Testing Bipartitness in an Augmented VDF Bounded-Degree Graph Model*

Oded Goldreich†

May 8, 2019

Abstract

In a recent work (*ECCC*, TR18-171, 2018), we introduced models of testing graph properties in which, in addition to answers to the usual graph-queries, the tester obtains *random vertices drawn according to an arbitrary distribution* $\mathcal{D}$. Such a tester is required to distinguish between graphs that have the property and graphs that are far from having the property, *where the distance between graphs is defined based on the unknown vertex distribution* $\mathcal{D}$. These ("vertex-distribution free" (VDF)) models generalize the standard models in which $\mathcal{D}$ is postulated to be uniform on the vertex-set, and they were studies both in the dense graph model and in the bounded-degree graph model.

The focus of the aforementioned work was on testers, called strong, whose query complexity depends only on the proximity parameter ϵ. Unfortunately, in the standard bounded-degree graph model, some natural properties such as Bipartiteness do not have strong testers, and others (like cycle-freeness) do not have strong testers of one-sided error (whereas one-sided error was shown inherent to the VDF model). Hence, it was suggested to study general (i.e., non-strong) testers of "sub-linear" complexity.

In this work, we pursue the foregoing suggestion, but do so in a model that augments the model presented in the aforementioned work. Specifically, we provide the tester with an evaluation oracle to the unknown distribution $\mathcal{D}$, in addition to samples of $\mathcal{D}$ and oracle access to the tested graph. Our main results are testers for Bipartitness and cycle-freeness, in this augmented model, having complexity that is almost-linear in the square root of the "effective support size" of $\mathcal{D}$.

Keywords: Property Testing, Graph Properties, Effective Support Size, One-Sided versus Two-Sided Error.

Contents

*Preliminary version; comments are most welcome.

†Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel. Email: oded.goldreich@weizmann.ac.il.

1 Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., a recent textbook [3]). Loosely speaking, property testing typically refers to sub-linear time probabilistic algorithms for deciding whether a given object has a predetermined property or is far from any object having this property. Such algorithms, called testers, obtain local views of the object by making adequate queries; that is, the object is seen as a function and the testers get oracle access to this function (and thus may be expected to work in time that is sub-linear in the size of the object).

A significant portion of the foregoing research was devoted to testing graph properties in two different models: the dense graph model (introduced in [5] and reviewed in [3, Chap. 8]) and the bounded-degree graph model (introduced in [6] and reviewed in [3, Chap. 9]). In both models, it was postulated that the tester can sample the vertex-set uniformly at random[1] (and, in both models, distances between graphs were defined with respect to this distribution).

In a recent work [4], we considered settings in which uniformly sampling the vertex-set of the graph is not realistic, and asked what happens if the tester can obtain random vertices drawn according to some distribution $\mathcal{D}$ (and, in addition, obtain answers to the usual graph-queries). The distribution $\mathcal{D}$ should be thought of as arising from some application, and it is not known *a priori* to the (application-independent) tester. In this case, it is also reasonable to define the distance between graphs with respect to the distribution $\mathcal{D}$, since this is the distribution that the application uses.

These considerations led us to introduce models of testing graph properties in which the tester obtains *random vertices drawn according to an arbitrary vertex distribution* $\mathcal{D}$ (and, in addition, obtains answers to the usual graph-queries). Such a tester is required to distinguish between graphs that have the property and graphs that are far from having the property, *where the distance between graphs is defined based on the unknown vertex distribution* $\mathcal{D}$. These ("vertex-distribution free" (VDF)) models generalize the standard models in which $\mathcal{D}$ is postulated to be uniform on the vertex-set, and they were studies both in the dense graph model and in the bounded-degree graph model (see [4, Sec. 2] and [4, Sec. 3], respectively).

The focus of [4] was on testers, called strong, whose query complexity depends only on the proximity parameter ϵ. Unfortunately, in the standard *bounded-degree graph model*, some natural properties such as bipartiteness do not have strong testers, and others (like cycle-freeness) do not have strong testers of one-sided error (whereas one-sided error was shown inherent to the VDF model [4, Thm. 1.1]). Hence, it was suggested in [4, Sec. 5.2] to study general (i.e., non-strong) testers of "sub-linear" complexity, especially for the VDF bounded-degree graph model.

In this work, we pursue the foregoing suggestion, but do so in a model that augments the model presented in [4]. Specifically, we provide the tester with an evaluation oracle to the unknown distribution $\mathcal{D}$, in addition to samples of $\mathcal{D}$ and oracle access to the tested graph.

1.1 The vertex-distribution-free model and its augmentation

We start by recalling the vertex-distribution free (VDF) model that generalizes the bounded-degree graph model. Essentially, this model differs from the standard bounded-degree graph model in that the tester cannot obtain uniformly distributed vertices, but rather random vertices drawn according to an arbitrary distribution $\mathcal{D}$ (which is unknown *a priori*)). (In addition, the tester obtains answers to the usual graph-queries.) As usual, the tester is required to accept (whp) graphs that have the predetermined property and reject (whp) graph that that are far from the property, but the distance between graphs is defined in terms of the distribution $\mathcal{D}$. Specifically, when generalizing the standard bounded-degree model, we define the distance between graphs as the sum of the weights of the edges

[1] Actually, in all these models, it is postulated that the vertex-set consists of $[n] = \{1, 2, ..., n\}$, where n is a natural number that is given explicitly to the tester, enabling it to sample $[n]$ uniformly at random.

in their symmetric difference, where *the weight of an edge is proportional to the sum of the probability weights of its end-points according to* $\mathcal{D}$.

Recall that the bounded-degree model (both in its standard and VDF incarnations) refers to a fixed degree bound, denoted d, and to graphs that are represented by their incidence functions; that is, the graph $G = (V, E)$ is represented by the incidence function $g : V \times [d] \to V \cup \{\perp\}$, where $g(v, i) = u$ if u is the i^{th} neighbour of v and $g(v, i) = \perp$ if v has less than i neighbours. Fixing a vertex distribution $\mathcal{D}$, we say that the graph G or rather its incidence function g is ϵ-far from the graph property Π if, for every $g' : V \times [d] \to V \cup \{\perp\}$ that represents a graph in Π, it holds that $\Pr_{v \leftarrow \mathcal{D}, i \in [d]}[g(v, i) \neq g'(v, i)] > \epsilon$.

Hence, following [4, Def. 3.1], a tester of Π in the VDF bounded-degree graph model is given a proximity parameter ϵ, samples drawn from an arbitrary distribution $\mathcal{D}$, and oracle access to the incidence function of the graph, $G = (V, E)$. It is required that, for every vertex-distribution $\mathcal{D}$ (and every $\epsilon > 0$ and G), the tester accepts (whp) if G is in Π and rejects (whp) if G is ϵ-far from Π (where the distance is defined according to $\mathcal{D}$).

The augmentation. Here, we augment the foregoing model by providing the tester also with an *evaluation oracle* to the vertex distribution; that is, an oracle that on query v returns $\mathcal{D}(v) = \Pr_{x \leftarrow \mathcal{D}}[x = v]$. This augmentation is introduced because we could not obtain our results without it (or without some relaxation of it), but it can be justified as feasible in some settings (see brief discussion in Section 1.4). At this point, we spell out the resulting definition of a tester.

Definition 1.1 (the augmented VDF testing model): *For a fixed* $d \in \mathbb{N}$*, let* Π *be a property of graphs of degree at most* d*. An* augmented VDF tester for the graph property Π (in the bounded-degree graph model) *is a probabilistic oracle machine* T *that satisfies the following two conditions* (for all sufficiently large V)*, when given access to the following three oracles: an incidence function* $g : V \times [d] \to V \cup \{\perp\}$*, a device – denoted* $\texttt{samp}_{\mathcal{D}}$ *– that samples in* V *according to an arbitrary distribution* $\mathcal{D}$*, and an evaluation oracle – denoted* $\texttt{eval}_{\mathcal{D}}$ *– for* $\mathcal{D}$*.*

1. *The tester accepts each* $G = (V, E) \in \Pi$ *with probability at least* $2/3$*; that is, for every* $g : V \times [d] \to V \cup \{\perp\}$ *representing a graph in* Π *and every* $\mathcal{D}$ (and $\epsilon > 0$)*, it holds that* $\Pr[T^{g, \texttt{samp}_{\mathcal{D}}, \texttt{eval}_{\mathcal{D}}}(\epsilon) = 1] \geq 2/3$.

