Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 171 (2018)

Testing Graphs in Vertex-Distribution-Free Models*

Oded Goldreich†

March 8, 2019

Abstract

Prior studies of testing graph properties presume that the tester can obtain uniformly distributed vertices in the tested graph (in addition to obtaining answers to the some type of graph-queries). Here we envision settings in which it is only feasible to obtain random vertices drawn according to an arbitrary distribution (and, in addition, obtain answers to the usual graph-queries). We initiate a study of testing graph properties in such settings, while adapting the definition of distance between graphs so that it reflects the different probability weight of different vertices. Hence, the distance to the property represents the relative importance of the "part of the graph" that violates the property. We consider such "vertex-distribution free" (VDF) versions of the two most-studied models of testing graph properties (i.e., the dense graph model and the bounded-degree model).

In both cases, we show that VDF testing within complexity that is independent of the distribution on the vertex-set (of the tested graph) is possible only if the same property can be tested in the standard model with one-sided error and size-independent complexity. We also show that this necessary condition is not sufficient; yet, we present size-independent VDF testers for many of the natural properties that satisfy the necessary condition.

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

*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 Unions Horizon 2020 research and innovation programme (grant agreement No. 819702).

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

ISSN 1433-8092

Contents

1 Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., a recent textbook [14]). 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 [17] and reviewed in [14, Chap. 8]) and the bounded-degree graph model (introduced in [19] and reviewed in [14, Chap. 9]).[1] 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).

1.1 The new models

Envisioning settings in which uniformly sampling the vertex-set of the graph is not realistic, we ask 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. (See motivational discussion in Section 1.4.) The foregoing suggestion may become more clear when focusing on specific graph testing models, as we do next.

The bounded-degree graph model. In the standard model of testing bounded-degree graphs, for a constant degree bound d, graphs (over the vertex-set $[n] = \{1, 2, ..., n\}$) are represented by incidence functions of the form $g : [n] \times [d] \to [n] \cup \{\perp\}$ such that $g(v, i)$ is the i^{th} neighbor of v (and $g(v, i) = \perp$ if v has less than i neighbors). The tester is given n (along with a proximity parameter ϵ) and oracle access to g, and is required to accept (whp) if g represents a graph having a predetermined property and reject (whp) if g is far from representing such a graph. Specifically, g is considered ϵ-far from the graph property Π if every $g' : [n] \times [d] \to [n] \cup \{\perp\}$ that represents a graph in Π differs from g on more than an ϵ fraction of the domain (i.e., $\Pr_{(v,i)\in[n]\times[d]}[g(v, i) \neq g'(v, i)] > \epsilon$).

In the new model, the vertex-set of the graph is arbitrary, and the tester is given a sampling device that returns vertices in the graph according to some distribution $\mathcal{D}$. In addition, the tester is given oracle access to the incidence function of the graph, $G = (V, E)$, which has the (analogous) form $g : V \times [d] \to V \cup \{\perp\}$. As before, the tester has to accept (whp) in case the graph has the property, and reject (whp) if the graph is far from the property, but here distances are measured according to $\mathcal{D}$; that is, g is considered 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 $\Pr_{v\leftarrow\mathcal{D},i\in[d]}[g(v, i) \neq g'(v, i)] > \epsilon$.

We stress that, unlike in the standard presentations of the bounded-degree graph model, the tester is not given a description of the vertex-set. Such a description is implicit in the standard presentations, which provide the tester with the size parameter n (while postulating that the vertex-set equals $[n]$). As discussed in [15], a more flexible model may provide the tester with (1) a device that samples the vertex-set uniformly at random, along with (2) partial information regarding the vertex-set, ranging from a full specification of the vertex-set to nothing. Here, we chose the latter option, because it seems most compatible with the settings that are envisioned above (where the vertex-set may not be known).

[1] A third model, called the general graph model (introduced in [30, 25] and reviewed in [14, Chap. 10]), was also studied.

[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.

The foregoing model may be called a distribution-free version of the bounded-degree graph testing model. Note, however, that the distribution on the domain of the incidence function is not arbitrary; it is rather a product of an arbitrary distribution $\mathcal{D}$ over the vertex-set and the uniform distribution over $[d]$. For this reason, we use the (more cumbersome) term vertex-distribution-free, abbreviated VDF.

The dense graph model. Analogously, we present a vertex-distribution-free version of the dense graph testing model. Again, the vertex-set of the graph is arbitrary, and the tester is given a sampling device that returns vertices in the graph according to some distribution $\mathcal{D}$. In addition, the tester is given oracle access to the adjacency predicate $g : V \times V \to \{0,1\}$ of the graph, $G = (V, E)$. The tester has to accept (whp) in case the graph has the property, and reject (whp) if the graph is far from the property, but here distances are measured according to $\mathcal{D}$; that is, g is considered ϵ-far from the graph property Π if, for every $g' : V \times V \to \{0,1\}$ that represents a graph in Π, it holds that $\Pr_{u,v \leftarrow \mathcal{D}}[g(u,v) \neq g'(u,v)] > \epsilon$. (Note that the standard model corresponds to the special case in which $\mathcal{D}$ is guaranteed to be uniform over $V = [n]$, and the tester is also given n.)[3]

We stress that the distribution induced on the domain of the adjacency predicate is not arbitrary, but is rather the Cartesian product of an arbitrary distribution $\mathcal{D}$ with itself. We mention that this model was briefly discussed by Goldreich, Goldwasser, and Ron [17, Sec. 10.1], where it was viewed as a restricted case of distribution-free testing (which was briefly discussed in the subsequent paragraph in [17, Sec. 10.1]).

Focus: Query complexity that only depends on the proximity parameter. In this work, we focus on testers having query complexity that only depends on the proximity parameter ϵ. We refer to such testers as strong testers.[4] Note that in the standard testing models this means that the complexity is independent of the size of the graph (and in [14] the term "size-oblivious complexity" is used), but in the VDF testing models this also means that the complexity is independent of the vertex-distribution $\mathcal{D}$. We stress that strong testability does not exhaust the scope of property testing, and that going beyond strong testability is of interest also in the vertex-distribution-free context (see discussion in Section 5.2).

Additional sampling device. We believe that providing the tester with uniformly distributed vertices, in addition to samples drawn from $\mathcal{D}$, is not compatible with the settings that we envision. Still, in light of the fact that some of our lower bounds rely on the tester's inability to sample the vertex-set uniformly, we consider also the case that such uniform samples are provided to the tester. We discuss this secondary model in Section 4.

1.2 Our main results

We first observe that in the vertex-distribution-free (VDF) versions of both the dense and bounded-degree graph models, strong testability implies strong testability with one-sided error.[5] We mention that this result stands in sharp contrast to the standard models where there are significant gaps between the complexities of one-sided error and general testers (see, for example, the case of ρ-Clique in the standard dense graph model [17], and the case of cycle-freeness in the standard bounded-degree graph model [19]).[6]

[3] Again, a more flexible model is discussed in [15].

[4] In some sources (cf., e.g. [4]), the term "strongly testable" refers only to one-sided error testers of such complexity. In [10], "strong testability" is defined as having a proximity oblivious tester (with linear detection probability function).

[5] A tester is said to have one-sided error if it always accepts objects that have the property (rather than accept them with high probability.

[6] Recall that, in the standard dense graph model, the sets of n-vertex graphs having a clique of size $\rho \cdot n$ has a poly$(1/\epsilon)$-query tester (with two-sided error) but no $o(n)$-query testers with one-sided error. Likewise, in the standard

Theorem 1.1 (strong testability in the VDF models implies one-sided error testability):[7] *If a graph property is strongly testable in the VDF version of the dense graph model* (resp., bounded-degree graph model)*, then it is strongly testable with one-sided error in the same model. Furthermore, in the case of the dense graph model* (resp., bounded-degree graph model)*, the query complexity of VDF testing with one-sided error is at most polynomial* (resp., exponential) *in the complexity of VDF testing with two-sided error.*

Since the standard testing models (for dense and bounded-degree graphs) are (essentially) special cases of the VDF models, it follows that *only graph properties that are strongly testable with one-sided error in the standard model are strongly testable in the corresponding VDF model.* A natural question is whether the necessary condition for strong testability in the VDF models is a sufficient one.[8] We show that this is not the case.

Theorem 1.2 (strong testability with one-sided error in the standard model does not suffice for strong testing in the VDF model):[9] *There are graph properties that are strongly testable with one-sided error in the standard dense graph model* (resp., bounded-degree graph model)*, but are not strongly testable in the VDF version of the dense graph model* (resp., bounded-degree graph model)*.*

Theorem 1.2 is manifested by natural properties such as Connectivity. In light of Theorem 1.2, we confine ourselves to showing that some natural classes of graph properties that are strongly testable with one-sided error in the standard model are strongly testable in the corresponding VDF model. In the case of the dense graph model, we consider two such classes.

Theorem 1.3 (classes that are strongly testable in the VDF dense graph model):[10] *The following properties are strongly testable in the VDF version of the dense graph model.*

1. *Any general graph partition problem that is strongly testable with one-sided error in the standard dense graph model.*

2. *Any subgraph-freeness property.*

Furthermore, in the case of general partition problems as well as in the case that the subgraph is bipartite, the complexity is polynomial in $1/\epsilon$*.*

The framework of graph partition problems was introduced in [17], where it was shown that all these properties are strongly testable (possibly with two-sided error). The subclass that admits strong testers with one-sided error was characterized in [23], and it contains problems such as k-coloring. (Actually, we consider a natural extension of the framework and the aforementioned results, presented recently in [28].) Subgraph-freeness properties were first considered in [3], where it was shown that all these properties (e.g., triangle-freeness) are strongly testable with one-sided error.

