}$, and that t_{12} is either an endpoint of $\mathsf{Bunch}(s_1)$ or $\mathsf{Bunch}(s_2)$ or both.*

Proof. If $t_{12} \in \pi(s_1, t_{1j}, \widehat{G}) \cap \pi(s_2, t_{2\ell}, \widehat{G})$, then we claim that t_{12} must be an endpoint of $\mathsf{Bunch}(s_1)$ or $\mathsf{Bunch}(s_2)$ or both. Suppose otherwise; that is t_{12} is not an endpoint of either $\mathsf{Bunch}(s_1)$ or $\mathsf{Bunch}(s_2)$. It means that there are two paths $\pi(s_1, t_{1a}, \widehat{G})$ and $\pi(s_2, t_{2b}, \widehat{G})$ such that $\{t_{12}\} \subseteq \pi(s_1, t_{1a}, \widehat{G}) \cap \pi(s_2, t_{2b}, \widehat{G})$ and that $\{a, b\} \cap \{1, 2\} = \varnothing$; see Figure 1 (d). The existence of such two paths contradicts Claim 2.

We are now proving that $\pi(s_1, t_{1j}, \widehat{G}) \cap \pi(s_2, t_{2\ell}, \widehat{G}) \subseteq \{t_{12}\}$. Observe that if $j = 2$ and $\ell = 1$, then $t_{12} = \pi(s_1, t_{1j}, \widehat{G}) \cap \pi(s_2, t_{2\ell}, \widehat{G})$ by Claim 1. Thus, Claim 3 follows. It remains to consider two other cases: (i) $j = 2, \ell \neq 1$ or (ii) $j \neq 2, \ell = 1$. Both cases are symmetric, and hence w.l.o.g, we only consider case (i).

Suppose that there exists $x \neq t_{12}$ such that $\{x\} \in \pi(s_1, t_{12}, \widehat{G}) \cap \pi(s_2, t_{2\ell}, \widehat{G})$ ($j = 2$ now); see Figure 1(e). We define $a_1, a_2, a_3, a_4, b_1, b_2$ as in Claim 2, specifically:

$$\begin{aligned} a_1 &= d_G(t_{12}, x) & a_2 &= d_G(x, s_1) \\ a_3 &= d_G(t_{2\ell}, x) & a_4 &= d_G(x, s_2) \\ b_1 &= d_G(v_{12}, x) & b_2 &= d_G(v_{2\ell}, x) \end{aligned}$$

By definition of t_{12} (Item (a) in Definition 4), $b_1 + a_1 + a_2 \leq \Delta_1 + d_G(v_{12}, s_0)$. By the same argument, $b_2 + a_3 + a_4 \leq \Delta_2 + d_G(v_{2\ell}, s_0)$. Thus,

$$b_1 + b_2 + a_1 + a_2 + a_3 + a_4 \leq \Delta_1 + \Delta_2 + d_G(v_{12}, s_0) + d_G(v_{2\ell}, s_0). \tag{10}$$

Since $(s_1, \Delta_1) \notin \widehat{X}_{v_{2\ell}}$, $d_G(v_{2\ell}, s_1) > \Delta_1 + d_G(v_{2\ell}, s_0)$. By the triangle inequality, we have that $b_2 + a_3 + a_2 > \Delta_2 + d_G(v_{2k}, s_0)$. Thus, by Equation (10), $b_1 + a_1 + a_4 \leq \Delta_2 + d_G(v_{12}, s_0)$. In summary, we have:

$$\begin{aligned} b_1 + a_1 + a_2 &\leq \Delta_1 + d_G(v_{12}, s_0) \\ b_1 + a_1 + a_4 &\leq \Delta_2 + d_G(v_{12}, s_0) \end{aligned}$$

By the triangle inequality, we have that $d_G(v_{12}, x) \leq b_1 + a_1$. Recall that $a_2 = d_G(x, s_1)$ and $a_4 = d_G(x, s_2)$. It follows that:

$$\begin{aligned} d_G(v_{12}, x) + d_G(x, s_1) &\leq \Delta_1 + d_G(v_{12}, s_0) \\ d_G(v_{12}, x) + d_G(x, s_2) &\leq \Delta_2 + d_G(v_{12}, s_0) \end{aligned} \tag{11}$$

Thus, x satisfies Item (a) of Definition 4. We now show that $d_{\widehat{G}}(x,s_1)+d_{\widehat{G}}(x,s_2) < d_{\widehat{G}}(t_{12},s_1)+d_{\widehat{G}}(t_{12},s_2)$, which will give a contradiction by the choice of t_{12} in Item (b) of Definition 4.

Observe that by a triangle inequality, $d_G(x,s_2) \le d_G(x,t_{12}) + d_G(t_{12},s_2)$. Since shortest paths are unique in $\widehat{G}$, $d_{\widehat{G}}(x,s_2) < d_{\widehat{G}}(x,t_{12}) + d_{\widehat{G}}(t_{12},s_2)$. (See Figure 1(e).) Since $\pi(t_{12},s_{k_1},\widehat{G})[x,s_1]$ is a shortest path in $\widehat{G}$, $d_{\widehat{G}}(x,s_1) = d_{\widehat{G}}(t_{12},s_1) - d_{\widehat{G}}(x,t_{12})$. It follows that:

$$\begin{aligned} d_{\widehat{G}}(x,s_1) + d_{\widehat{G}}(x,s_2) &< d_{\widehat{G}}(t_{12},s_1) - d_{\widehat{G}}(x,t_{12}) + d_{\widehat{G}}(x,t_{12}) + d_{\widehat{G}}(t_{12},s_2) \\ &= d_{\widehat{G}}(t_{12},s_1) + d_{\widehat{G}}(t_{12},s_2), \end{aligned}$$

as desired. □

We observe that Lemma 3 follows directly from Claim 2 and Claim 3. □

We now continue the proof of Theorem 3. Consider the subgraph $H = \cup_{1\le i\le h}\mathsf{Bunch}(s_i)$. We construct a K_h minor of H as follows (see Figure 1(f)). Let $\pi(s_i,t_{ij},G)$ be a path in $\mathsf{Bunch}(s_i)$ such that $i<j$. If t_{ij} is an endpoint of $\mathsf{Bunch}(s_j)$, then we contract $\pi(s_i,t_{ij},G)$ to s_i. Otherwise, we contract $\pi(s_i,t_{ij},G)\setminus\{t_{ij}\}$ to v_i; Lemma 3 implies that t_{ij} will be contracted to v_j. The resulting graph is a K_h-minor of H as the paths $\pi(s_i,t_{ij},G)$ and $\pi(s_j,t_{ij},G)$ for any $i\neq j$ are internally disjoint by Claim 1. This completes the proof of Theorem 3.

Remark 1. *We remark the following regarding Theorem 3:*

- *The proof of Theorem 3 breaks down if we apply it to the set system by Li and Parter in Definition 1. Specifically, in Equation* (7)*, $d_G(v_{1j},s_0)+d_G(v_{2\ell},s_0)$ will be replaced by $d_G(v_{1j},s_0)+d_G(v_{2\ell},s_1)$ while in Equation* (9)*, $d_G(v_{1j},s_0)+d_G(v_{2\ell},s_0)$ will be replaced by $d_G(v_{1j},s_1)+d_G(v_{2\ell},s_0)$, and hence we could not obtain a contradiction in the proof of Claim 2. The same happens to the proof of Claim 3.*

- *The VC dimension bound obtained by Li and Parter [LP19] is* 3 *for the setting of S on the outer face of a planar graph G, while our Theorem 3 gives VC dimension* 4*. However, we can modify the proof slightly to improve the VC dimension to* 3 *by only requiring that G excludes a K_h-minor where each vertex of the clique minor must correspond to a connected subgraph of G containing at least one vertex in S. We say that G is S-*restricted K_h-minor-free*. (A K_h-minor-free graph is S-restricted K_h-minor-free graph for any subset S.) The graph and the vertex set S considered in the setting of Li and Parter is S-restricted K_4-minor-free and hence Theorem 3 gives VC dimension bound of* 3*, matching the original bound of Li and Parter.*

3.2 Algorithmic Applications

In this section, we explore algorithmic applications of Theorem 3. *Graphs in this section are* unweighted *and hence the distances are unweighted distances.* We will use the notion of *patterns*, introduced by Fredslund-Hansen, Mozes, and Wulff-Nilsen [FHMWN20], though our pattern is defined slightly differently. Specifically, our definition rests on the VC set system $\widehat{\mathcal{LP}}_{G,M}$, while Fredslund-Hansen, Mozes, and Wulff-Nilsen relied on the VC set system $\mathcal{LP}_{G,M}$ by Li and Parter [LP19].

Let H be a connected, induced subgraph of G. Recall that ∂H denotes the set of all boundary vertices of H. Fix an arbitrary *sequence* σ_H of vertices of ∂H, which is a linear order of ∂H. We

write $\sigma_H = \langle s_0, s_1, \ldots, s_{|\partial H|-1} \rangle$. For each vertex $v \in V$, we define a *pattern* of v w.r.t σ_H, denoted by $\mathbf{p}_v$ be a $|\partial H|$ dimensional vector where:

$$\mathbf{p}_v[i] = d_G(v, s_i) - d_G(v, s_0) \quad \text{for every } 0 \le i \le |\partial H| - 1 \tag{12}$$

Note that $\mathbf{p}_v[0] = 0$ by definition. We bound the the number of all possible patterns w.r.t. σ_H.

Lemma 4. *H be a connected, induced subgraph of a K_h-minor-free graph G, and σ_H be an arbitrary sequence of vertices in ∂H. Let $P = \{\mathbf{p}_v : v \in V\}$ be the set of all patterns w.r.t. σ_H. Then $|P| = O((|\partial H| \cdot |V(H)|)^{h-1})$.*

Proof. Since H is connected, by the triangle inequality, $-(|V(H)| - 1) \le d_G(v, s_i) - d_G(v, s_0) \le |V(H)| - 1$. Let $M = \{-(|V(H)| - 1), \ldots, -1, 0, 1, \ldots, (|V(H)| - 1)\}$ and S be the set of all boundary vertices of H. Let $\bar{p}_v$ be a set obtained by *flattening* $\mathbf{p}_v$; that is, for each $i \in [1, |\partial H| - 1]$, we add to the set $\bar{p}_v$ a pair (i, Δ) for every $\Delta \in M$ such that $\Delta \ge \mathbf{p}_v[i]$. Observe by definition of $\widehat{X}_v$ in Equation (2) that $\bar{p}_v = \widehat{X}_v$. Thus, there is a bijection between the set of patterns P and $\widehat{\mathcal{LP}}_{G,M}$.

By the Sauer–Shelah Lemma (Lemma 2), we have $|\widehat{\mathcal{LP}}_{G,M}| = O((|S||M|)^{h-1}) = O((|\partial H| \cdot |V(H)|)^{h-1})$ as claimed. □

Let v be a vertex in H, and $\mathbf{p}$ be a pattern (of some vertex u) w.r.t. σ_H. We define the distance between v in $\mathbf{p}$, denoted by $d(\mathbf{p}, v)$, to be:

$$d(\mathbf{p}, v) = \min_{0 \le i \le |\partial H|-1} \{d_G(v, s_i) + \mathbf{p}[i]\} \tag{13}$$

The distance between a vertex and a pattern can be used to compute the distance between two vertices as shown by the following lemma, due to Fredslund-Hansen, Mozes, and Wulff-Nilsen [FHMWN20]. Since our definition of a distance between a pattern and a vertex in Equation (13) is slightly different from that of [FHMWN20], we include a proof for completeness.

Lemma 5 (Fredslund-Hansen, Mozes, and Wulff-Nilsen, Lemma 7 [FHMWN20])**.** *Let $u \in V \setminus V(H)$ be a vertex not in H, and $\mathbf{p}_u$ be the pattern of u w.r.t σ_H. Let v be a vertex in H. Then:*

$$d_G(u, v) = d_G(u, s_0) + d(\mathbf{p}_u, v) \tag{14}$$

Proof. Observe that for each boundary vertex s_i for $0 \le i \le |\partial H|$, $d_G(u, s_i) = \mathbf{p}_u[i] + d_G(u, s_0)$. Let s_ℓ be the boundary vertex in $\pi(u, v, G) \cap \partial H$; s_ℓ exists since H is an induced subgraph, and $u \notin V(H)$, $v \in V(H)$. Then:

$$\begin{aligned}
d_G(u, v) &= d_G(u, s_\ell) + d_G(s_\ell, v) = \min_{0 \le i \le |\partial H|-1} \{d_G(u, s_i) + d_G(s_i, v)\} \\
&= \min_{0 \le i \le |\partial H|-1} \{d_G(u, s_0) + \mathbf{p}_u[i] + d_G(s_i, v)\} \\
&= d_G(u, s_0) + \min_{0 \le i \le |\partial H|-1} \{d_G(v, s_i) + \mathbf{p}_u[i]\} \\
&= d_G(u, s_0) + d(\mathbf{p}_u, v) \ ,
\end{aligned}$$

as desired. □

In Section 3.2.1 and Section 3.2.2, we present algorithms to compute the diameter, eccentricities and the Wiener index. Our algorithm builds on an earlier algorithm by Wulff-Nilsen [WN09]. Here we use Lemma 4 to improve the running time to truly subquadratic time. In Section 3.2.3, we construct a distance oracle with truly subquadratic space and constant query time. The algorithm is almost the same as the algorithm by Fredslund-Hansen, Mozes, and Wulff-Nilsen [FHMWN20], except that we will use Lemma 4. In Section 3.2.4, we mention other algorithmic applications.