2. *Given* $\epsilon > 0$ *and oracle access to any* $G = (V, E)$ *and* $\mathcal{D}$ *such that* G *is* ϵ*-far from* Π *according to* $\mathcal{D}$*, the tester rejects with probability at least* $2/3$*; that is, for every* $\epsilon > 0$ *and distribution* $\mathcal{D}$*, if* $g : V \times [d] \to V \cup \{\perp\}$ *satisfies* $\delta^{\Pi}_{\mathcal{D}}(g) > \epsilon$*, then it holds that* $\Pr[T^{g, \texttt{samp}_{\mathcal{D}}, \texttt{eval}_{\mathcal{D}}}(\epsilon) = 0] \geq 2/3$*, where* $\delta^{\Pi}_{\mathcal{D}}(g)$ *denotes the minimum of* $\delta_{\mathcal{D}}(g, g')$ *taken over all incidence functions* $g' : V \times [d] \to V \cup \{\perp\}$ *that represent graphs in* Π*, and*

$$\delta_{\mathcal{D}}(g, g') \stackrel{\text{def}}{=} \Pr_{v \leftarrow \mathcal{D}, i \in [d]}[g(v, i) \neq g'(v, i)]. \tag{1}$$

   (That is, $\delta_{\mathcal{D}}(g, g') = \sum_{v \in V} \mathcal{D}(v) \cdot |\{i \in [d] : g(v, i) \neq g'(v, i)\}|/d$.)

The tester is said to have one-sided error probability *if it always accepts graphs in* Π*; that is, for every* $g : V \times [d] \to V \cup \{\perp\}$ *representing a graph in* Π (and every $\mathcal{D}$ and $\epsilon > 0$)*, it holds that* $\Pr[T^{g, \texttt{samp}_{\mathcal{D}}, \texttt{eval}_{\mathcal{D}}}(\epsilon) = 1] = 1$.

At times, we shall identify the incidence function g with the graph G that g represents, and simply say that we provide the tester with oracle access to G.

The query complexity of a tester is the maximum number of queries it makes to its (three) oracles as a function of ϵ and parameters of the vertex-distribution $\mathcal{D} : V \to [0, 1]$. The parameters we have in mind are label-invariant, where a parameter ψ is label invariable if $\psi(\mathcal{D}) = \psi(\mathcal{D}')$ for any two distributions $\mathcal{D}$ and $\mathcal{D}'$ that have the same histogram (i.e., for every $p > 0$ it holds that $|\{v : \mathcal{D}(v) = p\}| = |\{v :$

$\mathcal{D}'(v)=p\}|)$.[2] Recall that in the standard testing model, the complexity could depend on the size of the vertex-set, which is a special case of a parameter of $\mathcal{D} : V \to [0.1]$. However, we are interested in more refined parameters of $\mathcal{D}$. The first parameter that comes to mind is the support size of $\mathcal{D}$, but this parameter is too sensitive to insignificant changes in $\mathcal{D}$ (e.g., any distribution over V is infinitesimally close to having support size $|V|$). A more robust parameter is the "minimum effective support size"; that is, being "close" to a distribution with the specified support-size (cf., [1]).

Definition 1.2 (effective support size): *We say that the distribution $\mathcal{D}$ has* η-effective support of size n *if $\mathcal{D}$ is η-close to a distribution that has support size at most n. The* minimal η-effective support size of $\mathcal{D}$ *is the minimal n such that $\mathcal{D}$ has η-effective support of size n.*

The notion of effective support size is much more robust that the support size; in particular, if $\mathcal{D}$ is infinitesimally close to a distribution that has η-effective support of size n, then $\mathcal{D}$ that has η-effective support of size $n+1$ (where the additional unit is needed only in pathological cases).[3] In general, if $\mathcal{D}$ is $o(\epsilon)$-close to a distribution that has ϵ-effective support size n, then $\mathcal{D}$ that has $(1+o(1))\cdot\epsilon$-effective support size n.

An initial observation and an open problem. Recall that it was shown in [4, Prop. 3.2] that, without loss of generality, any tester in the VDF model only queries vertices that were provided as answers to prior sample and graph queries. The argument extends to the augmented VDF model. In contrast, it is unclear whether [4, Thm. 3.3], which asserts that strong testability in the VDF model yields strong testability with one-sided error, holds in the augmented VDF model.

Open Problem 1.3 (does one-sided error testing reduce to general testing): *For $q : (0,1] \to \mathbb{N}$, suppose that Π is a graph property that can be tested using $q(\epsilon)$ queries in the augmented VDF model, where ϵ denotes the proximity parameter. Does there exists a function $q' : (0,1] \to \mathbb{N}$ such that Π has a one-sided error tester of query complexity $q'(\epsilon)$ in the augmented VDF model.*

(Recall that in the original VDF model, an upper bound of $q'(\epsilon) = \exp(O(q(\epsilon)))$ was shown [4, Thm. 3.3].)

1.2 Our results

Our main result is testing Bipartitness in the augmented VDF model within complexity that matches the complexity of the known tester in the standard (bounded-degree graph) model [7], which in turn is almost optimal [6].

Theorem 1.4 (testing bipartitness in the augmented VDF model): *Bipartiteness can be tested in the augmented VDF testing model* (of Definition 1.1) *in expected time $\widetilde{O}(\sqrt{n})\cdot\mathrm{poly}(1/\epsilon)$, where n denotes*

[2] Indeed, in this case there exists a permutation $\pi : V \to V$ such that $\mathcal{D}'(v) = \mathcal{D}(\pi(v))$ for every $v \in V$.

[3] Let $\mathcal{D}'$ be the foregoing distribution that has η-effective support of size n. Then, the typical case is that, for some $\eta' < \eta$, the distribution $\mathcal{D}'$ has η'-effective support of size n. In this case, any distribution that is $(\eta-\eta')$-close to $\mathcal{D}'$ has η-effective support of size n. The pathological case is that $\mathcal{D}'$ has η-effective support of size n, but for every $\eta' < \eta$ the minimal η'-effective support size of $\mathcal{D}'$ is larger than n. We claim that in this case, for some $\eta' < \eta$, the distribution $\mathcal{D}'$ has η'-effective support of size $n+1$ (and it follows that any distribution that is $(\eta-\eta')$-close to $\mathcal{D}'$ has η-effective support of size $n+1$). Suppose that $\mathcal{D}'$ is η-close to a distribution $\mathcal{D}''$ of support size n. We prove the claim by considering two cases.

1. If the support of $\mathcal{D}'$ is contained in the support of $\mathcal{D}''$, then the claim is trivial (since then $\mathcal{D}'$ has support size n).
2. Otherwise, let v be in the support of $\mathcal{D}'$ but not in the support of $\mathcal{D}''$, and consider modifying $\mathcal{D}''$ by moving a probability mass of $\mathcal{D}'(v) > 0$ from $\{u : \mathcal{D}''(u) > \mathcal{D}'(u)\}$ to v. Then, the modified distribution $\mathcal{D}'''$ has support size $n+1$ and is $(\eta - \mathcal{D}'(v))$-close to $\mathcal{D}'$, and so the claim follows with $\eta' = \eta - \mathcal{D}'(v)$.

the minimal $\epsilon/5$-effective support size of the vertex distribution $\mathcal{D}$, used by the tester. Furthermore, the tester has one-sided error.

The tester that we use when proving Theorem 1.4 starts by obtaining a rough approximation of the minimal effective support size of $\mathcal{D}$. In fact, a rough approximation of the effective support size of $\mathcal{D}$ is implicit in the running-time of the asserted tester. We comment that it is essential to make queries to $\texttt{eval}_{\mathcal{D}}$ in order to obtain such an approximation, since obtaining such an approximation by making queries only to $\texttt{samp}_{\mathcal{D}}$ has complexity $n^{1-o(1)}$ (cf. [9]).[4] In fact, using both types of queries, we obtain a good approximation in polylogarithmic time (see Section 2.2).

Next, we extend the known reduction of testing cycle-freeness to testing Bipartiteness, presented in [2] for the standard (bounded degree graph) model, to the (augmented) VDF model. Combining this reduction with Theorem 1.4, we obtain

Theorem 1.5 (testing cycle-freeness in the augmented VDF model): *Cycle-freeness can be tested in the augmented VDF testing model* (of Definition 1.1) *in expected time $\widetilde{O}(\sqrt{n})\cdot\text{poly}(1/\epsilon)$, where n denotes the minimal $\epsilon/5$-effective support size of the vertex distribution $\mathcal{D}$ used by the tester. Furthermore, the tester has one-sided error.*

A begging open problem is whether the foregoing results can also be obtained in the original VDF model (of [4]), at least when providing the tester with the effective support size.