In the case of the bounded-degree graph model, our results are more sporadic in nature. This is an artifact of starting with less rich body of results regarding the standard model. Specifically, our

bounded-degree graph model, cycle-freeness has a poly$(1/\epsilon)$-query tester (with two-sided error) but no $o(\sqrt{n})$-query testers with one-sided error.

[7] The "dense graph model" part (resp., "bounded-degree graph model" part) of Theorem 1.1 appears as Theorem 2.3 (resp., Theorem 3.3).

[8] We mention that a characterization of the class of "natural" graph properties that are strongly tested with one-sided error in the standard dense graph model was provided in [5]. No such result is known for the bounded-degree graph model.

[9] The "dense graph model" part (resp., "bounded-degree graph model" part) of Theorem 1.2 is established both by Proposition 2.9 and Proposition 2.11 (resp., both by Proposition 3.5 and Proposition 3.6).

[10] The first (resp., second) part of Theorem 1.3 appears as Theorem 2.7 (resp., Theorem 2.8).

starting point is provided by the few strong testing results obtained in [19], and the fact (established in [9]) that a minor-free property is strongly testable with one-sided error if and only if the minor is cycle-free (i.e., is a forest).[11]

Theorem 1.4 (properties that are strongly testable in the VDF bounded-degree graph model):[12] *The following properties are strongly testable in the VDF version of the bounded-degree graph model.*

- *Subgraph-freeness, degree-regularity, and being Eulerian.*
- *For every $k \geq 1$, having a path of length k, and containing a tree with k leaves.*[13]

A begging question is whether the strong testability results (asserted in Theorem 1.4) can be extended to other tree (and forest) minors, which are all strongly testable in the standard model (see [9]). We refrain from studying this question here, since the corresponding analyses of the strong testers in the standard model are already fairly complex.

Additional results. The foregoing overview does not cover all results presented in this work. Notable omissions include:

- A study of two classes of graph properties that are easy to test in the standard dense graph model (see Section 2.4). We show that one class, which is trivial to test with one-sided error in the standard model, is hard to test in the current VDF model, whereas for the other class (which consists of properties of sparse graphs) the situation is mixed. In particular, we establish the hardness of testing (in the VDF model) whether a graph consists of two stars each containing at least one third of the graph's vertices (see Proposition 2.11), and show that any minor-free graph property can be strongly tested in the VDF model (see Theorem 2.13).
- A study of a generalized version of the VDF bounded-degree graph model that is briefly discussed at the end of Section 1.4. The bottom-line is that the positive results of the VDF (bounded-degree graph) model extend to the generalized model (see Section 3.5).
- A study of the secondary models mentioned at the end of Section 1.1. The most important take-home message is that the relation to one-sided error testing does not extend to these models (see Section 4).

1.3 Techniques

Throughout the paper, we denote by $\mathcal{D}$ both a distribution over a vertex-set and a device that outputs samples drawn from this distribution. We also use the notation $\mathcal{D}(v) \stackrel{\text{def}}{=} \Pr_{x \leftarrow \mathcal{D}}[x = v]$. (As usual, throughout the paper, ϵ denotes the proximity parameter.)

1.3.1 Strong testability and one-sided error

Theorem 1.1 is proved by transforming any VDF tester into one that operates with one-sided error. The resulting (VDF) tester takes a sample of $O(s^2)$ vertices, where $s = s(\epsilon)$ is the sample complexity of the original tester, and invokes the original tester on each possible choice of a sub-sample (of size s) and coins for the original tester, ruling by majority. Note that the resulting tester makes no random

[11]The fact (establshed in [7]) that all minor-free properties are strongly testable (in the standard bounded-degree graph model) is irrelevant here, since we need strong one-sided error testers.

[12]The VDF-testability of the various properties is established in Sections 3.2–3.4.

[13]These properties correspond to two minor-freeness properties, where the minors are a k-path (i.e., a path of k edges) and a k-star (i.e., a $(k+1)$-vertex tree with k leaves).

choices, and its randomization is due to the choice of the vertex-sample it uses. Indeed, in addition to obtaining samples, both the original and resulting tester issues queries, but these queries must refer to vertices that were already seen (in prior samples (or answers to prior queries in the bounded-degree graph model)). Hence, for the dense graph model the query complexity of the resulting (VDF) tester is $O(s^2)^2$, and for the bounded-degree graph model the resulting (VDF) query complexity is $\exp(O(s))$.

The key observation is that each sample of $O(s^2)$ vertices for the resulting tester gives rise to a distribution over $O(s^2)$ vertices (i.e., the uniform distribution over this sample), and the one-sided feature follows by the hypothesis that (with probability at least 2/3) the original tester rules correctly on any distribution. Hence, for any yes-instance and any sample of $O(s^2)$ vertices of any vertex-distribution $\mathcal{D}$, the majority vote (over all s-sized sub-samples) is in favor of accepting, and hence the resulting tester always accepts (i.e., it has one-sided error).[14] The claim that the resulting tester rejects no-instances (w.h.p.) is proved by using the fact that the original VDF tester must reject these instances (w.h.p.) when getting a sample of $\mathcal{D}$. Indeed, we use the fact that (with very high probability) a random invocation of the original tester uses a sample of $\mathcal{D}$ (i.e., s elements drawn independently from $\mathcal{D}$, where the difference is due to the difference between selecting s elements *with and without repetition* among $O(s^2)$ elements).

We mention that a similar argument was employed in the proof of [22, Thm. 7.2], which referred to a different setting (i.e., sample-based testers that obtain "labeled samples" but make no queries).[15] The resulting tester there did not invoke the original tester, but rather ruled according to whether the labeled-sample viewed by it is consistent with a object that has the predetermined property. We cannot afford this option here, since the size of the vertex-set is not *a priori* known to the testers in the current model. Furthermore, the time complexity of the testers we derive is independent of the size of the tested object, and seems typically smaller than that size.

Theorem 1.2 (i.e., "insufficiency of the necessary condition") can be proved by using properties that are trivial to test in the standard models; see Propositions 2.9 and 3.5, respectively. Specifically, such properties do not contain the empty graph, but do contain some graph with very few edges (e.g., $O(n)$ edges in case of the dense graph model, and $O(\sqrt{n})$ edges in case of the bounded-degree graph model). The impossibility of strongly testing in the VDF models relies on the fact that the tester does not know the size of the graph and cannot distinguish between a sample taken from an empty graph and a sample taken from a distribution that is concentrated on an independent set (or an isolated set) in a larger graph. This raises the question of what happens if we either provide the VDF tester with the size of the graph (which is not compatible with the settings that we envision) or restrict the standard-model testers in a similar manner (as done in [5]); the second option is discussed in Section 5.1. An alternative proof of the dense graph model part of Theorem 1.2, which does not rely on the testers' obliviousness of the size of the graph, is presented in the proof of Proposition 2.11.

1.3.2 Strong testability in the VDF dense graph model

Theorem 1.3 asserts strong tester for the VDF dense graph model for specific properties that have a strong tester in the standard dense graph model. The basic idea that underlies its proof is emulating an execution of these strong testers (for the standard model) on an auxiliary graph that is a suitable (generalized) blow-up of the input graph, where the varying amount of blow-up is determined by the vertex-distribution. Specifically, given query access to the input graph $G = (V, E)$ and samples from the vertex-distribution $\mathcal{D}$, we consider an auxilary graph in which each vertex $v \in V$ is replaced by a cloud of size proportional to $\mathcal{D}(v)$, and edges are replaced by complete bipartite graphs (between the corresponding clouds). This idea, which originates in [17, Sec. 10.1], works well provided that the

[14] Here we used the fact that the original VDF tester accepts (whp) when given s samples drawn uniformly from any multi-set of size $O(s^2)$.

[15] That is, when testing an object represented by a function $f : D \to R$, the tester obtains labeled samples of the form $(x, f(x))$, where x is drawn from a distribution over D.

same vertex is not sampled twice (equiv., the collision probability of $\mathcal{D}$ is $o(1/q^2)$, where $q = q(\epsilon)$ is the number of queries made by the original tester). But we have to deal with the general case (in which the same vertex may be sampled twice). This raises different problems in the two parts of Theorem 1.3.

In Part 1 (i.e., when testing generalized graph partition properties), the problem is whether or not to put an edge between copies of the same vertex. Not putting an edge will do when testing k-colorability, but in general an oblivious decision will not do (since in the specific graph k-partition property that we handle some parts may mandate internal edges and other parts may mandate no internal edges). To solve this problem, we first find (by sampling) all vertices that are likely to appear more than once in our main sample (i.e., all vertices v such that $\mathcal{D}(v) = \Omega(1/q(\epsilon/O(1))^2)$). Loosely speaking, assuming we found t such "heavy" vertices, we invoke 2^t copies (of an error-reduced version) of the original tester such that in the i^{th} copy we place edges between copies of the j^{th} heavy vertex if and only if the j^{th} bit in the binary extension of i equals 1. We accept if and only if at least one of these 2^t copies accepted.