3.2.1 Diameter and Eccentricities

In this section, we show how to compute all-vertices eccentricities in truly subquadratic time as described in Corollary 2. Computing the diameter trivially follows by finding the maximum eccentricity in $O(n)$ time. The algorithm has three steps:

- **(Step 1).** Construct an r-division $\mathcal{R}$ of G for $r = n^{2/(3h-1)}$. For each cluster $R \in \mathcal{R}$, form a sequence of boundary vertices σ_R in an arbitrary way. Then compute the set of patterns w.r.t σ_R: $P_R = \{u \in V : \mathbf{p}_u\}$. We store P_R in a table $T_R^{(1)}$.
- **(Step 2).** For each cluster $R \in \mathcal{R}$ and each pattern $\mathbf{p} \in P_R$, find $v = \arg\max_{v \in V(R)} d(v, \mathbf{p})$. That is, v is the vertex that has maximum distance to $\mathbf{p}$ over all vertices in $V(R)$; we say that v is the *furthest vertex* from $\mathbf{p}$. We then store the distance $d(\mathbf{p}, v)$ in a table $T_R^{(2)}$ of R; the key to access $T_R^{(2)}$ is (the ID of) $\mathbf{p}$.
- **(Step 3).** We now compute $\mathsf{ecc}(u)$ for each vertex $u \in V$. For each cluster $R \in \mathcal{R}$, we compute the distance from u to the vertex $v \in R$ furthest from u, denoted by $\Delta(u, R)$, as follows. If $u \in R$, we then simply compute $\Delta(u, R)$ by using BFS. If $u \notin R$, let $\mathbf{p}_u$ be the pattern of u w.r.t σ_R computed in (Step 1). Let v be the furthest vertex from $\mathbf{p}_u$, computed in (Step 2). Then we return $\Delta(u, R) = d_G(u, s_0) + d(\mathbf{p}_u, v)$ where s_0 is the first vertex of σ_R. Finally, we compute $\mathsf{ecc}(u) = \max_{R \in \mathcal{R}} \Delta(u, R)$.

By Lemma 5 and the computation in (Step 2), $\Delta(u, R)$ computed in (Step 3) is the distance from u to its furthest vertex in R for every cluster R. Thus, in (Step 3) above, $\mathsf{ecc}(u)$ is correctly computed.

We now implement each step of the algorithm efficiently, assuming that h is a constant. We can assume $h \geq 4$, as (connected) K_3-minor-free graphs are trees and hence all problems mentioned here can be solved in linear time. Let B be the set of boundary vertices of the r-division $\mathcal{R}$: $B = \cup_{R \in \mathcal{R}} \partial R$. By Lemma 1, $|B| = \tilde{O}(n/\sqrt{r})$. Thus, we can find all BFS trees, each rooted at a vertex of B, in $\tilde{O}(n^2/r)$ time.

Observation 2. *Let $D(B, V) = \{d_G(b, v) : (b, v) \in B \times V\}$. Then $D(B, V)$ can be computed in time $\tilde{O}(n^2/\sqrt{r})$.*

By Lemma 1, each cluster $R \in \mathcal{R}$ has at most r vertices and $\tilde{O}(\sqrt{r})$ boundary vertices. The following is a direct corollary of Lemma 4.

Corollary 5. *$|P_R| = \tilde{O}(r^{3(h-1)/2})$ for every $R \in \mathcal{R}$.*

Proof. By Lemma 4, the number of patterns is $O((|\partial R| \cdot |V(R)|)^{h-1}) = \tilde{O}(r^{3(h-1)/2})$. □

Next, we bound the running time of (Step 1).

Lemma 6. *Given $D(B, V)$, we can implement (Step 1) in $\tilde{O}(n^2/\sqrt{r})$ time.*

Proof. First, by Lemma 1, $\mathcal{R}$ can be constructed in time $O(n^{1+\epsilon}\sqrt{r})$ for any fixed constant $\epsilon > 0$. As $h \geq 4$, $n^2/\sqrt{r} = n^{2-1/(3h-1)} = \Omega(n^{1.5})$ while $n^{1+\epsilon}\sqrt{r} = n^{1+\epsilon+1/(3h-1)} = n^{1.2+\epsilon}$. Thus, by choosing $\epsilon = 0.1$, we have $n^{1+\epsilon}\sqrt{r} = O(n^2/\sqrt{r})$. That is, $\mathcal{R}$ can be constructed in $\tilde{O}(n^2/\sqrt{r})$ time.

Next we compute P_R, which is initialized to be $\varnothing$. Then for each $u \in V$, we look up the distance from all vertices of ∂R to u in $D(B, V)$. Then we compute the pattern $\mathbf{p}_u$ from u to R, in $O(|\partial R|)$

time. We then add $\mathbf{p}_u$ to P_R if $\mathbf{p}_u$ is currently not in P_R; this check can be done in $O(|\partial R|)$ time using a trie data structure, say. The total running time to compute P_R is $O(n|\partial R|)$. Thus, the total running time of this step is $O(n\sum_{R\in\mathcal{R}}|\partial R|) = \tilde{O}(n^2/\sqrt{r})$ by Lemma 1. □

Lemma 7. *Given $D(B,V)$ and $\{P_R\}_{R\in\mathcal{R}}$, we can implement (Step 2) in $\tilde{O}(nr^{(3h-2)/2})$ time.*

Proof. For each pattern $\mathbf{p} \in P_R$, we can compute the distance $d(\mathbf{p},v)$ for each $v \in V(R)$ in time $O(|\partial R|) = \tilde{O}(\sqrt{r})$. Thus, finding the furthest vertex from $\mathbf{p}$ takes $\tilde{O}(\sqrt{r}|V(R)|)$ time. By Corollary 5, the running time to compute the table $T_R^{(2)}$ is $\tilde{O}(r^{1/2}|V(R)|r^{3(h-1)/2}) = \tilde{O}(r^{(3h-2)/2})|V(R)|$. Thus, the total running time of (Step 2) is $\tilde{O}(r^{(3h-2)/2})\sum_{R\in\mathcal{R}}|V(R)|) = \tilde{O}(nr^{(3h-2)/2})$, as claimed. □

Lemma 8. *(Step 3) can be implemented in $\tilde{O}(n^2/\sqrt{r})$ time given the information computed in (Step 1) and (Step 2).*

Proof. First we bound the running time to compute $\mathsf{ecc}(u)$ for a given vertex $u \in V$. For the cluster $R \in \mathcal{R}$ such that $u \in R$, computing $\Delta(u,R)$ takes $O(|V(R)|) = O(r)$ time. If $u \notin R$, we can look up (the ID of) the pattern $\mathbf{p}_u$ in $T_R^{(1)}$ in $O(1)$ time. Given $\mathbf{p}_u$, we can lookup $d(\mathbf{p}_u,v)$ in $O(1)$ time from $T_R^{(2)}$ constructed in (Step 2). Furthermore, $d_G(u,s_0)$ can be found directly from $D(B,V)$ in $O(1)$ time. Thus, the running time to compute $\Delta(u,R)$ is $O(1)$ and hence the total running time to compute $\Delta(u,R)$ is $O(r+|\mathcal{R}|)$. By Lemma 1,

$$O(r+|\mathcal{R}|) = \tilde{O}(r+n/\sqrt{r}) = \tilde{O}(n/\sqrt{r})$$

as $r = n^{2/(3h-1)} \le n^{0.2}$ with $h \ge 4$. This means that the total running time to compute all the eccentricities is $\tilde{O}(n^2/\sqrt{r})$. □

By Lemmas 6 to 8, the total running time to compute all the eccentricities (and hence the diameter) of G is:

$$\tilde{O}(\frac{n^2}{\sqrt{r}} + nr^{(3h-2)/2}) = \tilde{O}(n^{2-\frac{1}{3h-1}}) \tag{15}$$

when $r = n^{2/(3h-1)}$.

3.2.2 Wiener Index

We show how to compute Wiener index in truly subquadratic time as described in Corollary 2. For any two set of vertices $X,Y \subseteq V$, let $W(X,Y) = \sum_{x\in X}\sum_{y\in Y} d_G(x,y)$. The Wiener index of G is $\frac{1}{2}W(V,V)$, and thus our goal is to compute $W(V,V)$. Let $\mathcal{R}$ be an r-division of G computed by Lemma 1 for $r = n^{2/(3h-1)}$. Let $R^\circ = V(R)\setminus\partial R$. Recall that in Section 3.2.1, we define $B = \cup_{R\in\mathcal{R}}\partial R$. Observe that:

$$\begin{aligned} W(V,V) &= W(B,V) + \sum_{R\in\mathcal{R}} W(R^\circ,V) \\ &= W(B,V) + \sum_{R\in\mathcal{R}} W(R^\circ,V(R)) + \sum_{R\in\mathcal{R}} W(R^\circ,V\setminus V(R)) \end{aligned} \tag{16}$$

First, we focus on computing $W(B,V)$ and $\sum_{R\in\mathcal{R}} W(R^\circ,V(R))$.

Lemma 9. *$W(B,V)$ can be computed in time $\tilde{O}(n^2/\sqrt{r})$ and $\sum_{R\in\mathcal{R}} W(R^\circ, V(R))$ can be computed in time $O(nr)$.*

Proof. By Observation 2, all the distances from vertices in B to vertices in V (the set $D(B,V)$) can be computed in time $\tilde{O}(n^2/\sqrt{r})$, which also is the running time to compute $W(B,V)$.

By Lemma 1, each cluster has size at most r. Furthermore, computing the distance from a vertex in R to all other vertices R can be done in $O(r)$ time using BFS. Thus, the running time to compute $\sum_{R\in\mathcal{R}} W(R^\circ, R)$ is $\sum_{R\in\mathcal{R}} O(r|R^\circ|) = O(nr)$. □

Next, we bound the running time to compute $\sum_{R\in\mathcal{R}} W(R^\circ, V \setminus V(R))$.

Lemma 10. *$\sum_{R\in\mathcal{R}} W(R^\circ, V \setminus V(R))$ can be computed in time $\tilde{O}(nr^{(3h-2)/2} + n^2/\sqrt{r})$,*

Proof. First we compute the set of patterns $\{P_R\}_{R\in\mathcal{R}}$ of all clusters in $\mathcal{R}$ in time $\tilde{O}(n^2/\sqrt{r})$ by Lemma 6. Next, we observe that:

$$\sum_{R\in\mathcal{R}} W(R^\circ, V \setminus V(R)) = \sum_{R\in\mathcal{R}} \sum_{u\in V\setminus V(R)} W(u, R^\circ) \tag{17}$$

Furthermore, by Lemma 5,

$$W(u, R^\circ) = \sum_{v\in R^\circ} d_G(u,v) = |R^\circ| d_G(u, s_0) + \sum_{v\in R^\circ} d(\mathbf{p}_u, v)$$

where s_0 is the first vertex in the boundary sequence σ_R of cluster R. The distance $d_G(u, s_0)$ is already computed, i.e, $d_G(u, s_0) \in D(B,V)$.

In Lemma 7, we find the furthest vertex from each pattern $\mathbf{p} \in P_R$ by iterating over all vertices of R in total time $\tilde{O}(\sqrt{r}|V(R)|)$ time. Thus, we can compute the sum $\sum_{v\in R^\circ} d(\mathbf{p}, v)$ in time $\tilde{O}(\sqrt{r}|V(R)|)$, and running time for to compute all the sums of *all patterns* in P_R is $\tilde{O}(\sqrt{r}|V(R)|r^{3(h-1)/2}) = \tilde{O}(r^{(3h-2)/2})|V(R)|$. We can think of this as preprocessing time for computing $W(u, R^\circ)$. Over all clusters in $\mathcal{R}$, the total preprocessing time is:

$$\sum_{R\in\mathcal{R}} \tilde{O}(r^{(3h-2)/2})|V(R)| = \tilde{O}(nr^{(3h-2)/2}) \tag{18}$$

which is the first term in the running time.

Once the sums of distances for all patterns in P_R are given, we can store them in a table keyed by the ID of the patterns, and then we can look up $\sum_{v\in R^\circ} d(\mathbf{p}_u, v)$ in $O(1)$ time. As a result, we can compute $W(u, R^\circ)$ in $O(1)$ time, and hence by Equation (17), $W(R^\circ, V \setminus V(R))$ can be computed in time:

$$\sum_{R\in\mathcal{R}} O(n) = O(n|\mathcal{R}|) = \tilde{O}(n^2/\sqrt{r})$$

by Lemma 1, which is the second term in the running time. □

By Lemma 9 and Lemma 10, the total running time to compute $W(V,V)$ is $\tilde{O}(\frac{n^2}{\sqrt{r}} + nr^{(3h-2)/2}) = \tilde{O}(n^{2-\frac{1}{3h-1}})$ when $r = n^{2/(3h-1)}$ as claimed in Corollary 2.

3.2.3 Exact Distance Oracle

We construct the first exact distance oracle for unweighted minor-free graphs with subquadratic space-query time trade-off, and subquadratic preprocessing time as described in Corollary 1.

Construction. The construction has two steps:

- **(Step 1).** Construct an r-division $\mathcal{R}$ of G with $r = n^{2/(3h-1)}$, and for each cluster $R \in \mathcal{R}$, store a set of patterns P_R w.r.t an (arbitrary) sequence of boundary vertices σ_R in a table T_R. We also store the exact distances of all pairs of vertices in R.
- **(Step 2).** For each vertex u and a region $R \in \mathcal{R}$: (2a) if $u \in R$, we store $d(\mathbf{p}, u)$ for every pattern $\mathbf{p} \in P_R$; (2b) if $u \notin R$, we store a pointer from u to its pattern $\mathbf{p}_u$ in table T_R and the distance from u to the first vertex in the sequence of boundary vertices σ_R.