Open Problem 1.6 (the complexity of testing Bipartiteness in the VDF model): *What is the query complexity of testing Bipartiteness in the* (original) *VDF model. In particular, can Bipartite be tested in this model in time $\widetilde{O}(\sqrt{n})\cdot\text{poly}(1/\epsilon)$, where n denotes the $\Omega(\epsilon)$-effective support size of the vertex distribution $\mathcal{D}$ used by the tester? The question holds both for the VDF model* (as defined in [4])*, and in a model in which the tester is provided with the effective support size.*[5]

Since the reduction of testing cycle-freeness to Bipartiteness does not make queries to $\mathcal{D}$ (and is in fact oblivious of $\mathcal{D}$), any upper bound regarding testing Bipartiteness in the (original) VDF model would yield a similar result for testing cycle-freeness. However, the latter problem may be easier.

1.3 Techniques

The testers asserted in Theorems 1.4 and 1.5 constitute testers for the standard (bounded-degree graph) model that meet the best results known in that model. Given that the proofs of the latter results are quite complex (see, e.g., [7]), it is fortunate that we can proceed by reducing the current results to the known ones.

Proving Theorem 1.4. The Bipartite tester asserted in Theorem 1.4 is obtained by a natural adaptation of the corresponding tester for the standard (bounded-degree graph) model [7]. The latter tester operates by taking many short random walks from few randomly selected vertices, where, in each step of a random walk, the next vertex is selected uniformly among the neighbors of the current vertex. Instead, our tester will select the the next vertex (among the neighbors of the current vertex) with probability that is *proportional to the probability weight of the corresponding incident edges* (according to $\mathcal{D}$); that is, being at vertex v we move to a neighbor w with probability proportional to $\mathcal{D}(v)+\mathcal{D}(w)$.

[4] Specifically, at least $n^{1-o(1)}$ queries are necessary to distinguish an n-grained distribution of support size $n/11$ from an n-grained distribution with support size n/f, for any $f = n^{o(1)}$, where a distribution is called n-grained if all probabilities are multiples of $1/n$.

[5] Note that the result of [9] does not rule out the possibility of approximating n to a factor of n^c using $n^{0.5+c'}$ queries, for some $c, c' \in (0, 0.5)$.

Here is where we make use of the evaluation oracle $\mathtt{eval}_{\mathcal{D}}$. (The start vertices will be selected with probability that is proportional to the $\mathcal{D}$-weight of their incident edges.)

Since the analysis of the foregoing tester in the standard (bounded-degree graph) model is quite complex, we wish to use this analysis (of [7]) as a black-box. Towards this end, we view the foregoing tester as emulating the tester of [7] on an auxiliary graph in which weighted edges are replaced by a proportional number of parallel edges, while recalling that the analysis of [7] holds also for (non-simple) graphs having parallel edges (see [8]). (We stress that this is a mental experiment performed in the analysis; the actual algorithm is essentially as outlined above.)

Indeed, this strategy may yield an auxiliary graph in which the average degree is much smaller than the maximum degree, whereas the analysis in [7] relies on the two quantities being of the same order of magnitude (see [8]). This issue can be addressed by using the transformation of [8], but doing so will yield a slightly less natural tester (than the one outlined above). So instead of applying the transformation of [8] as a black-box, we use an alternative implementation of its underlying ideas, which capitalizes on the fact that we can set the number of parallel edges (in our mental experiment) to be much larger than the number of vertices in the tested graph. We stress that the complexity of the tester of [7] is dominated by the number of the vertices in the graph, and hardly depends on the number of edges in it.[6]

The forgoing description presumes that we have a good upper bound on the support size of $\mathcal{D}$. (Note that such an upper bound (along with oracle access to the auxiliary graph) suffices for emulating the tester of [7].) Actually, aiming at complexity bounds that depend on the effective support size of $\mathcal{D}$ rather than on its support size, we "trim" the graph by ignoring edges of weight $o(\epsilon/n)$, where n is an upper bound on the $\epsilon/4$-effective support size of $\mathcal{D}$. Lastly, we show how to obtain such a good upper bound by using both the sampling and evaluation oracles of $\mathcal{D}$.

As for approximating the effective support size of $\mathcal{D}$, the basic observation is that if n is the minimal integer such that an η fraction of the weight of $\mathcal{D}$ resides on elements of weight at most η/n, then the $\Theta(\eta)$-support size of $\mathcal{D}$ is between $\Omega(n)$ and $O(n/\eta)$. Such a rough approximation suffices for the foregoing application, and we obtain it obtained by using a doubling procedure. We actually obtain a better approximation (see Theorem 2.2) by using the rough approximation as a starting point and approximating the total weight assigned to each set $\{v : 2^{-i+1} \leq \mathcal{D}(v) < 2^{-i}\}$ for $i \in [\log(n/\eta) + O(1)]$.

Proving Theorem 1.5. The cycle-freeness tester asserted in Theorem 1.5 is obtained by using the known reduction of testing cycle-freeness to testing Bipartiteness, which was presented and analyzed for the standard (bounded degree graph) model in [2]. Recall that this reduction replaces each edge at random either by a path of length two or by a path of length one (equiv., leaves the edge intact). It was shown in [2, Lem 3.1] that this transformation translates a graph that is far from being cycle-free to a graph that is far from being bipartite, where the distance refers to the number of edges that should be omitted to make the graph satisfy the property. The challenge is to extend this claim to weighted graphs (or rather to distances as measured by the sum of the weights of edges that should be omitted to make the graph satisfy the property).

We meet this challenge by observing that the edges that should be omitted in order to make the original graph cycle-free are those that do not reside on a *maximal spanning forest of the graph.* We then bucket these edges according to their approximate weight, and consider only the edges that are in the union of the heaviest bucket and said spanning forest. Considering the two-connected components of the corresponding graph, we observe that the remaining edges of the forest (i.e., those that reside inside these two-connected components) cannot be lighter than the edges in the bucket, which reduces the problem of lower-bounding the weight of edges to lower-bounding their number. At this point, we invoke [2, Lem 3.1] and are done.

[6] The complexity is almost-linear in a square root of the number of vertices, and only grows logarithmically with the number of edges.

1.4 Discussion

As admitted upfront, the augmentation of the VDF model captured by Definition 1.1 is made for opportunistic reasons: This augmentation (or some relaxation of it) seems essential to the testers asserted in Theorems 1.4 and 1.5. Nevertheless, one may envision setting in which an evaluation oracle as postulated in the augmentation can be implemented or at least be well-approximated.

Recall that the VDF model was motivated by settings in which some process (or application) of interest refers to (or embeds or emulates) a huge graph; in particular, the process generates random vertices according to some unknown distribution $\mathcal{D}$, and answers incidence queries regarding the graph. Furthermore, in case the vertices of the graph are real sites, they may maintain a count of the number of times they were visited by the foregoing process (or application). This yields a good approximation of the visiting probabilities that underly the vertex distribution $\mathcal{D}$ in question.

Indeed, the vertex distribution represents the "importance" of the various vertices from the application's point of view; that is, the application encounters vertices according to the distribution $\mathcal{D}$, and the relative "importance" of a vertex (to the application) is captured by the probability that it is encountered (by the application). Hence, the distance of a graph to the property represents the relative importance of the "part of the graph" that violates the property.

A VDF tester offers an application-independent way of determining whether the huge graph (embedded or emulated by an application) has some predetermined property or is far from having the property, where the distance that is relevant here is one that is induced by the vertex-distribution $\mathcal{D}$ (arising from the application).

2 The Bipartiteness Tester

For starters, we consider a model in which the tester is further augmented by providing it with the effective support size of the vertex distribution $\mathcal{D}$. Specifically, on input proximity parameter ϵ, we also provide the tester with an upper bound, denoted n, on the minimal $\epsilon/4$-effective support size of $\mathcal{D}$. The complexity of the following tester depends on that upper bound, and at a latter stage we shall show how the tester can obtain a good upper bound by itself.

