Testing Bipartiteness in an Augmented VDF Bounded-Degree Graph Model

Oded Goldreich*

October 13, 2021

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.

The focus of the aforementioned work was on testers, called strong, whose query complexity depends only on the proximity parameter ϵ, and such testers were studied both in the dense graph model and in the bounded-degree graph model. 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 Bipartiteness and Cycle-freeness, in this augmented model, having complexity that is almost-linear in the square root of the "effective support size" of $\mathcal{D}$.

The foregoing testers use an algorithm for approximating the effective support size of $\mathcal{D}$, which triggered a general study of this computational problem, which we conducted in a subsequent work (*ECCC*, TR19-088, 2019).

A preliminary version of this work was posted in May 2019 on `arXiv` (as report number 1905.03070). Unfortunately, the original presentation contained several errors, which are corrected in the current version.[1] In general, the presentation in the current version is more accurate and more detailed.

1 Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., a relatively 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

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

[1]Most importantly, the graph G'' used in the analysis of the Bipartitness tester of [8] was wrongly defined before. Likewise, the claim that one could have just applied the tester of [9] to the graph G' was incorrect.

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 [6] and reviewed in [3, Chap. 8]) and the bounded-degree graph model (introduced in [7] and reviewed in [3, Chap. 9]). In both models, it was postulated that the tester can sample the vertex-set uniformly at random[2] (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 appropriate, and asked what happens if the tester obtains 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 reasonable to define the distance between graphs with respect to the distribution $\mathcal{D}$, since this is the distribution that the application uses. (See further discussion in Section 1.4.)

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 studied 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 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

[2] 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.

$g(v,i)=u$ 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 $\mathcal{D}:V\to[0,1]$, 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 $\mathcal{D}:V\to[0,1]$, 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 distribution $\mathcal{D}$* (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 graph $G=(V,E)$ and distribution $\mathcal{D}$ 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 $|\{v:\mathcal{D}(v)=p\}|=|\{v:\mathcal{D}'(v)=p\}|$).[3] 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

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

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, where $\mathcal{D}$ is η-*close *to $\mathcal{D}'$ if their total variation distance is at most η. 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).[4] Furthermore, 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 exist 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 in [4, Thm. 3.3]. The derived one-sided error tester uses a sample of $O(q(\epsilon)^2)$ vertices and invokes the original tester while emulating a vertex-distribution that is uniform over this sample.[5] The analysis relies on the fact that, in the original VDF model, the tester's behavior on a fixed sample of vertices is oblivious of the distribution from which this sample is taken. This fact is no longer valid in the augmented model, since the tester may make evaluation queries.)

[4]Let $\mathcal{D}'$ be the foregoing distribution that has η-effective support of size n. Then, the typical case is that this value of η is not critical (w.r.t having η-effective support size n); that is, 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 $n+1$). To prove this claim, suppose that $\mathcal{D}'$ is η-close to a distribution $\mathcal{D}''$ of support size n, and consider the following 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 $n+1$ (i.e., the support of $\mathcal{D}'''$ is contained in the union of the support of $\mathcal{D}''$ and v) and is $(\eta-\mathcal{D}'(v))$-close to $\mathcal{D}'$. Hence, the claim follows with $\eta'=\eta-\mathcal{D}'(v)$.

[5]Actually, using a sample S of $O(q(\epsilon)^2)$ vertices, the derived tester emulates $\binom{|S|}{q(\epsilon)}$ different executions of the original tester, where in each execution the elements of a different $q(\epsilon)$-subset of S are provided as answers to the original tester's sampling requests. Note that these invocations make at most $|S|\cdot d^{q(\epsilon)}$ different incidence queries.

1.2 Our results

Our main result is testing Bipartiteness in the augmented VDF model within complexity that matches the complexity of the known tester in the standard (bounded-degree graph) model [8], which in turn is almost optimal [7].

Theorem 1.4 (testing Bipartiteness in the augmented VDF model): *For any constant d, Bipartiteness can be tested in the augmented VDF bounded-degree graph 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 $\mathcal{D}$ faced 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 $\mathcal{D}$, where by a rough approximation of the minimal η-effective support size we mean that the value $\widetilde{O}(n)$ is admissible if the minimal poly(η)-effective support size is n. 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 seems essential to make queries to $\texttt{eval}_{\mathcal{D}}$ in order to obtain such a rough approximation, since obtaining such an approximation by making queries only to $\texttt{samp}_{\mathcal{D}}$ has complexity $n^{1-o(1)}$ (cf. [10] and [5, Cor. 1.7]).[6] We mention that, using both types of queries to $\mathcal{D}$, we obtain a good (i.e., constant factor) 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 $\mathcal{D}$ faced 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.[7]

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 $\mathcal{D}$ faced 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. Ditto for testing Cycle-freeness.*

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.

[6] Specifically, for any $c : \mathbb{N} \to (0, 0.06)$, at least $\Omega(n^{1-c(n)})$ queries are necessary to distinguish an n-grained distribution of support size $n/11$ from an n-grained distribution with support size $n^{1-3c(n)^{1/2}}$, where a distribution is called n-grained if all probabilities are integer multiples of $1/n$.

[7] Indeed, obtaining a rough approximation of the effective support size (i.e., n) requires at least $n^{1-o(1)}$ queries to $\texttt{samp}_{\mathcal{D}}$, but this refers to a setting in which only $\texttt{samp}_{\mathcal{D}}$ queries are available. In contrast, the tester can also make incidence queries to the graph, and the combination of such queries with $\texttt{samp}_{\mathcal{D}}$ queries may potentially allow a more query-efficient approximation of the effective support size of $\mathcal{D}$ (i.e., n). Furthermore, as noted in [5, Prob. 1.8], the results of [10] do not rule out the possibility of approximating n to a factor of n^c when using $n^{0.5+c'}$ queries, for some $c, c' \in (0, 0.5)$.