Querying distances. Given two vertices u and v, if there is a region R containing both u and v, we can simply look up their distance stored at R in $O(1)$ time. Otherwise, let R_v be the region containing v. First, we look up the distance from u to the first vertex in the boundary sequence σ_{R_v}, say s_0, in $O(1)$ time. Then, we look up the pattern $\mathbf{p}_u \in P_{R_v}$ of u in R_v, and the distance $d(v, \mathbf{p}_u)$ in total $O(1)$ time due to the construction in (Step 2). Finally, we return:

$$d_G(u, s_0) + d(\mathbf{p}_u, v) \ . \tag{19}$$

Lemma 5 implies that the returned distance is $d_G(u, v)$. The total query time is $O(1)$.

Space analysis. By Corollary 5, the number of patterns is $\tilde{O}(r^{3(h-1)/2})$ and by Lemma 1, $|\mathcal{R}| = \tilde{O}(n/\sqrt{r})$. The total space of (Step 1) and (Step 2(a)) is:

$$\tilde{O}(\sum_{R \in \mathcal{R}} (r^{3(h-1)/2}|V(R)| + |V(R)|^2)) = \tilde{O}(\sum_{R \in \mathcal{R}} (r^{3(h-1)/2}|V(R)| + r|V(R)|)) = \tilde{O}(nr^{3h/2-2}) \tag{20}$$

as $h \geq 4$. The total space of Step 2(b) is $O(n|\mathcal{R}|) = \tilde{O}(n^2/\sqrt{r})$ by Lemma 1. Thus, the total space of the oracle is:

$$\tilde{O}(nr^{3h/2-2} + n^2/\sqrt{r}) = \tilde{O}(n^{2-\frac{1}{3h-1}}) \tag{21}$$

with $r = n^{2/(3h-1)}$.

Construction Time. We observe that the amount of information we need to construct the distance oracle is exactly the amount of information we need to compute the diameter and the Wiener index. Thus, the running time to compute all the information is $\tilde{O}(n^{2-\frac{1}{3h-1}})$.

Remark 2. *We can further reduce the space of the oracle by increasing the construction time by choosing r differently, or by increasing the query time using the nested r-division following the line of reasoning in [FHMWN20].*

3.2.4 Other Applications

Here we discuss other algorithmic applications of our Theorem 3.

Metric compression. Li and Parter [LP19] showed that for any two sets of vertices S, T in an unweighted *planar* graph of diameter D such that S is on the boundary of the outer face of the graph, then one can compress all the distances from T to S using only $\tilde{O}(|S|^3 D + |T||\log(|S|D)|)$ bits. Here it is instructive to think of a canonical regime where D is a constant and $1 \ll \text{poly}(|S|) \ll |T|$. In this canonical regime, the compression scheme has space $\tilde{O}(\text{poly}(|S|)+|T|)$ bits instead of $\tilde{O}(|S|\cdot|T|)$ bits by simply storing all the distances from T to S. This compression scheme has an application in computing diameter of planar graphs in the distributed CONGEST model.

Using Theorem 3, we improve the compression scheme by Li and Parter in two aspects: S is no longer restricted, and G could be any minor-free graphs. In the canonical regime, the space of our compression scheme is $\tilde{O}(\text{poly}(|S|) + |T|)$ which is the same as Li-Parter space bound up to a factor of $\text{poly}(|S|)$ in the additive term.

Now we give a more formal description of our result. Let $S = \{s_0, s_1, \ldots, s_{k-1}\}$ and $T = \{t_1, t_2, \ldots, t_\ell\}$ where $k = |S|$ and $\ell = |T|$. For each vertex $v \in V$, we define

$$\mathsf{Tuple}(v) = \langle d_G(v, s_0), \ldots, d_G(v, s_{k-1}) \rangle \ ,$$

which is called a *distance tuple* of v w.r.t S. Li and Pater showed in their Theorem 2.2 [LP19] that the set $\mathsf{Tuple}(V) = \{\mathsf{Tuple}(v) : v \in V\}$ has size $O(|S^3|D)$ when S is on the outer face of a planar graph of diameter D. Hence, to compress the distances from T to S, one only needs to store $\mathsf{Tuple}(V)$ using $\tilde{O}(|S|^3 D)$ bits and then for each $t \in T$, one stores a pointer from t to its corresponding distance tuple $\mathsf{Tuple}(t) \in \mathsf{Tuple}(V)$. Here we show that in our more general setting where S has no restriction and G is K_h-minor-free, the number of tuples is bounded by $O((|S| \cdot D)^{O(h)}) = O(\text{poly}(|S|))$ for fixed h and D. This implies our result on the metric compression.

Lemma 11. *$|\mathsf{Tuple}(V)| = O(|S|^{h-1} \cdot D^h)$ when G is a K_h-minor-free graph and has diameter at most D.*

Proof. The proof is the same as the proof of Lemma 4. The only difference is that now $-D \le d_G(v, s_i) - d_G(v, s_0) \le D$ by the triangle inequality. Let's fix the distance from $d_G(v, s_0)$, and $M = \{-D, \ldots, -1, 0, 1, \ldots D\}$. Let $\bar{p}_v$ be a set obtain by adding pairs (i, Δ) for $\Delta \in M$ such that $d_G(v, s_i) - d_G(v, s_{i-1}) \le \Delta$ to the set $\bar{p}_v$ for each $i \in [1, k-1]$. Then there is a bijection between the set $\{\bar{p}_v\}_{v \in V}$ and $\widehat{\mathcal{LP}}_{G,M}$. By the Sauer–Shelah Lemma (Lemma 2), we have $|\widehat{\mathcal{LP}}_{G,M}| = O((|S||M|)^{h-1}) = O((|S| \cdot D)^{h-1})$. As we have D choices for $d_G(v, s_0)$, the number of different distance tuples is at most $O((|S| \cdot D)^{h-1} \cdot D) = O(|S|^{h-1} D^h)$. □

Computing diameter and all-vertices eccentricities in low-treeewith minor-free graphs. Abboud, Williams, and Wang [AWW16] studied the problem of computing diameter in unweighted graphs of treewidth k. They showed surprisingly that, there exists a constant $c > 0$ such that for any $k \le c \cdot \log n$, under the Strong Exponential Time Hypothesis (SETH), there is no algorithm with running time $n^{2-\epsilon} 2^{\Omega(k)}$ to compute the diameter for any fixed $\epsilon > 0$. That is, if one insists on having an algorithm with truly subquadratic time, one has to pay an *exponential dependency* on the treewidth. They also presented an algorithm for distinguishing diameter 2 vs diameter 3 graphs with running time $n^{1+o(1)} 2^{O(k \log k)}$. Husfeldt [Hus17] designed an improved algorithm with running time $O(d^{O(k)} n)$ where d is the diameter using dynamic programming. An open question is to design in algorithm with running time $O(d^{O(1)} 2^{O(k)} n)$.

We show that if the input graph G has treewidth k, and in addition, is K_h-minor-free, for a fixed constant h, then one can find the diameter of G in time $O((dk)^{O(1)} n)$. Notably, the dependency

on the treewidth k is polynomial instead of exponential. We note that the class of K_h-minor-free graphs of treewidth k includes well-studied classes of graphs, such as k-outerplanar graphs, Halin graphs, and series-Parallel graphs.

Here we sketch our argument. The basic idea is to use Theorem 3 to optimize the running time of the dynamic programming algorithm by Husfeldt [Hus17] (for computing all-vertices eccentricities and hence diameter). For each bag $B = \{s_0, s_1, \ldots, s_{k-1}\}$ of size k in the tree decomposition, the dynamic program keeps track of all the distance tuples of vertices in the graph induced by vertices in descendant bags of B (and including B). The maximum number of distance tuples is $d^{O(k)}$, which results in running time $d^{O(k)}n$. When G is K_h-minor-free, then by Lemma 11, the number of distance tuples is $\text{poly}(k \cdot d)$, and hence the running time of the dynamic program becomes $\text{poly}(k \cdot d)n$.

Approximate distance oracles in planar graphs. In [Le23], Le constructed a $(1+\epsilon)$-approximate distance oracle for planar graphs with $\tilde{O}(n/\epsilon^{o(1)})$ space and $\tilde{O}(1)$ query time. That is, the space-query product trade-off depends sublinearly on $1/\epsilon$. A key ingredient of the construction is a polynomial bound on the number of (approximate) distance tuples by Li and Parter [LP19]. Our set system $\widehat{\mathcal{LP}}_{G,M}$ also gives a polynomial bound on the number of such distance tuples, and hence could be used in the same way to derive the result in [Le23].

4 VC Dimension of Digraphs and Applications

In this section, $G = (V, E)$ denotes a K_h-minor-free *digraphs*. G could be weighted or unweighted. In bounding the VC-dimension, we allow edges of G to have arbitrary non-negative weights, while in the algorithmic applications, G is unweighted.

4.1 VC dimension of $\vec{\mathcal{LP}}_{G,M}$

In this section, we prove Theorem 4, which we restate below.

Theorem 4. *Let S be any set of vertices on an edge weighted K_h-minor-free digraph G. Let $M \subseteq \mathbb{R}$ be any set of real numbers. Then $\overrightarrow{\mathcal{LP}}_{G,M}(S)$ has VC-dimension at most h^2.*

Suppose that $\overrightarrow{\mathcal{LP}}_{G,M}$ shatters a set $X = \{(s_1, \Delta_1), (s_1, \Delta_2) \ldots, (s_q, \Delta_q)\}$ of size q. Our goal is to show that (the undirected counterpart of) G has a clique minor of size at least $\lfloor \sqrt{q} \rfloor$, which gives the bound on the VC dimension of $\overrightarrow{\mathcal{LP}}_{G,M}$, as $\lfloor \sqrt{q} \rfloor \leq h - 1$. The major difficulty in the proof is that, in digraphs, we do not have strong properties of $\mathsf{Bunch}(s_i)$—we construct $\mathsf{Bunch}(s_i)$ in the same way—as we do in the proof of Theorem 3 in Section 3.1. More precisely, Lemma 3 no longer holds. This makes the construction of the clique minor more difficult, and as a result, we could not show the linear bound on the VC dimension. On the other hand, we show that an analog of Claim 1 suffices for our construction of a clique minor of size $\sqrt{q}$.

We now present the proof. By the same reasoning in Observation 1, we have that $s_i \neq s_j$ for all $i \neq j$. For every pair $(s_i, \Delta_j), (s_j, \Delta_j)$, let t_{ij} be such that $\overrightarrow{X}_{t_{ij}} \cap (V \times M) = \{(s_i, \Delta_j), (s_j, \Delta_j)\}$.

Lemma 12. *If $i, j, \ell, p \in [q]$ are pairwise different, then $\pi(t_{ij} \to s_i, G) \cap \pi(t_{\ell p} \to s_\ell, G) = \emptyset$.*

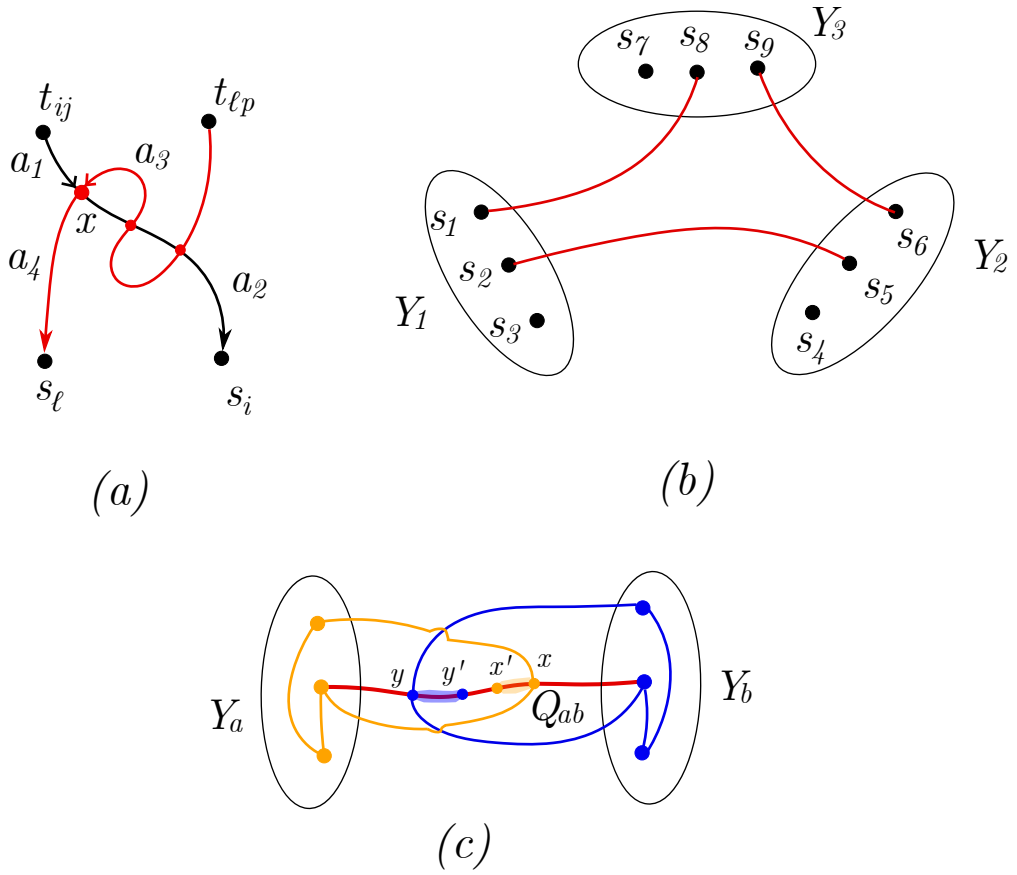

Figure 2: (a) Assume that $x \in \pi(t_{ij} \to s_i, G) \cap \pi(t_{\ell p} \to s_\ell, G)$; (b) three sets Y_1, Y_2, Y_3 when $k = 9$ and three paths Q_{12}, Q_{23}, Q_{13} where no two paths share the same endpoint; (c) $Q_{ab}[x, x']$ is added to H_a (orange) and $Q_{ab}[y, y']$ is added to H_b (blue).

