Every Property of Outerplanar Graphs is Testable

Jasine Babu[1], Areej Khoury[2], and Ilan Newman[3]

1 **Department of Computer Science and Engg, Indian Institute of Technology Palakkad, India,**
jasine@iitpkd.ac.in
2 **Department of Computer Science, University of Haifa, Haifa, Israel,**
areejkhoury@csweb.haifa.ac.il
3 **Department of Computer Science, University of Haifa, Haifa, Israel,**
ilan@cs.haifa.ac.il

Abstract

A D-disc around a vertex v of a graph $G = (V, E)$ is the subgraph induced by all vertices of distance at most D from v. We show that the structure of an outerplanar graph on n vertices is determined, up to modification (insertion or deletion) of at most ϵn edges, by a set of D-discs around the vertices, for $D = D(\epsilon)$ that is independent of the size of the graph. Such a result was already known for planar graphs (and any hyperfinite graph class), in the limited case of *bounded degree* graphs (that is, their maximum degree is bounded by some fixed constant, independent of $|V|$). We prove this result with no assumption on the degree of the graph.

A pure combinatorial consequence of this result is that two outerplanar graphs that share the same local views are close to be isomorphic.

We also obtain the following property testing results in the sparse graph model:

- graph isomorphism is testable for outerplanar graphs by $poly(\log n)$ queries.
- every graph property is testable for outerplanar graphs by $poly(\log n)$ queries.

We note that we can replace outerplanar graphs by a slightly more general family of k-edge-outerplanar graphs. The only previous general testing results, as above, where known for forests (Kusumoto and Yoshida), and for some power-law graphs that are extremely close to be bounded degree hyperfinite (by Ito).

1998 ACM Subject Classification G.3

Keywords and phrases Property testing, Isomorphism, Outerplanar graphs

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2016.<article-no>

1 Introduction

We study property testing and the related learning problem for some classes of sparse graphs. The theory of property testing in the dense graph model is quite well understood (see [**?**] and bibliography therein). The theory of sparse graphs is less understood, and, in particular, there is no characterization of what properties can be tested, even for the bounded degree model.

Our starting point is the result in Newman-Sohler [**?**] stating roughly that every graph property can be tested by constantly many queries for *bounded* degree planar[1] graphs. The result follows a long line of previous results, and uses heavily a basic idea of Onak [**?**], and Hassidim et. al [**?**], (a.k.a. "local partition oracle") showing that a bounded degree graph G

[1] The result in[**?**] is for the larger family of hyperfinite graphs containing planar graphs.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016).
Editors: Klaus Jansen, Claire Matthieu, José D. P. Rolim, and Chris Umans; Article No. <article-no>; pp. <article-no>:1–<article-no>:??

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

can be approximated, up to the deletion of ϵn edges, by a graph G' whose components are all of constant size. Moreover, the graph G' (or a short description of it), can be obtained by making a constant number of queries to the original graph G.

This result is essentially equivalent to two other formulations: the first is that every bounded degree planar graph can be *learned* up to the deletion of ϵn edges, by making a constant number of queries to it. For the other formulation, let D be a constant natural number. The D-local views of a graph G on n vertices is the collection (multiset) of the n unlabelled discs (balls) of radius D around the n vertices. The other, purely combinatorial result, states that for any ϵ, there is a constant $D = D(\epsilon, d)$, such that if two n-vertices d-bounded degree planar graphs G, H, have the same[2] D-local neighbourhoods, then changing at most ϵn edges in G makes it isomorphic to H (we will say that G is ϵ-close to H in this case).

The results above restrict the graphs they are applicable to, in two conceptually different ways. The first is being planar (or hyperfinite). Indeed it is known that this is essential; namely, we know that similar statements as above are wrong for e.g., bounded degree, but otherwise general graphs. The other restriction is being *bounded degree*. The results above (specifically the distance measure) are defined so to be used for sparse graphs - namely of bounded (constant) *average* degree. Bounding the maximum degree is essential for the proof machinery in the papers above, but does not seem to be essential for the results. However, as of now, the only general results for non-bounded degree families of sparse graphs are known only for the much simpler family of Forests [**?**], and the special power-law graphs of Ito [**?**] (that are very close to be hyperfinite). In particular, the following, purely combinatorial question proposed by Sohler [**?**] is still wide open: Suppose that two n-vertex planar graphs H, G have identical D-local views (for some large enough constant D), is it true that the graphs are ϵ-close to be isomorphic? (ϵ-close means that we can change at most ϵn edges in one to make it isomorphic to the other).

We answer this question positively for a subclass of planar graphs that includes forests and outerplanar graphs (and k-edge-outerplanar graphs - to be defined later). We follow coarsely the route used by Newman-Sohler [**?**], and the generalization of it to non-bounded degree forests by Kusumoto and Yoshida [**?**]. As an outcome, we also obtain three other results as well: (a) every graph property is testable for this subclass. (b) isomorphism is testable for any two graphs of this subclass. (c) every such graph G can be "learned", namely one can infer a graph H that is ϵ-close to G. All results using $poly(\log n)$ many queries.

The presentation is arranged as follows: In Section **??** we present the essential definitions, and the tools we use. We then state the formal results in Section **??**, along with a road map to the structure of the proof.

2 Notations and Tools

In this paper we consider labelled undirected graphs without multiple edges and self-loops. We use $G = (V, E)$ to denote a graph with vertex set V and edge set E. We will assume by default that $V(G) = [n]$, unless otherwise stated. We will say that a graph is d-bounded degree if its maximum degree is at most d. For a set $S \subseteq V$ we denote by $E(S) = \{(u, v) \in E(G) |\ u \in S,\ v \notin S\}$, and $e(S) = |E(S)|$. A block in a graph G is a maximal 2-connected subgraph of G.

[2] The result are asserted even when the the D-local neighbourhoods are not the same, but just close enough.

The subclass of planar graphs that is discussed in this paper is that of k-edge-outerplanar graph for some fixed constant k.

▶ **Definition 2.1** (k-Edge-Outerplanar)**.** A graph G is 1-edge-outerplanar if it has a planar embedding in which all vertices of G are on the outer face.

We say that G is k-edge-outerplanar if G has a planar embedding such that if all edges on the exterior face are deleted, the connected components of the remaining graph are all $(k-1)$-edge-outerplanar.

Note: Being 1-outerplanar coincides with the standard definition of being *outerplanar*. However, for $k > 1$, being k-edge-outerplanar is a weaker notion than the standard notion of being k-outerplanar - namely, graphs that have a planar embedding such that the removal of the *vertices* on the outer face results in a $(k-1)$-outerplanar graph. In particular, a graph may be 2-outerplanar, but not k-edge-outerplanar for any given constant k.

Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are said to be *isomorphic*, if there is a bijective mapping $\Phi : V_1 \to V_2$ such that $(u, v) \in E_1$, if and only if $(\Phi(u), \Phi(v)) \in E_2$. A graph property is a (possibly infinite) collection of graphs, which is closed under isomorphism. We will consider graph properties of graphs with fixed number of vertices (n in what follows), where the number is growing to infinity. The graphs that are discussed in this paper are all sparse graphs, specifically, they are planar, and hence their average degree is at most 6.

2.1 Property Testing

▶ **Definition 2.2** (Graph distance)**.** Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be planar graphs on n vertices. The distance $dist(G_1, G_2)$ is the number of edges that needs to be deleted and/or inserted from G_1 in order to make it isomorphic to G_2.

We extend the definition of $dist(G_1, G_2)$ for the case where G_1 and G_2 have different number of vertices, by adding a sufficient number of isolated vertices to the graph with the lesser number of vertices.

We say that G_1, G_2 are ϵ-far from being isomorphic (or G_1 is ϵ-far from G_2), if $dist(G_1, G_2) > \epsilon n$, where $n = \min\{|V_1|, |V_2|\}$. Otherwise, we say that they are ϵ-close (to being isomorphic).

▶ **Definition 2.3** (ϵ-far)**.** Let Π be any (non-empty) graph property. A graph $G = (V, E)$ is said to be *ϵ-far* from Π, if it is ϵ-far from every $G' \in \Pi$. If G is not ϵ-far from Π, it is said to be *ϵ-close* to Π.

For algorithms in the model that we will discuss in this paper, the input graph, $G = (V, E)$, is given but not known to the algorithm. The vertex set $V = [n]$ is known. The neighbours of each vertex $v \in [n]$ are assumed to be ordered, namely by a list $u_1, \ldots, u_{d(v)}$, where $d(v) = deg(v)$ is the degree of v. The access of the algorithm to the input graph is via 'neighbourhood' queries: A query is to specify a vertex name $v \in [n]$, and $i \in [n]$, on which the answer to the query is the name of the i-th neighbour of v, or a special indication if $deg(v) < i$. We further augment this standard model with an additional type of queries: On a queried vertex v, one gets $deg(v)$. It is easy to see that $deg(v)$ can be determined using the standard model at the cost of $O(\log n)$ queries[3].