In Part 2 (i.e., when testing subgraph freeness), the problem is that we should consider only subgraphs (of the blown-up graph) in which the vertices belong to different clouds (and so correspond to different vertices of the original graph). This is not a problem when the subgraph is a clique (and we place no internal edges in the emulated clouds), since in this case the subgraph cannot contain two vertices from the same cloud, but for other subgraphs we do have a problem. The problem is resolved by extending the original tester to the setting of r-colored graphs (studied in [2]); this task is undertaken in Claim 2.8.4, which builds on [26, Thm. 1.18] (which generalizes Szemerédi's regularity lemma [33] to the setting of r-colored graphs).[16] We then reduce the problem of finding subgraphs with at most one vertex in each cloud to finding 2-colored cliques in a suitable 2-colored version of the blown-up graph (in which all pairs of clouds are connected by edges, and the colors of these edges indicate whether or not the corresponding edges existed in the uncolored version of the blown-up graph).

Another problem that arises when testing subgraph freeness is relating the weighted distance of the input graph from being subgraph-free to the distance of the blow-up graph (from the same property). The relevant lower bound is proved in Claim 2.8.1.

Lastly, we mention that when proving the furthermore part of Theorem 1.3, which refers to testing H-freeness when H is bipartite, we cannot just rely on the fact that the blow-up is far from being H-free, since this may be the case even if the tested graph has no copies of H (e.g., an edge connecting two heavy vertices yields a complete bipartite graph with many edges). Instead, we distinguish edges that connect light vertices from the other edges, and deal with each case separately and differently. Edges that connect light vertices can be dealt with by a reduction to the standard case (since the sample is unlikely to hit the corresponding clouds twice), whereas the number of heavy vertices is small (and so enumerating all relevant configurations is feasible). A similar strategy is employed in the proof of Theorem 2.13, which refers to testing minor-freeness (in the VDF dense graph model).

1.3.3 Strong testability in the VDF bounded-degree graph model

The strong testers asserted in Theorem 1.4 are obtained by adaptations of the algorithms (and the analyses) that are used in the standard model. A common theme in all the original testers is starting several searches at random vertices. Here, depending on the property, we sometimes start these searches not at a vertex obtained from the vertex-sampling device but rather at a related vertex (e.g., a uniformly selected neighbor of the sampled vertex). The reason for this modification is that, in the analysis, the cost of omitting or adding an edge is related to the probability weight of both its endpoints, whereas the violation of a tested condition may occur only when starting the search at one of these endpoints.

For example, the distance from having no vertex of degree d' is not proportional to the weight of

[16] Given [26, Thm. 1.18] leaves us with the task of generalizing the rest of the analysis of the subgraph tester (cf., e.g., [14, pp. 190-194]).

such vertices, but is rather proportional to the weight of these vertices and their neighbors. Hence, the tester should not be confined to checking the degree of sampled vertices; it should rather check also the degree of random neighbors of sampled vertices.

We note that modifying the starting point of the search may not suffice. In some cases, the search itself is modified: For example, in the case of testing k-path freeness, rather than conducting a single k-step random walk, we conduct two random walks of total length summing up to k.

In general, the fact that the analysis refers to the probability weight of certain vertices rather than to their number complicates the analysis of almost all testers. For example, unlike in the standard model (cf., [14, Exer. 9.5]), it does not hold that *if a graph is close to being connected and to being Eulerian, then it is close to a connected Eulerian graph* (see Footnote 59).

1.4 Discussion

The (standard) study of testing property of graphs, as reviewed in [14, Chap. 8-10], is extremely idealized. It postulates that the vertex-set of the graph equals a set of the form $[n]$, where n is known *a priori* (i.e., is given as an explicit input to the tester). This is the case both in the dense graph model (introduced in [17] and reviewed in [14, Chap. 8]), and in the bounded-degree graph model (introduced in [19] and reviewed in [14, Chap. 9]).[17] But it is hard to imagine any realistic setting in which the vertex-set is actually of this form (or can be easily put in 1-1 correspondence to this form).

Addressing this concern, it was suggested (by the author [15]) to relax the model and only require that the tester be given the size of the vertex-set as well as a sample of uniformly (and independently) distributed elements in this set, and it was shown that this model is essentially equivalent to the standard one.[18] Here, we take an additional step towards realistic applications, by waiving the requirement that the samples of the vertex-set be uniformly distributed in it.

As hinted at the very beginning of Section 1.1, the models suggested in this paper are motivated by settings in which uniformly sampling the vertex-set of the graph of interest is not feasible. Instead, it is feasible to obtain random vertices drawn according to some distribution $\mathcal{D}$ (and, in addition, obtain answers to the usual graph-queries). This is because some process (or application) of interest refers to (or embeds or emulates) a huge graph; in particular, the process generates random vertices according to the distribution $\mathcal{D}$, and answers adjacency (or incidence) queries regarding the graph. We would like to know whether this huge graph 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 $\mathcal{D}$.

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 encounted (by he application). Hence, the distance of a graph to the property represents the relative importance of the "part of the graph" that violates the property. When the application supports adjacency queries (to a dense graph), it is reasonable to postulate that the importance of an edge/non-edge is proportional to the product of the (probability) weights of its endpoints. When the application supports incidence queries (to a bounded-degree graph), it is reasonable to postulate that the importance of an edge/non-edge is proportional to the sum of the (probability) weights of its endpoints.

The discrepancy between these two cases reflects the discrepancy between the two types of models (equiv., queries): In the dense graph model one encounters a edge/non-edge by visiting both its endpoints, whereas in the bounded-degree graph model an edge/non-edge is encountered when visiting one of its two endpoints. Indeed, in the dense graph model we envision applications that refer to a symmetric relation, represented as a graph, whereas in the bounded-degree graph model we envision a

[17] The same hols also and in the general graph model (introduced in [30, 25] and reviewed in [14, Chap. 10]).

[18] Actually, the foregoing is just one incarnation of a more general framework suggested in [15]. In general, the tester may be given some partial information about the vertex-set rather than its exact size. Other examples may include an approximation to the size of the vertex-set or nothing.

network with a bounded number of “ports” at each site (vertex). We do not claim that our definition of distance (equiv., level of importance of edges/non-edges) is the only reasonable one, but rather than it is a reasonable one. (Still, a generalization of the definition of importance of edges/non-edges, for the bounded-degree graph model, is presented below.)

We stress that we seek universal testers for the setting envisioned above; that is, testers that perform well for any vertex-distribution $\mathcal{D}$ (rather than testers that are tailored to a specific distribution $\mathcal{D}$). We call such testers vertex-distribution-free (abbrev., VDF), and would welcome suggestions for a less cumbersome term (which led us to use the abbreviation in most places).

Note that, in our models, the tester is not given the size of the vertex-set as an explicit input. This is quite natural given the foregoing motivation. Furthermore, it seems that in natural cases, the VDF testers are unlikely to benefit from knowledge of the size of the vertex-set.

A generalized version of the bounded-degree graph model. We have motivated the definition of the importance of an edge/non-edge in our (bounded-degree graph VDF) model by implicitly referring to the probability (under $\mathcal{D}$) that one of its endpoints is encountered by the application. This presumes that the application encounters vertices only by generating them at random according to the distribution $\mathcal{D}$. However, in the bounded-degree graph setting, the application can encounter a vertex also by selecting a neighbor of a previously encountered vertex. Postulating that the application takes walks of length at most t, we suggest the t-removed VDF model in which the importance of a vertex v is defined as proportional to the sum of the probabilities of all vertices that are at distance at most t from v. We extend our positive results regarding the VDF bounded-degree graph model to the t-removed VDF model, incurring an overhead that is exponential in t. For details see Section 3.5.

2 The Dense Graph Model

In this section, we generalize the notion of property testing in the dense graph model (a.k.a. the adjacency predicate model, which was introduced in [17] and is reviewed in [14, Chap. 8]). The generalized model is tentatively called the vertex-distribution-free (VDF) dense graph model.

In this model, a graph of the form $G = (V, E)$ is represented by its adjacency predicate $g : V \times V \to \{0, 1\}$; that is, $g(u, v) = 1$ if and only if u and v are adjacent in G (i.e., $\{u, v\} \in E$). Indeed, since g represents a (simple) graph, it holds that $g(u, v) = g(v, u)$ and $g(v, v) = 0$ for every $u, v \in V$.

The tester is given oracle access to the representation of the input graph (i.e., to the adjacency predicate g) as well as to a device, denoted $\mathcal{D}$, that returns identically and independently distributed elements in the graph's vertex-set. This distribution is also denoted $\mathcal{D}$. In addition, the tester gets the proximity parameter, ϵ, as explicit input. Following [15], we consider the case that the tester does not obtain any information about V as explicit input; this means that the tester better query the graph only on pairs of vertices that have been provided before by the sampling device $\mathcal{D}$ (see Proposition 2.2).

Distance between graphs is measured in terms of their foregoing representation and with reference to the distribution $\mathcal{D}$; that is, the distance between the graphs that are represented by the adjacency predicates $g : V \times V \to \{0, 1\}$ and $g' : V \times V \to \{0, 1\}$ is defined as

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

Note that the distance also accounts for the reflexive pairs (i.e., (v, v) for $v \in V$),[19] which means that $\delta_{\mathcal{D}}(g, g') < 1$ holds whenever both g and g' represent simple graphs. For a graph property Π and a graph represented by the adjacency predicates $g : V \times V \to \{0, 1\}$, we let $\delta_{\mathcal{D}}^{\Pi}(g)$ denote the minimum of $\delta_{\mathcal{D}}(g, g')$ taken over all adjacency predicates $g' : V \times V \to \{0, 1\}$ that represent graphs in Π. (We assume

[19]This convention is also used in many other sources that refer to the standard dense graph model; see discussion in [14, Sec. 8.2.1].

for simplicity that Π contains some graphs with vertex-set V; otherwise, one may define $\delta_{D}^{\Pi}(g) > 1$.) When G is the graph represented by g, we may write $\delta_{\mathcal{D}}^{\Pi}(G)$ instead of $\delta_{\mathcal{D}}^{\Pi}(g)$. When the property Π is clear from the context, we may omit it from the notation (and write $\delta_{\mathcal{D}}(\cdot)$ instead of $\delta_{\mathcal{D}}^{\Pi}(\cdot)$).