Proof. Suppose otherwise, there exists $x \in \pi(t_{ij} \to s_i, G) \cap \pi(t_{\ell p} \to s_\ell, G)$; see Figure 2(a). Let:

$$a_1 = d_G(t_{ij} \to x) \qquad a_2 = d_G(x \to s_i)$$
$$a_3 = d_G(t_{\ell p} \to x) \qquad a_4 = d_G(x \to s_\ell)$$

Then $a_1 + a_2 \le d_G(t_{ij} \to s_0) + \Delta_i$ and $a_3 + a_4 \le d_G(t_{\ell p} \to s_0) + \Delta_\ell$. Thus, we have:

$$a_1 + a_2 + a_3 + a_4 \le d_G(t_{ij} \to s_0) + d_G(t_{\ell p} \to s_0) + \Delta_i + \Delta_\ell$$

Furthermore, since $(s_\ell, \Delta_\ell) \notin X_{t_{ij}}$, we have $d_G(t_{ij} \to s_\ell) > d_G(t_{ij} \to s_0) + \Delta_\ell$. This implies that $a_1 + a_4 > d_G(t_{ij} \to s_0) + \Delta_\ell$. By the same argument, $a_2 + a_3 > d_G(t_{\ell p} \to s_0) + \Delta_i$. Thus, $a_1 + a_2 + a_3 + a_4 > d_G(t_{ij} \to s_0) + d_G(t_{\ell p} \to s_0) + \Delta_i + \Delta_\ell$, a contradiction. □

We now ignore the direction of G and focus on constructing a clique minor of size $\lfloor \sqrt{q} \rfloor$. Let $\pi(t_{ij} \not\to s_i, G)$ be the undirected path obtained by ignoring the direction of edges in $\pi(t_{ij} \to s_i, G)$. For every $i \ne j$ we denote by P_{ij} the path from s_i to s_j obtained by simplifying the (undirected) walk from s_i to s_j obtained by gluing two paths $\pi(t_{ij} \not\to s_i, G)$ and $\pi(t_{ij} \not\to s_j, G)$ at t_{ij}. Lemma 12 implies:

Corollary 6. *$P_{ij} \cap P_{\ell p} = \varnothing$ when i, j, ℓ, p are pairwise different.*

That is the two paths between two pairs of vertices in X can intersect if and only if they share one endpoint. In this case, they could intersect in an arbitrarily complicated way.

We partition X into $\sqrt{q}$ subsets $Y_1, \ldots, Y_{\sqrt{q}}$ each contains $\sqrt{q}$ vertices in X; for ease of notation, we assume that $\sqrt{q}$ is an integer. For every pair (Y_a, Y_b) for $a, b \in [\sqrt{q}], a \neq b$, let $\mathcal{P}_{ab} = \{P_{ij} : s_i \in Y_a, s_j \in Y_b\}$ be the set of paths between Y_a and Y_b. We then choose a path $Q_{ab} \in \mathcal{P}_{ab}$ such that the set of chosen paths, denoted by $\mathcal{Q} = \{Q_{ab}\}_{(a,b)\in[\sqrt{q}]\times[\sqrt{q}],a<b}$, has no two paths sharing the same endpoint; we can pick $\mathcal{Q}$ in a greedy manner. $\mathcal{Q}$ exists since each Y_a has $\sqrt{q}$ vertices while we only need $\sqrt{q}-1$ paths in $\mathcal{Q}$ to connect Y_a to other sets. See Figure 2(b).

We now construct a $K_{\sqrt{q}}$-minor as follows. For each Y_a, $a \in [\sqrt{q}]$, let $H_a = \cup_{s_i, s_j \in Y_a} P_{ij}$. Clearly, H_a is connected and furthermore, by Corollary 6, $V(H_a) \cap V(H_b) = \varnothing$. Between H_a and H_b, we have a path $Q_{ab} \in \mathcal{Q}$ that is vertex disjoint from all other paths in $\mathcal{Q}$. (H_a and H_b could contain vertices of Q_{ab} other than its endpoints.) Since $V(H_a) \cap V(H_b) = \varnothing$, there must be a subpath $Q_{ab}[x, y]$ from a vertex x to a vertex y such that $x \in H_a$ and $y \in H_b$ and no other vertex in $Q_{ab}[x, y] \setminus \{x, y\}$ belongs to $H_a \cup H_b$. (It could be that $Q_{ab}[xy]$ is an edge.) Pick an arbitrary edge $e_{ab} = (x', y') \in Q[x, y]$; we assume w.l.o.g that $x' \in Q_{ab}[x, y']$. Then we add $Q_{ab}[x, x']$ to H_a and $Q_{ab}[y', y]$ to H_b. See Figure 2(c). Let H'_a be the graph H_a after applying this process to all pairs $(a, b) \in [\sqrt{q}] \times [\sqrt{q}], a < b$. Then $\{H'_a\}_{a\in[\sqrt{q}]}$ are pairwise vertex-disjoint, and there is an edge connecting every pair of graphs. These graphs induce a $K_{\sqrt{q}}$ of G, as desired.

4.2 VC dimension of $\vec{\mathcal{B}}(G)$

We show Theorem 5, which states that the set system of balls $\overrightarrow{B}(G)$ defined in Equation (4) is a VC set system. We tailor the proof by Bousquet and Thomassé [BT15] for the undirected case to the directed case.

Theorem 5. *If G is a K_h-minor-free digraph, then $\overrightarrow{B}(G)$ has VC-dimension at most $h-1$.*

The proof follows the presentation of the proof of Theorem 3 though several details are different. Specifically, we assume for contradiction that $\overrightarrow{B}(G)$ shatters a set $X = \{v_1, v_2, \ldots, v_h\} \subseteq V$ of size h. Then for every $i \neq j$, there is a ball $\overrightarrow{B}(t_{ij}, r_{ij})$ such that $\overrightarrow{B}(t_{ij}, r_{ij}) \cap X = \{v_i, v_j\}$. We choose t_{ij} and r_{ij} such that

$$r_{ij} \text{ is minimum.} \tag{22}$$

We then can assume that $r_{ij} = \max\{d_G(t_{ij} \to v_i), d_G(t_{ij} \to v_j)\}$ as otherwise, $r_{ij} > \max\{d_G(t_{ij} \to v_i), d_G(t_{ij} \to v_j)\}$ and we can always set r_{ij} to be $\max\{d_G(t_{ij} \to v_i), d_G(t_{ij} \to v_j)\}$. Our goal is to construct a K_h-minor of G as we did in the proof of Theorem 3. We observe that Claim 1 remains true in this setting of digraphs.

Observation 3. *$\pi(t_{ij} \to v_i, G)$ and $\pi(t_{ij} \to v_j, G)$ are internally disjoint.*

Proof. Suppose otherwise; there would be a vertex $x_{ij} \in (\pi(t_{ij} \to v_i, G) \cap \pi(t_{ij} \to v_j, G)) \setminus \{t_{ij}\}$. Observe that $\overrightarrow{B}(x_{ij}, r_{ij} - d_G(t_{ij} \to x_{ij})) \cap X = \{v_i, v_j\}$, contradicting the choice of r_{ij} in Equation (22); see Figure 3(a). □

For each v_i, we define $\mathsf{Bunch}(v_i)$ as in Equation (6), ignoring the directions of the paths.

$$\mathsf{Bunch}(v_i) = \cup_{j\neq i} \pi(t_{ij} \not\to v_i, G) \tag{23}$$

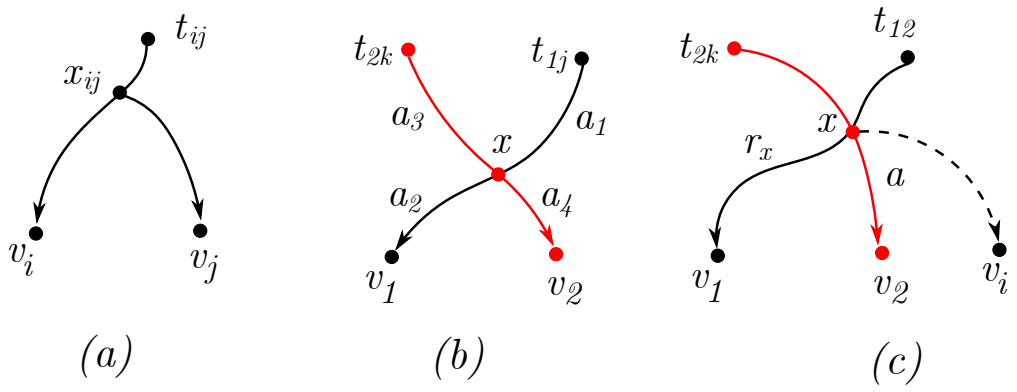

Figure 3: Illustration for the proof of Theorem 5.

Here $\pi(t_{ij} \not\to v_i, G)$ is an undirected path obtained by ignoring the directions of edges in $\pi(t_{ij} \to v_i, G)$. The proof of Theorem 3 in Section 3.1 implies that the existence of a K_h-minor is reduced to showing the following lemma.

Lemma 13. *For every $a \neq b$, $\mathsf{Bunch}(v_a) \cap \mathsf{Bunch}(v_b) = \{t_{ab}\}$. Furthermore, t_{ab} is either an endpoint of $\mathsf{Bunch}(v_a)$, or an endpoint of $\mathsf{Bunch}(v_b)$, or both.*

Proof. We follow the same proof strategy of Lemma 3: considering $a = 1$ and $b = 2$. Let $\pi(t_{1j} \to v_1, G)$ and $\pi(t_{2k} \to v_2, G)$ be paths whose undirected counterparts are in $\mathsf{Bunch}(v_1)$ and $\mathsf{Bunch}(v_2)$, respectively. The following claim is analogous to Claim 2.

Claim 4. *If $\{j,k\} \cap \{1,2\} = \varnothing$, then $\pi(t_{1j} \to v_1, G) \cap \pi(t_{2k} \to v_2, G) = \varnothing$.*

Proof. Suppose otherwise, there exists $x \in \pi(t_{1j} \to v_1, G) \cap \pi(t_{2k} \to v_2, G)$; see Figure 3(b). Let:

$$a_1 = d_G(t_{1j} \to x) \qquad a_2 = d_G(x \to v_1)$$
$$a_3 = d_G(t_{2k} \to x) \qquad a_4 = d_G(x \to v_2)$$

Since $v_1 \in \overrightarrow{B}(t_{1j}, r_{1j})$ and $v_2 \in \overrightarrow{B}(t_{2k}, r_{2k})$, $a_1 + a_2 \leq r_{1j}$ and $a_3 + a_4 \leq r_{2k}$. This implies that

$$a_1 + a_2 + a_3 + a_4 \leq r_{1j} + r_{2k} \tag{24}$$

On the other hand, $v_2 \notin \overrightarrow{B}(t_{1j}, r_{1j})$ and $v_1 \notin \overrightarrow{B}(t_{2k}, r_{2k})$, which gives $a_1 + a_4 > r_{1j}$ and $a_2 + a_3 > r_{2k}$. This implies that $a_1 + a_2 + a_3 + a_4 > r_{1j} + r_{2k}$, contradicting Equation (24). Thus, x does not exist. □

The proof of the lemma follows directly from the following claim.

Claim 5. *If $\{j,k\} \cap \{1,2\} \neq \varnothing$, then $\pi(t_{1j} \to v_1, G) \cap \pi(t_{2k} \to v_2, G) \subseteq \{t_{12}\}$, and that t_{12} is either an endpoint of $\mathsf{Bunch}(v_1)$ or $\mathsf{Bunch}(v_2)$ or both.*

Proof. W.l.o.g., we assume that $j = 2$ and $k \neq 1$. Suppose that there exists $x \in \pi(t_{12} \to v_1, G) \cap \pi(t_{2k} \to v_2, G)$ such that $x \neq t_{12}$. Let $a = d_G(x \to v_2)$. Then $d_G(x \to v_1) > a$ as otherwise, $v_1 \in \overrightarrow{B}(t_{2k}, r_{2k})$, a contradiction. Let $r_x = d_G(x \to v_1)$. Then $\{v_2, v_1\} \subseteq \overrightarrow{B}(x, r_x) \cap X$. We claim that $\overrightarrow{B}(x, r_x) \cap X$ contains no other vertex other than v_1, v_2; see Figure 3(c).

Suppose otherwise, there exists $v_i \in \overrightarrow{B}(x, r_x) \cap X$ for $v_i \neq v_1, v_2$. Then $v_i \in \overrightarrow{B}(x, r_x)$ and hence $d_G(x \to v_i) \leq r_x = d_G(x \to v_1)$. This implies that $d_G(t_{12}, v_i) \leq d_G(t_{12} \to v_1) \leq r_{12}$; that is, v_i also belongs to the ball $\overrightarrow{B}(t_{12}, r_{12})$ contradicting the fact that $\overrightarrow{B}(t_{12}, r_{12})$ only shatters $\{v_1, v_2\}$.

Since $\overrightarrow{B}(x, r_x) \cap X$ contains no other vertex other than v_1, v_2 and $r_x < r_{12}$, we obtain a contradiction to the choice of t_{12} in Equation (22), as $\max\{d_G(x \to v_1), d_G(x \to v_2)\} < \max\{d_G(t_{12} \to v_1), d_G(t_{12} \to v_2)\}$. □

The lemma then follows directly from Claim 4 and Claim 5. □

4.3 Algorithmic Applications

In this section, we explore algorithmic applications of two VC set systems $\overrightarrow{\mathcal{LCP}}_{G,M}$ and $\overrightarrow{B}(G)$. *Digraphs in this section are* unweighted *and hence the distances are unweighted directed distances.* A central concept in the algorithmic applications of $\widehat{\mathcal{LCP}}_{G,M}$ in undirected graphs in Section 3.2 is the notion of patterns and polynomial bounds on the number of patterns in a connected subgraph in Lemma 4. The same bound on the number of patterns *completely breaks down* in digraphs, as the triangle inequality no longer holds. Only an asymmetric version fo the triangle inequality holds in digraphs, but this is not enough for deriving Lemma 4 in digraphs. Indeed, we believe that Lemma 4 does not hold in digraphs. The implication of not having a polynomial bound on the number of patterns is clear: we could not easily derive analogous algorithmic results presented in Section 3.2 for digraphs. Instead, obtain similar results using $\overrightarrow{\mathcal{LCP}}_{G,M}$ and $\overrightarrow{B}(G)$.