The notion of property testing was introduced by Rubinfeld and Sudan [**?**] and then formally defined by Goldreich, Goldwasser and Ron [**?**]. A property testing algorithm for

[3] A good enough approximation at a better cost would suffice for all our purposes; but we do not use this here, as we do not expect to optimize the query complexity to better than $poly(\log n)$.

property Π, for the model of *sparse* graphs, or *bounded degree* graph model is a (randomized) algorithm that, given query access to a graph G as described above, accepts every graph from Π with probability at least 2/3, and rejects every graph that is ϵ-far from Π with probability at least 2/3. If the graph neither has property Π nor is ϵ-far from Π, then a property tester may accept or reject.

2.2 Partitions and the local views of the graph

For a graph $G = (V, E)$, and a set of vertices $S \subseteq V(G)$, $G[S]$ denotes the subgraph induced by S. A partition of a set V is a set of pairwise disjoint non-empty subsets of V whose union is V. For a partition $P = (C_1, C_2, ..., C_r)$ of $V(G)$ we denote by $G[P]$ the graph that is the union of $G[C_i]$. Note that $G[P]$ is disconnected if $r \geq 2$ and is obtained from G by deleting all edges whose endpoints are in different partition classes of P.

Every d-bounded degree planar graph admits a partitioning into small (constant size) connected components by removing a fraction of the edges, by using recursively the Lipton-Tarjan separator [**?**]. To be useful for property testing and sub-linear approximation algorithms, it would be nice if the features of such partitions could be obtained by some local sampling. Hassidim, Kelner, Nguyen, and Onak in a seminal work, [**?**], following an earlier work of Benjamini, Schramm and Shapira[**?**], showed how to construct an oracle to such a partition, that takes a vertex as input and returns in *constant time* the partition class that vertex belongs to.

We will use extensively the local-partition-oracle for d-bounded degree planar graphs, and the related results which we present in what follows.

A connected graph $G = (V, E)$ with a specially identified vertex v, is called *rooted graph* and we sometimes say that G is rooted at v. A rooted graph $G = (V, E)$ has radius D, if every vertex in V has distance at most D from v. Two rooted graphs G and H are isomorphic, if there is a graph isomorphism between H and G that identifies the roots with each other. For a graph $G = (V, E)$, an integer D and a vertex $v \in V$, let $B_G(v, D)$ be the subgraph rooted at v that is induced by all vertices of G that are at distance less or equal to D from v. $B_G(v, D)$ is a graph of radius at most D with root v, and we call it the D-disc around v. The collection (multiset) of the n unlabelled D-discs of G is called the D-local views of G. Note that for d-bounded degree graphs, the number of possible non-isomorphic D-discs is a constant depending on D and d, and does not depend on n.

▶ **Definition 2.4** (Frequency Vector)**.** For integers $d \geq 1$ and $s \geq 1$, a graph is called (d, s)-graph if it is d-bounded degree and has at most s vertices.

Let $\mathcal{F}(d, s) = F^{(1)}, F^{(2)}, ..., F^{(f(d,s))}$ be the family of all non-isomorphic (d, s) planar graphs and let $f(d, s) = |\mathcal{F}(d, s)|$.

For a graph $G = (V, E)$ and a partition P of V that gives a collection of (d, s) components $G[P]$, the P-frequency vector $Freq(G[P])$ is the $f(d, s)$-dimensional vector whose i-th coordinate is the number of (d, s)-components of G that are isomorphic to $F^{(i)}$. Let the normalized P-frequency vector $freq(G[P])$ be the ℓ_1-unit vector $\frac{1}{m} \cdot Freq(G[P])$, where m is the number of (d, s) components[4] in $G[P]$.

We will use the following theorem considered and proved first by Onak [**?**], and by Hassidim, Kelner, Nguyen, and Onak [**?**] on partitions of bounded degree hyperfinite graphs.

[4] In [**?**] the frequency vector is for rooted components, hence normalization is by dividing into n - the number of vertices. These two notions are interchangeable in terms of the ability to approximate them.

For the statement below, we use the better bounds achieved by Levi-Ron [**?**] and we state it here only for the restricted case of planar graphs.

▶ **Lemma 2.5.** *[**?**] [Onak's local-partition oracle]*
Let $\epsilon > 0$, and $d \geq 2$. Then there is an $s = s_{??}(\epsilon, d) = O(d^2/\epsilon^2)$ and a randomized algorithm (a.k.a "local partition oracle"), $\mathcal{A}$, such that for every d-bounded degree planar graph $G = (V, E)$, algorithm $\mathcal{A}$ produces an implicit partition P so that $G[P]$ is a collection of (d, s)-components.

Algorithm $\mathcal{A}$ provides a "neighbourhood oracle" to $G[P]$, namely, for a query to a vertex $v \in V(G)$, the algorithm returns the name of a component of $G[P]$ in which v lies in, by doing $(d/\epsilon)^{O(log(1/\epsilon))}$ queries to the graph G, and more specifically to vertices in $B_G(v, poly(1/\epsilon))$.

The total time complexity of a sequence of q queries to the oracle is $q \log q \cdot (d/\epsilon)^{O(log(1/\epsilon))}$ and with success probability $9/10$, the answers are all consistent with a partition P such that $G[P]$ is ϵ-close to G.

Using the local partition oracle, Newman-Sohler [**?**] proved that the normalized P-frequency vector of $G[P]$, for a (d, s) partition P of a d-bounded degree hyperfinite graph G can be estimated with an additive error of ϵ in its l_1-norm, simply by querying the $D = D(s)$-neighbourhoods (D-discs) around some constant number of randomly chosen vertices in G.

▶ **Definition 2.6.** Let $f, g \in \mathbb{R}^n$ be two vectors. We say that g λ-approximates f if $|f - g|_1 = \sum_1^n |g_i - f_i| \leq \lambda$.

The following lemma is a restatement of Lemma 5.2 of [**?**] for the restricted case of planar graphs (originally stated in [**?**] for hyperfinite graphs). Here we do not specify the function types explicitly, so to make the lemma more readable.

▶ **Lemma 2.7.** *[**?**] Let $G = (V, E)$ be a d-bounded degree planar graph, and $\epsilon \in (0, 1)$ any constant. Let $s = 100d^2/\epsilon^2$. Then there are values $D_{??} = D_{??}(\epsilon, d)$, $q_{??} = q_{??}(\epsilon, d)$ and a randomized algorithm* Sampler*, that accesses the graph G by querying independently $q_{??}$ random vertices of G and exploring the $D_{??}$-discs around them. The algorithm outputs a frequency vector $\tilde{f}$ with the following properties.*

With probability at least $4/5$ (over the internal coins of the algorithm) the following two events occur simultaneously: (a) the output vector $\tilde{f}$ ϵ-approximates the normalized (d, s)-frequency vector $freq(G[P])$ of the graph $G[P]$, where P is a partition of G into (d, s)-components. (b) $G[P]$ is ϵ-close to G.

Finally, to close the cycle, the following simple claim shows why an approximation of $freq(G[P])$ as above is useful.

▶ Claim 2.1. [**?**] Let $s \geq 1$ be an integer and let $0 < \lambda < 1$. Let G and H be two graphs that are each a union of (d, s)-graphs on n vertices such that their normalized frequency vectors (for the corresponding partitions into components) f, g respectively have $|f - g|_1 \leq \lambda$. Then G and H are λ-close.

For non-bounded degree outerplanar graphs it is not always possible to delete ϵn edges so that in the resulting graph all components are of constant sizes. E.g., consider the star of n vertices. Hence, we allow some more complex pieces in the partitions. This motivates the following definition that is introduced in [**?**] for partitions of forests.

▶ **Definition 2.8** ((d, s)-union)**.** A graph $G = (V, E)$ is a (d, s)-rooted graph if G contains a (unique) vertex v with $deg(v) \geq d + 1$ and each connected component of $G \setminus \{v\}$ is (d, s)

graph. The unique vertex v (with $deg_G(v) \geq d+1$) is called the *root* vertex of G and is denoted by $root(G)$.

A graph is a (d,s)-union if it is a vertex disjoint union of (d,s)-rooted components and (d,s)-components.