Theorem 2.1 (a tester that gets an upper bound on the effective support size): *There exists an oracle machine T that, on input ϵ and n such that n is an upper bound on the minimal $\epsilon/4$-effective support size of $\mathcal{D}$, and oracle access to $G, \texttt{samp}_{\mathcal{D}}, \texttt{eval}_{\mathcal{D}}$, runs in time $\widetilde{O}(\sqrt{n}) \cdot \mathrm{poly}(1/\epsilon)$ and constitutes a bipartite tester* (of one-sided error) *as defined above. That is:*

1. *If G is bipartite, then* $\Pr[T^{G,\texttt{samp}_{\mathcal{D}},\texttt{eval}_{\mathcal{D}}}(\epsilon, n) = 1] = 1$.
2. *If G is ϵ-far from being bipartite according to $\mathcal{D}$, then* $\Pr[T^{G,\texttt{samp}_{\mathcal{D}},\texttt{eval}_{\mathcal{D}}}(\epsilon, n) = 0] \geq 2/3$.

The Bipartite tester asserted in Theorem 1.4 is obtained by combining Theorem 2.1 with a procedure that approximates the effective support size, which in turn is provided in Section 2.2.

2.1 Proof of Theorem 2.1

Our starting point is the Bipartiteness tester of [7] that works in the standard (bounded-degree graph) model. On input an n-vertex graph, this tester operates by taking many (i.e., $\widetilde{O}(\sqrt{n})$) short (i.e., $\mathrm{poly}(\log n)$) random walks from few (i.e., $O(1)$) randomly selected vertices (where the O-notation hide a polynomial dependence on $1/\epsilon$). In [7], in each step of each random walk, the next vertex is selected uniformly among the neighbors of the current vertex. Our basic idea is to adapt this tester to the current setting by selecting the next vertex among the neighbors of the current vertex with probability that is *proportional to the probability weight of the corresponding incident edges* (according to $\mathcal{D}$).

Wishing to use the analysis of [7] as a black-box, we present the foregoing tester as emulating the tester of [7] on an auxiliary graph in which weighted edges are replaced by a proportional number of parallel edges, while recalling that *the analysis of [7] holds also for* (non-simple) *graphs having* (many) *parallel edges* (see [8]). That is, on input G, we consider an auxiliary multi-graph G' in which the number of edges between vertices is approximately proportional to the weight of the corresponding edge in G. Indeed, this may yield a graph with a large discrepancy between the average degree and the maximum degree, but this can be fixed using the technique of [8]. In fact, in order to present a simpler and more natural algorithm, we use an alternative implementation of the foregoing technique, which suffices for our setting.

Fixing a vertex distribution $\mathcal{D}$, on input $G = (V, E)$, proximity parameter ϵ and the value n (provided as an effective support size of $\mathcal{D}$), we consider the corresponding auxiliary multi-graph $G' = (V', E')$. For sake of simplicity, we shall assume first that *for each* $v \in V$ *either* $\mathcal{D}(v) = 0$ *or* $\mathcal{D}(v) \geq \rho$, where we later set $\rho = \Theta(\epsilon/n)$ and reduce the general case to this special case. Recalling that the weight of the edge $\{u, v\} \in E$ under $\mathcal{D}$ is $2\cdot(\mathcal{D}(u)+\mathcal{D}(v))/d$, we place $m_{u,v} \stackrel{\rm def}{=} \lfloor(\mathcal{D}(u) + \mathcal{D}(v)) \cdot N\rceil \gg 1$ parallel edges between u and v in G', where $N = \text{poly}(d/\epsilon\rho)$ will be determined later. We keep only the non-isolated vertices in V'; that is, $v \in V'$ if and only if $\sum_u m_{u,v} > 0$, which holds if and only if either $\mathcal{D}(v) \geq \rho$ or $\mathcal{D}(u) \geq \rho$ for some neighbor u of v. Hence, $|V'| \leq d/\rho$.

Indeed, the number of parallel edges between u and v is either zero or at least $\rho \cdot N - 1$, and the total the number of edges in G' (i.e., $\sum_{\{u,v\}\in E} \lfloor(\mathcal{D}(u) + \mathcal{D}(v)) \cdot N\rceil$) is roughly $\sum_{w\in V} d_w \cdot \mathcal{D}(w) \cdot N = \Theta(N)$, where d_v is the degree of v in G. Hence, if G is ϵ-far from being bipartite, under $\mathcal{D}$, then G' is $(\epsilon - o(\epsilon))$-far from being bipartite (in the sense that more than $(\epsilon - o(\epsilon)) \cdot |E'|$ edges must be omitted from G' to make it bipartite). Specifically, we have:

Claim 2.1.1 (weight under $\mathcal{D}$ versus number of edges in G'): *If edges of total $\mathcal{D}$-weight at least δ must be removed from G in order to make it bipartite, then at least $\delta \cdot N - d|V'|$ edges must be removed from G' in order to make it bipartite.*

Indeed, if $N = \omega(|V'|/\delta)$, then $\delta \cdot N - d|V'| = (\delta - o(\delta)) \cdot N$.

Proof: Removing a weighted edge of G is equivalent to removing the corresponding parallel edges of G'. Recalling that an edge $\{u, v\} \in E$ of weight $w_{u,v} = 2 \cdot (\mathcal{D}(u) + \mathcal{D}(v))/d \leq \mathcal{D}(u) + \mathcal{D}(v)$ yields $m_{u,v} = \lfloor(\mathcal{D}(u) + \mathcal{D}(v)) \cdot N\rceil \geq w_{u,v} \cdot N - 1$ parallel edges and using $|E| < d|V'|$, the claim follows.[7] ■

At this point, we could have emulated an execution of the bipartite tester of [8] on $G' = (V', E')$, which boils down to emulating the tester of [7] on an auxiliary graph $G'' = (V'', E'')$, where $|V''| = O(|V'|)$. (Such an emulation would have resulted in an algorithm that is slightly different from the one we outlined above.) Instead, we prefer to capitalize on the specific features of G', and emulate the tester of [7] on a different auxiliary graph $G'' = (V'', E'')$ that is more closely related to G' (and to G). This allows us to invoke a minor variant of the tester of [7] (indeed, the variant outlined upfront) directly on G. In particular, we shall use the fact that the number of parallel edges (between a pair of connected vertices) in G' is large (e.g., much larger than the square of the ratio between the maximum and average degrees in G'); that is, $M = \omega((N/M)^2/\epsilon)$, where $M = \min_{\{u,v\}\in E: m_{u,v}>0}\{m_{u,v}\} \geq \rho \cdot N - 1$. (Recall that the maximum degree in G' is at most N, whereas the average degree is at least M, since each vertex in G' has degree at least M.)

[7]For any $E' \subseteq E$, we have

$$\begin{aligned}\sum_{\{u,v\}\in E'} m_{u,v} &\geq \sum_{\{u,v\}\in E'} (w_{u,v} \cdot N - 1) \\ &= \left(\sum_{\{u,v\}\in E'} w_{u,v}\right) \cdot N - |E'|.\end{aligned}$$

The auxiliary graph $G'' = (V'', E'')$. As in [8], we wish to transform $G' = (V', E')$ to a graph with a maximum degree that is only a constant factor larger than its average degree, while preserving the relative distance from the set of bipartite graphs. Denoting the degree of vertex v (in G') by $d'_v = \sum_u m_{v,u}$ and letting $d' = \sum_{v \in V'} d'_v/|V'| = 2|E'|/|V'|$ denote the average degree of G', we replace each verterx in G' by a cloud, denoted C_v, of $c_v \stackrel{\rm def}{=} \lceil d'_v/d' \rceil$ vertices. Hence, $V'' = \bigcup_{v \in V'} C_v$ and $|V''| = \sum_{v \in V'} |C_v| < \sum_{v \in V'}((d'_v/d') + 1) = 2|V'|$. (Unlike in [8], we don't augment the graph with bipartite expanders of degree d' between C_v and an auxiliary set of $|C_v|$ vertices, but rather capitalize on the fact that the number of parallel edges between clouds is much larger than the product of the size of the cluds; that is, $m_{u,v} = \omega(|C_u| \cdot |C_v|/\epsilon)$ for every $u, v \in V'$ such that $m_{u,v} > 0$.) Next, for each $u, v \in E'$ such that $m_{u,v} > 0$, we partition the $m_{u,v}$ parallel edges that connect u and v (almost) equally among the pairs of vertices in $C_u \times C_v$. That is, for every $\langle u, i \rangle \in C_u$ and $\langle v, j \rangle \in C_v$, we connect $\langle u, i \rangle$ and $\langle v, j \rangle$ by $m^{i,j}_{u,v}$ parallel edges, where $m^{i,j}_{u,v} \in \{\lfloor m_{u,v}/|C_u \times C_v| \rfloor, \lceil m_{u,v}/|C_u \times C_v| \rceil\}$.