First, we devise a new way to exploit the set system of balls $\overrightarrow{B}(G)$ to design a distance oracle for digraphs with truly subquadratic space and logarithmic query time. The VC set system of balls is very hard to manipulate, as evidenced in the work of Ducoffe, Habib, and Viennot [DHV20] since it does not encode distances directly into the system. Thus, we believe that our technique is of independent interest; the details are in Section 4.3.1.

Second, we modify the notion of patterns to include $\pm\infty$, called *infinite patterns*, as a marker for the failure of triangle inequality. We then are able to bound the number of infinite patterns, obtaining a lemma analogous to Lemma 4. We note that we still do not know how to exploit infinite patterns in constructing distance oracles in digraphs, as they do not enjoy the same properties as their (finite) counterpart. However, we are able to exploit infinite patterns to design truly subquadratic time algorithms for computing all-vertices eccentricities and the diameter of digraphs. The technical details are in Section 4.3.1.

4.3.1 Distance oracle in digraphs.

In this section, we construct an exact distance oracle for unweighted minor-free *digraphs* with $\tilde{O}(n^{2-\frac{1}{2(h-2)}})$ space and $O(\log(n))$ query time as described in Corollary 4. We will use a well-known property of VC set system restricted to a subset, as described in the following lemma.

Lemma 14. *Let $\mathcal{F}$ be a set system of a ground set U of VC-dimension $d \geq 1$. Let X be any subset of U. Then $\mathcal{F}_X = \{Y \cap X : Y \in \mathcal{F}\}$ has VC dimension at most d. We call $\mathcal{F}_X$ the X-restriction of $\mathcal{F}$.*

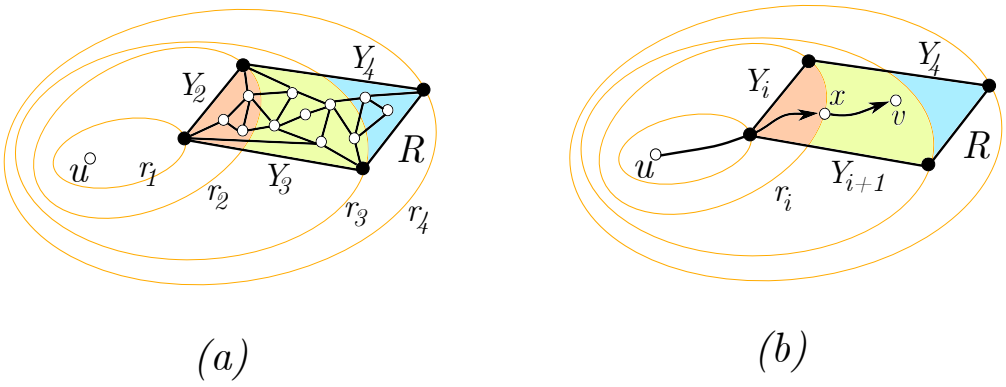

Figure 4: (a) A region R with 4 boundary vertices; the set $Y_1 = \vec{B}(u, r_1) \cap R$ only contains the boundary vertex closest to u. We ignore the directions of edges in R in this figure for a better visualization. (b) Querying distance from u to v.

Construction. The construction has three steps.

- **(Step 1).** Construct an r-division $\mathcal{R}$ of G with $r = n^{2/(2h-1)}$, and for each cluster $R \in \mathcal{R}$, we store the exact distances of all pairs of vertices in R. Let $\overrightarrow{\mathcal{B}}_R$ be the $V(R)$-restriction of $\overrightarrow{\mathcal{B}}(G)$. We store (the IDs of) the sets of $\overrightarrow{\mathcal{B}}_R$ in a table.
- **(Step 2).** For each cluster $R \in \mathcal{R}$ and each vertex $v \in R$, we store: (2a) the distance $d_G(s \to v)$ from each vertex $s \in \partial R$ to v; (2b) for each set $Y \in \overrightarrow{\mathcal{B}}_R$, store $d_G(Y \to v) \stackrel{\text{def.}}{=} \min_{y \in Y} d_G(y \to v)$.
- **(Step 3).** For each cluster $R \in \mathcal{R}$ and each vertex $u \notin R$, let $k_R = |\partial R|$. Let $\overrightarrow{B}(u, r_1), \ldots, \overrightarrow{B}(u, r_{k_R})$ be a sequence of nest balls centered at u where $r_1 \le r_2 \le \ldots \le r_{k_R}$ such that $\overrightarrow{B}(u, r_1)$ is the smallest ball containing at least one vertex of ∂R, and $\overrightarrow{B}(u, r_i)$ is the smallest ball containing at least one vertex of $\partial R \setminus \overrightarrow{B}(u, r_{i-1})$; see Figure 4(a). (The number of balls could be smaller than k_R; for simplicity, we assume that there are exactly k_R balls.) Then we store at u the radius r_i and (the IDs of) the restriction $Y_i = \overrightarrow{B}(u, r_i) \cap V(R)$ for all $i \in [k_R]$ in a list $L(u, R)$. Note that $Y_i \in \overrightarrow{\mathcal{B}}_R$ by the construction in (Step 1). We also store the distance $d_G(u \to s)$ from u to every boundary vertex $s \in \partial R$.

Querying distances. Given two vertices u and v, if there is a cluster R containing both u and v, we can simply look up their distance stored at R in $O(1)$ time. Otherwise, let R be the cluster containing v. Let $Y_i = \overrightarrow{B}(u, r_i) \cap V(R)$. We then do a binary search on the list $L(u, R)$ to find the first radius r_i such that $v \notin Y_i$ and $v \in Y_{i+1}$; see Figure 4(b). Note that we can check whether v is in Y_i or not in $O(1)$ time by the construction in (Step 2), in particular (2b) since $v \in Y_i$ if and only if $d_G(Y_i \to v) = 0$. We then return:

$$r_i + d_G(Y_i \to v) \tag{25}$$

as the distance from u to v. We note that $d_G(Y_i \to v)$ is stored in (2b) of (Step 2), so we can look up this distance in $O(1)$ time.

The query time is dominated by the time to do binary search on $L(u, R)$, which is $O(\log |L(u, R)|) = O(\log |\partial R|) = O(\log r) = O(\log n)$.

Correctness. By the definition of Y_i and Y_{i+1}, the shortest path from u to v must go through a vertex in $\partial R \cap Y_i$. Let x be the last vertex on $\pi(u \to v, G)$ that is contained in Y_i; see Figure 4(b). Then $d_G(u \to v) = d_G(u \to x) + d_G(x \to v)$. Since G is unweighted, it must be that $d_G(u \to x) = r_i$. Furthermore, $d_G(x \to v) = d_G(Y_i \to v)$ since otherwise, $d_G(x \to v) > d_G(Y_i, v)$ which means there is a path from u to v of length less than $d_G(u \to v)$, a contradiction. Thus, $d_G(u \to v) = r_i + d_G(Y_i \to v)$ as desired.

Space analysis. By Theorem 5 and Lemma 14, $\overrightarrow{\mathcal{B}}_R$ has VC-dimension at most $h-1$. By Lemma 2, $|\overrightarrow{\mathcal{B}}_R| = O(r^{h-1})$ and hence the total space of Step 1 is $\tilde{O}((n/\sqrt{r})(r^{h-1}+r^2) = \tilde{O}(nr^{h-3/2}))$. The total space of Step 2 is $\tilde{O}((n/\sqrt{r})(r \cdot r^{h-1})) = \tilde{O}(nr^{h-3/2})$. For each vertex u and cluster R in Step 3, the total space is $O(|\partial R|)$. Thus, the total space of Step 3 is n times the total number of boundary vertices, which is $\tilde{O}(n^2/\sqrt{r})$ by Lemma 1. In summary, the total space of the oracle is:

$$\tilde{O}(nr^{h-3/2} + n^2/\sqrt{r}) = \tilde{O}(n^{2-\frac{1}{2(h-2)}}) \tag{26}$$

when $r = n^{1/(h-2)}$.

4.3.2 Computing all-vertices eccentricities and diameter.

Infinite patterns. Let H be an induced sub-digraph of G; H might or might not be (even weakly) connected. Recall that ∂H is the set of all boundary vertices of H. Let $r = |V(H)|$ and $b = |\partial H|$. Fix an arbitrary *sequence* σ_H of vertices of ∂H, which is a linear order of ∂H. We write $\sigma_H = \langle s_0, s_1, \ldots, s_{b-1} \rangle$. For each vertex $v \in V$, we define an *infinite pattern* of v w.r.t σ_H, denoted by $\mathbf{p}_v$ be a b-dimensional vector where for each $i \in [0, b-1]$

$$\mathbf{p}_v[i] = \begin{cases} -\infty & \text{if } d_G(v \to s_i) - d_G(v \to s_0) \le -r \\ d_G(v \to s_i) - d_G(v \to s_0) & \text{if } -(r-1) \le d_G(v \to s_i) - d_G(v \to s_0) \le r-1 \\ +\infty & \text{if } d_G(v \to s_i) - d_G(v \to s_0) \ge r \end{cases} \tag{27}$$

In particular, two values $-\infty$ and $+\infty$ are used to mark that the distance $d_G(v \to s_i)$ is far smaller or larger than $d_G(v \to s_0)$. We have the following lemma analogous to Lemma 4.

Lemma 15. *H be an induced sub-digraph of a K_h-minor-free digraph G, and σ_H be an arbitrary sequence of vertices in ∂H. Let $P = \{\mathbf{p}_v : v \in V\}$ be the set of all infinite patterns w.r.t. σ_H. Then $|P| = O((|\partial H| \cdot |V(H)|)^{h^2})$.*

Proof. The proof follows the same line of the proof of Lemma 4: we show that there is a bijection between the set of patterns P and $\widehat{\mathcal{LP}}_{G,M}(\partial H)$ for an appropriate choice if M. Let $M = \{-r, -(r-1), \ldots, (r-1), +r\}$. Observe that $|M| \le 2r+1$. Consider the VC set system $\widehat{\mathcal{LP}}_{G,M}(\partial H)$, which has VC dimension at most h^2 by Theorem 4. By the Sauer–Shelah Lemma (Lemma 2), we have $|\widehat{\mathcal{LP}}_{G,M}(\partial H)| = O((br)^{h^2})$. To see that there is a bijection between the set of patterns P and $\widehat{\mathcal{LP}}_{G,M}(\partial H)$, we simply flatten each pattern $\mathbf{p}_v$ to obtain a set $\widehat{X}_v \in \widehat{\mathcal{LP}}_{G,M}(\partial H)$ in exactly the same way we did in Lemma 4. □

We now define the distance from an infinite pattern to a vertex. Let v be a vertex in H, and $\mathbf{p}$ be a pattern (of some vertex u) w.r.t. σ_H. Let $\mathsf{reach}(v, \partial H)$ be the set of boundary vertices of H

that can reach v (via directed paths) *in* H. (We do not count boundary vertices that can reach v in G.) We define the distance from $\mathbf{p}$ to v, denoted by $d(\mathbf{p} \to v)$, to be:

$$d(\mathbf{p} \to v) = \begin{cases} \text{undefined} & \text{if } \mathbf{p}[i] = +\infty \text{ for some } i \\ \min_{s_i \in \mathsf{reach}(v, \partial H)} \{d_G(s_i \to v) + \mathbf{p}[i]\} & \text{otherwise} \end{cases} \tag{28}$$

In undirected graphs, we show in Lemma 5 that if $u \notin V(H)$ and $v \in V(H)$, then $d_G(u, s_0) + d_G(\mathbf{p}_u, v) = d_G(u, v)$ where $\mathbf{p}_u$ is the pattern of u. This no longer holds in digraphs. In particular, $d_G(u \to s_0) + d_G(\mathbf{p}_u \to v)$ now may be undefined or larger than $d_G(u \to v)$. However, we are still able to extract information by looking at all distances $\{d_G(\mathbf{p}_u \to v)\}_{v \in V(H)}$. In particular, we show in the following lemma that we can recover *the maximum distance* from u to a vertex in H, via $\{d_G(\mathbf{p}_u \to v)\}_{v \in V(H)}$, *provided that* $d_G(u \to s_0)$ *is the maximum among all boundary vertices.*

Lemma 16. *Let $u \in V \setminus V(H)$ be a vertex not in H, and $\mathbf{p}_u$ be the pattern of u w.r.t σ_H. Define:*

$$\Delta(u \to H) = d_G(u \to s_0) + \max_{v \in V(H)} \{d(\mathbf{p}_u \to v)\} \tag{29}$$

If $d_G(u \to s_0) = \max_{0 \le i \le |\partial H| - 1} \{d_G(u \to s_i)\}$, then $\Delta(u \to H) = \max_{v \in V(H)} d_G(u \to v)$.

Proof. As $d_G(u \to s_0)$ is maximum, $d_G(u, s_i) - d_G(u \to s_0) \le 0$ for every $i \in [0, b-1]$. Thus, no entry of $\mathbf{p}_u$ is $+\infty$. Therefore, $d(\mathbf{p}_u \to v)$ is defined (but could still be $-\infty$).