▶ **Definition 2.9** (Multiway-Cut)**.** For a graph G and a set of vertices $T \subseteq V(G)$ a T-multiway cut is a set of edges $E' \subseteq E(G)$ such that in the graph $G \setminus E'$ no two vertices from T are in the same connected component.

3 Global Partitions

In this section we prove the main structural theorem stating that every k-edge outerplanar graph is close to a k-edge outerplanar (d,s)-union, for some constants d, s (which depend only on ϵ and k). For clarity we present in this version the statements, and results for outerplanar graphs (rather than k-edge outerplanar graphs). The generalization to k-edge-outerplanar graphs is immediate, but will not be presented here. Note, however that the constants d and s will depend also on k when the generalization is done.

▶ **Theorem 3.1.** *Every outerplanar graph G is ϵ-close to a graph G' that is an outerplanar (d,s)-union for some $d = d(\epsilon)$ and $s = s(\epsilon)$.*

We note that this does not immediately imply that every such G has a 'short' (constant size) description, as each component of G' may have a root of different and unbounded degree. It does not imply also, that such a "close" graph G' can be "learned" from the local views in G. Thus, this is not directly applicable for property testing, but could be of independent interest. We will prove the theorem, and provide positive answer for the two additional properties above, namely that G' can be learned from the local views, and that it has a "short" description.

Before we present the proof of Theorem **??** we make some observations about outerplanar graphs which provides the core tool for the proof as well as the motivation for the definition of k-edge outerplanar graphs.

For a graph $G = (V,E)$ and $a, b \in V$ let $c(a,b)$ denote the minimum edge cut in G, separating a and b. The following basic Lemma **??** is used, via a chain of reductions, to prove Corollary **??** (See appendix for further details and proofs).

▶ **Lemma 3.2.** *Let $G(V,E)$ be 2-connected outerplanar graph, $s,t \in V$ such that (s,t) is an edge of the outer face in the embedding of G as an outerplanar graph. Then $c(s,t) \leq \lfloor(\log(|V|+1))\rfloor$.*

▶ **Corollary 3.3.** *Let $G(V,E)$ be a connected outerplanar graph and $U, W \subsetneq V$ be disjoint subsets of vertices. Suppose that U is an independent set in G Then, there is a U-multiway cut of size at most $2(|U|-1)\log(2|W|+1)$ in the graph $G[W \cup U]$.*

▶ Claim 3.1. Let G be a bipartite outerplanar graph with bipartition A, B. If degree of each vertex in B is at least two, then $|B| \leq 4|A|$ and hence, G has at most $15|A|$ edges.

We note that reducing the constants 4 and 15 in the above claim can be reduced by a factor of two, but this is of little interest in our context. We prefer the current proof of Claim **??** in the appendix, because its generalization to handle k-edge-outerplanar graphs is easy.

Now we are ready to prove Theorem **??**. The proof will be algorithmic, namely, *Algorithm 1* below will produce the required G' that is close to G.

Algorithm 1 Given d, ϵ, and an outerplanar graph $G(V, E)$ this algorithm returns an outerplanar (d, s)-union graph $G'(V, E')$, such that G' is obtained from G by removing $f(s, d, \epsilon) \cdot n$ edges, where $s = s_{??}(\epsilon/4, d)$.

1: **procedure** *GlobalPartition*(G)
2: Let $V^h = \{v \in V \mid deg_G(v) > d\}$, and $V^l = V \setminus V^h$.
3: Let $E_1 = E(G[V^h])$ be the set of edges with both endpoints in $G[V^h]$, and let G_1 be the graph obtained from G by deleting E_1.
4: Find a $(\epsilon/4, s)$-partition of $G[V^l]$. Such a partition exists, and, in particular, as asserted in Lemma **??**, a (local) oracle to such partition can be found. Let G_2 be the graph resulting from $G[V^l]$ after partitioning. G_2 is a disjoint union of (d, s)-components.
5: Replace $G[V^l]$ by G_2 in G_1, that is: let $G_3 = (V, E_2 \cup F)$, where $E_2 = E(G_2)$ and $F = E(G) \cap (V^h \times V^l)$. Namely F contains all edges of G with exactly one endpoint in V^h and one endpoint in V^l.
6: Finally, obtain G' from G_3 by removing for each component C of G_2 a minimum size V^h-multiway cut in the graph $G_3[C \cup (V^h \cap N(C))]$.
7: **end procedure**

▶ **Theorem 3.4.** *Let $\epsilon \in (0, 1)$ be any constant. Let $G = (V, E)$ be an outerplanar graph, $d = d(\epsilon) = O(\frac{1}{\epsilon^2})$, $s = s_{??}(\epsilon/4, d) = O(d^2/\epsilon^2)$. Then Algorithm 1 produces a (d, s)-union subgraph G' of G, that is ϵ-close to G with probability better than* 0.9.

Proof. Since G is planar, it follows that $|E(G)| \leq 3n$. This implies that $|V^h| \leq 6n/d$. Since the graph is planar then $G[V^h]$ is planar too. Hence, at most $3|V^h| \leq 18n/d$ edges are removed in step 3 of the algorithm. We fix $d = O(\frac{1}{\epsilon^2})$ to be sufficiently high, so to make sure that at most $\epsilon^2 n/10$ edges are removed in step 3.

Applying the global partition in step 4 with parameters $\epsilon_1 = \frac{\epsilon}{4}$ and d we obtain, with success probability 0.9, a graph G_2 that is a a union of (d, s)-components, and that is ϵ_1-close to $G[V^l]$. This defines the graph G_3 in step 5 of the algorithm.

By Claim **??**, the number of connected components in G_2 with at least two neighbours in V^h in the graph G_3 is at most $4|V^h|$. If each of these components is contracted to a single representative vertex for the component, after removal of parallel edges and self loops, there are only $15|V^h|$ edges between the representative vertices and V^h.

For step 6, observe that if for each component C of $G_3[V^l]$ we get a $N(C) \cap V^h$-multiway cut M_C in $G_3[V(C) \cup (N(C) \cap V^h)]$, then $M = \cup_C M_C$ will be a V^h-multiway cut in G_3. Moreover, we can restrict our attention to only components which have at least two neighbours in V^h. As explained in the paragraph above, the number of such components is only $4|V^h|$ and $\sum_C |N(C) \cap V^h|$ is at most $15|V^h|$. For each such component C, we have $|M_C| \leq |N(C) \cap V^h| \cdot (2\log(2s + 1))$ by Corollary **??**. From this, it follows that $|M| \leq 15|V^h|(2\log(2s + 1))$. Since $s = O(d^2/\epsilon^2)$, a proper choice of d ensures that $|M| \leq \epsilon n/3$.

Thus, the total number of edges removed in all steps of the algorithm is at most ϵn, implying that the resultant graph G' is ϵ-close to G.

After applying the partitioning oracle in step 4 the size of every connected component in G_2 is at most s. Since V^h becomes an independent set after step 3, after executing step 6, no two vertices in V^h have a path between them in G'. Therefore, each component of G' has at most one vertex of degree greater than d. Therefore G' is a (d, s)-union for d and s as above. ◀

We note that in the proof above, step 4 of the algorithm, which is the only random part, may be replaced with any deterministic partition (e.g., recursively removing edges connected to a good enough separator). We used random local partition, in the spirit of Onak [**?**], looking ahead, to hint to the fact that the partition can actually be done in a *distributed* manner, and hence "approximated" locally. The same is also true with respect to step 6, where a global multiway cut could be taken. It is possible to do step 6 in a distributed way and locally, because a component C of G_2 has at most $d \cdot s$ (which is a constant) neighbours in V_h, and hence $G_3[C \cup (V^h \cap N(C))]$ is a graph of constant size.

4 From global partition to Local partition

Let $G = (V, E)$ be an outerplanar graph. Recall that our goal is two fold: the first is to roughly "learn" G from its *local views*. Learning here means to find a graph G' that is a (d, s)-union and that is close to G, as asserted by Theorem **??**. Conceptually this implies that two graphs with the same local views are close to be isomorphic (some extra care should be taken here). The 2nd goal is to find the above approximating G' using a small number of queries. Conceptually this immediately implies a property testing mechanism for all properties.

This is summed up in the following theorems.