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., [8]), 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 [8]. 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 $\mathcal{D}(v)+\mathcal{D}(w)$. Here is where we make use of the evaluation oracle $\texttt{eval}_{\mathcal{D}}$. (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 [8]) as a black-box. Towards this end, we view our VDF-model tester as emulating the tester of [8] on an auxiliary graph in which weighted edges are replaced by a proportional number of parallel edges, and the vertices are replaced by sets of vertices, called clouds, that are of size that is proportional to the sum of the weights of the edges incident at the original vertices.[8] The parallel edges are distributed uniformly between the relevant pairs of clouds, while recalling that the analysis of [8] holds also for (non-simple) graphs having parallel edges (see [9]). (We stress that this is a mental experiment performed in the analysis; the actual algorithm is essentially as outlined in the previous paragraph.)

We stress that the complexity of the tester of [8] is dominated by the number of the vertices in the graph, and hardly depends on the number of edges in it.[9] In our (VDF) context, the complexity of the adapted tester depends on the effective support size of the distribution $\mathcal{D}$.

The foregoing description presumes that we have a good upper bound on the support size of $\mathcal{D}$. Indeed, such an upper bound (along with oracle access to the auxiliary graph) suffices for emulating the tester of [8]. 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 $\mathcal{D}$.

Approximating the effective support size of $\mathcal{D}$. We observe that if n is the minimal integer such that at most an η fraction of the weight of $\mathcal{D}$ resides on elements of weight smaller than 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 by using a doubling procedure; that is, we output n if a sample of $m = O(\eta^{-1} \log n)$ hits less than $2\eta \cdot m$ elements of weight smaller than η/n, and double n otherwise. (We can 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 $i \in [\log(n/\eta) + O(1)]$. Even better approximations were obtained in our subsequent work [5].)

[8] Specifically, an edge $\{u, v\}$ of weight $w_{u,v}$ is replaced by $m_{u,v} \approx w_{u,v} \cdot N$ parallel edges, where N is a sufficiently large number (which is possibly larger than the original graph). The vertex v is replaced by a cloud C_v of size proportional to $\sum_u w_{u,v}$, and the $m_{u,v}$ parallel edges connecting u and v are distributed uniformly among the vertex pairs in $C_u \times C_v$.

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

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 the said spanning forest. Considering the two-connected components of the corresponding graph, we observe that the edges of the spanning forest yield maximal spanning trees of each of the two-connected components. Hence, the edges that should be omitted from each 2-connected component are all of approximately the same weight, and so lower-bounding their weight reduces to lower-bounding their number. At this point, we invoke [2, Lem 3.1] and are done.

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 $G, \mathtt{samp}_\mathcal{D}, \mathtt{eval}_\mathcal{D}$, runs in time $\widetilde{O}(\sqrt{n}) \cdot \mathrm{poly}(1/\epsilon)$ and constitutes a Bipartite tester* (of one-sided error) *as defined in Section 1.1. That is:*

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

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 [8] 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., of length $\mathrm{poly}(\log n)$) random walks from few (i.e., $O(1)$) randomly selected vertices, where the various notations ignore a polynomial dependence on $1/\epsilon$. In [8], 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 the distribution $\mathcal{D}$.

Wishing to use the analysis of [8] as a black-box, we present the foregoing tester as emulating the tester of [8] on an auxiliary graph that is derived in two steps: First, we replace the aforementioned weighted edges by a proportional number of parallel edges, where the weight of an edge $\{u,v\}$ equals $2 \cdot (\mathcal{D}(u) + \mathcal{D}(v))/d$. Next, we replace the vertices by sets of vertices, called clouds, that are of size that is proportional to the sum of the weights of the edges incident at the original vertices, and distribute the parallel edges uniformly among the pairs of vertices in the relevant pair of clouds. This yields a regular multi-graph (i.e., having parallel edges), yet *the analysis of [8] holds also for multi-graphs having* (many) *parallel edges* (see [9]). Furthermore, the complexity and analysis of the tester are oblivious of the degree of the vertices (provided the maximal degree is approximately equal the average degree).

The first step: The multi-graph G'. Fixing a vertex distribution $\mathcal{D}$, on input a graph $G = (V, E)$, proximity parameter ϵ and a value n (provided as an effective support size of $\mathcal{D}$), we consider the following 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 (see Claim 2.1.4). 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 $m_{u,v} \stackrel{\rm def}{=} \lfloor (\mathcal{D}(u) + \mathcal{D}(v)) \cdot N \rfloor \gg 1$ parallel edges between u and v in G', where $N = \mathrm{poly}(1/\epsilon\rho)$ will be determined later. We keep in V' only the non-isolated vertices; that is, $v \in V'$ if and only if $\sum_u m_{u,v} > 0$ (equiv., $\sum_{u:\{u,v\}\in E} (\mathcal{D}(u) + \mathcal{D}(v)) \geq \rho$), 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,

$$|V'| \leq d/\rho \text{ and } |E'| = 0.5 \cdot \sum_{u,v\in V'} m_{u,v} \leq d \cdot \sum_{v\in V'} \mathcal{D}(v) \cdot N = dN. \tag{2}$$

Next, we relate the distance of G from being bipartite to the number of edges that should be removed from G' in order to make it bipartite.

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 $0.5d \cdot \delta \cdot N - d|V'|$ edges must be removed from G' in order to make it bipartite.*

Indeed, if $N = \omega(|V'|/\delta)$, then $0.5d\delta \cdot N - d|V'| = 0.5d \cdot (\delta - o(\delta)) \cdot N$.