The key observation is that a random walk on G' is closely related to a random walk on G'' in the sense that the walk on G' moves from u to v with (almost) the same probability that the walk on G'' moves from a vertex in C_u to a vertex in C_v. This is the case since for every $i \in [c_u]$ it holds that $\frac{m_{u,v}}{\sum_w m_{u,w}} \approx \frac{m^{i,\cdot}_{u,v}}{\sum_w m^{i,\cdot}_{u,w}}$, where $m^{i,\cdot}_{u,w} = \sum_{j \in [c_w]} m^{i,j}_{u,w}$. The approximation holds because $m^{i,\cdot}_{u,w} \approx m_{u,v}/c_u$ which is due to $c_u \leq N/M \ll M$, which in turn is due to $M \geq \rho N - 1$ and $N = \omega(1/\rho^2)$. Actually, to get an approximation factor of $1 \pm \rho$, we need $N/M \ll \rho \cdot M$, which holds when $N = \omega(1/\rho^3)$. (Note that $\frac{m_{u,v}}{\sum_w m_{u,w}} \approx \frac{c_v \cdot m^{i,j}_{u,v}}{\sum_w m^{i,\cdot}_{u,w}}$, holds if $(N/M)^2 \ll M$, but we don't use this fact.)

Indeed, $|E''| = |E'|$, and so the average degree (which is $2|E''|/|V''| < 2|E''|/2|V'| = d'/2$) is at least a third of the maximum degree (which is at most $\max_{\langle v,j \rangle \in V''}\{\sum_{\langle u,i \rangle \in V''} \lceil m_{u,v}/c_u c_v \rceil\} \leq d' + (d/\rho) < 1.5d'$, since $d' \geq M \geq \rho \cdot N - 1 = \omega(d/\rho)$).[8]

We conclude that one can employ the tester of [7] to G'', since the analysis in [7] presumes a constant ratio between the maximum and the average degrees. Of course, we have to guaranteed that the distance of G to being bipartite is reflected by the distance of G'' to being bipartite. This follows by combining Claims 2.1.1 and 2.1.2.

Claim 2.1.2 (number of violating edges in G' versus G'): *If at least Δ edges must be removed from G' in order to make it bipartite, then at least $\Delta - |V''|$ edges must be removed from G'' in order to make it bipartite.*

The slackness is due to the fact that $m^{i,j}_{u,v}$ equals $\frac{m_{u,v}}{|C_u \times C_u|}$ up to at most one unit, rather than being equal to it. Loosely speaking, this means that 2-partitions of G'' with fewer inter-part edges are obtained by placing all vertices of each cloud on the same side. Intuitively, this holds because the edges between each pair of clouds are partitioned equally between the corresponding pairs of vertices, and so one does not benefit by treating vertices of the same cloud differently (i.e., placing them on different sides of the 2-partition).

Proof: For every 2-partition $\chi'' : V'' \to \{1,2\}$, we present a 2-partition $\chi' : V' \to \{1,2\}$ such that the number of χ'-monochromatic edges in G' is at most $|V''|$ units larger than the number of χ''-monochromatic edges in G''; that is,

$$\sum_{u,v \in V' : \chi'(u) = \chi'(v)} m_{u,v} \leq |V''| + \sum_{\langle u,i \rangle, \langle v,j \rangle \in V'' : \chi''(\langle u,i \rangle) = \chi''(\langle v,j \rangle)} m^{i,j}_{u,v}.$$

This is shown by using the probabilistic method. Specifically, fixing $\chi'' : V'' \to \{1,2\}$, we consider assigning each vertex of G' a color chosen at random according to the colors of the vertices in the

[8] To see the first inequality, note that $\sum_{\langle u,i \rangle \in V''} m_{u,v}/c_u c_v$ equals $\sum_{u \in V'} m_{u,v}/c_v = d'_v/c_v \leq d'$, since $c_v = \lceil d'_v/d' \rceil \geq d'_v/d'$. Hence, $\sum_{\langle u,i \rangle \in V''} \lceil m_{u,v}/c_u c_v \rceil$ is at most $d' + |V''| \leq d' + (d/\rho)$.

corresponding cloud; that is, $\chi'(v) = 1$ with probability $|\{i \in [c_v] : \chi''(\langle v, i\rangle) = 1\}|/|C_v|$, and $\chi'(v) = 2$ otherwise. Letting $X_{u,v}$ be a random variable indicating whether or not $\chi'(u) = \chi'(v)$ (i.e., $X_{u,v} = 1$ if $\chi'(u) = \chi'(v)$ and $X_{u,v} = 0$ otherwise), we have

$$\begin{aligned} \text{Exp}\left[\sum_{u,v\in V'} m_{u,v} \cdot X_{u,v}\right] &= \sum_{u,v\in V'} m_{u,v} \cdot \Pr[\chi'(u) = \chi'(v)] \\ &= \sum_{u,v\in V'} \frac{m_{u,v}}{|C_u \times C_u|} \cdot |\{(i,j) \in [c_u] \times [c_v] : \chi''(\langle u, i\rangle) = \chi''(\langle v, j\rangle)\}|. \end{aligned}$$

Recalling that the absolute difference between $m_{u,v}^{i,j}$ and $\frac{m_{u,v}}{|C_u \times C_u|}$ is at most one unit, it follows that the expected number of χ'-monochromatic edges in G' differs from the number of χ''-monochromatic edges in G'' by at most $|V''|$ units. ■

The actual tester. Recall that, when given oracle access to an n-vertex graph (and proximity parameter ϵ), the tester of [7] selects uniformly $O(1/\epsilon)$ (start) vertices, and starts $\text{poly}(1/\epsilon) \cdot \widetilde{O}(\sqrt{n})$ random ℓ-step walks from each vertex, where $\ell = \text{poly}(\epsilon^{-1} \log n)$ and each step in the random walk moves uniformly to one of the neighbors of the current vertex. The tester accepts if and only if the explored subgraph is bipartite (i.e., contains no odd-length cycles). When seeking to test G under the vertex-distribution $\mathcal{D}$, we shall emulate a testing of G'' as follows.

- The $O(1/\epsilon)$ start vertices will be selected according to the degrees in G' (and in G''), which in turn reflect the weight of the incident edges. Hence, we wish to select v with probability that is proportional to $\sum_{u\in\Gamma(v)}(\mathcal{D}(u) + \mathcal{D}(v))$, where $\Gamma(v)$ denotes the set of neighbors of v in G. We do so by employing "rejection sampling" as follows. First, we obtain a sample of $\mathcal{D}$; that is, $s \leftarrow \mathcal{D}$. Next, we output s with probability $|\Gamma(s)|/2d$, and halt with no output with probability $(2d - 2|\Gamma(s)|)/2d$, Otherwise (i.e., with probability $|\Gamma(s)|/2d$), we output a uniformly selected neighbor of s (i.e., each neighbor of s is output with probability $1/2d$).

  Hence, the probability that v is output (in a single trial) equals $p(v) \stackrel{\text{def}}{=} \mathcal{D}(v) \cdot \frac{|\Gamma(v)|}{2d} + \sum_{u\in\Gamma(v)} \mathcal{D}(u) \cdot \frac{1}{2d}$, where the first term is due to selecting $s = v$ and the second term is due to selecting $s \in \Gamma(v)$. Note that $p(v) = \sum_{u\in\Gamma(v)}(\mathcal{D}(v) + \mathcal{D}(u))/2d$, whereas the degree of v in G' is proportional to $\sum_{u\in\Gamma(v)} m_{u,v} \approx \sum_{u\in\Gamma(v)}(\mathcal{D}(u) + \mathcal{D}(v)) \cdot N$. (Indeed, we repeat trying till some vertex is output, while noting that each trial succeeds at least with probability $1/d$.)

- A step in a random walk is made by selecting a neighbor of the current vertex with probability that is proportional to the weight of the edge leading to it. Specifically, when we reside at $v \in V$, we move to the neighbor u with probability proportional to $\mathcal{D}(v) + \mathcal{D}(u)$.

Note that for implementing the foregoing actions it suffices to be able to sample $\mathcal{D}$, and have oracle access to the evaluation of $\mathcal{D}$ (and to the incidence function of G). Actually, the evaluator of $\mathcal{D}$ can be replaced by an oracle that answers (w_1, w_2) with $\mathcal{D}(w_1)/\mathcal{D}(w_2)$, provided $\mathcal{D}(w_2) > 0$ (and a special symbol otherwise).