Claim 6. *If there exists a boundary vertex $s_i \in \mathsf{reach}(v, \partial H)$ such that $\mathbf{p}_u[i] = -\infty$, then $d_G(u \to v) < d_G(u \to s_0)$.*

Proof. $\mathbf{p}_u[i] = -\infty$ implies that $d_G(u, s_i) \le d_G(u \to s_0) - r$. As v is reachable from s_i in H, there is a path of length at most $|V(H)| - 1 = r - 1$ from s_i to v, meaning that $d_G(s_i \to v) \le r - 1$. Thus, $d_G(u \to v) \le d_G(u \to s_i) + d_G(s_i \to v) \le d_G(u \to s_0) - r + (r-1) < d_G(u \to s_0)$ as claimed. □

By definition of the distance in Equation (28), if there exists a boundary vertex $s_i \in \mathsf{reach}(v, \partial H)$ such that $\mathbf{p}_u[i] = -\infty$, then $d(\mathbf{p}_u \to v) = -\infty$ and hence would have no effect in the computation of $\Delta(u \to H)$. And by Claim 6, such a vertex v also do not contribute to $\max_{v \in V(H)} d_G(u \to v)$. Thus, we only need to consider vertices v such that for every boundary vertex $s_i \in \mathsf{reach}(v, \partial H)$, $\mathbf{p}_u[i] \ne -\infty$. We claim that for such vertices, $d_G(u \to s_0) + d(\mathbf{p}_u \to v)$ is the distance from u to v.

Claim 7. *If for every boundary vertex $s_i \in \mathsf{reach}(v, \partial H)$, $\mathbf{p}_u[i] \ne -\infty$, then $d_G(u \to v) = d_G(u \to s_0) + d(\mathbf{p}_u \to v)$.*

Proof. The assumption of the claim implies that $d_G(u \to s_i) = d_G(u \to s_0) + \mathbf{p}_u[i]$ for every boundary vertex $s_i \in \mathsf{reach}(v, \partial H)$. Let s_ℓ for some $\ell \in [0, b-1]$ be the boundary vertex on the path $\pi(u \to v, G)$ *furthest* from u. That is, the subpath from s_ℓ to v of $\pi(u \to v, G)$ lies entirely in H. Thus, $s_\ell \in \mathsf{reach}(v, \partial H)$ and $d_G(s_\ell \to v) = d_H(s_\ell \to v)$. Then:

$$\begin{aligned} d_G(u \to v) &= d_G(u \to s_\ell) + d_G(s_\ell \to v) \\ &= \min_{s_i \in \mathsf{reach}(v, \partial H)} \{d_G(u \to s_i) + d_G(s_i \to v)\} \\ &= \min_{s_i \in \mathsf{reach}(v, \partial H)} \{d_G(u \to s_0) + \mathbf{p}_u[i] + d_G(s_i \to v)\} \\ &= d_G(u \to s_0) + \min_{s_i \in \mathsf{reach}(v, \partial H)} \{d_G(v \to s_i) + \mathbf{p}_u[i]\} \\ &= d_G(u \to s_0) + d_G(\mathbf{p}_u \to v) \ , \end{aligned}$$

as desired. □

Let $\delta = \max_{v \in V(H)} d_G(u \to v)$ and $v^* \in V(H)$ be such that $d_G(u \to v^*) = \delta$. Observe that $\delta \geq d_G(u \to s_0)$ since s_0 is an eligible choice for v^*. We now show that for every boundary vertex s_j such that $s_j \in \mathsf{reach}(v^*, \partial H)$, $\mathbf{p}_u[j] \neq -\infty$. If so, by Claim 7, $\delta = d_G(u \to s_0) + d(\mathbf{p}_u \to v^*)$, which implies the lemma.

To see that $\mathbf{p}_u[j] \neq -\infty$, first observe that $d_G(s_j \to v^*) \leq r - 1$ as $s_j \in \mathsf{reach}(v, \partial H)$, and that $d_G(u \to s_j) + d_G(s_j \to v^*) \geq d_G(u \to v^*) = \delta$. Thus, we have:

$$d_G(u \to s_j) \geq \delta - d_G(s_j \to v^*) \geq \delta - (r-1) \geq d_G(u \to s_0) - (r-1)$$

which gives $d_G(u \to s_j) - d_G(u \to s_0) \geq -(r-1)$. Furthermore, by definition of s_0, $d_G(u \to s_j) - d_G(u \to s_0) \leq 0$. Thus, $\mathbf{p}_u[j] \neq -\infty$ as desired. □

We call the first boundary vertex s_0 in a sequence of boundary vertex σ_H of H the *base* of σ_H. We remark that in Lemma 16, it is important that the distance from u to the base vertex satisfies $d_G(u \to s_0) = \max_{0 \leq i \leq |\partial H|-1}\{d_G(u \to s_i)\}$, we call this condition the *maximum base condition*. In general, for any fixed sequence σ_H, if the maximum base condition is satisfied for u, it might not be satisfied for some vertex v. Thus, in the following algorithm for computing all-vertices eccentricities, we have to consider $|\partial H|$ different boundary sequences, each has a different boundary vertex as the base. We note that only the base vertex is important; the order of remaining vertices in a sequence σ_H could be arbitrary.

The algorithm. The algorithm for computing all-vertices eccentricities has 3 steps. Here we focus on presenting the ideas and then discuss the implementation later.

- **(Step 1).** Construct an r-division $\mathcal{R}$ of G for $r = n^{2/(3h^2+6)}$. For each cluster $R \in \mathcal{R}$, we construct a set, denoted by Γ_R, of $|\partial R|$ different sequences of boundary vertices of R such that each sequence in Γ_R admits a different boundary vertex as the base. We write $\Gamma_R = \{\sigma_R^1, \sigma_R^2, \ldots, \sigma_R^{|\partial R|}\}$. Then for each sequence σ_R^t for $t \in [|\partial R|]$, we construct a set of infinite patterns w.r.t σ_R^t: $P_R^t = \{u \in V : \mathbf{p}_u^t\}$ where $\mathbf{p}_u^t$ is the infinite pattern of u w.r.t σ_R^t. Let $\mathcal{P}_R = \{P_R^t\}_t$.
- **(Step 2).** For each cluster $R \in \mathcal{R}$, each pattern $\mathbf{p} \in (\bigcup_{P_R \in \mathcal{P}_R} P_R)$, find $v = \arg\max_{\tilde{v} \in V(R)} d(\mathbf{p} \to \tilde{v})$; we exclude undefined distances in the search for v. That is, v is the vertex that has maximum distance from $\mathbf{p}$ over all vertices in $V(R)$; we say that v is the *furthest vertex* from $\mathbf{p}$. We then store the distance $d(\mathbf{p} \to v)$ in a table.
- **(Step 3).** We now compute $\mathsf{ecc}(u)$ for each vertex $u \in V$. For each cluster $R \in \mathcal{R}$, we compute the distance from u to a vertex $v \in R$ furthest from u, denoted by $\Delta(u \to R)$, as follows. If $u \in R$, we then simply compute $\Delta(u \to R)$ by using BFS. If $u \notin R$, let s_t be the furthest boundary vertex in R: $d_G(u \to s_t) = \max_{s \in \partial R} d_G(u \to s)$. Let P_R^t be the set of infinite patterns w.r.t the boundary sequence that has s_t as the base computed in (Step 1). Let $\mathbf{p}_u^t$ be the pattern of u in P_R^t. Let v be the furthest vertex from $\mathbf{p}_u^t$, computed in (Step 2). Then we return $\Delta(u \to R) = d_G(u, s_t) + d(\mathbf{p}_u^t, v)$. By Lemma 16, $\Delta(u \to R)$ is the maximum distance from u to vertices in R. Finally, we compute $\mathsf{ecc}(u) = \max_{R \in \mathcal{R}} \Delta(u \to R)$.

As discussed in (Step 3), Lemma 16 implies that the computed value $\mathsf{ecc}(u)$ is the eccentricity of u. We now show an efficient implementation and analyze its running time.

Efficient implementation. Implementing the algorithm for digraphs shown above in truly subquadratic time turns out harder than the algorithm for undirected graphs in Section 3.2.1. One reason is that each vertex u now is associated with up to $\sqrt{r}$ different pattern vectors, each for one boundary sequence, in the same cluster R. As each pattern vector has size up to $\sqrt{r}$, the total amount of information per vertex u, and per cluster R is $O(r)$. The number of clusters is $\tilde{O}(n/\sqrt{r})$ in Lemma 1. The number of clusters can indeed be improved to $\tilde{O}(n/r)$ if one is willing to pay more running time. Even in the best case on the size of the number of clusters, the total amount of computation, if done carelessly, is $\tilde{O}(nr \cdot (n/r) \cdot r) = \tilde{O}(n^2)$, which is larger than permitted.

The key idea in the implementation is not to compute all the patterns of all vertices in the graph. As we see in (Step 3), we only need to compute a pattern $\mathbf{p}_u$ associated with a specific boundary sequence where the base of the sequence is the furthest boundary vertex; other boundary sequences are not relevant to computing $\Delta(u \to R)$. And this is what we will do: we *will not* compute all the sets Γ_R and $\mathcal{P}_R$ as described in (Step 1) up front. Instead, we will implement (Step 3) directly first, and then add patterns to $\mathcal{P}_R$ along the way we examine each vertex u.

Recall that B is the set of boundary vertices of the r-division $\mathcal{R}$: $B = \cup_{R \in \mathcal{R}} \partial R$. Let $D(B \to V) = \{d_G(s \to v) : (s, v) \in B \times V\}$, $D(V \to B) = \{d_G(v \to s) : (v, s) \in V \times B\}$. Observation 2 remains true here:

Observation 4. *$D(B \to V)$ and $D(V \to B)$ can be computed in time $\tilde{O}(n^2/\sqrt{r})$.*

We also obtain a polynomial bound on the number of infinite patterns as a corollary of Lemma 15.

Corollary 7. *$\sum_{P_R \in \mathcal{P}_R} |P_R| = \tilde{O}(r^{(3h^2+1)/2})$ for every $R \in \mathcal{R}$.*

Proof. The number of infinite patterns per boundary sequence σ_R is $O((|\partial R| \cdot |V(R)|)^{h^2}) = \tilde{O}(r^{3h^2/2})$. The corollary follows from the fact that we have up to $\sqrt{r}$ different boundary sequences. □

Now we show the detailed implementation of the algorithm, given $D(B \to V)$ and $D(V \to B)$. In (Step 1), we now only form all $|\partial R|$ boundary sequences – the set Γ_R– for each cluster R. The total running time per region is $O(|\partial R|^2) = \tilde{O}(r)$. By Lemma 1, the running time to find all $\{\Gamma_R\}_{R \in \mathcal{R}}$ is:

$$\tilde{O}(n \cdot r/\sqrt{r}) = \tilde{O}(n \cdot r^{1/2}) \tag{30}$$

Now we jump to (Step 3). For each vertex u and each cluster R, we first find the furthest boundary vertex s_t, in $O(|\partial R|)$ time by looking through all the distances from u to vertices ∂R stored in $D(V \to B)$. Thus, the total running time of finding all furthest boundary vertices over all vertices and all clusters is:

$$n \sum_{R \in \mathcal{R}} |\partial R| = \tilde{O}(n^2/\sqrt{r}) \tag{31}$$

by Lemma 1. Now we know the boundary sequence σ_R^t, computed in (Step 1), as we know the furthest vertex s_t. We can compute $\mathbf{p}_u^t$, which is the pattern satisfying the maximum base condition, in $O(|\partial R|)$ time; the same time it takes to find s_t. Thus, the total running time to find all infinite patterns satisfying the maximum base condition over all vertices and clusters is $\tilde{O}(n^2/\sqrt{r})$ by Equation (31).

Finally, we have to compute $\max_{\tilde{v} \in R} d(\mathbf{p}_u^t \to \tilde{v})$, and find the vertex v which is furthest from $\mathbf{p}_u^t$ in R. We could not naively iterate over all vertices in R every time we examine a vertex u. Recall that the number of distinct infinite patterns per cluster R is $\tilde{O}(r^{(3h^2+1)/2})$, and hence many vertices will share the same infinite patterns. We then could store results computed before in table

and do table look up if we encounter the same pattern again. More specifically, we store a trie data structure $\mathcal{L}$: for a vertex u, if $\mathbf{p}_u^t$ is not in $\mathcal{L}$, which we can check in $O(|\partial R|)$ time, then we iterate over all vertices in R to find v, and store v and $d(\mathbf{p}_u^t \to v)$ in $\mathcal{L}$, keyed by $\mathbf{p}_u^t$. Otherwise, we simply lookup v and $d(\mathbf{p}_u^t \to v)$ from $\mathcal{L}$. Modulo the running time to find v and $d(\mathbf{p}_u^t \to v)$ when $\mathbf{p}_u^t \notin \mathcal{L}$, the total time to look up v and $d(\mathbf{p}_u^t \to v)$ will be $|\partial R|$, which is also the time to find s_t. Thus, the total running time is $\tilde{O}(n^2/\sqrt{r})$ by Equation (31).

We now bound the running time to find v and $d(\mathbf{p} \to v)$ for every pattern $\mathbf{p} \in (\bigcup_{P_R \in \mathcal{P}_R} P_R)$. For each given pattern $\mathbf{p}$, computing the distance from $\mathbf{p}$ to a vertex $\tilde{v} \in R$ can be done in $O(|\partial R|) = \tilde{O}(\sqrt{r})$ time by definition in Equation (28). Then finding $\max_{v \in R} d(\mathbf{p} \to v)$ can be done in $|V(R)|\tilde{O}(\sqrt{r}) = \tilde{O}(r^{3/2})$ time. Over all patterns in $(\bigcup_{P_R \in \mathcal{P}_R} P_R)$, by Corollary 7, the total running time is $\tilde{O}(r^{(3h^2+1)/2} \cdot r^{3/2}) = \tilde{O}(r^{(3h^2+6)/2})$. Over all clusters in $\mathcal{R}$, the running time is:

$$\tilde{O}(n/\sqrt{r}) \cdot \tilde{O}(r^{(3h^2+6)/2}) = \tilde{O}(nr^{(3h^2+5)/2}) \tag{32}$$

In summary, by Equation (30), Equation (31), and Equation (32), the total running time to compute all-vertices eccentricities is:

$$\tilde{O}(nr^{(3h^2+5)/2}) + \tilde{O}(n^2/\sqrt{r}) = \tilde{O}(n^{2-1/(3h^2+6)}) \tag{33}$$

when $r = n^{2/(3h^2+6)}$, as claimed in Corollary 3. This is also the running time to compute the directed diameter of G.