Proof: Removing a weighted edge of G is equivalent to removing the corresponding parallel edges of G'. Recall that an edge $\{u, v\} \in E$ of weight $w_{u,v} = 2\cdot(\mathcal{D}(u)+\mathcal{D}(v))/d$ yields $m_{u,v} = \lfloor(\mathcal{D}(u) + \mathcal{D}(v)) \cdot N\rfloor \geq (d/2) \cdot w_{u,v} \cdot N - 1$ parallel edges. The claim follows because, for any $R \subseteq \{\{u, v\} : E : w_{u,v} > 0\}$, we have

$$\begin{aligned}\sum_{\{u,v\}\in R} m_{u,v} &\geq \sum_{\{u,v\}\in R} ((d/2) \cdot w_{u,v} \cdot N - 1)\\ &= \frac{d}{2} \cdot \left(\sum_{\{u,v\}\in R} w_{u,v}\right) \cdot N - |R|\end{aligned}$$

whereas $|R| \leq d \cdot |V'|$. ■

As in [9], 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. This is done by replacing the vertices of G' by vertex-sets, called clouds, and distributing the edges incident at each vertex $v \in V'$ among the vertices of the cloud that replaces v (so that all vertices in this cloud have approximately the same degree). However, unlike in [9], we also wish the sizes of these clouds to reflect the degrees of the different vertices of G' (so that sampling vertices of G' according to their degree translates to sampling vertices uniformly in the new graph). (In contrast, in [9] the size of the cloud that replaces v is only upper-bounded by the ratio of the degree of v and the average degree; and no lower bound is provided.)[10] Here we use the fact that each vertex in G' has degree at least $\rho \cdot N - 1$ and at most N, whereas $N = \omega(1/\rho^2)$.

The auxiliary multi-graph $G'' = (V'', E'')$. Denoting the degree of vertex v (in G') by $d'_v = \sum_u m_{v,u}$ and letting $t = \Theta(1/\epsilon)$, we replace each vertex in G' by a cloud, denoted C_v, of $c_v \stackrel{\text{def}}{=} \lceil t \cdot d'_v/\rho N\rceil$ vertices. Hence, $V'' = \bigcup_{v\in V'} C_v$ and

$$\begin{aligned}|V''| = \sum_{v\in V'} |C_v| &= \sum_{v\in V'} \lceil t \cdot d'_v/\rho N\rceil && (3)\\ &< \sum_{v\in V'} \left(\frac{t \cdot d'_v}{\rho N} + 1\right)\\ &= 2t \cdot \frac{|E'|}{\rho N} + |V'| < 3td/\rho, && (4)\end{aligned}$$

since $|E'| \leq dN$ and $|V'| \leq d/\rho$. Unlike in [9], where the edges incident at v were distributed *arbitrarily* (and equally) among the vertices of C_v (and the focus was on graphs with no parallel edges), here we distribute the parallel edges connecting v and u equally among the vertex pairs in $C_v \times C_u$. (We can afford doing so because the number of parallel edges between clouds is much larger than the product of the sizes of the clouds (i.e.., $m_{u,v} = \omega(|C_u| \cdot |C_v|/\epsilon)$ for every $u, v \in V'$ such that $m_{u,v} > 0$).)[11] 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 $m^{i,j}_{u,v} \in \{\lfloor m_{u,v}/|C_u \times C_v|\rfloor, \lceil m_{u,v}/|C_u \times C_v|\rceil\}$. Hence, all vertices of C_v are equivalent in the sense that each of them has approximately the same number of neighbors in each other cloud (i.e.,

[10] On the other hand, the priority in [9] is increasing the number of vertices as little as possible. Hence, in [9], the size of the cloud replacing v is $\lceil d'_v/d'\rceil$, where d'_v is v's degree (in G') and d' is the average degree in G'. Needless to say, this does not yield clouds of size proportional to the vertex degree in case $d'_v \ll d'$.

[11] In addition, unlike in [9], we do not augment the graph with gadgets that guarantee that a 2-coloring of the graph must assign the same color to all vertices of a cloud (and similarly for 2-partitions that are close to proper 2-colorings).

$\sum_{j\in[c_v]} m_{u,v}^{i,j} \approx m_{u,v}/c_u$, for each $\langle u,i\rangle \in V''$ and $v \in V'$). Indeed, $|E''| = |E'|$ and the resulting graph is almost regular; that is, the degree of each vertex $\langle v,j\rangle \in V''$ equals

$$\begin{aligned} \sum_{\langle u,i\rangle\in V''} m_{u,v}^{i,j} &\approx \sum_{\langle u,i\rangle\in V''} \frac{m_{u,v}}{c_u c_v} && (5)\\ &= \sum_{u\in V'} \frac{m_{u,v}}{c_v} \\ &= \frac{d'_v}{\lceil t\cdot d'_v/\rho N\rceil} = (1\pm\Theta(1/t))\cdot\rho N/t, && (6)\end{aligned}$$

where the first approximation is up to an additive term of $|V''| = O(t/\rho) = o(\rho N/t^2)$, since $N = \omega(t^3/\rho^2)$. 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 probability (i.e., $\frac{m_{u,v}}{\sum_w m_{u,w}}$) that (almost) equals the probability that the walk on G'' moves from any fixed vertex in C_u to some vertex in C_v (i.e., $\frac{m_{\langle u,i\rangle,v}}{\sum_w m_{\langle u,i\rangle,w}}$, where $m_{\langle u,i\rangle,w} = \sum_{j\in[c_w]} m_{u,w}^{i,j}$). These probabilities are approximately equal because $m_{u,v}^{i,j} \approx m_{u,v}/c_u c_v$, which is due to $m_{u,v} \geq \rho N - 1$ and $c_w \leq (t\cdot d'_w/\rho N)+1 \leq (t/\rho)+1 \ll \sqrt{\rho N}$, where the last inequality is due to $N = \omega(t^2/\rho^3)$.

We conclude that one can employ the tester of [8] to G'', since the analysis in [8] presumes only a constant ratio between the maximum and the average degrees (cf. [9]). 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 Claim 2.1.1 with the following claim.