Definition 2.1 (VDF testing in the dense graph model): *Let Π be a property of graphs. A* VDF tester for the graph property Π (in the dense graph model) *is a probabilistic oracle machine T that is given access to two oracles, an adjacency predicate $g : V \times V \to \{0,1\}$ and a device* (denoted $\mathcal{D}$) *that samples in V according to an arbitrary distribution $\mathcal{D}$, and satisfies the following two conditions* (for all sufficiently large V).[20]

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

2. *Given $\epsilon > 0$ and oracle access to any graph $G = (V, E)$ that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, if $g : V \times [d] \to V \cup \{\perp\}$ satisfies $\delta_{\mathcal{D}}^{\Pi}(g) > \epsilon$, then it holds that $\Pr[T^{g,\mathcal{D}}(\epsilon)=0] \geq 2/3$.*

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

Definition 2.1 was outlined by Goldreich, Goldwasser, and Ron [17, Sec. 10.1], where it was viewed as a restricted case of truly distribution-free testing. That is, the conceptual focus and starting point of relevant paragraph of [17, Sec. 10.1] is the distribution on vertex-pairs, whereas our focus and starting point is the distribution on vertices. Hence, in [17, Sec. 10.1] the foregoing definition is viewed as special case in which the distribution on pairs is a product of some distribution with itself (and the tester gets samples of that product distribution). In contrast, we view the distribution on vertices as the pivot of the definition, and derive the distance measure from it.

The query complexity of a tester accounts for the total number of queries made to both the graph G and the sampling device $\mathcal{D}$. This complexity is measured in terms of the proximity parameter $\epsilon > 0$ and label-invariant parameters of the distribution $\mathcal{D}$ (e.g., its support-size, its collision probability, its min-entropy, etc).[21] Indeed, the latter parameters replace the number of vertices of the graph, which is used as a size parameter in the standard model. We shall focus on testers having query complexity that depends only on ϵ; we tentatively call such testers strong.

As hinted above, it makes little sense for the tester to query the graph on pairs of vertices that have not been provided before by the sampling device $\mathcal{D}$. In fact, we may assume, without loss of generality, that the tester makes queries only to pairs of vertices that appeared as answers of the sampling device.

Proposition 2.2 (avoiding illegal queries in the VDF dense graph model): *Suppose that Π can be tested by making at most s sampling requests and at most q queries to the graph, where both complexities depend on the proximity parameter and on label-invariant parameters of the distribution. Then, Π can be tested by making at most $3s$ sampling requests and making only queries to pairs of vertices that appear in these samples.*

[20] This relaxation, which allows the tester to fail on some tiny graphs, was not made in the first posting of this work [16]. It allows discarding small exceptional input graphs, which are of no interest to us anyhow (per our motivation). Such provision is unnecessary in the standard model, since there the tester obtains the size of the graph (and can afford to retrieve the entire graph in case it is small). The relaxation can be captured within the framework of [15, Def. 1.1], by providing as partial information a bit indicating whether or not the graph is large enough.

[21] A parameter p of (discrete) distributions is called label-invariant if for every distribution $\mathcal{D}$ over $\{0,1\}^*$ and every bijection π of $\{0,1\}^*$ to itself it holds that $p(\mathcal{D})$ equals $p(\pi \circ \mathcal{D})$. In other words, the parameter is a function of the histogram of the distribution, where the histogram of $\mathcal{D}$ is the multi-set $\{\mathcal{D}(e) : \mathcal{D}(e) > 0\}$.

The specific query complexity bound, denoted q, is immaterial here; what matters is that the query complexity (like the sampling complexity) depends only (on the proximity parameter and) on label-invariant parameters of the distribution.

Proof: Let T be a tester as in the hypothesis. We construct a tester T' that, on input $g' : V' \times V' \to \{0,1\}$ and $\mathcal{D}'$, invokes T on related inputs $g : V \times V \to \{0,1\}$ and $\mathcal{D}$, which are generated on-the-fly based on g' and V'. Specifically, T' generates on the fly a random bijection π of $V' \uplus [100q \cdot |V'|]$ to $[(1+100q) \cdot |V'|]$, while defining $V = \pi(V') \stackrel{\rm def}{=} \{\pi(v') : v' \in V'\}$ and $g(\pi(v'), \pi(w')) = g'(v', w')$ and $\mathcal{D}(\pi(v')) = \mathcal{D}'(v')$. It essentially emulates an execution of T with oracles g and $\mathcal{D}$ as follows.

- When T asks for a sampled vertex, T' obtains $v' \leftarrow \mathcal{D}'$ and answers with $\pi(v')$ if π is defined on v' (which means that v' did not appear as an answer to some previous sampling request), and with a random unused value otherwise. That is, if U denotes the set of vertices used so far and $\pi(v') \in U$, then the answer is $\pi(v')$; otherwise, the answer v is selected uniformly at random in $[(1+100q) \cdot |V'|] \setminus U$, and $\pi(v') = v$ is added to U.

- When T makes the query (u, w), we return $g'(\pi^{-1}(u), \pi^{-1}(w))$ if the relevant π^{-1}-values are defined (which means that the values v and w appeared as answers to previous sampling requests), and return a special error symbol otherwise (indicating that (u, w) is not in the domain of g). We may assume that T rejects in such a case. (Alternatively, we can define T' to reject in this case.)

For the sake of the analysis, we assume that when the emulation is completed, the undefined values of π are selected at random (among the unused ones).

The key observation is that, on input g' and $\mathcal{D}'$, with very high probability (over the choice of π), algorithm T' emulates an execution of T on input g and $\mathcal{D}$ (which are defined via a random bijection π such that $\mathcal{D}(v) = \mathcal{D}'(\pi^{-1}(v))$ and $g(u, w) = g'(\pi^{-1}(u), \pi^{-1}(w))$). The deviation is due to the case that a query (u, w) was made (by T) although some $v \in \{u, w\}$ was not obtained as a sample and yet $\pi^{-1}(v) \in V'$ (whereas T' behaved as if $v \notin \pi(V')$). The probability of this event is at most $2q \cdot |V|/(100q|V|)$. It follows that the probability that T' accepts a graph G' is sandwiched between $p - 0.02$ and p, where p denotes the probability that T accepts the random homomorphic copy $G = \pi(G')$ of G'.

Note that T' never makes a query to a pair of vertices that were not provided by the sampling device. Using mild error reduction to compensate for the additional error of 0.02, the claim follows. (Indeed, invoking T' three times and ruling by majority will do.) ■

Organization of the rest this section. We first show that it is possible to transform testers for the VDF version of the dense graph model into ones that have one-sided error, while incurring an overhead that is much lower than what is possible in the standard (dense graph) model. Using lower bounds on the complexity of testing with one-sided error in the standard model, this yields lower bounds on the testability in the VDF model. Next, focusing on properties that are strongly testable with one-sided error in the standard (dense graph) model, we try to extend these testers to the VDF model. Specifically, in Section 2.2, building on the results of [17], we present strong testers for the relevant subclass of the graph partition problems (e.g., k-colorability). Likewise, in Section 2.3, building on results of [3], we present strong testers for subgraph-freeness properties (e.g., triangle-freeness). Put together, these results establish (the two parts of) Theorem 1.3. Lastly, in Section 2.4, we study two classes that are easy to test in the standard model, using some of them to establish the "dense graph model" part of Theorem 1.2 (i.e., strong testability with one-sided error in the standard model does not suffice for strong testing in the VDF model).

2.1 One-sided error in the VDF model

In contrast to the situation in the standard model (see, e.g., the gap between the complexities of general versus one-sided error testing of ρ-clique [17]), one-sided error comes almost for free in the VDF model. This is the case since any strong tester in the VDF model can be transformed into a one-sided error tester (for this model) while at most squaring the sample complexity. For sake of simplicity, we state the transformation for the case of strong testers, but the assertion holds for general testers (with complexity that may depend both on ϵ and on some natural parameters of the vertex distribution such as support size).

Theorem 2.3 (one-sided error VDF testing reduces to general VDF testing): *Let Π be a graph property that can be tested using $s(\epsilon)$ (vertex) samples in the VDF dense graph model, where ϵ denotes the proximity parameter. Then, Π has a one-sided error tester of sample complexity $O(s(\epsilon)^2)$ in the VDF dense graph model.*

Theorem 2.3 implies that properties that do not have a strong one-sided error tester in the standard model cannot be strongly tested in the VDF model (see Corollary 2.4).

Proof: Let T be a (general) tester of sample complexity $s(\epsilon)$ for Π in the current (VDF) model. Recall that by Proposition 2.2, we may assume, without loss of generality, that T does not query the graph on unsampled vertices. Relying on this fact, we present a one-sided error tester for Π in the current model. On input parameter $\epsilon > 0$, and oracle access to a graph $G = (V, E)$ and a sampling device $\mathcal{D}$, the claimed tester operates as follows.