▶ **Theorem 4.1.** *For every $\epsilon > 0$ there is a $D = D_{??}(\epsilon)$, $s = s_{??}(\epsilon), d = d_{??}(\epsilon)$, $q = q_{??}(\epsilon, n) = O(poly(\log n))$, and a randomized algorithm* Approx *that on an outerplanar graph $G = (V, E)$ on n vertices:*

- Approx *outputs an outerplanar (d, s)-union graph G^*.*
- Approx *does random queries to q vertices in G, and only inside the D-disc around the above vertices.*
- *With success probability at least 0.9, G^* will be ϵ-close to G.*

▶ **Theorem 4.2.** *For every $\epsilon > 0$ there is a $D_{??} = D(\epsilon)$, $q = q_{??}(\epsilon, n) = O(poly(\log n))$, and a randomized algorithm $Tester$, that on two outerplanar graphs G, H on n vertices, it accepts if H is isomorphic to G and rejects if H is ϵ far from G with error probability at most $1/3$.*

The algorithm $Tester$ does q random queries to q vertices in G and H, only inside the D-disc around (some of) the above vertices.

▶ **Theorem 4.3.** *For every $\epsilon > 0$ and a graph property Π, of graphs on n vertices, there is a $D = D_{??}(\epsilon)$, $q = q_{??}(\epsilon, n) = O(poly(\log n))$, and a randomized algorithm T_Π, that accepts every outerplanar graph G having the property, and rejects every outerplanar graph G that is ϵ-far from Π, with error probability $1/3$.*

The algorithm T_Π does q random queries to q vertices in G, and only inside the D-disc around (some of) the above vertices. Moreover, the queries to G are oblivious of Π: only the final decision once the q queries are done, is dependent on Π.

▶ **Theorem 4.4.** *For every $\epsilon > 0$ there is a $D = D_{??}(\epsilon)$ such that if two outerplanar graphs G, H on n vertices, have identical D-views then H is ϵ-close to G.*

We note that analogue theorems for planar d-bounded-degree graphs are given in [**?**]. However, unlike the case for d-bounded-degree planar graphs, that have constant size approximations in form of a union of (d, s)-components, a (d, s)-union graph does not necessarily has a short description. This is due to the fact that the degree of the root of every component may be arbitrary number in $[n-1]$, and hence there are non-constant many types of possible components (let alone their number). To overcome this difficulty we define

an ϵ-net for (d,s)-union graphs, namely, a set $\mathcal{G}(d,s)$ of (d,s)-union graphs (of relatively short description), and show that for every G' as above, there is a graph $G'' \in \mathcal{G}(d,s)$ that is close to G'.

Further we will show that G'' can be obtained from the original G by sampling. As will turn out, this sampling can be restricted to randomly sampling a relatively small number of vertices ($poly(\log n)$), in some constant-diameter discs in G. Hence this will provide the "locality" that is stated as desirable above. A similar method in nature, was used by Kusumoto and Yoshida, [**?**], for unbounded degree forests.

We need the following definitions.

▶ **Definition 4.5.** [γ-layered (d,s) union graphs] Let $\gamma > 1$ be a constant. A γ-layered (d,s) union graph is a (d,s) union graph in which all high-degree vertices have degrees that are γ-powers, namely, in the set $\{\gamma^i\}_{i=\alpha}^{L}$ where $L = \lfloor \log_\gamma n - 1 \rfloor$ and $\alpha = \min\{i | \gamma^i \geq d+1\}$. We denote by G^0 the components of a (d,s) union G that are (d,s) graphs.

In the above definition, all γ^i are assumed to be integral. This is achieved by rounding if necessary. We do not explicitly write this rounding to increase readability. The extra rounding will not affect any of our results.

The role of γ-layered graphs is obvious from the following claim.

▶ Claim 4.1. For every $\epsilon > 0$ there is a $\gamma = \gamma_{??}(\epsilon, d) \in (1,2)$ such that every (d,s)-union graph G is ϵ-close to a γ-layered (d,s)-union graph.

Proof. Let $\gamma \in (1,2)$ to be defined later, and let $G^0, , ..., G^L$ be a partition of G, where $G^i,\ i = \alpha, \ldots, L$ contains all (d,s)-rooted components C with $deg(root(C)) \in (\gamma^{i-1}, \gamma^i]$ and G^0 contains the (d,s) components. Let n_i be the number of components in G^i.

Consider each G^i separately. For $i \geq \alpha$, and for each component C in G^i we add at most $\gamma^i - \gamma^{i-1}$ isolated vertices and link them to $root(C)$. This obviously makes the graph γ-layered. Thus for G^i we added at most $n_i \cdot (\gamma^i - \gamma^{i-1}) = n_i(\gamma - 1)\gamma^{i-1}$ edges. Note, however, that $n_i \cdot \gamma^{i-1} \leq |V(G^i)|$, as every component in G^i contains a vertex with degree larger than γ^{i-1}.

For G^0 we do not need to change anything. This results in a total number of edge changes bounded by $(\gamma - 1) \cdot \sum_{i \geq \alpha} |V(G^i)| \leq (\gamma - 1)n$. Hence setting $\gamma \leq (1 + \epsilon)$ implies the claim. ◀

Now, to define a short description (a.k.a. "sketch") for a γ-layered (d,s) union graph, all we need is to define the structure of $G^i,\ i = \alpha, \ldots, L$, and that of G^0. For the latter, a good sketch is the (d,s)-frequency vector of G^0, (or a good approximation of it), as being done in [**?**]. This will also become clear as a special case in what follows. For $G^i, i \geq \alpha$, we only need to define the structure of C for each component $C \in G^i$.

Note that $C \setminus root(C)$ is a union of (d,s) components, each with some marked subset of vertices, indicating the neighbour set of $root(C)$. Since there are constantly many possible (d,s) graphs, there are also constantly many (d,s)-graphs with marked vertices. Hence, each component C of G^i is defined by the (d,s) frequency vectors of the marked components, $\{Freq(C \setminus root(C))\}_{C \in G^i}$. Still, computing for each $C \in G^i$ its frequency vector would be too demanding. Instead we will approximate this vector, using the easy Claim **??**. Doing this will bring us two advantages; the first is that we will still get a component C' which is close enough to C, but which we will be able to afford (in terms of number of queries). The second and more important feature is the reduction in the number of types of components to a constant, thereby making it possible to approximate G^i by estimating the number of components of each of these constantly many types.

This is summed up in what follows:

Recall that for fixed d and s we set $f(d,s) = |\mathcal{F}(d,s)|$ (which is a constant), where $\mathcal{F}(d,s)$ is the set of all possible outerplanar(d,s)-graphs. We now add a boolean marking of vertices in each (d,s)-graph. This boolean marking will be used later to indicate which vertices in the component are connected to its root in a rooted component (if at all). Hence the histogram, and the frequency vector, is of dimension $2^s \cdot f(d,s)$, since corresponding to each graph in $\mathcal{F}(d,s)$, we have also to specify which subset of vertices in it are marked (have 1-marking).

For fixed $\gamma > 1$ and d, s, let $G = G^0 \cup (\cup_{i=\alpha}^{L} G^i)$ be a γ-layered (d,s) union graph, and fix an $i \in \{\alpha, \ldots, L\}$. As explained above, each component $C \in G^i$ is completely defined by its (d,s) frequency vector $Freq(C) \in [n]^{2^s f(d,s)}$, where the marked vertices in each (d,s)-component of $C \setminus root(C)$ are the vertices that are connected to $root(C)$. Let $freq(C) = Freq(C)/(\text{sum of coordinates of } Freq(C))$. Note that $||freq(C)||_1 = 1$.

Let $0 < \delta < 1$ be small enough constant (to be defined later), and $N(\delta)$ be a δ-net for the ℓ_1-unit ball of dimension $2^s f(d,s)$. Obviously such an $N(\delta)$ whose size is a constant that depends only on δ, d, s exists. For example, take $N(\delta) = \{\delta\overline{x} \mid \overline{x} \text{ is a } (2^s f(d,s))\text{-dim vector of integral coordinates whose absolute values sum up to } 1/\delta\}$.

For $Freq(C)$ as above, we define its *δ-normalized approximation* as a closest vector in $N(\delta)$ to $freq(C)$ (in case of tie choose an arbitrary closest vector). Thus, we have a mapping that maps each component C of G^i to a constant size alphabet (of size $|N(\delta)|$), and hence G^i is mapped into a vector $LFreq(G^i) \in [n]^{|N(\delta)|}$, where the jth coordinate is the number of components C in G^i that have type$=j$ as their δ-normalized approximation. Again, we normalize as follows: Let n_i be the number of components in G^i, we let $lfreq(G^i) = \frac{1}{n_i} \cdot LFreq(G^i)$.