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 $\Delta - d\cdot|V'|$ edges must be removed from G'' in order to make it bipartite.*

The slackness is due to the fact that $m_{u,v}^{i,j}$ equals $\frac{m_{u,v}}{|C_u\times C_u|}$ up to at most one unit, rather than being equal to it. Loosely speaking, Claim 2.1.2 means that 2-partitions of G'' with a minimal number of edges on the same side 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 $d|V'|$ units larger than the number of χ''-monochromatic edges in G''; that is,

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

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 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), it holds that the expected number of χ'-monochromatic edges equals

$$\mathrm{Exp}\left[\sum_{u,v\in V'} m_{u,v}\cdot X_{u,v}\right] = \sum_{u,v\in V'} m_{u,v}\cdot \Pr[\chi'(u)=\chi'(v)]$$

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

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'' (i.e., $\sum_{\langle u,i\rangle,\langle v,j\rangle \in V'' : \chi''(\langle u,i\rangle) = \chi''(\langle v,j\rangle)} m_{u,v}^{i,j}$) by at most $|E| \leq d \cdot |V'|$ units. ■

The actual tester of G and the emulated testing of G''. Recall that, when given oracle access to an n-vertex graph (and proximity parameter ϵ), the tester of [8] 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, denoted T, accepts if and only if the explored subgraph is bipartite (i.e., contains no odd-length cycles). When seeking to test the graph $G = (V, E)$ under the vertex-distribution $\mathcal{D}$ (such that $\min_{v\in V : \mathcal{D}(v) > 0}\{\mathcal{D}(v)\} \geq \rho$), we essentially emulate a testing of the auxiliary graph $G'' = (V'', E'')$ as follows.

- We set $n = |V''| < 3t \cdot d/\rho$. We mimic an execution of T, on proximity parameter $\epsilon/2$ and size parameter n, while emulating the actions on G'' as follows.

- We actually do not select vertices of G'' uniformly at random, although this is possible to do too, but rather select random clouds in G'' with probability that that is proportional to their sizes. (Recall that the size of each cloud is proportional to the number of the edges that are incident to it in G'', and that all vertices in each cloud are essentially equivalent.)

  Selecting a cloud in G'' with probability that that is proportional to the number of the edges that are incident to it in G'' is almost equivalent to selecting each vertex in G with probability that that is proportional to the weight of the edges that are incident to v in G, which is done by employing "rejection sampling". Specifically, we obtain a sample s of $\mathcal{D}$ (i.e., $s \leftarrow \mathcal{D}$), and proceed as follows.

  1. we output s with probability $|\Gamma(s)|/2d$, where $\Gamma(v)$ denotes the set of neighbors of v in G;
  2. with probability $|\Gamma(s)|/2d$, we output a uniformly selected neighbor of s; that is, each neighbor of s is output with probability $1/2d$;
  3. otherwise (i.e., with probability $(2d - 2|\Gamma(s)|)/2d$), we halt with no output.

  (Indeed, we repeat iterating the foregoing process till some vertex is output, while noting that each trial succeeds with probability at least $1/d$. Hence, we can abort and accept if $\omega(\log n)$ attempts failed.)

  The probability that vertex 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 $s \in \Gamma(v)$ (and then picking $v \in \Gamma(s)$). Note that $p(v) = \sum_{u\in\Gamma(v)}(\mathcal{D}(v) + \mathcal{D}(u))/2d$, whereas the size of the cloud C_v (in G'') is proportional to the degree of v in G', which in is proportional to $\sum_{u\in\Gamma(v)} m_{u,v} \approx \sum_{u\in\Gamma(v)}(\mathcal{D}(u) + \mathcal{D}(v)) \cdot N$. Hence, $p(v)$ is proportional to the degree of v in G'.

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

$$\frac{\mathcal{D}(v) + \mathcal{D}(u)}{\sum_{w \in \Gamma(v)}(\mathcal{D}(v) + \mathcal{D}(w))} \approx \frac{m_{v,u}}{\sum_{w \in \Gamma(v)} m_{v,w}}$$

which approximately equals the probability that a (uniformly) random walk (in G'') moves from any vertex in cloud C_v to the cloud C_u, since for every $i \in [c_v]$ it holds that $\frac{m_{\langle v,i\rangle,u}}{\sum_{w \in \Gamma(v)} m_{\langle v,i\rangle,w}} \approx \frac{m_{v,u}}{\sum_{w \in \Gamma(v)} m_{v,w}}$, where $m_{\langle v,i\rangle,w} = \sum_{j \in [c_w]} m^{i,j}_{v,w}$

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 time complexity of the foregoing algorithm is $\text{poly}(1/\epsilon) \cdot \widetilde{O}(\sqrt{|V''|})$, which equals $\text{poly}(1/\epsilon) \cdot \widetilde{O}(\rho^{-1/2})$, since $|V''| < 3td/\rho$ and $t = O(1/\epsilon)$. (Note that the algorithm does not actually emulate G''; this emulation is merely a thought experiment that is used in the analysis.)[12]

Claim 2.1.3 (analysis of the special case): *Suppose that $G = (V, E)$ is ϵ-far from being bipartite under the vertex-distribution $\mathcal{D}$ such that $\min_{v \in V:\mathcal{D}(v)>0}\{\mathcal{D}(v)\} \geq \rho$. Then, the foregoing algorithm rejects G with high probability.*

Needless to say, the foregoing algorithm always accepts any graph G that is bipartite.