1. The tester obtains $t = O(s(\epsilon)^2)$ samples, denoted $v_1, ..., v_t$, from the distribution $\mathcal{D}$.

   Note that the v_i's need not be distinct; that is, we may have $v_i = v_j$ for some $i \neq j$.

2. Letting $s = s(\epsilon)$, for every sequence $(i_1, ..., i_s)$ over $[t]$ and every possible random-pad r of T, the algorithm invokes $T(\epsilon)$ on randomness r and oracle access to G, while providing v_{i_j} as the j^{th} sampled vertex (i.e., as an answer to the j^{th} sampling request). That is, T is invoked on input ϵ, and provided access to G, but its randomness is set to r and the s samples it expects to receive from the sampling device are set to $v_{i_1}, ..., v_{i_s}$.

3. The algorithm accepts if and only if a majority of the invocations performed in Step 2 accept.

Since T only queries pairs of vertices that are provided by its sampling device, our algorithm queries the graph on pairs of vertices that have appeared in its own sample, $v_1, ..., v_t$. Hence, the sample complexity of our algorithm is $t = O(s(\epsilon)^2)$ and its query complexity is $\binom{t}{2} = O(s(\epsilon)^4)$. (Note that no claim is made regarding the time complexity of our algorithm, which is indeed exponential in the sample and randomness complexities of T.) We now show that this algorithm constitutes a one-sided error tester for Π in the current model.

Suppose that $G \in \Pi$. Note that our algorithm performs no random choices, and the probability space of its possible executions consists solely of the t samples drawn from $\mathcal{D}$. Hence, we have to show that, under each such choice, our tester accepts. Fixing such a sequence of samples $\overline{v} = (v_1, ..., v_t)$, we consider the distribution, denoted $\mathcal{U}(\overline{v})$, defined by selecting uniformly $i \in [t]$ and outputting v_i. By the hypothesis regarding T, we have $\Pr[T^{G,\mathcal{U}(\overline{v})}(\epsilon) = 1] \geq 2/3$. Note that the probability space here consists of all possible choices of s samples drawn (with repetitions) from $\mathcal{U}(\overline{v})$ (which correspond to all choices of $(i_1, ..., i_s) \in [t]^s$) and all possible random-pads of T. Hence, at least two third of the invocations performed in Step 2 are accepting, and our algorithm accepts.

We now turn to the case that $G = (V, E)$ is ϵ-far from Π (with respect to the distribution $\mathcal{D}$), and consider the execution of our algorithm when given samples drawn from $\mathcal{D}$. By the hypothesis regarding T, we have $\Pr[T^{G,\mathcal{D}}(\epsilon) = 1] \leq 1/3$. Assuming that $t \geq 100 \cdot s(\epsilon)^2$, it follows that selecting

$s = s(\epsilon)$ random samples (with repetitions) from a random t-sequence of samples drawn from $\mathcal{D}$ yields a distribution that is 0.01-close to the distribution obtained by selecting s random samples from $\mathcal{D}$, since the first (resp., second) case corresponds to selecting uniformly at random a multi-set of size s in $[t]$ (resp., a set of size s in $[t]$).[22] Hence,

$$\mathrm{Exp}_{v_1,...,v_t \leftarrow \mathcal{D}}\left[\Pr[T^{G,\mathcal{U}(v_1,..,v_t)}(\epsilon)=1\right] \leq \Pr[T^{G,\mathcal{D}}(\epsilon)=1] + 0.01 < 0.35.$$

(This observation originates from the proof of [17, Cor. 7.2].) An averaging argument implies that

$$\Pr_{v_1,...,v_t \leftarrow \mathcal{D}}\left[\Pr[T^{G,\mathcal{U}(v_1,...,v_t)}(\epsilon)=1] > 0.5\right] \leq 0.7.$$

Hence, for at most 70% of the samples obtained in Step 1, more than half of the invocations performed in Step 2 are accepting, which implies that our algorithm accepts with probability at most 0.7. Using moderate error reduction, the theorem follows. ■

Using Theorem 2.3 towards establishing lower bounds in the VDF model. As noted above, Theorem 2.3 implies that properties that do not have a strong one-sided error tester in the standard model cannot be strongly tested in the VDF model.

Corollary 2.4 (lower bounds via reduction from one-sided error testing): *Let Π be a graph property that can be tested using $s(\epsilon)$ vertex samples in the VDF dense graph model, where ϵ denotes the proximity parameter. Then, Π has a one-sided error tester of query complexity $\mathrm{poly}(s(\epsilon))$ in the standard dense graph model. The claim holds even if the VDF model tester is given the size of the graph as auxiliary input. Furthermore, if the sample complexity of the VDF tester depends also on the support size and has the form $s(m, \epsilon)$, where m is the size of the support, then Π has a one-sided error tester of query complexity $\max_{i\in[n]} \mathrm{poly}(s(i, \epsilon))$ in the standard dense graph model, where n denotes the number of vertices in the tested graph.*

Hence, lower bounds on the complexity of one-sided error testers in the standard dense graph model yield lower bounds on testers in the VDF (dense graph) model.

Proof: Theorem 2.3 yield one-sided error tester for Π in a variant of the standard model in which the tester obtains uniformly distributed samples of the vertex-set (rather than being given a succinct description of the vertex-set). The discrepancy between this tester and the standard model is that in the standard model the tester is given n (and ϵ) as explicit inputs as well as oracle access to $G = ([n], E)$, but does not get access to a sampling device for $\mathcal{U}([n])$. Nevertheless, given n, one can easily emulate a sampling device for $\mathcal{U}([n])$, obtaining a tester in the standard model.

The furthermore-part is proved by first observing that Theorem 2.3 can be extended to testers of sample complexity that depends both on the support size and the proximity parameter. The key observation is that the support size of the distribution $\mathcal{U}(v_1, ..., v_t)$, where $t = \max_{i\in[n]}\{O(s(i, \epsilon)^2)\}$, is at most n (i.e., the support size of $\mathcal{D} = \mathcal{U}([n])$, which in turn is the number of vertices in the graph). Furthermore, letting m denote the support size of $\mathcal{U}(v_1, ..., v_t)$, it holds that $m \leq n$ and $O(s(m, \epsilon)^2) \leq t$ follows. ■

Concrete lower bounds. Combining the furthermore-part of Corollary 2.4 with the linear lower bound on the complexity of one-sided error testing of ρ-Clique and ρ-CUT in the standard dense graph model (see [17, Sec. 10.1.6]), we get the following

[22] In the first case, we select $v_1, ..., v_t \leftarrow \mathcal{D}$ and $i_1, ..., i_s \in [s]$ and output $v_{i_1}, ..., v_{i_s}$, whereas the second case may be viewed as selecting $v_1, ..., v_t \leftarrow \mathcal{D}$ and outputting $v_{i'_1}, ..., v_{i'_s}$ such that $i'_1, .., i'_s$ are s distinct elements selected arbitrarily in $[t]$.

Corollary 2.5 (lower bounds on testing ρ-Clique and ρ-CUT): *Testing ρ-Clique and ρ-CUT in the VDF dense graph model requires query complexity that is polynomially related to the size of the graph, even for $\epsilon = 0.1$.*

A similar lower bound hold for all non-trivial graph partition problems that do not satisfy the conditions of Theorem 2.7. Similarly, stepping beyond the class of graph partition problems, we obtain

Corollary 2.6 (lower bounds on testing degree-regularity) *Testing degree-regularity in the dense graph VDF model requires query complexity that is polynomially related to the size of the graph, even for $\epsilon = 0.1$.*

2.2 Testing graph partition properties

Definition 2.1 was outlined by Goldreich, Goldwasser, and Ron [17, Sec. 10.1], who observed that some of their results extend to this model. Specifically, they claimed that their tester of k-Colorability extends to the current setting, and sketched a proof of this claim.[23] We show that the claimed result (regarding testing k-Colorability) can be extended to any graph k-partition problem that is strongly testable with one-sided error in the standard (dense graph) model.

Loosely speaking, a graph partition problem calls for partitioning the graph into a specified number of parts such that the sizes of the parts fit the specified bounds and ditto with respect to the number of edges between parts. More specifically, each graph partition problem (resp., property) is specified by a number $k \in \mathbb{N}$ and a sequence of intervals (which serve as parameters of the problem). A graph $G = (V, E)$ is a yes-instance of this problem (resp., has the corresponding property) if there exists a k-partition, $(V_1, ..., V_k)$, of V such that

1. For each $i \in [k]$, the density of V_i fits the corresponding interval (specified in the sequence of parameters).

2. For each $i, j \in [k]$ (including the case $i = j$), the density of edges between V_i and V_j fits the corresponding interval.

This framework was presented in [17], where only absolute density bounds were considered (e.g., $\frac{|V_i|}{|V|} \leq 0.1$ and $\frac{|E(V_i,V_j)|}{|V|^2} \geq 0.1$). Here, following [14, Sec. 8.3] and [28], we consider also relative edge-density bounds (e.g., $\frac{|E(V_i,V_j)|}{|V_i|\cdot|V_j|} \geq 0.8$).