The general case. Recall that the foregoing analysis presumes that for each $v \in V$ either $\mathcal{D}(v) = 0$ or $\mathcal{D}(v) \geq \rho$. Using $\rho = \Theta(\epsilon/n)$, we now reduce the general case of $\mathcal{D}$ that has $\epsilon/4$-effective support size n to the foregoing case. We first note that, by the following Claim 2.1.3, the vertex-distribution $\mathcal{D}$ is $\epsilon/2$-close to a vertex-distribution $\mathcal{D}'$ that satisfiers the foregoing condition with $\rho = \epsilon/4n$. Hence, if G is ϵ-far from being bipartite under $\mathcal{D}$, then it is 0.5ϵ-far from being bipartite under $\mathcal{D}'$, and we can test G by providing the tester with oracle access to $\mathcal{D}'$ (i.e., to $\texttt{samp}_{\mathcal{D}}{}'$ and $\texttt{eval}_{\mathcal{D}'}$) and setting the proximity parameter to $\epsilon/2$. Actually, it suffices to provide the foregoing tester with sampling access to $\mathcal{D}'$, which can be emulated by "rejection sampling" via $\mathcal{D}$, and with an evaluator of the ratios between the $\mathcal{D}'$-values (which, in turn, equal the corresponding the ratios between the $\mathcal{D}$-values).

Claim 2.1.3 (effective support size and minimal weight): *Suppose that $\mathcal{D}$ has an η-effective support of size n. Then, $\mathcal{D}$ is 2η-close to a distribution $\mathcal{D}'$ that satisfies $\min_{e:\mathcal{D}'(e)>0}\{\mathcal{D}'(e)\} > \eta/n$. Furthermore, $\mathcal{D}'(e) > 0$ if and only if $\mathcal{D}(e) > \eta/n$, and if $\mathcal{D}'(e) > 0$ then $\frac{\mathcal{D}'(e')}{\mathcal{D}'(e)} = \frac{\mathcal{D}(e')}{\mathcal{D}(e)}$ for every e'.*

Combining the foregoing analysis with Claim 2.1.3, Theorem 2.1 follows.

Proof: We may assume, without loss of generality, that the support of $\mathcal{D}$ has $s > n$ elements, and arrange these elements according to their $\mathcal{D}$-value; specifically, let $e_1, ..., e_s$ such that $\mathcal{D}(e_i) \geq \mathcal{D}(e_{i+1}) > 0$ for every $i \in [s-1]$. Then, $\sum_{i=n+1}^{s} \mathcal{D}(e_i) \leq \eta$, because the distance of $\mathcal{D}$ from the class of distributions of support size at most n is at least $\sum_{i=n+1}^{s} \mathcal{D}(e_i)$. Letting $P(e) = \mathcal{D}(e)$ if $\mathcal{D}(e) > \eta/n$ and $P(e) = 0$, we define $\mathcal{D}'(e) = \frac{P(e)}{\sum_{i\in[s]} P(e_i)}$, and notice that

$$\begin{aligned}\sum_{e:P(e)=0} \mathcal{D}(e) &\leq \sum_{i\in[n]:\mathcal{D}(e_i)\leq\eta/n} \mathcal{D}(e_i) + \sum_{i\in[s]\setminus[n]} \mathcal{D}(e_i) \\ &\leq 2\eta.\end{aligned}$$

Noting that $\mathcal{D}'(e) \geq \mathcal{D}(e)$ for every $e \notin P^{-1}(0)$, the main claim follows. The furthermore claim follows by the definition of $\mathcal{D}'$. ■

2.2 Approximating the effective support size of $\mathcal{D}$

The foregoing Bipartite tester, which establishes Theorem 2.1, presupposes that the tester is given an upper bound on the minimum $\epsilon/4$-effective support size of $\mathcal{D}$ as auxiliary input. Proving Theorem 1.4 requires getting rid of that auxiliary input, or rather approximating it when using oracle access to $\mathcal{D}$. Indeed, we shall show that given sampling and evaluation oracles to $\mathcal{D}$, it is relatively easy to approximate its effective support size. (In contrast, as commented in the introduction, obtaining such an approximation while using only samples of $\mathcal{D}$ is too expensive for our application.)

Note that the notion of approximating the effective support size is not robust in the sense that, for every $\eta < \eta'$, a distribution $\mathcal{D}$ may be have a minimal η-effective support size that is much larger than its minimal η'-effective support size (e.g., consider $\mathcal{D}$ that assigns a total probability mass of $1-\eta$ to very few elements and is otherwise uniform on a huge set). Hence, on input η, our goal is to approximate the $\Theta(\eta)$-effective support size of $\mathcal{D}$; say, output a number between the minimal 2η-effective support size of $\mathcal{D}$ and its minimal η-effective support size (or a good approximation of such a number).

A simple approximation of the effective support size. We first present a simple algorithm for obtaining a rather rough (but sufficiently good for our purposes) approximation. Given an "effectiveness" parameter η, we proceed in iterations such that in the i^{th} iteration we *take a sample of $m = O(\eta^{-1} \log i)$ elements of $\mathcal{D}$, and halt outputting $2^i/\eta$ if the number of samples that have $\mathcal{D}$-value below $\eta/2^i$ is at most $3\eta \cdot m$.* Observer that (in iteration i), with probability at least $1 - 0.01/i^2$, the sample approximates the total weight of the "light elements" (i.e., elements having $\mathcal{D}$-value below $\eta/2^i$) in the sense that if $\Pr_{v\leftarrow\mathcal{D}}[\mathcal{D}(v) < \eta/2^i] < 2\eta$, (resp., if $\Pr_{v\leftarrow\mathcal{D}}[\mathcal{D}(v) < \eta/2^i] \geq 4\eta$), then the number of samples that have $\mathcal{D}$-value below $\eta/2^i$ is at most $3\eta \cdot m$ (resp., is greater than $3\eta \cdot m$).

Now, letting n be an upper bound on the η-effective support size of $\mathcal{D}$ (and assuming for simplicity that n is a power of two), observe that, with high (constant) probability, we halt by iteration $i^* = \log_2 n$, because $\Pr_{v\leftarrow\mathcal{D}}[\mathcal{D}(v) < \eta/2^{i^*}] < n \cdot \eta/2^{i^*} + \eta = 2\eta$, where the first (resp., second) term is due to elements that are (resp., are not) in the effective support of $\mathcal{D}$. Hence, with high (constant) probability, the algorithm outputs a value that is at most n/η. (Actually, if we reach iteration i^*, we halt in it with probability at least $1 - 0.01/(i^*)^2$; it follows that with probability at $1 - 0.01/\log^2 n$, the algorithm outputs a value that is at most n/η.)

On the other hand, assuming that the minimal 4η-effective support size of $\mathcal{D}$ is at least n' (and assuming that n' is also a power of two), with high (constant) probability, we do not halt before iteration $i^+ = \log_2(\eta \cdot n')$, because otherwise $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < \eta/2^{i^+-1}] < 4\eta$, which implies that $\mathcal{D}$ has 4η-effective support size at most $n'/2$ (since $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < 2/n'] < 4\eta$ implies that $\mathcal{D}$ is 4η-close to a distribution of support size at most $n'/2$). Hence, with high (constant) probability (i.e., with probability at least $1 - \sum_{i<i^+} 0.01/i^2 > 0.98$), the algorithm outputs a value that is at least n'.

It follows that, with high (constant) probability, the algorithm outputs a number that lies between n' and n/η (i.e., between the half the minimal 4η-effective support size of $\mathcal{D}$ and $2/\eta$ times its minimal η-effective support size). Furthermore, with high (constant) probability, this algorithm runs for $\sum_{i \le i^*} O(\eta^{-1} \cdot \log i) = \widetilde{O}(\log n)/\eta$ steps (and its expected number of steps can be similarly bounded). Combining this result with Theorem 2.1, we essentially infer Theorem 1.4 (except that $\epsilon/5$ is replaced by $\epsilon/16$).