Proof: Combining Claims 2.1.1 and 2.1.2, we observe that more than $0.5d \cdot \epsilon \cdot N - 2 \cdot d \cdot |V'|$ edges must be removed from G'' in order to make it bipartite. Hence, G'' is ϵ'-far from being bipartite (in the standard bounded-degree graph model), where $\epsilon' = 2 \cdot (0.5d \cdot \epsilon \cdot N - 2 \cdot d \cdot |V'|)/dN$. Using $0.5d \cdot \epsilon \cdot N > 4d|V'|$, which holds since $|V'| \leq d/\rho$ (and since we may assume that $N = \omega(1/\epsilon\rho)$), it follows that $\epsilon' \geq \epsilon/2$. Consequently, using proximity parameter ϵ', the tester T (of [8]) would have rejected G'' with high probability (e.g., probability at least $1 - o(1)$), but our emulation of T is not perfect. Specifically, our vertex-sampling procedure produces output that is a factor of $1 \pm O(1/t)$ off from the uniform distribution, whereas each step that we take on the random walks is a factor of $1 \pm \max_{\{u,v\} \in E'} \left\{\frac{c_u c_v}{m_{u,v}}\right\}$ off from the uniform distribution. Recalling that $\frac{c_u c_v}{m_{u,v}} = O((t/\rho)^2/\rho N)$, we conclude that our emulation of T accepts G'' with probability at most

$$\left((1 + O(1/t)) \cdot \left(1 + O(t^2/\rho^3 N)\right)^{\text{poly}(1/\epsilon)\cdot\widetilde{O}(\rho^{-1/2})}\right)^{O(1/\epsilon)} \cdot o(1)$$
$$< \left(\left(1 + \frac{O(1/\epsilon)}{t}\right) \cdot \left(1 + \text{poly}(1/\epsilon) \cdot \frac{t^2}{\rho^4 N}\right)\right) \cdot o(1)$$

which is $o(1)$ when using sufficiently large $t = O(1/\epsilon)$ and $N = \text{poly}(1/\rho\epsilon)$. Hence, our algorithm rejects G with probability $1 - o(1)$. ■

[12] The last comment is inessential, since we could afford emulating G'' at the cost of a factor of $O(\log |E''|)$. Recall that $|E''| \leq dN$, whereas $N = \text{poly}(1/\epsilon\rho)$.

The general case. Recall that the foregoing analysis (which is summarized in Claim 2.1.3) 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 2.1.4, the vertex-distribution $\mathcal{D}$ is $\epsilon/2$-close to a vertex-distribution $\mathcal{D}'$ that satisfies 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 $\texttt{samp}_{\mathcal{D}}$, 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).[13]

Claim 2.1.4 (effective support size and minimal weight): *Suppose that the distribution $\mathcal{D} : U \to [0, 1]$ has an η-effective support of size n. Then, $\mathcal{D}$ is 2η-close to a distribution $\mathcal{D}'$ that satisfies $\min_{e \in U : \mathcal{D}'(e) > 0}\{\mathcal{D}'(e)\} > \eta/n$. 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 $e' \in U$.*

Combining Claims 2.1.3 and 2.1.4, 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 observe that

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

Hence, $\sum_{e \in P^{-1}(0)} \mathcal{D}(e) \leq 2\eta$ whereas $\sum_{e \in P^{-1}(0)} \mathcal{D}'(e) = 0$. Noting that $\mathcal{D}'(e) \geq \mathcal{D}(e)$ for every $e \in U \setminus P^{-1}(0)$, the main claim follows. The furthermore claim follows by the definition of $\mathcal{D}'$. ■

Caveat. Theorem 2.1 was stated for a fixed value of the degree bound d; the running-time of the tester that we present actually grows (polynomially) with d. For starters, in our analysis we emulate an execution of the tester of [8] on a graph that may have $O(d/\epsilon) \cdot n$ vertices, which means that we explore a subgraph of size $\text{poly}(1/\epsilon) \cdot \widetilde{O}(\sqrt{dn})$. In addition, each step in the random walk requires obtaining the weights of all edges that are incident at the current vertex, which means that each random step may have complexity $O(d)$. In contrast, the tester of [8], which works in the standard bounded-degree graph model, has complexity that is independent of the degree bound.

2.2 Approximating the minimal 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

[13] Note that it is not clear how to emulate $\texttt{eval}_{\mathcal{D}'}$, when given restricted oracle access to $\texttt{eval}_{\mathcal{D}}$. Hence, it is crucial that we can emulate the tester by using (instead) and with an evaluator of the ratios between the $\mathcal{D}'$-values. An evaluator of $\mathcal{D}'$-ratios can be implemented using $\texttt{eval}_{\mathcal{D}}$ by relying on the furthermore part of Claim 2.1.4.

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).

2.2.1 A simple approximation of the minimal 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* $3\eta \cdot m$. Here and below, the sampling oracle $\mathtt{samp}_{\mathcal{D}}$ is used to generate samples, whereas the evaluation oracle $\mathtt{eval}_{\mathcal{D}}$ is used to classify elements according to their $\mathcal{D}$-values.

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 minimal 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 η-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 n' is a lower bound on the minimal 4η-effective support size of $\mathcal{D}$ (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* $2^{\lfloor \log_2 n' \rfloor}$ *and* $2^{\lceil \log_2 n \rceil}/\eta$ (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 $\sum_{i \leq i^*} O(\eta^{-1} \cdot \log i) = \widetilde{O}(\log n)/\eta$ steps (and its expected number of steps can be similarly bounded). Combining this result with Theorem 2.1, we essentially obtain Theorem 1.4 (except that $\epsilon/5$ is replaced by $\epsilon/16$).

2.2.2 Better approximations of the minimal 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 foregoing version, we shall 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 $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < (\beta - 1) \cdot \eta/2^i] \geq \beta^3 \cdot \eta$), 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 $\beta^2 \cdot \eta \cdot m$). When analyzing the probability that this algorithm halts by iteration $i^* = \log_2 n$ (where n upper-bounds the minimal η-effective support size of $\mathcal{D}$) 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 when analyzing the probability that the algorithm halts before iteration $i^+ = \log_2(\eta \cdot n')$ we use the fact that $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < (\beta-1) \cdot \eta/2^{i^+-1}] < \beta^3 \cdot \eta$ implies that $\mathcal{D}$ is $\beta^3 \cdot \eta$-close to a distribution with support size at most $n'/2$. 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.* Hence, the slackness factor in the "effectiveness" parameter is reduced from 4 to β^3, for any constant $\beta > 1$.