Recall that each graph partition problem can be tested within query complexity $\text{poly}(1/\epsilon)$ in the standard dense graph model (see [17], extended by [28]). However, as shown in [28] (which extends [23, Thm. 3]), only some of these properties can be strongly tested with *one-sided error* in the standard dense graph model (i.e., can be tested with one-sided error within query complexity that depends only on ϵ). Our result refers to this class, which consists of sets of graphs that can be k-partitioned such that each part is either required to be a clique or an independent set, and each pair of parts is either required to have all possible edges or have no edges at all or be arbitrary. (This class is effectively defined in the hypothesis of Theorem 2.7; it consists of all the properties $\Pi_{\alpha,\beta}$ defined there.)

Theorem 2.7 (testing a subclass of graph partition problems): *Let $k \in \mathbb{N}$. For $\alpha = (\alpha_i)_{i\in[k]} \in \{0,1\}^k$ and $\beta = (\beta_{i,j})_{i<j} \in \{0,1,2\}^{\binom{k}{2}}$, let $\Pi_{\alpha,\beta}$ denote the set of all graphs $G = (V, E)$ such that V can be*

[23] They also claimed that the results regarding ρ-Clique and ρ-Cut can be similarly extended, but it seems that the properties that they referred to are weighted versions of ρ-Clique and ρ-Cut (which refer to the weights of cliques and cuts with respect to the probability distribution $\mathcal{D}$). That is, these properties depends on the distribution over the vertex-set, whereas in Definition 2.1 the graph property is fixed, regardless of the said distribution. Note that Corollary 2.4 implies that the positive result does not extend to the unweighted versions of ρ-Clique and ρ-Cut.

partitioned into k (possibly empty) *parts, denoted* $V_1,..,V_k$*, that satisfy the following conditions:*[24]

1. *For each* i*, if* $\alpha_i = 1$ *then the subgraph of* G *induced by* V_i *is a clique, and otherwise* (i.e., if $\alpha_i = 0$) *it is an independent set.*

2. *For every* $i < j$*, the set of edges connecting* V_i *and* V_j *is empty if* $\beta_{i,j} = 0$*, contains all edges if* $\beta_{i,j} = 1$*, and is arbitrary otherwise* (i.e., if $\beta_{i,j} = 2$).

Then, $\Pi_{\alpha,\beta}$ *can be tested in the VDF dense graph model in query complexity* $\text{poly}(1/\epsilon)$ *and time complexity* $\exp(\text{poly}(1/\epsilon))$ (with one-sided error).

We stress that the foregoing graph properties (along with the two trivial properties)[25] are the only *graph partition properties* that are strongly testable with one-sided error in the standard dense graph model (see [28], which extends [23, Thm. 3]). Furthermore, the stated complexities match the best results known in the standard dense graph model.

Proof: We reduce testing $\Pi = \Pi_{\alpha,\beta}$ in the VDF dense graph model to testing Π in the standard dense graph model. The reduction is basically the one outlined in [17, Sec. 10.1]; that is, in the case of k-Colorability, it suffices to check that a sample of $\text{poly}(1/\epsilon)$ vertices drawn from $\mathcal{D}$ induces a k-colorable subgraph. As shown next, this strategy works in the case of arbitrary Π, provided that the collision probability of $\mathcal{D}$ is $o(Q(\epsilon/2)^{-2})$, where Q is the query complexity (or rather the sample complexity) of the original tester, and needs to be somewhat revised otherwise (i.e., to handle the case of arbitrary $\mathcal{D}$).

By [23, Thm. 2], we may assume (w.l.o.g.) that the standard model tester, denoted T, takes a sample of $q = O(Q(\epsilon/2))$ vertices and decides according to the (unlabeled) subgraph that is induced by these vertices. Now, when given access to a graph $G = (V, E)$ and a vertex-sampling device $\mathcal{D}$, our tester takes a sample of q vertices from $\mathcal{D}$, queries G on the corresponding $\binom{q}{2}$ vertex pairs, and invokes the residual decision procedure of T on the corresponding q-vertex subgraph. (This description presumes that no vertex appears twice in the sample.)

To analyze the behavior of our tester, we consider a graph G' that is obtained by replacing each vertex v of G with a set (or a "cloud") C_v of $\lfloor \mathcal{D}(v) \cdot N \rceil$ vertices, for a sufficiently large N (e.g., $N = 4|V|/\epsilon$), and placing a complete bipartite graph between C_u and C_v if u and v are connected in G. (Indeed, if u and v are not connected, then no edges are placed between C_u and C_v.) Note that we avoided specifying the subgraph induced by the individual C_v's; it will be either a clique or an independent set.

Assuming that no vertex of G has appeared twice in the sample of q vertices (drawn from $\mathcal{D}$), our tester emulates an execution of T on G', since the subgraph of G' induced by a sample of q uniformly distributed vertices is distributed identically to the subgraph of G induced by a sample of q vertices drawn from $\mathcal{D}$ (with no repetitions). It remains to show that the distance of G from Π (under $\mathcal{D}$) equals the distance of the best G' from Π (up to an additive term of $\epsilon/2$). (That is, at this point, we consider all possible G' that fit the foregoing definition; that is, for each set C_v, we let C_v be either a clique or an independent set.)

Claim 2.7.1 (relating the distances of G and G' to Π): *For every* $\sigma \in \{0,1\}^V$*, we let* G'_σ *denote the graph obtained from* G *by replacing each vertex* v *with a* $\lfloor \mathcal{D}(v) \cdot N \rceil$*-vertex set* C_v *such that the subgraph of* G'_σ *induced by* C_v *is a clique if* $\sigma_v = 1$ *and is an independent set otherwise, and replacing each edge* $\{u, v\}$ *in* G *with a complete bipartite graph between* C_u *and* C_v*. Then, the following hold.*

1. *If* $G \in \Pi$*, then for some* $\sigma \in \{0,1\}^V$ *it holds that* $G'_\sigma \in \Pi$.

[24] If for some $i \in [k]$ the set V_i is empty, then the corresponding conditions are considered satisfied regardless of the values of α_i and the $\beta_{i,j}$'s.

[25] The trivial graph properties are the property consisting of all graphs and the empty property.

2. *In general,*

$$\delta_{\mathcal{D}}^{\Pi}(G) = \min_{\sigma\in\{0,1\}^V}\{\delta_{\mathcal{D}'}^{\Pi}(G'_\sigma)\} \pm \left(\frac{|V|}{N} + \sum_{v\in V}\mathcal{D}(v)^2\right) \qquad (2)$$

where $\mathcal{D}'$ denotes the uniform distribution over $V' \stackrel{\text{def}}{=} \cup_{v\in V} C_v$. Furthermore, for every $\sigma \in \{0,1\}^V$, it holds that

$$\delta_{\mathcal{D}}^{\Pi}(G) \le 7k \cdot \delta_{\mathcal{D}'}^{\Pi}(G'_\sigma) + \frac{|V|}{N}. \qquad (3)$$

Hence, assuming that the collision probability of $\mathcal{D}$ is small (i.e., $\sum_{v\in V}\mathcal{D}(v)^2 \le \epsilon/4$) and that N is sufficiently large (i.e., $N \ge 4|V|/\epsilon$), the deviation in Eq. (2) is upper-bounded by $\epsilon/2$. We stress that the upper-bound in Eq. (3) is independent of the collision probability of $\mathcal{D}$. (Although the factor of $7k$ lost in this upper-bound is immaterial, we wonder if it can be avoided (or at least be replaced by a factor independent of k).)

Proof: Suppose that $G \in \Pi$ and let $(V_1, .., V_k)$ be a k-partition $(V_1, ..., V_k)$ that witnesses this fact. Then, for every $i \in [k]$ and $v \in V_i$, setting $\sigma_v = \alpha_i$ we infer that $G'_\sigma \in \Pi$ by using the corresponding k-partition of V' (i.e., $V'_i = \bigcup_{v\in V_i} C_v$ for every $i \in [k]$), where α_i is the i^{th} intra-part parameter of $\Pi = \Pi_{\alpha,\beta}$. Applying the same argument to the partition that witnesses the value of $\delta_{\mathcal{D}}^{\Pi}(G)$, shows that there exists $\sigma \in \{0,1\}^V$ such that $\delta_{\mathcal{D}'}^{\Pi}(G'_\sigma) \le \delta_{\mathcal{D}}^{\Pi}(G) + |V|/N$, where the error term is due to rounding errors (that arise in determining the sizes of the C_v's). Hence, we focus on establishing the opposite inequality; that is, for any $\sigma \in \{0,1\}^G$, we upper-bound $\delta_{\mathcal{D}}^{\Pi}(G)$ in terms of $\delta_{\mathcal{D}'}^{\Pi}(G'_\sigma)$.

Fixing σ and a k-partition $(V'_1, ..., V'_k)$ that witnesses the value of $\delta_{\mathcal{D}'}^{\Pi}(G'_\sigma)$, consider a random k-partition of V, denoted $(X_1, ..., X_k)$, obtained by assigning each vertex $v \in V$ to the i^{th} cloud with probability $|V'_i \cap C_v|/|C_v|$. Letting $X'_i = \bigcup_{v\in X_i} C_v$, observe that the expected number of vertex-pairs in the k-partition $(X'_1, ...X'_k)$ that belong to different C_v's and violate the constraints of Π equals the number of vertex-pairs in the k-partition $(V'_1, ...V'_k)$ that belong to different C_v's and violate the constraints of Π.[26] This ignores the contribution of pairs of vertices that belong to the same C_v, which is at most $\sum_{v\in V}|C_v|^2$. Hence, there exists a partition $(X_1, ..., X_k)$ that supports the claim $\delta_{\mathcal{D}}^{\Pi}(G) \le \delta_{\mathcal{D}'}^{\Pi}(G'_\sigma) + |V|/N + \sum_{v\in V}\mathcal{D}(v)^2$, where the $|V|/N$ term is due to rounding errors.