Obtaining better approximations of the effective support size. For starters, we present a tighter analysis of (a minor variant of) the foregoing algorithm. Specifically, for any constant $\beta > 1$, in the i^{th} iteration, we *take a sample of* $m = O(\eta^{-1} \log i)$ *elements of* $\mathcal{D}$*, and halt outputting* $2^i/\eta$ *if the number of samples that have* $\mathcal{D}$*-value below* $(\beta - 1) \cdot \eta/2^i$ *is at most* $\beta^2 \cdot \eta \cdot m$. In analyzing the probability that this algorithm halts by iteration $i^* = \log_2 n$, we use the fact that $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < (\beta - 1) \cdot \eta/2^{i^*}] < n \cdot (\beta - 1) \cdot \eta/2^{i^*} + \eta = \beta \cdot \eta$, whereas in analyzing the probability that the algorithm does halts before iteration $i^+ = \log_2(\eta \cdot n')$ implies $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < (\beta - 1) \cdot \eta/2^{i^+-1}] < \beta^3 \cdot \eta$ (where here n' is the minimal $\beta^3\eta$-effective support size).[9] It follows that, with high (constant) probability, the algorithm outputs a number, denoted $\widetilde{n}$, that lies between half the minimal $\beta^3 \cdot \eta$-effective support size of $\mathcal{D}$ and $2/\eta$ times its minimal η-effective support size.

To obtain an even better approximation of the effective support size, we use the rough estimate $\widetilde{n}$ obtained above in order to approximate the number of elements that have $\mathcal{D}$-value approximately $\beta^{i-0.5}$ for every $i \in [O(\log \widetilde{n}/\eta)]$. Indeed, our first step is ignoring elements having $\mathcal{D}$-value below $\eta/\widetilde{n}$ or so. Specifically, setting $\eta' = \beta^3 \cdot \eta$, recall that if $\mathcal{D}$ has an η'-effective support of size $\widetilde{n}$, then $\mathcal{D}(H) \ge 1 - \beta\eta'$ for $H \stackrel{\text{def}}{=} \{v : \mathcal{D}(v) \ge (\beta - 1) \cdot \eta'/\widetilde{n}\}$ (see prior paragraph, while replacing i^* by $\log_2 \widetilde{n}$).[10] Hence, assuming that $\mathcal{D}(H) \le 1 - \eta$ and letting $\eta'' = \beta \cdot \eta' = \beta^4 \cdot \eta$, it holds that $|H|$ lies between the minimal η''-effective support size of $\mathcal{D}$ and its minimal η-effective support size, and so providing a good approximation of the "effective size" of H will do. (In the case that $\mathcal{D}(H) > 1 - \eta$ additional steps will be needed.)

To (effectively) approximate $|H|$, we let $W_i = \{v : \beta^{i-1} \le \mathcal{D}(v) < \beta^i\}$, and observe that it suffice the approximate $\mathcal{D}(W_i)$ for $i = 1, ..., \ell$, where $\ell \stackrel{\text{def}}{=} \log_\beta(\widetilde{n}/(\beta - 1) \cdot \eta') = O((\beta - 1)^{-1} \cdot \log(\widetilde{n}/\eta))$. Actually, letting $I = \{i \in [\ell] : \mathcal{D}(W_i) \ge (\beta - 1)\eta''/\ell\}$, it suffices to approximate $\mathcal{D}(W_i)$ for every $i \in I$, which yields approximations of the corresponding $|W_i|$'s (using $|W_i| \approx \mathcal{D}(W_i)/\beta^{i-0.5}$). That is, we do not actaully approximate $|H|$ but rather approximate $H' \stackrel{\text{def}}{=} \bigcup_{i \in I} W_i \subseteq H$, which capitalizing on $\mathcal{D}(H \setminus H') \le (\beta - 1) \cdot \eta''$. Hence, we will approximate the minimal η'''-support size for some $\eta \le \eta''' \le \beta\eta'' = \beta^5\eta$. Specifically, for each $i \in [\ell]$, using a sample of $O(t\ell/(\beta - 1)^2 \cdot \eta')$ samples, we obtain (with probability $1 - 2^{-t}$) a β-factor approximation of $\mathcal{D}(W_i)$ for each $i \in I$, which yields a

[9] Here we use the fact that (in iteration i), with probability at least $1 - 0.01/i^2$, the sample approximates the total weight of the "light elements" (i.e., elements having $\mathcal{D}$-value below $\eta/2^i$) in the sense that if $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < (\beta - 1) \cdot \eta/2^i] < \beta \cdot \eta$, (resp., if $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < (\beta - 1) \cdot \eta/2^i] \ge \beta^3 \cdot \eta$), then the number of samples that have $\mathcal{D}$-value below $(\beta - 1) \cdot \eta/2^i$ is at most $\beta^2 \cdot \eta \cdot m$ (resp., is greater than $\beta^2 \cdot \eta \cdot m$).

[10] In other words, observe that $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < (\beta - 1) \cdot \eta'/\widetilde{n}] < \widetilde{n} \cdot (\beta - 1) \cdot \eta'/\widetilde{n} + \eta' = \beta \cdot \eta'$, where the first (resp., second) term is due to elements that are (resp., are not) in the effective support of $\mathcal{D}$.

β^2-factor approximation of $|W_i|$ (since $|W_i| \in [\beta^{i-1}\mathcal{D}(W_i), \beta^i \cdot \mathcal{D}(W_i))$). Note that the rough estimate of the effective support size of $\mathcal{D}$ (i.e., $\widetilde{n}$) is only used in order to determine ℓ.

Recall that we have assumed that $\mathcal{D}(H) \leq 1-\eta$, whereas this is not necessarily the case. Needless to say, we can easily estimate $1-\mathcal{D}(H)$ up to any desired constant factor (using $O(1/\eta)$ samples of $\mathcal{D}$). In case we are quite sure that $\mathcal{D}(H) > 1-\eta$ (which will happen if $\mathcal{D}(H) > 1-\beta\eta$ but not if $\mathcal{D}(H) < 1-\beta^{-1}\eta$), we can reduce the estimate obtained for $|H|$ by disposing an adequate weight of H; that is, we dispose as many as the the lightest elements as possible till reaching a subset of H' that has $\mathcal{D}$-value that is most likely to be below $1-\eta$. Note that the foregoing process is performed without making any samples or queries; it is solely based on the estimated values of $\mathcal{D}(W_i)$ for $i \in I$ already obtained. Hence, we get.

Theorem 2.2 (approximating the effective support size): *There exists an oracle machine M that, on input η and $\beta > 1$, satisfies the following condition for every distribution $\mathcal{D}$. Given oracle access to $\mathtt{samp}_{\mathcal{D}}$ and $\mathtt{eval}_{\mathcal{D}}$, with probability at least $2/3$, the machine outputs a number n after making* $\mathrm{poly}(1/(1-\beta)) \cdot \widetilde{O}(\eta^{-1}\log n)$ *steps such that n is at least the minimal $\beta^5 \cdot \eta$-effective support size of $\mathcal{D}$ and is at most β^2 times the minimal $\beta^{-1} \cdot \eta$-effective support size of $\mathcal{D}$. Furthermore, the expected number of steps performed by M is* $\mathrm{poly}(1/(1-\beta)) \cdot \widetilde{O}(\eta^{-1}\log n)$.

Needless to say, by a change in parameters we can make n lie between the minimal $\beta \cdot \eta$-effective support size of $\mathcal{D}$ and β times its minimal η-effective support size. Combining Theorems 2.1 and 2.2, we infer Theorem 1.4.

The approximation algorithm of Theorem 2.2 provides a rather good approximation of the effective support size, but its complexity depends (mildly) on the effective support size. We wonder whether it is possible to obtain a meaningful approximation of the effective support size within complexity that is independent of it.

3 The Cycle-freeness Tester

Following [2], we reduce testing cycle-freeness to testing bipartiteness, where in both cases we refer to VDF testing in the bounded degree graph model. While the reduction is identical to the one presented in [2] (for the standard bounded-degree model), our analysis presented in Lemma 3.1 addresses issues that arise only in the VDF model.

Specifically, we use the presentation provided in [2, Sec. 8.1] rather than the one provided in [2, Sec. 3]. The pivot of this presentation is the following *generalization of 2-colorability* in which edges of the graph are labeled by either `eq` or `neq`. That is, an instance of this problem is a graph $G=(V,E)$ along with a labeling $\tau : E \to \{\mathtt{eq}, \mathtt{neq}\}$. We say that $\chi : V \to \{0,1\}$ is a legal 2-coloring of this instance if for every $\{u,v\} \in E$ it holds that $\chi(u) = \chi(v)$ if and only if $\tau(\{u,v\}) = \mathtt{eq}$. That is, a legal 2-coloring (of the vertices) is one in which every two vertices that are connected by an edge labeled `eq` (resp., `neq`) are assigned the same color (resp., opposite colors).