Obtaining a better approximation factor. To obtain an even better approximation of the minimal 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 $i \in [O(\log(\widetilde{n}/\eta))]$, where $\widetilde{n}/\eta$ is upper-bounded by $2/\eta^2$ times the minimal η-effective support size of $\mathcal{D}$. The key observation here is that the hypothesis that $2\widetilde{n}$ upper-bounds the $\beta^3\eta$-effective support size of $\mathcal{D}$ implies that the total weight of elements having a $\mathcal{D}$-value that is smaller than $(\beta-1) \cdot \beta^3\eta/2\widetilde{n}$ is at most $\beta^4 \cdot \eta$ (cf., prior paragraph, while replacing 2^{i^*} by $2\widetilde{n}$).[14] Indeed, the rough estimate of the effective support size of $\mathcal{D}$ (i.e., $\widetilde{n}$) is only used in order to determine the set of i's for which we approximate the number of elements that have $\mathcal{D}$-value approximately $\beta^{-(i-0.5)}$.

In light of the foregoing, our first step is ignoring elements having $\mathcal{D}$-value below $\eta/\widetilde{n}$ or so. Specifically, letting $\eta' = \beta^3 \cdot \eta$ and $H \stackrel{\text{def}}{=} \{v : \mathcal{D}(v) \geq (\beta-1) \cdot (\eta'/2\widetilde{n})\}$, recall that $\mathcal{D}$ has η'-effective support size at most $2\widetilde{n}$, which implies that $\mathcal{D}(H) \stackrel{\text{def}}{=} \sum_{v \in H} \mathcal{D}(v) \geq 1 - \beta\eta'$ (see prior paragraph). Hence, for $\eta'' = \beta \cdot \eta' = \beta^4 \cdot \eta$, the size of H provides an upper bound on the minimal η''-effective support size of $\mathcal{D}$. If $\mathcal{D}(H) \leq 1-\eta$, then $|H|$ is a lower bound on the minimal η-effective support size of $\mathcal{D}$. Otherwise, we shall obtain such a lower bound by using the information gathered for the upper bound. Details follow.

To approximate $|H|$, we let $W_i = \{v : \beta^{-i} \leq \mathcal{D}(v) < \beta^{-(i-1)}\}$, and observe that it suffice the approximate $\mathcal{D}(W_i)$ for $i = 1, ..., \ell$, where $\ell \stackrel{\text{def}}{=} \log_\beta(\widetilde{n}/(\beta-1) \cdot \eta') = O((\beta-1)^{-1} \cdot \log(\widetilde{n}/\eta))$, since $|W_i| \approx \mathcal{D}(W_i)/\beta^{-(i-0.5)}$. Actually, letting $I = \{i \in [\ell] : \mathcal{D}(W_i) \geq (\beta-1)\eta''/\ell\}$, it suffices to approximate $\mathcal{D}(W_i)$ for every $i \in I$, which yields approximations of the corresponding $|W_i|$'s (using $|W_i| \approx \beta^{i-0.5} \cdot \mathcal{D}(W_i)$). That is, we do not actually approximate $|H|$ but rather approximate the size of $H' \stackrel{\text{def}}{=} \bigcup_{i \in I} W_i \subseteq H$, while capitalizing on $\mathcal{D}(H \setminus H') \leq (\beta-1) \cdot \eta''$, which implies that $\mathcal{D}(H') \geq 1 - \eta'' - (\beta-1) \cdot \eta''$, which in turn means that the minimal $\beta \cdot \eta''$-effective support size of $\mathcal{D}$ is at most $|H'|$. Now, 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 β^2-factor approximation of $|W_i|$ (since $|W_i| \in [\beta^{i-1}\mathcal{D}(W_i), \beta^i \cdot \mathcal{D}(W_i))$). Hence, we obtain a β^2-factor approximation of $|H'|$.

If $\mathcal{D}(H) \leq 1-\eta$, then $|H'|$ also provides the desired lower bound (on the minimal η-effective support size of $\mathcal{D}$). Otherwise (i.e., $1 - \mathcal{D}(H) \leq \eta$), we approximate the minimal η-effective support size of $\mathcal{D}$ by using the foregoing approximations of the $|W_i|$'s. For starters, note that we can easily estimate $1 - \mathcal{D}(H)$ up to any error term of the form $\Omega(\eta)$ (using $O(1/\eta^2)$ samples of $\mathcal{D}$). Letting $\widetilde{\delta}$ denote our approximation of $1 - \mathcal{D}(H)$, we wish to dispose of the maximal number of elements of H' having total probability weight $w \stackrel{\text{def}}{=} \max(\eta - \widetilde{\delta}, 0)$ according to $\mathcal{D}$. In other words, we wish to dispose of as many of the lightest elements of H' till we reach a total weight of w. We do so by finding a maximal $t \in I$ such that $\sum_{i \in I : i \geq t} \mathcal{D}(W_i) \geq w$ and dispose of all the elements in $\cup_{i \in I : i > t} W_i$ as well as of $\frac{w - w'}{\beta^{-(t-0.5)}}$ of the elements of W_t, where $w' \stackrel{\text{def}}{=} \sum_{i \in I : i > t} \mathcal{D}(W_i) \leq w$. (We stress that the foregoing process is performed

[14] That is, letting $\eta' = \beta^3\eta$, observe that $\Pr_{v \leftarrow \mathcal{D}}[\mathcal{D}(v) < (\beta-1) \cdot (\eta'/2\widetilde{n})] < 2\widetilde{n} \cdot (\beta-1) \cdot (\eta'/2\widetilde{n}) + \eta' = \beta \cdot \eta'$, where the first (resp., second) term is due to elements that are (resp., are not) in the η'-effective support of $\mathcal{D}$.