▶ Claim 4.2. Let G^i be the ith layer, $i > 0$, of a γ-layered (d,s)-union graph as above. Let $\epsilon > 0$. Then there is a constant $\nu = \nu_{??} = \nu_{??}(d,s,\delta,\epsilon) \in (0,1)$ such that if $|\tilde{n_i} - n_i|\gamma^i \leq \nu \cdot \max\{n_i\gamma^i, n/L\}$, and $|\tilde{f} - lfreq(G^i)|_1 \leq \nu$, the graph $\tilde{G^i}$ that is defined as stated below has $dist(G^i, \tilde{G^i}) \leq \epsilon \cdot \max\{n_i\gamma^i, n/L\}$.

Here $\tilde{G^i}$ is the following graph: let $F = \tilde{n_i} \cdot \tilde{f} = (\tilde{m}_1, \ldots, \tilde{m}_{|N(\delta)|})$ and for $j = 1 \ldots, |N(\delta)|$, let C_j be a rooted component whose frequency vector is the j-type frequency vector. Then for $j = 1 \ldots, |N(\delta)|$ we include $\lceil \tilde{m}_j \rceil$ disjoint copies of C_j in $\tilde{G^i}$.

Note that the claim only asserts an *additive* error between G^i and $\tilde{G^i}$ that is not necessarily proportional to the size of G^i. However, since there are L "layers", the average G^i has n/L vertices. For G^i larger than the average, the above approximation is with a ν-multiplicative error. For G^i smaller than the average, the additive error is a fraction of the average, which we will be able to afford.

Proof. Let $G^i, \tilde{G^i}$ as above , and let $m_j = LFreq(G^i)_j = f_j \cdot n_i$ be the number of components in G^i of type j. Namely $lfreq(G^i) = (f_1, \ldots, f_{|N(\delta)|})$. Let $\Delta = \max\{n_i\gamma^i, n/L\}$. A close isomorphism between $G^i, \tilde{G^i}$ is clear: we map for each type j, the corresponding matching components, leaving $|\tilde{m}_j - m_j|$ components unmatched. For the unmatched components we remove all edges and make the corresponding nodes isolated points. Hence the contribution of type j to the distance (edge-count) is bounded by $|m_j - \tilde{m_j}| \cdot \gamma^i \cdot e(d,s)$ (disregarding here errors due to non-integrality), where $e(d,s)$ is the maximum number of edges in a (d,s) graph (which is constant).

Summing this over all $j \in [|N(\delta)|]$ one gets:

$$\begin{aligned} dist(\tilde{G}^i, G^i) &\leq \sum_{j=1}^{|N(\delta)|} |m_j - \tilde{m}_j| \cdot \gamma^i \cdot e(d,s) \leq e(d,s)\gamma^i \sum_j |\tilde{n}_i \tilde{f}_j - n_i f_j| \\ &\leq e(d,s)\gamma^i \sum_j |\tilde{n}_i \tilde{f}_j - \tilde{n}_i f_j + \tilde{n}_i f_j - n_i f_j| \\ &\leq e(d,s)\gamma^i \tilde{n}_i \cdot \nu + e(d,s)\gamma^i |\tilde{n}_i - n_i| \cdot |lfreq(G^i)|_1 \\ &\leq e(d,s) \cdot 2\nu\Delta + e(d,s)\nu\Delta \end{aligned}$$

Now if we set $\nu \leq \frac{\epsilon}{3e(d,s)}$ we get the asserted claim. ◀

We now restate Theorem **??** in a more detailed version, and present its proof.

▶ **Theorem 4.6.** *For every $\epsilon > 0$ there is a $D_{??} = D(\epsilon)$, $s = s_{??}(\epsilon), d = d_{??}(\epsilon)$, $\gamma_{??}(\epsilon)$, $q = q_{??}(\epsilon, n) = O(poly(\log n))$, there is a randomized algorithm* Approx*, that on an outerplanar graph $G = (V, E)$ on n vertices, outputs an outerplanar γ-layered graph G^*.*

The algorithm Approx *does q random queries to q vertices in G, and only inside the D-disc around (some of) the above vertices.*

It holds that with success probability at least 0.9, G^* will be ϵ-close to G.

Proof of Theorem ??. Sketch

We start by defining G' as the (d,s)-union graph obtained by *Algorithm 1*, for $\epsilon' = \epsilon/10$. We do not know G', but we know how it would have been formed by *Algorithm 1*. We also know that with high probability it would be $\epsilon/10$-close to G. By Claim **??**, this implicitly defines a γ-layered graph G^* that is close to G', for a suitably small γ. Let $G^* = G^0 \cup \cup_{i=\alpha}^{L} G^i$, and for $i = 0, \alpha, \ldots L$, let n_i be the number of components of G^i, and $f_i = lfreq(G^i)$. Let $roots(G')$ be the set of high-degree vertices in G', namely the roots of the (d,s)-components of G'.

The main part of the algorithm, is algorithm Sampler that is described in the appendix. Algorithm Sampler aims at choosing a vertex y that is distributed uniformly at random among the $roots(G')$ that are in any given layer of G^*. For such y it will also approximate its degree in G' accurately enough, and while doing this it will also approximate $freq(y)$, the approximated frequency vector of y (although this is defined w.r.t G^* rather than G').

Once this is done, approximation $n_i, lfreq(G^i)$ as required by Claim **??**, for every $i \leq L$ is straight forwards: we just sample q random y's as above, for q large enough ($q = poly(\log n)$) and for each obtain its $freq$ and degree. Then, by normalizing, the proportion of such vertices that are in any interval $[\gamma^{i-1}, \gamma^i)$ is a good approximation of n_i, while the proportion of each type of $freq(y)$ gives an approximation of $lfreq(G^i)$. Finally, having these estimates, Claim **??** ends the proof.

The idea behind the Sampler is also simple. We choose a high-degree vertex y at random from V^h by sampling uniformly an edge (v, y) where $v \in V^l$ and $y \in V^h$. Once we have such y, we sample a random neighbour v of it of small degree, discover the component of v, in $G'[V^l]$ by running the local partition oracle for d-bounded graphs, and deleting the multiway cut. As a result, a random (d,s) component connected to y in G' is found (or a conclusion that v is not in the (d,s)-component connected to y). Having found such a random component, we repeat the process for q independent times, which allows us to estimate (again by Chernoff), the degree of y in G', and its frequency $freq(y)$.

Some extra care should be taken since the Sampler cannot succeed for every y that is a root of G'. Consider a vertex y for which $deg_G(y) >> deg_{G'}(y)$. For such y, for most

neighbours v of y, their components in the (d, s) partition of $G[V^l]$ (possibly after deleting the edges in the relevant multiway cut) will not be connected to y, and $\deg_{G'}(y)$ might not be estimated correctly. Such vertices we call "bad". In the proof of correctness of the algorithm Approx that outputs G^* we will show that while bad vertices contribute some additional increase in the distance between the estimated G^* and G, this increase in distance is small enough, so that the produced G^* will be (w.h.p) as needed.

We end this very high level description of the sampling process by two notes. The first is that we need to approximate every (large) G^i. Namely, we need to decrease the failure probability in each large G^i to $O(1/\log n)$.

The second remark is that, a similar estimation in spirit (although starting from a forest rather than the more general outerplanar graph), is done in [**?**], but using a different and finer metric.

For further details, see the algorithm Approx in the appendix. ◀

Remark: It is a suitable point here to note the difference of the results in this paper up to this point, and the results for d-bounded hyperfinite (or planar) graphs of [**?**, **?**]. In the cited papers, a local oracle (in the sense of Onak, as described above) is obtained for the (d, s)-graph H that is close to G. This local oracle is used to approximate the frequency vector of the components in a straight forwards way, by sampling. In our case, a local oracle to the (d, s) union graph (γ-layered) graph G^* is not obtained; instead it is only "nearly" obtained. It fails to produce a local oracle exactly for the bad y's as explained in the proof. Namely, let u be a high-degree vertex in G^* (and hence in G too). It could be that many edges adjacent to u in G are absent from G^*. Hence when asking for a random neighbour of u, the sampler above may not succeed in finding one.

We present now the proofs of Theorems **??** and **??**. These proofs follow from Theorem **??** exactly as in [**?**]. Before getting into the proof, it should be noted that in the standard model, we are concerned only about the number of queries to the input graph and not about the running time of the algorithm. The number of vertices in the input graphs is also an information available.

Proof of Theorem ??. : Let Π be any graph property, and let Π_n be its restriction to graph on n vertices. An ϵ-tester T_Π for Π_n for outerplanar graphs on n vertices is the following: We first run the randomized algorithm Approx that is guaranteed in Theorem **??** with parameter $\epsilon/2$, to produce a graph G^* that is a (d, s) union and is $\epsilon/2$-close to G with high probability. Having a full knowledge of G^*, without further queries to G, Algorithm T_Π checks if G^* is $\epsilon/2$-close to Π_n. It accepts if the answer is yes, and reject otherwise.