The alternative bound of Eq. (3) is proved by first modifying the partition $(V'_1, ..., V'_k)$ such that, for every $v \in V$ and $i \in [k]$, either $|C_v \cap V'_i| \ge |C_v|/3k$ or $|C_v \cap V'_i| = 0$. The modification is performed by distributing, for every $v \in V$, the elements of the $C_v \cap V'_i$'s that violate the claim (i.e., i's such that $|C_v \cap V'_i| < |C_v|/3k$) among the V'_j's that satisfy the claim (in proportion to the sizes of these $C_v \cap V'_j$'s). This may increase the number of violations of the partition $(V'_1, ..., V'_k)$ by a factor of at most $(1 + \frac{1/3}{2/3})^2 = 9/4$. Next, we consider the random assignment performed in the previous paragraph, and observe that the expected contribution of pairs of vertices that belong to the same C_v to violations w.r.t $(X'_1, ..., X'_k)$ is at most $3k$ times larger than their contribution to violations w.r.t $(V'_1, ..., V'_k)$, since the expected contribution of pairs in C_v is at most $\sum_{i\in[k]} \frac{|C_v\cap V'_i|}{|C_v|} \cdot |C_v|^2 \le 3k \cdot \sum_{i\in[k]} |C_v \cap V'_i|^2$. (Indeed, here we use the fact that either $|C_v \cap V'_i| \ge |C_v|/3k$ or $|C_v \cap V'_i| = 0$.)[27] Hence, $\delta_{\mathcal{D}}^{\Pi}(G) \le \frac{9}{4} \cdot 3k \cdot \delta_{\mathcal{D}'}^{\Pi}(G'_\sigma) + |V|/N$, and the claim follows. ■

The assumption that no vertex of G appears twice in the sample (which holds when the collision probability of $\mathcal{D}$ is $o(1/q^2)$) was crucial to asserting that our tester emulates the execution of T on G',

[26] For every distinct $u, v \in V$, the expected contribution of the pairs in $C_u \times C_v$ to violations wrt $(X'_1, ..., X'_k)$ equals the number of violations wrt $(V'_1, ..., V'_k)$. Specifically, fixing u and v, let $\gamma_{i,j} \in \{0,1\}$ indicate whether the density constraint regarding parts i and j (or regarding part $i = j$) is violated by the edge relation between u and v (e.g., $\gamma_{i,j} = 1$ if $\{u, v\} \in E$ and $\beta_{i,j} = 0$), the latter number equals $\sum_{i,j\in[k]:\gamma_{i,j}=1} |C_u \cap V'_i| \cdot |C_v \cap V'_j|$ whereas the expectation equals $\sum_{i,j\in[k]:\gamma_{i,j}=1} p_i q_j \cdot |C_u| \cdot |C_v|$, where $p_i = |C_u \cap V'_i|/|C_u|$ and $q_j = |C_v \cap V'_j|/|C_v|$.

[27] Indeed, either way $|C_v \cap V'_i| \cdot |C_v| \le 3k \cdot |C_v \cap V'_i|^2$ holds.

regardless of the choice of the subgraph (of G') induced by individual C_v's, since under this assumption each C_v is hit at most once (in the q trials). In contrast, when obtaining two samples of the same vertex v, our emulation should return two random elements of C_v, and the problem is that we should also tell T whether these vertices are adjacent or not. We solve this problem by trying both possibilities for each vertex v that may appear multiple times in a sample of q vertices. This requires figuring out the list of relevant (i.e., heavy) vertices, and ignoring executions of T in which vertices that are not in this list occur more than once. Details follow (starting with the following easy claim).

Claim 2.7.2 (listing all high probability vertices): *There exists an algorithm that given $t \in \mathbb{N}$ and a sampling device $\mathcal{D}$, works in $\widetilde{O}(t)$ time and outputs a list that, with probability at least* 0.99*, contains all elements in $\{v : \mathcal{D}(v) \geq 1/t\}$.*

The algorithm just takes $O(t \log t)$ samples and outputs the set containing all of them. (Alternatively, it may output all elements that appeared more than once.) Furthermore, intending to use the bound provided by Eq. (3) (rather than the one of Eq. (2)), we set $q = O(Q(\epsilon/8k))$ (rather than $q = O(Q(\epsilon/2))$). Our revised tester proceeds as follows.

Algorithm 2.7.3 (given oracle access to the adjacency predicate of a graph G and to a corresponding vertex-sampling device $\mathcal{D}$):

1. *Letting $q = O(Q(\epsilon/8k))$ and $t = 100q^2$, we invoke the algorithm of Claim 2.7.2, and denote the obtained list by L. Note that $|L| = \widetilde{O}(q^2)$.*
2. *We run, in parallel, $2^{|L|}$ copies of an amplified version of the tester T, where each amplified version runs $|L|$ sequential copies of T, while using the same sequence of $|L| \cdot q$ samples in all the parallel copies. The parallel copies are associated with different $\tau \in \{0,1\}^L$. In the copy associated with a fixed τ, we connect multiple occurrences of a vertex $v \in L$ in the sample if and only if $\tau_v = 1$, and discard executions* (of T) *in which a vertex not in L is sampled multiple times. Specifically:*
   - *We view the sample of $|L| \cdot q = \widetilde{O}(q^3)$ vertices drawn from $\mathcal{D}$ as consisting of $t' = |L|$ samples, denoted $S_1, ..., S_{t'}$, each of size q. The sample S_i* (viewed as a multi-set) *is called* good *if only vertices that appear in L occur multiple times in S_i. We let I denote the set of i's such that S_i is good.*
   - *For every $\tau \in \{0,1\}^L$ and $i \in I$, we invoke the residual decision procedure of T on a q-vertex graph obtained as follows: The vertices of this graph are the q samples in S_i, edges connect occurrences of the same vertex $v \in L$ if and only if $\tau_v = 1$, and edges connect occurrences of different vertices $u, v \in S_i$ if and only if these vertices are connected in G.*

   If for some $\tau \in \{0,1\}^L$, all $|I|$ invocation of T accepted, then we accept. Otherwise, we reject.

Note that the query complexity of Algorithm 2.7.3 is $\widetilde{O}(t) + t' \cdot \binom{q}{2} = \widetilde{O}(q^4)$, whereas its time complexity is $2^{t'} \cdot |I| \cdot \mathrm{Time}(T)$, which equals $\exp(\widetilde{O}(q^2)) \cdot \exp(O(q))$ (since the time complexity of T is exponential in its sample complexity [17, 28]).

We next observe that if $G \in \Pi$, then Algorithm 2.7.3 accepts with probability 1, since there exists a correct choice $\tau \in \{0,1\}^L$ that yields a corresponding graph $G' = G'_\tau \in \Pi$ that is being emulated in our invocations of T. (The string τ is determined by the location of the vertices of L in a good k-partition of G; that is, a k-partition that testifies for $G \in \Pi$.)

Lastly, we get to the case that G is ϵ-far from Π, and assume that the list L generated in Step 1 contains all vertices v such that $\mathcal{D}(v) \geq 1/t$. In this case, by Eq. (3), each possible $G' = G'_\tau$ is $\epsilon/8k$-far from Π, which implies that each of the parallel executions accepts with probability at most $0.35^{|I|}$, since considering only good samples biases the distribution on an execution by at most a 0.01 amount (because a sample is good with probability at least $1-\binom{q}{2}/t > 0.99$). Furthermore, with overwhelmingly high probability, it holds that $|I| > 0.9t'$. We conclude that our tester rejects G with probability at least $0.99 - 2^{t'} \cdot 0.35^{0.9t'} > 0.98$. ■

2.3 Testing subgraph-freeness properties

Corollary 2.4 guides us in the search for additional classes of properties that can be strongly tested in the VDF model: We should look at properties that are strongly testable with one-sided error in the standard dense graph model. We have already used a class of such properties in Section 2.2, and in this section we consider another such class – the class of subgraph-freeness properties, for which a strong tester with one-sided error (for the standard dense graph model) was presented in [3]. Adapting their result, we shall present a strong VDF-model tester for any subgraph-freeness property.

Actually, we present proximity oblivious testers (cf. [21]) for these properties, where the definition of proximity oblivious tester for a property Π is adapted to the VDF setting by requiring that the tester (makes a constant number of queries G and $\mathcal{D}$, and) rejects any graph G not in Π with probability that is related to $\delta_{\mathcal{D}}^{\Pi}(G)$, and always accepts graphs in Π. Specifically, the tester rejects with probability at least $\texttt{dpf}(\delta_{\mathcal{D}}^{\Pi}(G))$, where $\texttt{dpf} : (0,1] \to (0,1]$ is called the detection probability function. (Recall that $\delta_{\mathcal{D}}^{\Pi}(G) = \min_{G' \in \Pi}\{\delta_{\mathcal{D}}(G,G')\}$, where $\delta_{\mathcal{D}}$ is define in Eq. (1).)

Theorem 2.8 (testing subgraph-freeness properties): *For any fixed graph $H = ([h], F)$, let Π_H denote the set of H-free graphs; that is, graphs that do not contain a copy of H as a subgraph. Then, Π_H has a proximity oblivious testers, in the VDF dense graph model, with a detection probability function* $\texttt{dpf}(\delta) = 1/\texttt{T}(\text{poly}(1/\delta))$ *such that $\texttt{T}(m)$ is a tower of m exponents. Furthermore, if H is bipartite, then the detection probability function is* $\text{poly}(\delta)$.