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.) Denoting the resulting set by H'', it follows that $|H''|$ is a lower bound on the minimal $\beta^{-1} \cdot \eta$-effective support size of $\mathcal{D}$. Hence, we get.

Theorem 2.2 (approximating the effective support size): *There exists an oracle machine M that, on input parameters $\eta > 0$ 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 $\mathrm{poly}(1/(1-\beta)) \cdot \widetilde{O}(\eta^{-2} \log n)$ steps such that $n \in [L, U]$, where L equals the minimal $\beta^5 \cdot \eta$-effective support size of $\mathcal{D}$ and U equals β^3 times the minimal $\beta^{-1} \cdot \eta$-effective support size of $\mathcal{D}$. Furthermore, the expected number of steps performed by M is $\mathrm{poly}(1/(1-\beta)) \cdot \widetilde{O}(\eta^{-2} \log n)$.*

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 establish Theorem 1.4.

The algorithm underlying Theorem 2.2 provides a rather good approximation of the minimal effective support size, but its complexity depends (mildly) on the effective support size. In the original version of this work, we asked whether it is possible to obtain a meaningful approximation of the effective support size within complexity that is independent of it, and furthermore whether one can obtain good approximations (as in Theorem 2.2) within complexity $\mathrm{poly}(1/\eta)$. In a subsequent work [5], we addressed these questions, obtaining a positive answer to the first question but leaving the second question unresolved. Specifically, a $\widetilde{O}(\log^{O(1)}(n/\epsilon))$-factor approximation can be obtained in $o(1/\eta^2)$-time [5, Thm. 1.9], where $\log^{(i)}$ denotes i iterated logarithms.

3 The Cycle-Freeness Tester

Following [2], which operates in the standard bounded-degree model, we reduce (VDF) testing Cycle-freeness to (VDF) testing Bipartiteness, where in both cases we refer to testing in the augmented VDF model for bounded-degree graphs. While the reduction is identical to the one presented in [2], 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 the 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). When testing the instance $\langle G, \tau\rangle$ for generalized 2-Colorability, one gets oracle access both to (the incidence function of) G and to τ, and distances between instances are defined in the natural manner.[15]

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 [8] (as well as the ones of [9] and [6]) is easily extended to test this generalization of 2-Colorability. Specifically, instead of checking whether the explored subgraph is bipartite, one checks whether the

[15] Specifically, we replace $\delta_{\mathcal{D}}^{\Pi}(g)$ by the minimum of $\delta_{\mathcal{D}}(\langle g, \tau\rangle, \langle g', \tau'\rangle)$ taken over all incidence functions $g' : V \times [d] \to V \cup \{\perp\}$ representing graphs $G' = (V, E')$ and all functions $\tau' : E' \to \{\mathtt{eq}, \mathtt{neq}\}$ such that $\langle G', \tau'\rangle$ has a legal 2-coloring, where

$$\delta_{\mathcal{D}}(\langle g, \tau\rangle, \langle g', \tau\rangle) \stackrel{\mathrm{def}}{=} \mathrm{Pr}_{v \leftarrow \mathcal{D}, i \in [d]}[g(v, i) \neq g'(v, i) \quad \vee \quad (g(v, i) \neq \perp \;\&\; \tau(\{v, g(v, i)\}) \neq \tau'(\{v, g(v, i)\}))].$$

(If $g(v, i) \neq g'(v, i)$, then the condition is defined to hold even if either τ or τ' is undefined on $\{v, g(v, i)\}$; on the other hand, if $g(v, i) = g'(v, i) = \perp$, then the condition is defined to be violated.)

explored sub-instance (i.e., the explored subgraph along with the labeling of its edges) has a legal 2-coloring.

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 $\langle G, \tau\rangle$ 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 $\langle G, \tau\rangle$ is $\Omega(\epsilon)$-far from having a legal 2-coloring (under the distribution $\mathcal{D}$). We only prove a weaker result (in which $\Omega(\epsilon)$ is replaced by $\Omega(\epsilon/\log n)$), which suffices for our application (since $\widetilde{O}(\sqrt{n}) \cdot \mathrm{poly}(1/\Omega(\epsilon/\log n)) = \widetilde{O}(\sqrt{n}) \cdot \mathrm{poly}(1/\epsilon)$).

Lemma 3.1 (analysis of the randomized reduction): *Let $G = (V, E)$ be a simple bounded-degree graph* (i.e., G has neither parallel edges nor self-loops)[16] *and $\mathcal{D}$ be a distribution on V. If G is ϵ-far* (according to distribution $\mathcal{D}$) *from being Cycle-free, then, with probability at least $\Omega(1)$ over the random choice of $\tau : E \to \{\mathtt{eq}, \mathtt{neq}\}$, the instance $\langle G, \tau\rangle$ is $\Omega(\epsilon/\log|V|)$-far* (according to distribution $\mathcal{D}$) *from having a 2-legal coloring.*

Hence, referring to Definition 1.1, 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-freeness with respect to arbitrary graphs and distribution $\mathcal{D}$ that have $\epsilon/4$-effective support size n (see Corollary 3.2). Hence, combining Corollary 3.2 with the core of the proof of Theorem 2.1 (i.e., Claims 2.1.4 and 2.1.3), we obtain Theorem 1.5. (See details following Corollary 3.2.)