Note that the standard notion of 2-colorability corresponds to the case in which all edges are labeled `neq`. We first observe that the Bipartite tester provided in Section 2 can be extended to test this generalization of 2-colorability. This is the case because the (one-sided error) Bipartite tester of [7] (as well as the ones of [8] and [5]) can be extended to test this generalization of 2-colorability.

Next, as in [2, Sec. 8.1], we randomly reduce testing cycle-freeness of a graph $G=(V,E)$ to testing generalized 2-coloring of the instance obtained by coupling G with a uniformly selected labeling $\tau : E \to \{\mathtt{eq}, \mathtt{neq}\}$, which may be selected on the fly (i.e., whenever we encounter a new edge, we assign it a random label). Clearly, if G is cycle-free, then, for any choice of $\tau : E \to \{\mathtt{eq}, \mathtt{neq}\}$, the instance (G,τ) has a legal 2-coloring. We conjecture that, like in [2, Lem. 3.1], it holds that if G is ϵ-far from being cycle-free (under the distribution $\mathcal{D}$), then in expectation the random instance (G,τ) is $\Omega(\epsilon)$-far from having a legal 2-coloring (under the distribution $\mathcal{D}$). We only prove a weaker result, which suffices for our application (since $\widetilde{O}(\sqrt{n}) \cdot \text{poly}(\Omega(\epsilon/\log n)^{-1}) = \widetilde{O}(\sqrt{n}) \cdot \text{poly}(\epsilon^{-1})$).

Lemma 3.1 (analysis of the randomized reduction): *Let $G = (V, E)$ be a simple graph* (i.e., G has neither parallel edges nor self-loops)[11] *and $\mathcal{D}$ be a distribution on V. If G is ϵ-far from being cycle-free according to distribution $\mathcal{D}$, then, with probability at least $\Omega(1)$ over the random choice of $\tau : E \to \{\texttt{eq}, \texttt{neq}\}$, the instance (G, τ) is $\Omega(\epsilon/\log|V|)$-far from having a 2-legal coloring.*

Hence, we reduce testing cycle-freeness of n-vertex graphs with respect to the proximity parameter ϵ and vertex-distribution $\mathcal{D}$ to testing generalized 2-coloring of n-vertex graphs with respect to the proximity parameter $\Omega(\epsilon/\log n)$ (and vertex-distribution $\mathcal{D}$). An analogous assertion holds for testing cycle-freeneess with respect to arbitrary graphs and distribution $\mathcal{D}$ that have $\epsilon/5$-effective support size n. Hence, combining Lemma 3.1 and Theorem 1.4, we obtain Theorem 1.5.

Proof: Maintaining $\mathcal{D}$ unchanged, we first omit from G all edges that have probability weight smaller than $\epsilon/d|V|$ (i.e., we omit the edge $\{u, v\}$ if $2 \cdot \frac{\mathcal{D}(u)+\mathcal{D}(v)}{d} < \epsilon/d|V|$). Denoting the resulting graph by $G' = (V, E')$, observe that G' is $\epsilon/2$-far from being cycle-free according to distribution $\mathcal{D}$. Next, we consider a spanning forest of maximal weight of G', denote its edges by F', and note that the weight of the edges in $E' \setminus F'$ is at least $\epsilon/4$, because G' can be made cycle-free by omitting the edges $E' \setminus F'$. We partition the edges $E' \setminus F'$ to buckets according to their weight, letting the bucket $B_i \subseteq E' \setminus F'$ contain the edges of weight in $(2^{-i}, 2^{-i+1}]$. Letting $\ell = \lfloor \log_2(d|V|/\epsilon) \rfloor + 1$, we observe that $E' \setminus F' = \bigcup_{i \in [\ell]} B_i$, and it follows that there exists $i \in [\ell]$ such that the total weight of the edges in B_i is at least $\epsilon/4\ell$.

Fixing such an $i \in [\ell]$ and focusing on the graph $G'' = (V, F' \cup B_i)$, we consider the two-connected components of G'' (i.e., the maximal subgraphs of G'' in which every two vertices are connected by at least two edge-disjoint paths). We observe that *edges that connect such components must be in F'*, by considering two cases.

> In the first case, these components are not connected in G', and in this case there are also not connected in G'' (and the claim holds vacuously). Otherwise (which is the second case), these components are connected in G' and so there must be a path of edges of F' that connects (a pair of vertices in) them, since F' spans the connected components of G'. In fact, in this case, any pair of vertices in these two components is connected by a path of edges in F'. But, then having an edge of B_i that connects these components yields a contradiction to their definition (as two-connected components of G''), since it yields two enge-disjoint paths between the endpoints of this edge (one being the edge itself and the other being a path of edges in F').

Lastly, we omit the edges of F' that connect different two-connected components, and obtain a graph G''' in which each connected component is two-connected.

We next claim that each edge in $G''' = (V''', E''')$ has weight at least 2^{-i}. Since $E''' \subseteq F' \cup B_i$, we have to establish this claim only for edges in $E''' \cap F'$. We first observe that the edges in $E''' \cap F'$ induce a spanning tree of each connected component of G''', because otherwise we reach a contradiction to their definition (as two-connected components of G'').[12] Suppose towards the contradiction that one of the edges of such a spanning tree has weight smaller than 2^{-i}. Then, omitting this edge and adding an edge of B_i that connects the two resulting sub-trees, where such an edge must exist by two-connectivity of the component, we obtain a spanning tree of this component that has a larger weight than the original

[11] Alternatively, we consider a pair parallel edges (resp., a self-loop) as constituting a cycle.

[12] Specifically, vertices in a component of G''' that are not connected by edges of F' must be connected in G'' by an edge-disjoint path of edges of F' that contain vertices that are not in this component. But this contradicts the hypothesis that this (two-connected) component of G''' is a two-connected component of G'' (i.e., it contradicts its maximality in G'').

one. This yields a spanning forest of G' that has weight larger than the forest F', in contradiction to the definition of F'.

Finally, we apply [2, Lem. 3.1] to G''', while noting that $E''' \setminus F' = B_i$ and $|B_i| \geq \frac{\epsilon/4\ell}{2^{-i+1}} = \Omega(2^i \cdot \epsilon/\ell)$, it follows that (with probability at least $\Omega(1)$ over the choice of τ) at least $\Omega(|B_i|)$ edges must be omitted from G''' such that the resulting instance $(\cdot, \tau)$ has a legal 2-coloring. Recalling that each edge in G''' have weight at least 2^{-i}, it follows that the weight of edges that must be omitted from G''' in order to yield an instance $(\cdot, \tau)$ that has a legal 2-coloring is at least $\Omega(|B_i| \cdot 2^{-i}) = \Omega(\epsilon/\ell)$. The same holds with respect to G, since G''' is a subgraph of G, and the claim follows. ■

Acknowledgments

This project was partially supported by the Israel Science Foundation (grant No. 1146/18), and has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819702).

References

[1] E. Blais, C.L. Canonne, and T. Gur. Distribution Testing Lower Bounds via Reductions from Communication Complexity. In *32nd Computational Complexity Conference*, pages 28:1–28:40, 2017.

[2] A. Czumaj, O. Goldreich, D. Ron, C. Seshadhri, A. Shapira, and C. Sohler. Finding cycles and trees in sublinear time. *RS&A*, Vol. 45(2), pages 139–184, 2014.

[3] O. Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

[4] O. Goldreich. Testing Graphs in Vertex-Distribution-Free Models. *ECCC*, TR18-171, 2018. (See Revision Nr 1, March 2019.)

[5] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, pages 653–750, July 1998.

[6] O. Goldreich and D. Ron. Property Testing in Bounded Degree Graphs. *Algorithmica*, Vol. 32 (2), pages 302–343, 2002.

[7] O. Goldreich and D. Ron. A Sublinear Bipartitness Tester for Bounded Degree Graphs. *Combinatorica*, Vol. 19 (3), pages 335–373, 1999.

[8] T. Kaufman, M. Krivelevich, and D. Ron. Tight Bounds for Testing Bipartiteness in General Graphs. *SIAM Journal on Computing*, Vol. 33 (6), pages 1441–1483, 2004.

[9] S. Raskhodnikova, D. Ron, A. Shpilka, and A. Smith. Strong Lower Bounds for Approximating Distribution Support Size and the Distinct Elements Problem. *SICOMP*, Vol. 39 (3), pages 813–842, 2009.