5 Lower Bound for Directed VC-dim Edge Set System

In this section, we prove the lower bound in Theorem 6, which we restate below.

Theorem 6. *For any constant integer $r \geq 1$, there exists an unweighted planar digraph $G = (V, E)$ and a subset $X \subseteq E$ of size r such that X is shattered by $\overrightarrow{\mathcal{SP}}(G)$.*

We will construct a graph with a set of r directed edges $X = \{e_1, e_2, \ldots, e_r\}$ on a path P such that for every subset $Y \subseteq X$, there exists a vertex $v \notin P$ such that Y belongs to the shortest path tree rooted at v.

We construct P as follows (see Figure 5(a)). First, form the set X of r directed edges, where $e_i = (u_i \to v_i)$. Then add a path of length 2 between v_{i+1} and u_i consisting of two edges in different directions: $(v_{i+1} \to x_i), (u_i \to x_i)$. The idea is to ensure that no endpoint of e_i can reach (or be reached by) other endpoints of other edges by going along the path P. Set the weight of each edge to be 1.

Now we construct shortest path trees where each tree realizes a subset Y of X. By realizing Y, we mean the subset Y will be included in some shortest path tree, while other edges in $X \setminus Y$ will not be included by the same tree. We will add edges with *integer weights* and finally we can turn them into unweighted edges by subdividing them.

Choose a sufficiently large number M and 2^r other numbers: $1 \ll A_0 \leq A_1 \leq \ldots \leq A_{2^r-1}$ where $A_i = 4A_{i-1}$. M will be sufficiently larger than all $\{A_i\}$. The following inequalities will be helpful:

$$A_i \geq 2A_t + (A_{i-1} + A_{i-2} + \ldots + A_0 + 1) \quad \text{for any } t \leq i-1 \tag{34}$$

Now consider all 2^r bit strings $\{0,1\}^r$, the i-th string, denoted by s_i, is the binary representation of i for $i \in [0, 2^r - 1]$. Starting from s_0, for each string s_i, we will add a new vertex a_i to the graph

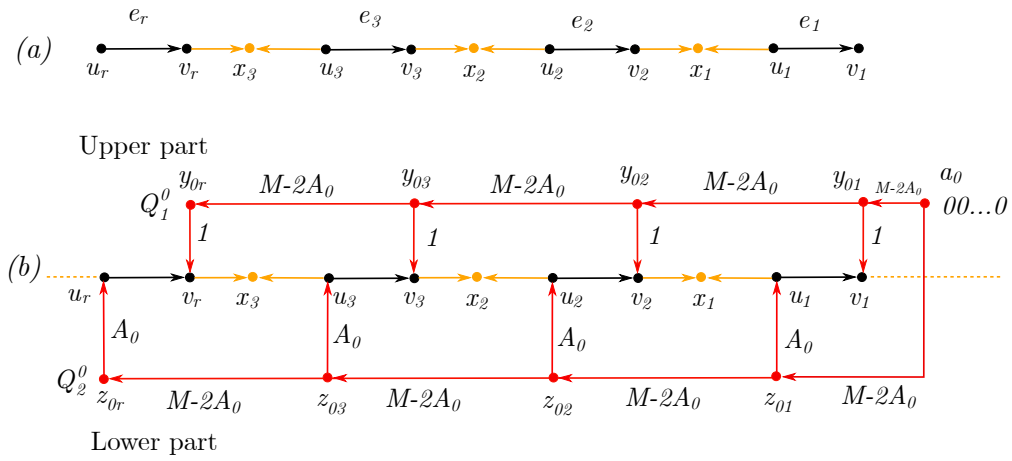

Figure 5: (a) the path P. (b) Vertex a_0, corresponding to the bit string $s_0 = 0 \ldots 0$ containing r bits of 0s, connected to P via directed edges.

along with some other vertices and directed edges. Some directed edges will be given weights based on M and A_i. Vertex a_i will be embedded outside the outer face of G_{i-1}, the directed graph constructed after step $i-1$. The final graph is $G = G_{2^r-1}$.

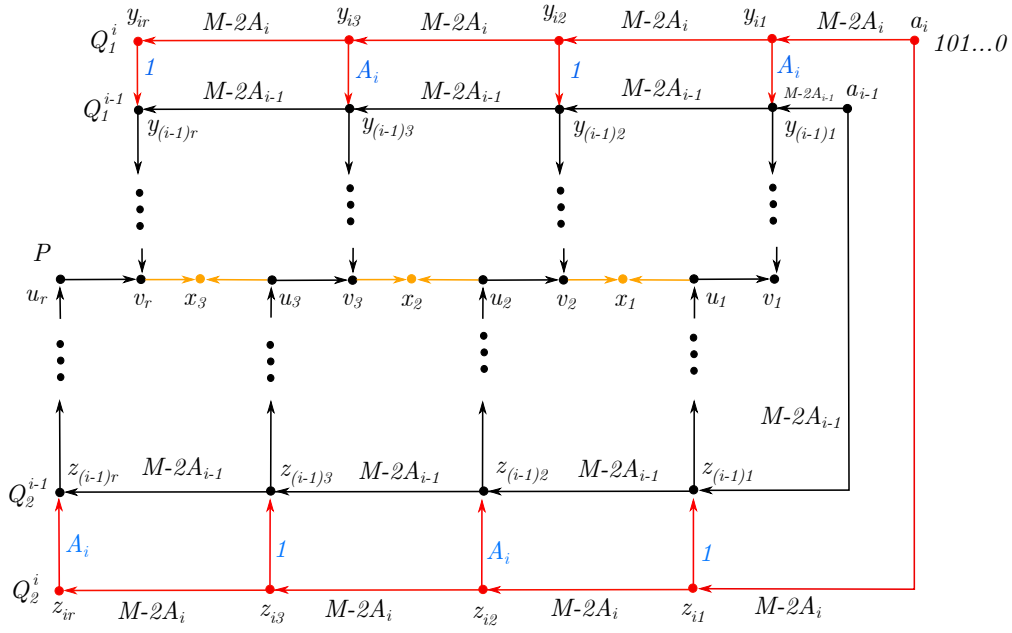

Figure 6: Adding a_i to the current graph G_{i-1}.

The path P will separate the plane into two parts: the upper part (or 0 part) and the lower part (or 1 part); see Figure 5(b). The upper part will realize the fact that some edges of X are NOT added to the shortest path tree of a_i and the lower part serves the opposite purpose. In particular, if the j-th bit of s_i is 0, then the shortest path from a_i to v_j will only contain edges from the upper part and hence does not contain e_j. Otherwise, the shortest path from a_i to v_j will only contain

edges from the lower part and e_j. The rule for adding a_i is as follows (see Figure 6):

1. Add two directed paths $Q_1^i = (a_i = y_{i0} \to y_{i1} \ldots \to y_{ir})$ and $Q_2^i = (a_i = z_{i0} \to z_{i1} \ldots \to z_{ir})$ directed away from a_i. Each path has exactly r edges. Each edge of the two paths is assigned a weight $M - 2 \cdot A_i$. Q_1^i is embedded in the upper part of the plane, separated by P, and Q_2^i is embedded in the lower part of the plane.

2. Look at the bit string s_i (see an example in Figure 6, the first 3 bits are $\{1, 0, 1\}$, and the last bit is 0 in the string of a_i). Assume that $i \geq 1$.

   - if the j-th bit is 1, add a directed edge $(y_{ij} \to y_{(i-1)j})$ of weight A_i, and add a directed edge $(z_{ij} \to z_{(i-1)j})$ of weight 1. Note that $y_{(i-1)j}$ and $z_{(i-1)j}$ are vertices on the paths Q_1^{i-1} and Q_2^{i-1}, respectively, of a_{i-1}. For example, in Figure 6, the first bit of s_i is 1 and hence we have an edge of weight A_i from y_{i1} to $y_{(i-1)1}$ and an edge of weight 1 from z_{i1} to $z_{(i-1)1}$. The same holds for y_{i3} and z_{i3}.
   - if the j-th bit is 0, add a directed edge $(y_{ij} \to y_{(i-1)j})$ of weight 1, and add a directed edge $(z_{ij} \to z_{(i-1)j})$ of weight A_i. For example, in Figure 6, the second bit of s_i is 0 and hence we have an edge of weight 1 from y_{i2} to $y_{(i-1)2}$ and an edge of weight A_i from z_{i2} to $z_{(i-1)2}$. The same holds for y_{ir} and z_{ir}.

   When $i = 0$ (see Figure 5(b)), we connect y_{0j} to v_j and z_{0j} to u_j for every $j \in [1, r]$. Note that, since s_0 only contains 0 bits, every edge $(y_{0j} \to v_j)$ has weight 1 and every edge $(z_{0j} \to u_j)$ has weight A_0. In this case, no edge in X will be included in the shortest path tree of a_0.

We now analyze the shortest path tree of a_i, denote by T_i. We say that an edge is *horizontal* if it belongs to P or Q_1^t or Q_2^t for some $t \in [0, 2^r - 1]$; otherwise, we say that the edge is *vertical*. Note that a vertical edge is either an edge from a vertex $y_{tj} \in Q_1^t$ *down* to some vertex in Q_1^{t-1} when $t \geq 1$ or down to some vertex in P when $t = 0$, or from a vertex $z_{tj} \in Q_2^t$ *up* to a vertex in Q_2^{t-1} when $t \geq 1$ or to a vertex in P when $t = 0$.

The following claim is the key to the proof; see Figure 7.

Claim 8. *For any $j \in [1, r]$, then the shortest path from a_i to v_j in $G \setminus \{e_j\}$ consists of: (a) the subpath from a_i to y_{ij} of Q_1^i, which only contains horizontal edges, and the directed paths from y_{ij} to v_j, which only contains vertical edges.*

Proof. The shortest path from a_i to v_j is the path highlighted green in Figure 7. Let W_1 be the path from a_i to v_j as described in the claim. Note that there is a unique directed path from y_{ij} to v_j, which only contains vertical edges. Similarly, there is a unique directed path from a_i to y_{ij} which only contains horizontal edges. Thus, if the shortest path from a_i to v_j goes through y_{ij}, the path must be W_1.

Let R be the shortest path from a_i to v_j; assume that $R \neq W_1$. Thus, $y_{ij} \notin R$. Observe that R can only include edges in the upper part of P since e_j is deleted from G. That is, R does not contain any z-vertex. Furthermore, R contains exactly j horizontal edges and i vertical edges. Note that W_1 also contains exactly j horizontal edges and i vertical edges. Additionally, the weight any horizontal edge of W_1 is at most the weight of any horizontal edge of R by the choice of $\{A_i\}_{i=0}^{2^r-1}$.

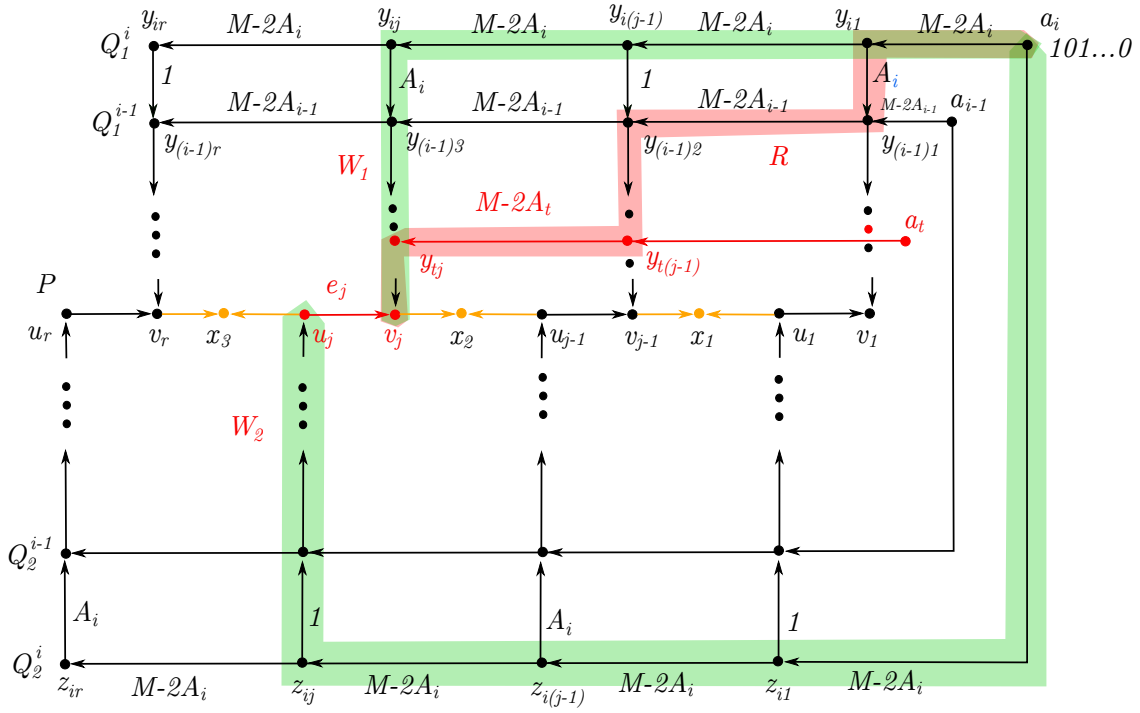

Figure 7: Two green-highlighted paths are shortest path from a_i to v_j and to u_j in graphs $G \setminus \{e_j\}$ and G_i, respectively.