Proof: Maintaining $\mathcal{D}$ unchanged, we first omit from G all edges that have probability weight smaller than $\epsilon/2d|V|$ (i.e., we omit the edge $\{u, v\}$ if $2 \cdot \frac{\mathcal{D}(u)+\mathcal{D}(v)}{d} < \epsilon/2d|V|$). 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 $\ell = \lfloor \log_2(2d|V|/\epsilon)\rfloor + 1$, 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 claim that *edges that connect such components must be in F'.*

Claim 3.1.1 (all bridges in G'' are in F'): *Any edge of G'' that does not reside in a two-connected component of G''* (i.e., a bridge of G'') *is in F'.*

Proof: Note that any edge e of G'' that does not resides in a two-connected components connects two such components, denoted C_1 and C_2, whereas each of the C_i's may consist of a single vertex. That is, e has one endpoint in C_1 and one endpoint in C_2. On the other hand, C_1 and C_2 must be connected in G' by a path of edges that belong to the spanning forest F', and this path must exist also in G''. Hence, e must be on this path (and so is in F'), since otherwise we obtain two different edge-disjoint paths that connect C_1 and C_2 in G'', which contradicts the maximality of the C_i's. ■

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

Next, we omit the edges of F' that connect different two-connected components (which, by Claim 3.1.1, is equivalent to omitting the bridges of G''), 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 $E''' \cap F'$.

Claim 3.1.2 (the weight of edges in G'''): *Each edge in $E''' \cap F'$ has weight at least 2^{-i}.*

Proof: 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 the fact that the connected components of G''' are two-connected components of G''. Specifically, vertices u and v in a connected component C of G''' that are not connected in G''' by a path of edges of F' must be connected in G'' by a path of edges of F' that passes through a vertex w that is not in C. But this contradicts the hypothesis that C is a two-connected component of G'', since u and v are connected in G'' both via a path that reside inside C and via a path that goes though $w \notin C$ (a fact that contradicts the maximality of C as a two-connected component in G'').

Now, let $T \subseteq F'$ be a spanning tree of one of the connected components of G''', and suppose towards the contradiction that one of the edges of T has weight smaller than 2^{-i}. Then, omitting this edge and adding an edge of B_i that connects the two resulting sub-trees of T, 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 one. Applying the same replacement to F' itself, this yields a spanning forest of G' that has weight larger than the forest F', in contradiction to the definition of F'. ■

Recalling that $E''' \setminus F' = B_i$ and that $F' \cap E'''$ is a spanning forest of G''', we observe that $|B_i|$ edges must be removed from G''' in order to make it cycle-free. Finally, we apply [2, Lem. 3.1] to G''', and infer 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 $\langle \cdot, \tau \rangle$ has a legal 2-coloring. Recalling that $|B_i| \geq \frac{\epsilon/4\ell}{2^{-i+1}} = \Omega(2^i \cdot \epsilon/\ell)$, and that each edge in G''' has weight at least 2^{-i}, it follows that the weight of edges that must be omitted from G''' in order to yield an instance $\langle \cdot, \tau \rangle$ 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. ■

Corollary 3.2 (analysis of the randomized reduction, revised): *Let $G = (V, E)$ be a simple bounded-degree graph and $\mathcal{D}$ be a distribution on V. Suppose that $\mathcal{D}$ is 0.5ϵ-close to a distribution $\mathcal{D}'$ that has support size at most n, and that $\mathcal{D}'(v) \geq \epsilon/4n$ for every v in the support of $\mathcal{D}'$. If G is ϵ-far according to distribution $\mathcal{D}$ from being cycle-free, then, with probability at least $\Omega(1)$ over the random choice of $\tau : E \to \{\mathtt{eq}, \mathtt{neq}\}$, the instance $\langle G, \tau \rangle$ is $\Omega(\epsilon/\log n)$-far according to distribution $\mathcal{D}'$ from having a 2-legal coloring.*

Recall that by Claim 2.1.4 if $\mathcal{D}$ has $\epsilon/4$-effective support size n, then it is $\epsilon/2$-close to a distribution $\mathcal{D}'$ as in the hypothesis (of Corollary 3.2). Invoking Corollary 3.2, we obtain an instance $\langle G, \tau \rangle$ that (w.p. $\Omega(1)$) is ϵ'-far according to distribution $\mathcal{D}'$ from having a 2-legal coloring, where $\epsilon' = \Omega(\epsilon/\log n)$. The extra condition regarding $\mathcal{D}'$ allows to invoke Claim 2.1.3 (or rather its extension to generalized 2-colorings), with vertex distribution $\mathcal{D}'$, the instance $\langle G, \tau \rangle$, proximity parameter ϵ', and $\rho = \epsilon/4n$. It follows that the corresponding tester rejects $\langle G, \tau \rangle$ with probability $\Omega(1)$. Note that the time spend by this tester on this instance is $\mathrm{poly}(1/\epsilon') \cdot \widetilde{O}(\sqrt{1/\rho\epsilon'})$, which is $\mathrm{poly}(1/\epsilon) \cdot \widetilde{O}(\sqrt{n})$, and Theorem 1.5 follows.

Proof: Noting that G is $0.5 \cdot \epsilon$-far according to $\mathcal{D}'$ from being cycle-free, we consider the subgraph $G' = (V', E')$ obtained by omitting from G all edges that have weight 0 under $\mathcal{D}'$ as well as all vertices that become isolated. Hence, $|V'| \leq d \cdot n = O(n)$. Applying Lemma 3.1, we infer that, with probability at least $\Omega(1)$ over the random choice of $\tau' : E' \to \{\mathtt{eq}, \mathtt{neq}\}$, the instance $\langle G', \tau' \rangle$ is $\Omega(\epsilon/\log n)$-far according to distribution $\mathcal{D}'$ from having a 2-legal coloring. In these cases, extending τ' arbitrarily to E, we obtain a labeling $\tau : E \to \{\texttt{eq}, \texttt{neq}\}$ such that the instance $\langle G, \tau \rangle$ is $\Omega(\epsilon/\log n)$-far according to distribution $\mathcal{D}'$ from having a 2-legal coloring. ■

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. On the Complexity of Estimating the Effective Support Size. *ECCC*, TR19-088, 2019.

[6] 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.

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

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

[9] 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.

[10] 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.