Note that T_π is oblivious of Π when performing the queries to G. Once the queries are made to G and G^* is obtained, a test for any property can be run (in parallel, say).

To analyse the error probability, assume that G^* is indeed $\epsilon/2$-close to G, as asserted by Theorem **??**. This happens with probability at least 0.9. Now if G has Π, then T_Π would accept, because $G \in \Pi_n$, and G^* is $\epsilon/2$-close G, which makes it $\epsilon/2$-close to Π_n. On the other hand, if T_Π accepts on account of finding an $H \in \Pi_n$, and such that G^* is $\epsilon/2$-close to H, then by triangle inequality G is ϵ-close to Π_n. Thus the error probability is bounded by 0.1. ◀

Proof of Theorem ??. : Let G, H be two outerplanar graphs on which we want to ϵ-test isomorphism. The ϵ-test will be as follows: It will first run the randomized algorithm Approx, as guaranteed by Theorem **??**, to produce a G^* that is (d, s)-union, with distance parameter $\epsilon/2$.

It will then consider the graph property of n-vertex graphs $\Pi(G^*)$ to be the following property: *the input graph is* $\epsilon/2$*-close to* G^*. By Theorem **??**, there is an $\epsilon/2$-tester T' for $\Pi(G^*)$. We just run T' on H, accept if it does and reject if it rejects. Note that the query complexity is just doubled. Note also that G^* and hence T' are not known in advance, but this does not matter, as we do not need to worry about the time complexity.

To analyse the success probability, assume that G^* is $\epsilon/2$-close to G, which is asserted to happen with probability 0.9. Now, assume that H is isomorphic to G, than G^* is also $\epsilon/2$-close to H, and hence H has property $\Pi(G^*)$. Thus, test T' will indeed accept H with probability at least 0.9. On the other hand, assume that T' accepts H, then with probability 0.9, H is $\epsilon/2$ close to $\Pi(G^*)$. Then, by the triangle inequality it is ϵ-close to G as required. The total error is hence bounded by the events, that either G^* is not $\epsilon/2$-close to G or that T' errs. As both are bounded by 0.1, the total error probability is at most 0.2. ◀

Proof of Theorem ??. - Sketch.

The proof in this case is somewhat more involved than the previous theorems. The basic idea, as in [**?**] is that if G, H have the same local view, then applying on both the sampler of Theorem **??**, one will get identical (or close enough), approximations G^*, H^* respectively, as the approximation is done based on the information in the local views, which is identical for both graphs. However, there was a difficulty in this argument, even in [**?**] for bounded-degree hyperfinite graphs. To understand what the difficulty is, let us start to formalize the proof.

Let D be as in Theorem **??**, and $R = f(D)$ be a constant depending on D, to be defined later. Let $q = poly(\log n)$ be the number of queries asserted by the algorithm Sampler that is used in the proof of Theorem **??**, for some fixed ϵ.

Let $V = V(G)$, and $V' = V(H)$. Assume that the R-views of G is identical to the R-views of H. Hence, we can fix a $1-1$ map $\phi : V \mapsto V'$ so that every v is mapped to $v' = \phi(v)$ such that the R-disc of v is identical to the R-disc of v'. Our aim is to simulate the process of the sampler on H, by observing its run in G: that is, if the sampler on G is making some q queries to discs around vertices $v_1, \ldots, v_q$, we will aim to use queries on identical discs of their images $v'_1, \ldots, v'_q$ in H, with the hope that the output graph G^* produced by the sampler on G will be identical to the output graph H^* obtained from H.

The first problem is that the sampler on G might be successful in G, in respect of obtaining a good approximation G^*, using the queries $v_1, \ldots, v_q$, while the output graph H^* obtained from H on the corresponding sequence $v'_1, \ldots, v'_q$ might not be a good approximation of H. However, as both process are assured to be successful with high probability, for most sequences $v_1, \ldots v_q$, the processes on G and the corresponding one on H are both going to be successful: on G with queries to discs of $v_1, \ldots v_q$, and on H with queries to discs of $v'_1 = \phi(v_1), \ldots, v'_q = \phi(v_q)$. It is not clear however, that they will produce the same approximation although they seemingly see the same view, due to the following reason.

The sampler needs to makes some q i.i.d queries, to components in G^i, in order to approximate $lfreq(G^i)$. Concentrate first on G^0 (which is an identical case to that considered in [**?**]). The sampler makes queries to a sequence $v_1, \ldots v_q$ (on which as explained above, we may assume it will succeed), and explore the D-disc around each v_i on which it can run the local partition oracle on the graph restricted to low-degree vertices, in order to define the component of each v_i in G^0. Now suppose that v_i is now being queried and that a vertex u in the D-disc(v_i) is also present in the D-disc(v_j), for some previously queried vertex v_j. Since the partition must be consistent, the neighbourhood around u that is discovered when v_j was queried, is the same when viewed exploring the disc around v_i. However, from the view point in H, while v'_j, v'_i have isomorphic discs of the appropriate size as v_j, v_i respectively,

they do not have to share a vertex $u' = \phi(u)$. Namely, the $u \in D\text{-}disc(v_j)$ is mapped to u' that is not necessarily in $D\text{-}disc(v_i)$.

The argument in [**?**] addressed this issue is the following way: since the degree is bounded by some constant d, a situation as above (that for two random D-discs there is a non-empty intersection) has very low probability, and hence will not occur on most random sequences.

Here this is not correct any more, as the degree of the roots may be as high as $\Omega(n)$ in higher layers. Then it could happen that any two discs around such high-degree vertices do intersect. To get rid of this problem assume first that there are no edges with two endpoints that are high-degree, both in G and H. This may be assumed, as for the map ϕ above, vertices are mapped to vertices with isomorphic discs even after deleting edges between two high degree vertices. Further assume that the set of edges $M = \{(u,y)|\ u \in V^l,\ y \in V^h\}$ is of size $|M| \geq \frac{\epsilon n}{q^2 \log n}$ (otherwise, there is no problem as no high-degree vertex is likely to be seen at all as a root (in the first item in phase 3 of the sampler - In addition, in this case the graph is close to a d-bounded degree, and hence the argument above for d bounded degree graphs implies that with high probability the sampler will produce an identical approximation for both G and H, when simulated as explained above.

Recall that our sampler makes a total of q queries (which is $poly(\log n)$ in our case). At the top level, it makes some queries to random edges in M (in phase 3 first item, sub-item (a)), in order to make independent queries to at most q randomly chosen root vertices in G^i for every level i, and explore the component in G^i under such roots, by random sampling.

Consider the bad case when a low-degree vertex v might be found while randomly exploring components formed by two such high-degree roots y and y'. Assume that y' is chosen after y, and that the random neighbours of y that are queried are $u_1, \ldots u_r$, where $r \leq q$. Then while forming the (d,s)-components in $G[V^l]$ for $u_1, \ldots, u_r$, they together involves querying a total of at most q low-degree vertices (v being among them). Further, these vertices have a total of $r \leq q$ edges that are queried and whose end points are high-degree vertices other than y. Call these edges "bad" with respect to y.

Now, for v to be queried while exploring y', the same should happen with y', i.e., v should be among the total of at most q queried edges once the above exploration is done with y'. However, v will not be queried while exploring y', if no bad edge with respect to y is queried while exploring y', and if the D-discs in $G[V^l]$ around the at most q queries from y to low-degree vertices do not intersect the $D\text{-}disc(v)$ in $G[V^l]$. Hence, when y' is chosen by a random edge, if none of its at most q queried random neighbours of y' are bad edges w.r.t. y, then v will not be queried while exploring y', due to the first reason, and conditioned on that, the D-discs around these random vertices will also be disjoint from $D\text{-}disc(v)$ as $D\text{-}disc(v)$ contains only a very small (constant) number of vertices .

Therefore, for y' such that $deg(y') \geq q^5$, while exploring y', encountering a low-degree vertex u that was encountered earlier while exploring from another such y will happen with probability at most $1/q^3$.

For y' such that $deg(y') \leq q^5$, the layer containing y' has a high mass (more than $n/\log^2 n$) only if that layer contains at least $\frac{n}{sq^5 \log^2 n}$ such high-degree vertices, where $s = O(d^2/\epsilon^2)$ is the bound set by the partitioning algorithm on the component size. Now, for such y', if none of the low-degree neighbours u of y' has the edge (u,y') that is bad w.r.t y, again exploration from y' will not query a v that was queried while exploring y. As there are at most q bad edges w.r.t. y, at most q high-degree vertices y's will have one of these bad edges incident on them, and hence the probability of picking such a y' for exploration is at most $\frac{s \cdot q^6 \log^2 n}{n}$ which is extremely low.