Let $(y_{t(j-1)} \to y_{tj})$ be the last horizontal edge of R; see Figure 7. As $y_{ij} \notin R$, $t < i$. Thus we have:

$$\begin{aligned} w(R) - w(W_1) &\geq w(y_{t(j-1)} \to y_{tj}) - w(y_{i(j-1)} \to y_{ij}) - (\text{ total weight of all vertical edges of } W_1) \\ &\geq M - 2A_t - (M - 2A_i) - (A_i + A_{i-1} + \dots A_0) \\ &\geq A_i - 2A_t - (A_{i-1} + \dots A_0) \geq 1 \quad \text{(by Equation (34))} \end{aligned} \tag{35}$$

This contradicts that R is a shortest path from a_i to v_j. □

The following claim is similar to Claim 8, except that the graph is G. The proof is exactly the same.

Claim 9. *For any $j \in [1, r]$, the shortest path from a_i to u_j in G consists of: (a) the subpath from a_i to z_{ij} of Q_2^i, which only contains horizontal edges, and the directed paths from z_{ij} to u_j, which only contains vertical edges.*

Lastly, we claim that e_j is in T_i or not is equivalent to whether the j-th bit in the bit string s_i is 1 or 0. This implies that the directed edge set system does not have a bounded VC dimension.

Claim 10. *For any $j \in [1, r]$, if $s_i[j] = 1$ then $e_j \in T_i$; otherwise, $e_j \notin T_i$*

Proof. Let W_1 be the shortest path from a_i to v in $G \setminus e_j$, and W_2 be the shortest path from a_i to u_j in G. Suppose that $s_i[j] = 1$, then $w(y_{ij} \to y_{(i-1)j}) = A_i$ and $w(z_{ij} \to z_{(i-1)j}) = 1$. See Figure 7.

Note that both W_1 and W_2 contain the same number of horizontal edges, each of the same weight $M - 2A_i$. Thus, we have:

$$\begin{aligned} w(W_1) - w(W_2 \circ e_j) &\geq A_i - (\text{ total weight of all vertical edges of } W_2) - 1 \\ &\geq A_i - (1 + A_{i-1} + \dots A_0) - 1 > 0 \quad \text{(by Equation (34))} \end{aligned} \tag{36}$$

Thus, $W_2 \circ e_j$ is the shortest path from a_i to v_j, implying the claim. The proof that if $s_i[j] = 0$ then $e_j \notin T_i$ follows the same line. □

6 Conclusion

In this work, we propose a systematic study of VC set systems in minor-free graphs, both directed and undirected. Our work leaves many open problems. First, could we establish a formal relationship between our set system $\widehat{\mathcal{LP}}_{G,M}(S)$ and the original set system $\mathcal{LP}_{G,M}(S)$ by Li and Parter [LP19] in the sense that if one has a bounded VC dimension, then the other also does. This will imply that $\mathcal{LP}_{G,M}(S)$ is a VC set system for any M, S and any minor-free graph G. The second question is to extend all results here to graphs beyond minor-free, such as graphs of polynomial expansion and nowhere dense graphs. The third question is, could we design a truly subquadratic space distance oracle with *constant* query time for minor-free digraphs? Our oracle in Corollary 4 has $O(\log n)$ query time. The fourth question is to obtain a similar metric compression result for digraphs. As far as we know, our Theorem 4 is not sufficient for metric compression as we do not have the triangle inequality in digraphs.

References

[AWW16] A. Abboud, V. V. Williams, and J. Wang. Approximation and fixed parameter subquadratic algorithms for radius and diameter in sparse graphs. In *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '16, page 377–391, 2016.

[Bak94] B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994, doi:10.1145/174644.174650.

[BBE+21] N. Bousquet, W. C. V. Batenburg, L. Esperet, G. Joret, W. Lochet, C. Muller, and F. Pirot. Packing and covering balls in graphs excluding a minor. *Combinatorica*, 41(3):299–318, 2021, doi:10.1007/s00493-020-4423-3.

[BC14] G. Borradaile and E. W. Chambers. Covering nearly surface-embedded graphs with a fixed number of balls. *Discrete & Computational Geometry*, 51(4):979–996, 2014, doi:10.1007/s00454-014-9594-5.

[BP21] G. Bodwin and M. Parter. Restorable shortest path tiebreaking for edge-faulty graphs. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*, 2021, doi:10.1145/3465084.3467913.

[BT15] N. Bousquet and S. Thomassé. VC-dimension and Erdős–Pósa property. *Discrete Mathematics*, 338(12):2302–2317, 2015, doi:10.1016/j.disc.2015.05.026.

[Cab18] S. Cabello. Subquadratic algorithms for the diameter and the sum of pairwise distances in planar graphs. *ACM Transactions on Algorithms*, 15(2), 2018. Announced at SODA'17, doi:10.1145/3218821.

[CADWN17] V. Cohen-Addad, S. Dahlgaard, and C. Wulff-Nilsen. Fast and compact exact distance oracle for planar graphs. In *IEEE 58th Annual Symposium on Foundations of Computer Science*, FOCS '17, pages 962–973, 2017, doi:10.1109/FOCS.2017.93.

[CC07] S. Cabello and E. W. Chambers. Multiple source shortest paths in a genus g graph. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 89–97. Society for Industrial and Applied Mathematics, 2007.

[CCE13] S. Cabello, E. W. Chambers, and J. Erickson. Multiple-source shortest paths in embedded graphs. *SIAM Journal on Computing*, 42(4):1542–1571, 2013, doi:10.1137/120864271.

[CEV07] V. Chepoi, B. Estellon, and Y. Vaxes. Covering planar graphs with a fixed number of balls. *Discrete & Computational Geometry*, 37(2):237–244, 2007, doi:10.1007/s00454-006-1260-0.

[CGMW19] P. Charalampopoulos, P. Gawrychowski, S. Mozes, and O. Weimann. Almost optimal distance oracles for planar graphs. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC '19, pages 138–151, 2019, doi:10.1145/3313276.3316316.

[CK97] V. Chepoi and S. Klavžar. The wiener index and the szeged index of benzenoid systems in linear time. *Journal of Chemical Information and Computer Sciences*, 37(4):752–755, 1997, doi:10.1021/ci9700079.

[CK09] S. Cabello and C. Knauer. Algorithms for graphs of bounded treewidth via orthogonal range searching. *Computational Geometry*, 42(9):815–824, 2009, doi:10.1016/j.comgeo.2009.02.001.

[DFHT05] E. D. Demaine, F. V. Fomin, M. Hajiaghayi, and D. M. Thilikos. Subexponential parameterized algorithms on bounded-genus graphs and h-minor-free graphs. *Journal of the ACM*, 52(6):866–893, 2005, doi:10.1145/1101821.1101823.

[DH05] E. D. Demaine and M. Hajiaghayi. Bidimensionality: New connections between FPT algorithms and PTASs. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'05, pages 590–601, 2005.

[DHM10] E. D. Demaine, M. Hajiaghayi, and B. Mohar. Approximation algorithms via contraction decomposition. *Combinatorica*, 30(5):533–552, 2010, doi:10.1007/s00493-010-2341-5.

[DHT04] E. D. Demaine, M. Hajiaghayi, and D. M. Thilikos. The bidimensional theory of bounded-genus graphs. In *Proceedings of the 29th Symposium on Mathematical Foundations of Computer Science*, MFCS '04, pages 191–203. 2004, doi:10.1007/978-3-540-28629-5_12.

[DHV20] D. Ducoffe, M. Habib, and L. Viennot. Diameter computation on h-minor free graphs and graphs of bounded (distance) vc-dimension. In *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '20, page 1905–1922, 2020.

[DPBF09] F. Dorn, E. Penninkx, H. L. Bodlaender, and F. V. Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut decompositions. *Algorithmica*, 58(3):790–810, 2009, doi:10.1007/s00453-009-9296-1.

[Epp03] D. Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '03, pages 599—-608, 2003, doi:10.5555/644108.644208.

[Eri10] J. Erickson. Maximum flows and parametric shortest paths in planar graphs. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010, doi:10.1137/1.9781611973075.65.

[Fed87] G. N. Federickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM Journal on Computing*, 16(6):1004–1022, 1987, doi:10.1137/0216064.

[FHMWN20] V. Fredslund-Hansen, S. Mozes, and C. Wulff-Nilsen. Truly subquadratic exact distance oracles with constant query time for planar graphs. *arXiv preprint arXiv:2009.14716*, 2020. https://arxiv.org/abs/2009.14716.

[FR01] J. Fakcharoenphol and S. Rao. Planar graphs, negative weight edges, shortest paths, and near linear time. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, 2001, doi:10.1109/sfcs.2001.959897.

[GKM+21] P. Gawrychowski, H. Kaplan, S. Mozes, M. Sharir, and O. Weimann. Voronoi diagrams on planar graphs, and computing the diameter in deterministic $\tilde{O}(n^{5/3})$ time. *SIAM Journal on Computing*, 50(2):509–554, 2021, doi:10.1137/18m1193402.

[GMWWN18] P. Gawrychowski, S. Mozes, O. Weimann, and C. Wulff-Nilsen. Better tradeoffs for exact distance oracles in planar graphs. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, number SODA '18, pages 515–529, 2018, doi:10.1137/1.9781611975031.34.

[Hus17] T. Husfeldt. Computing Graph Distances Parameterized by Treewidth and Diameter. In *11th International Symposium on Parameterized and Exact Computation (IPEC 2016)*, volume 63, pages 16:1–16:11, 2017, doi:10.4230/LIPIcs.IPEC.2016.16.

[HW87] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete & Computational Geometry*, 2(2):127–151, 1987, doi:10.1007/bf02187876.

[JR23] G. Joret and C. Rambaud. Neighborhood complexity of planar graphs. *arXiv preprint arXiv:2302.12633*, 2023.

[KKR+97] E. Kranakis, D. Krizanc, B. Ruf, J. Urrutia, and G. Woeginger. The VC-dimension of set systems defined by graphs. *Discrete Applied Mathematics*, 77(3):237–257, 1997, doi:10.1016/s0166-218x(96)00137-0.

[KKR12] K. Kawarabayashi, Y. Kobayashi, and B. Reed. The disjoint paths problem in quadratic time. *Journal of Combinatorial Theory, Series B*, 102(2):424–435, 2012, doi:10.1016/j.jctb.2011.07.004.

[Kle05a] P. N. Klein. A linear-time approximation scheme for planar weighted TSP. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '05, pages 647–657, 2005, doi:10.1109/SFCS.2005.7.

[Kle05b] P. N. Klein. Multiple-source shortest paths in planar graphs. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, page 146–155. Society for Industrial and Applied Mathematics, 2005.

[KR10] K. Kawarabayashi and B. Reed. A separator theorem in minor-closed classes. In *Proceedings of the 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, 2010, doi:10.1109/focs.2010.22.

[KTW18] K. Kawarabayashi, R. Thomas, and P. Wollan. A new proof of the flat wall theorem. *Journal of Combinatorial Theory, Series B*, 129:204–238, 2018, doi:10.1016/j.jctb.2017.09.006.

[KTW20] K. Kawarabayashi, R. Thomas, and P. Wollan. Quickly excluding a non-planar graph, 2020, doi:10.48550/ARXIV.2010.12397.

[Le23] H. Le. Approximate distance oracles for planar graphs with subpolynomial error dependency. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '23, pages 1877–1904. Society for Industrial and Applied Mathematics, 2023.

[LP19] J. Li and M. Parter. Planar diameter via metric compression. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 152–163, 2019, doi:10.1145/3313276.3316358.

[LP21] Y. Long and S. Pettie. Planar distance oracles with better time-space tradeoffs. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA'21*, pages 2517–2537, 2021.

[LT79] R. Lipton and R. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

[LT80] R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM Journal on Computing*, 9(3):615–627, 1980, doi:10.1137/0209046.

[MNNW18] S. Mozes, K. Nikolaev, Y. Nussbaum, and O. Weimann. Minimum cut of directed planar graphs in $o(n \log\log n)$ time. In *Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 477–494. 2018, doi:10.1137/1.9781611975031.32.

[MS12] S. Mozes and C. Sommer. Exact distance oracles for planar graphs. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA'12, pages 209–222, 2012, doi:10.1137/1.9781611973099.19.

[RS83] N. Robertson and P. D. Seymour. Graph minors. I. Excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983, doi:10.1016/0095-8956(83)90079-5.

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

[RS04] N. Robertson and P. D. Seymour. Graph minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory Series B*, 92(2):325–357, 2004, doi:10.1016/j.jctb.2004.08.001.

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

[Tho04] M. Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM*, 51(6):993–1024, 2004. Announced at FOCS' 01, doi:10.1145/1039488.1039493.

[VC71] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971, doi:10.1137/1116025.

[VV86] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47(0):85 – 93, 1986, doi:10.1016/0304-3975(86)90135-0.

[Wie47] H. Wiener. Structural determination of paraffin boiling points. *Journal of the American Chemical Society*, 69(1):17–20, 1947, doi:10.1021/ja01193a005.

[WN09] C. Wulff-Nilsen. Wiener index and diameter of a planar graph in subquadratic time. In *Proceedings of the 25th European Workshop on Computational Geometry*, pages 25–28, 2009.

[WN11] C. Wulff-Nilsen. Separator theorems for minor-free and shallow minor-free graphs with applications. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science*, FOCS '11, 2011, doi:10.1109/focs.2011.15.

[WN14] C. Wulff-Nilsen. Faster separators for shallow minor-free graphs via dynamic approximate distance oracles. In *Proceedings of the 41th International Colloquium on Automata, Languages, and Programming*, ICALP '14, pages 1063–1074. 2014, doi:10.1007/978-3-662-43948-7_88.