Over all, combining the two cases, the probability that for (y,y') as above, a common v

will be queried is lower than $1/q^3$, and hence by the union bound on all possible q^2 pairs, with very high probability, for no pair (y, y') a common vertex is queried.

Assuming that indeed for no pair (y, y') of high-degree vertices, there is a common low-degree vertex that is queried, the local views that are sampled for each root in $G^i, i \geq \alpha$ are distinct.

Hence we can couple the two sampling processes, the one for G and that for H in a consistent way, so to have the same views, and therefore will produce G^* and H^* which are identical.

Thus, if we run the sampler of Theorem **??** with parameter $\epsilon/2$ on G and H, it is ensured that with high probability the runs are successful and it produces G^* and H^* respectively with $dist(G, G^*) \leq \epsilon n/2$, $dist(H, H^*) \leq \epsilon n/2$ and moreover $G^* = H^*$. Therefore, we have $dist(G, H) = \epsilon$, as desired.

Finally, let us consider what is the disc radius needed to ensure that the above will indeed occur. Note that for low-degree vertices, we only need D-discs around them to be able simulate the sampler behaviour from the local information. For high-degree vertices, some $r \leq q$ random queries to neighbours are being made in the disc around them. While $q = poly(\log n)$ and not constant, note that all the possibly q such queries, are done to neighbours (namely, vertices of distance 1), from such high-degree vertices. Further queries are done in $G[V^l]$ to simulate the local partition on low-degree vertices, with an occasional query that discovers a high-degree vertex, but, in which case, no further exploration is done from this high-degree vertex. Hence taking $2D$-discs is enough to simulate the sampler behaviour.

We avoid further details in this version. ◀

5 Discussion

Our results are another step towards understanding the theory of property testing in the sparse graph model, and mainly for restricted subfamilies of planar graphs. Yet the main questions in this area are still open:

- Which graph properties are testable with sub-linear query complexity?
- Is it true that if two n vertices planar graphs H, G have their D-local views identical (for some large enough constant D), then the graphs are ϵ-close to be isomorphic? Is this true for bounded tree-width graphs ?

We note that the above questions are open, even for the class of 2-outerplanar graphs.

Appendix -A

Missing Proofs of Section ??

Proof of Lemma ??. We will prove the lemma by induction on $|V| = n$. Assume that $s, t \in V$ and $|V| \geq 4$. Since G is 2-connected and $(s,t) \in E$, then all the vertices are on a simple path between s and t. Enumerate the vertices along this path $v_1 = s, v_2, \ldots v_n = t$. Let $i < n$ be the largest such that $(s, v_i) \in E$, and let $j > 1$ be the smallest such that $(v_j, t) \in E$. Since G is outerplanar it follows that $i \leq j$, therefore either $i \leq \lceil \frac{n}{2} \rceil$ or $\lceil \frac{n}{2} \rceil \leq j \leq |V|$. Assume w.l.o.g that $i \leq \lceil n/2 \rceil$ and let $V_1 = \{s, v_2, ..., v_i\}$. Note that $G[V_1]$ is outerplanar, with (s, v_i) an edge on the outer face. If $i = 2$, then $C_1 = \{(s, v_i)\}$ will separate s from v_i in $G[V_1]$. If $i > 2$, $G[V_1]$ is 2-connected and by induction hypothesis, there is an edge-cut C_1 separating between s and v_i in $G[V_1]$, with $|C_1| \leq \lfloor \log(\lceil n/2 \rceil + 1) \rfloor$. It is easy to see that $C_1 \cup (s,t)$ is a $\{s,t\}$-multiway cut in G, of size as claimed. ∎

▶ **Lemma 5.1.** *Let G be 2-connected outerplanar graph. For any pair of vertices $s, t \in V(G)$, $c(s,t) \leq 2(\log(|V(G)| + 1))$.*

Proof of Lemma ??. Let G be 2-connected outerplanar graph, and $s, t \in V(G)$. Since G is outerplanar, then all vertices of G are on the unique Hamiltonian cycle C of G. We may assume that s, t are not neighbours on C, as otherwise, Lemma **??** immediately implies the result. Hence, C defines two vertex disjoint paths, from s to t: $P_1 = (s, v_1, \ldots, v_k = t)$, and $P_2 = (s, u_1, \ldots, u_\ell = t)$. Let $i \leq k$ be the largest such that $(s, v_i) \in E$ and $j < \ell$ the largest such that $(s, u_j) \in E$. Then $G_1 = G[\{s, v_1, \ldots, v_i\}]$ is outerplanar with (s, v_i) on its outer face. If $i = 1$, then $C_1 = \{(s, v_i)\}$ will separate s from v_i in G_1. Otherwise, G_1 is 2-connected and by Lemma **??**, there exist an edge cut C_1 in G_1 separating s and v_i with $|C_1| \leq \log(|V(G)| + 1)$. Similarly, $G_2 = G[\{s, u_1, \ldots, u_j\}]$ is outerplanar with (s, u_j) on its outer face and has an edge cut C_2 separating s and u_j with $|C_2| \leq \log(|V(G)| + 1)$. It is easy to see that $C_1 \cup (s,t)$ is an edge-cut in G separating s and t, of size as claimed. ∎

▶ **Lemma 5.2.** *Let $G(V, E)$ be a connected outerplanar graph and $U, W \subsetneq V$ be disjoint subsets of vertices. Suppose that $|U| \geq 2$ and U is an independent set in G. Then there exists an edge cut of size $2\log(2|W| + 1)$ in $G[W \cup U]$ separating some two points in U.*

Proof of Lemma ??. Let $G(V, E)$ be a connected outerplanar graph and $U, W \subsetneq V$ be disjoint subsets of vertices. Suppose $G[W]$ is connected and U is an independent set in G such that every vertex in U is a neighbour of some vertex in W and $|U| > 1$.

First consider the case when $G[W \cup U]$ is 2-connected. Since U is an independent set, then in the Hamiltonian cycle that is the boundary of the outer face of $G[W \cup U]$, between every two vertices of U, there must be at least one vertex from W. Hence $|U| \leq |W|$, which implies that $|U \cup W| \leq 2|W|$. Take any two arbitrary vertices $u_1, u_2 \in U$. By Lemma **??**, there exists an edge-cut C of size at most $2(\log(|U| + |W| + 1)) \leq 2(\log(2|W| + 1))$ separating u_1 and u_2 in $G[W \cup U]$.

The same argument as above applies also when $G[W \cup U]$ is not 2-connected, and there is a block $\mathcal{B}$ of $G[W \cup U]$ that contains two vertices from U. Hence we may assume that every block of G contains at most one vertex form U.

Let $\mathcal{B}$ be a block of $G[W \cup U]$ containing a single vertex u from U. Let u' be another vertex in U, which by the reasoning above, is in another block $\mathcal{B}'$ of $G[W \cup U]$. Let x be the cut-vertex in $\mathcal{B}$ which is a separation point between u and u' in $G[W \cup U]$. In this case,

$|V(\mathcal{B})| \leq |W| + 1$ and by Lemma **??**, u can be separated from x in $\mathcal{B}$ by removing at most $2(\log(|W| + 2))$ edges. Such a cut also separates u from u'. ∎

Proof of Corollary ??. We will prove this by induction on $|U|$. Let $t = 2\log(2|W| + 1)$. The proof is trivial when $|U| = 1$. If $|U| > 1$, by Lemma **??** an edge cut C of size t exists in $G[W \cup U]$ that separates a subset $S \subseteq U$ from $U \setminus S$ where $1 \leq |S| < |U|$. Let us now consider $G_1 = G[W \cup S]$ and $G_2 = G[W \cup (U \setminus S)]$. By the induction hypothesis there is a S-multiway cut C_1 of size $(|S| - 1)t$ in the graph G_1 and there is a $(U \setminus S)$ multiway cut of size C_2 of size $(|(U \setminus S)| - 1)t$ in G_2. Taking $C \cup C_1 \cup C_2$ we obtain a U-multiway cut in G of size $(|U| - 1)t$. ∎

Proof of Claim ??. For every vertex $b \in B$ remove all but exactly two edges. Hence we get a subgraph G_1 of G in which $deg(b) = 2$ for every $b \in B$. Hence, every $b \in B$ form a simple path of length 2 in G_1. Replace each such path by a single edge; this is equivalent to the contraction of a single edge in the neighbourhood of every $b \in B$. As a result we get a multigraph G_2 that is a minor of G, and hence outerplanar. In this multigraph, the number of parallel edges between two vertices is at most two, as otherwise in G, there would have been a $K_{2,3}$ minor, which is a contradiction. Hence $|E(G_2)| \leq 4|V(G_2)|$.

However, by construction, every edge $e \in E(G_2)$ corresponds to a vertex $b \in B$, with a $1 - 1$ correspondence. Further $V(G_2) = A$. Hence the claim follows. ∎

Algorithm Approx.

Let $G = (V, E)$ be an outerplanar graph, $\epsilon > 0$ the error parameter, and let G' be the (d, s)-union graph obtained by *Algorithm 1*, for $\epsilon' = \epsilon/10$ (and d accordingly).

The purpose of the algorithm Approx is to estimate the γ-layered graph G^* that is $\epsilon/10$-close to G' (and hence $\epsilon/5$-close to G). That is, for each $i \in [L]$ to give an approximation to n_i the number of high-degree components in G^i, and $lfeq[i]$ the frequency vector of G^i, where L is as defined in Definition **??**.

Algorithm Approx runs a main algorithm Sampler that samples high-degree vertices of G' accoring to a distribution in which all root vertices in the same layer are sampled with the same probability. Sampler runs another algorithm, Sampler2 that estimates the degree and the component frequency of the sampled vertex in order for Sampler to be able to update n_i and $lfreq[i]$ accordingly.

We now present the algorithm Sampler2, which given an outerplanar graph $G(V, E)$, and a vertex y, approximates $freq_{G'}(y)$ and $deg_{G'}(y)$.

Let $q = poly(\log n)$ (e.g., the reader may take $q = 10\log^3 n$ to get the right magnitude, the exact value will not be defined here).

Algorithm Sampler2(y)**:**

y is a vertex. The output is an estimate $\widetilde{deg_{G'}(y)}$, and an estimate $\widetilde{freq_{G'}(y)}$ for $deg_{G'}(y)$ and $freq_{G'}(y)$ respectively.

1. Obtain $deg_G(y)$ by one query. If $deg(y) \leq d$ stop; y is not a root of a (d, s)-rooted component in G'.
   Otherwise, let $c = 0$ (c will count the number of discovered (d, s)-components that are connected to y). Let $Freq(y)$ be the all-zero vector of dimension $f(d, s)$.
2. Repeat independently for q times: Choose a random low-degree neighbour $u \in_R V^l \cap N(y)$.

Look at the component of u in $G[V^l]$ (by applying Levi-Ron, as needed), take the multiway cut, and finally see if y is still in the same component as u in G' which is the graph obtained by the simulation of Algorithm 1 locally on v (some edges adjacent to y might be deleted, due to the multiway cut procedure, and due to the deletion of edges between two high-degree vertices).
If y is still in the same component as u in G^*, increment c. Also, depending on the type of the (d, s) component containing u and connected to y, increment the corresponding coefficient of $Freq(y)$.

3. Take $\widetilde{deg_{G'}}(y) = deg_G(y) \cdot c/q$, and $\widetilde{freq_{G'}}(y) = Freq(y)/c$.

It is easy to see that SAMPLER2 will estimate well the required parameters for y such that $deg_{G'}(y)$ is high enough. This motivates the following definition.

▶ **Definition 5.3.** Let $Bad = \{y|\ deg_{G'}(y) \leq deg_G(y)/(20 \log n)\}$.

We present the following lemma without proof.

▶ **Lemma 5.4.** *Let G' be the graph obtained from a graph G by the partition, and $y \notin Bad$. Then for the output of* SAMPLER2 *it holds that $Pr[|\widetilde{deg_{G'}}(y) - deg_{G'}(y)| \geq \epsilon^2 \cdot deg_{G'}(y)/100] \leq 1/\log^5 n$, and $Pr[|freq(y) - \widetilde{freq(y)}| \geq \epsilon^2/100] \leq 1/\log^5 n$.*

The proof is a standard application of Chernoff-Hoefding bound and will not be presented here.

We now present the algorithm SAMPLER, which given an outerplanar graph $G(V, E)$ as input, returns a high-degree vertex in G'. Among high-degree vertices in any fixed layer i, the probability of obtaining each one will be roughly the same and the total probability of returning any one of the roots in layer i will be roughly proportional to $n_i\gamma^i$, where n_i is the number of high-degree vertices in layer i.

Algorithm SAMPLER:

1. Sample uniformly at random, a vertex $v \in V(G)$, if $deg_G(v) \geq d$ reject.
2. Otherwise, if $deg_G(v) \leq d$, query all neighbours of v and if for all $y \in N(v)$, $deg_G(y) \leq d$ reject.
3. Otherwise, let $N^h(v) = \{y|\ deg_G(y) > d\}$ and let $deg^h(v) = |N^h(v)|$. Choose uniformly a random member $y \in N^h(v)$. Discard y and reject with probability $1 - \frac{deg^h(v)}{d}$. Otherwise (with probability $\frac{deg^h(v)}{d}$), we will consider y to be a candidate for a (random) root of G'.
4. Run SAMPLER2(y). If y not rejected, check if v is a neighbour of y in G' by simulating locally Algorithm 1, (as is done in SAMPLER2, for other random neighbours of y). If not reject.
5. With probability $\gamma^i/\widetilde{deg_{G'}}(y)$, return y, where i is the largest for which $\gamma^i \leq \widetilde{deg_{G'}}$.

We claim that for each layer of G^*, the distribution that SAMPLER2(y) indices on its roots is (nearly) uniform. Moreover, if there is enough "mass" in $G^i, i = \alpha, \ldots L$ then SAMPLER will produce a random root of G' (in some layer) w.h.p.

Before stating the corresponding lemma, we note that what appears to be a trivial sampling is not correct. Namely, choosing a random vertex $y \in V(G)$ and returning y if its degree is high enough is not likely to succeed, as it might be the case that the number of roots in G' is very small (possibly 1), while their degree is very high. In such a case, a random sampling will not find a random root, while the influence of the small number of roots on the structure of G' (in terms of distance to G), is very high.

▶ **Lemma 5.5.** *Suppose that* $\sum_{y \in roots(G')} \deg_{G'}(y) \geq \epsilon n / \log n$ *then* SAMPLER *produces a random* $y \in roots(G')$ *distributed uniformly on each layer of* G^**, with probability at least* $1/\log^2 n$*.*

Proof. We do not present the full proof of the lemma in this version. We will only prove that SAMPLER induces the uniform probability on each layer of G^* and that with high probability it will output such y.

Indeed, consider a fixed $y \in roots(G')$. The only way y is going to be accepted is if a v that is selected in step 1 is one of the $deg_{G'}(y)$ neighbours of y in G'. Denote this set of neighbours of y by N'. Each such v is chosen with probability $1/n$ at step 1, and will pass step 2. Now, conditioned on specific v chosen, the probability that y is chosen in step 3 and not discarded is $\frac{1}{deg^h(v)} \cdot \frac{deg^h(v)}{d} = \frac{1}{d}$. Finally, if y is chosen at step 3, and conditioned on the event that SAMPLER2(y) accepts y and estimates $\deg_{G'}(y)$ correctly, which happen with probability $1 - 1/\log^5 n$, y will be accepted with probability $\gamma^i / \widetilde{deg_{G'}(y)}$. Altogether, the probability of returning y is:

$$Prob(\text{SAMPLER returns } y) = \sum_{v \in N'} \frac{1}{n} \cdot \frac{1}{d} \cdot \frac{1}{\widetilde{deg_{G'}(y)}} = \frac{1}{nd} \cdot \frac{\gamma^i \cdot deg_{G'}(y)}{\widetilde{deg_{G'}(y)}}$$

Assuming the estimate is as good as we needed, this probability is very close to $\frac{\gamma^i}{nd}$ and hence to uniform on each layer of G^*, as it is essentially independent of y but just on the layer.

Finally, let i be such that $n_i \cdot \gamma^i = \Omega(n/\log n)$, namely, such that the ith layer in G^* has a large mass. The probability some y in the i layer of G^* is accepted the sum above for all y in the layer which is just $n_i \cdot \frac{\gamma^i}{nd} \geq \frac{\epsilon}{d \log n}$. By our choice of d, the proof is concluded. ∎

Algorithm APPROX is now self evident: it runs SAMPLER for $poly(\log n)$ times to generate enough random roots to hit all significant layers G^i. Would there be no vertices in Bad, the estimate would clearly be correct, by Chernoff-Hoefding bounds. The effect of vertices in Bad can be shown to be small, as the total number connected to vertices in Bad is small. We avoid further details in this version.