Proof: Assuming, without loss of generality, that H has no isolated vertices, we reduce testing $\Pi = \Pi_H$ in the VDF (dense graph) model to testing Π in the standard dense graph model.[28] The reduction is similar to the one used in the proof of Theorem 2.7, but its analysis is different.

Recall that, on input G, the proximity oblivious tester T of [3] selects uniformly a set of h vertices and accepts if and only if the subgraph of G induced by this set does not contain a copy of H. Our tester will do just the same, except that it will use h samples obtained from $\mathcal{D}$; that is, given access to a graph G and a vertex-sampling device $\mathcal{D}$, our tester takes a sample of h vertices from $\mathcal{D}$, queries G on the corresponding vertex pairs, and accepts if and only if the subgraph observed contains no copy of H.

Clearly, our tester always accepts any H-free graph (i.e., it never rejects $G \in \Pi$), and so we focus on lower-bounding the probability that our test rejects $G = (V, E)$ as a function of $\delta = \delta_{\mathcal{D}}^{\Pi}(G) > 0$. Suppose that $\sum_{v \in V} \mathcal{D}(v)^2 < \texttt{dpf}(\delta/h^2)/2h^2$, where $\texttt{dpf}$ is the detection probability function of the proximity oblivious tester in the standard dense graph model. Then, we consider a graph G' that is obtained by replacing each vertex v of G with an *independent set* C_v of $\lfloor \mathcal{D}(v) \cdot N \rceil$ vertices, for a sufficiently large N (e.g., $N = h^2|V|/\delta$ or better $N = h^2|V|/\min_{u:\mathcal{D}(u)>0}\{\mathcal{D}(u)\}^2$), and placing a complete bipartite graph between C_u and C_v if u and v are connected in G. (Indeed, if u and v are not connected, then no edges are placed between C_u and C_v.)

Assuming that no vertex of G has appeared twice in the sample of h vertices (drawn from $\mathcal{D}$), our tester emulates an execution of T on G', since in this case a sample of h vertices drawn from $\mathcal{D}$ with no repetitions is distributed identically to a sample of h vertices drawn uniformly from $V' = \bigcup_{v \in V} C_v$ such that at most one vertex is selected from each C_v. As shown next (in Claim 2.8.1), $\delta_{\mathcal{D}'}^{\Pi}(G') \geq \delta_{\mathcal{D}}^{\Pi}(G)/h^2$, where $\mathcal{D}'$ is the uniform distribution on $V' = \bigcup_{v \in V} C_v$, and it follows that our tester rejects G (under

[28] Recall that by Definition 2.1 (see also Footnote 20), we assume that the tested graph has at least h vertices.

$\mathcal{D}$) with probability at least

$$\texttt{dpf}(\delta_{\mathcal{D}}^{\Pi}(G)/h^2) - \sum_{v\in V}\mathcal{D}(v)^2 \;>\; \texttt{dpf}(\delta_{\mathcal{D}}^{\Pi}(G)/h^2)/2, \tag{4}$$

where the second term (in the l.h.s of Eq. (4)) accounts for the case that the sample (of $\mathcal{D}$) contain two occurrences of the same vertex, and the inequality relies on the postulated upper bound on $\sum_{v\in V}\mathcal{D}(v)^2$.

Claim 2.8.1 (relating the distances of G and G' to Π): *Let G and G' be as above. Then, $\delta_{\mathcal{D}'}^{\Pi}(G') \geq \delta_{\mathcal{D}}^{\Pi}(G)/h^2$.*

A loss factor that is proportional to the number of edges in the subgraph H is inherent in our argument, and in the first posting of this work [16] we asked whether this can be avoided. In subsequent work, Gishboliner and Shapira answered this question positively, and actually obtained a more general result (see [12, Lem 2.2]).[29]

Proof: We first assume for simplicity that all $\mathcal{D}(v)$'s are multiples of $1/N$. Let $P' \in \Pi$ be a graph closest to G'; that is, P' witnesses the value of $\delta' = \delta_{\mathcal{D}'}^{\Pi}(G')$. We consider a graph P obtained from G by omitting each edge $\{u,v\}$ such that at least $|C_u|\cdot|C_v|/\binom{h}{2}$ edges are missing in the bipartite graph that connects C_u and C_v in P'. Since the number of (directed) edges that are missing in such bipartite graphs in P' is at most $\delta'\cdot|V'|^2$, it follows that the relative distance between G and P (under $\mathcal{D}$) is at most $\binom{h}{2}\cdot\delta'$.

We next claim that $P \in \Pi$, and it follows that $\delta_{\mathcal{D}}^{\Pi}(G) \leq \binom{h}{2}\cdot\delta'$. This claim is proved by contradiction. Suppose that the subgraph of P induced by $\{v_1,...,v_h\}$ contains a copy of H, and observe that $C_{v_i} \neq \emptyset$ for each $i \in [h]$ (because $\mathcal{D}(v_i) > 0$ for each $i \in [h]$).[30] Then, we select $\{r_1,...,r_h\}$ at random such that r_i is uniformly distributed in C_{v_i}. For every $i,j \in [h]$ such that $\{v_i,v_j\}$ is an edge in P, it holds that $\{r_i,r_j\}$ is an edge in P' with probability exceeding $1-\binom{h}{2}^{-1}$ (since otherwise the edge $\{v_i,v_j\}$ would have been omitted from G). Using a union bound, it follows that with positive probability the subgraph of P' induced by $\{r_1,...,r_h\}$ contains a copy of H, and it follows that $P' \notin \Pi$ (contradicting our hypothesis).

Having established $\delta_{\mathcal{D}'}^{\Pi}(G') \geq \delta_{\mathcal{D}}^{\Pi}(G)/\binom{h}{2}$ in the case that all $\mathcal{D}(v)$'s are multiples of $1/N$, we note that in general it holds that $\delta_{\mathcal{D}'}^{\Pi}(G') > 2h^{-2}\cdot\delta_{\mathcal{D}}^{\Pi}(G) - |V|/N$. Using a sufficiently large N (e.g., $N = h^2|V|/\delta_{\mathcal{D}}^{\Pi}(G)$), the claim follows. ■

As stated above, combining Eq. (4) and Claim 2.8.1, it follows that our tester finds a copy of H in G (under $\mathcal{D}$) with probability at least $\texttt{dpf}(\delta_{\mathcal{D}'}^{\Pi}(G')) - \binom{h}{2}\cdot\sum_{v\in V}\mathcal{D}(v)^2$, which is at least $\texttt{dpf}(\delta_{\mathcal{D}}^{\Pi}(G)/h^2)/4$. Note that the upper bound on the collision probability of $\mathcal{D}$ (i.e., $\sum_{v\in V}\mathcal{D}(v)^2 < \texttt{dpf}(\delta/h^2)/2h^2$) is crucial to asserting that finding a copy of H in G' implies finding a copy of H in G (see Eq. (4)). In contrast, when obtaining two samples of the same vertex v, it is no longer the case that a copy of H (in G') that contains several vertices in C_v yields a copy of H in G. This implication still holds in case H is a h-clique, since in this case a copy of H cannot contain several vertices in the same independent set C_v, and so the theorem has just been established for that case (with a somewhat improved bound)[31], but we wish to handle the general case.

[29] A related question arose in [18] and was addressed in [31], but there the l.h.s (i.e., $\delta_{\mathcal{D}'}^{\Pi}(G')$) is replace by the distance to the set of graphs that are (equal-factor) blow-ups of graphs in Π (in which all clouds have the same size). On the other hand, the question in [18, 31] is more general: It refers to any property Π, not only to the subgraph-freeness ones.

[30] Here we use the fact that H does not contain isolated vertices. Using this fact, it follows that v_i is not an isolated vertex in P. But, on the other hand, we may assume, without loss of generality, that P contains no edges that are incident at the set $Z = \{v : \mathcal{D}(v) = 0\}$, since we can omit all edges incident at Z at no cost. It follows that $v_i \notin Z$ for each $i \in [h]$.

[31] In this case we can apply the improved graph removal lemma of Fox [11], which avoids Szemerédi's regularity lemma [33].

Towards this end, we reduce the problem of *testing whether G' contains a copy of H that touches h different C_v's* to the following generalization of the subgraph freeness problem. In this generalization, for some constant r, each edge is colored by one of r colors, and the (fixed) forbidden graphs are similarly colored and called spots. The analogous property consists of r-colored graphs that do not contain one of the spots as a r-colored subgraph. In our application, we consider a 2-colored graph similar to G' except that we place edges between each pair of the vertices that reside in different C_v's, and color the edge red if it is in G' and black otherwise. The spots, in our application, are 2-colored h-vertex cliques that contain a copy of H colored red. Following is the general definition (which generalizes the definition studied in [2]).[32]

Definition 2.8.2 (spot freeness): *For $r, h \in \mathbb{N}$, consider a set of r-colored h-vertex graphs, denoted $S_1, ..., S_t$ and called* spots*; that is, each S_i consists of the vertex-set $[h]$ and a set of edges E_i along with their r-coloring $\chi_i : E_i \to [r]$. A colored graph is called $\{S_i : i \in [t]\}$-free if it contains no colored subgraph that equals any of the S_i's. Distance between colored graphs is defined as the fraction of the vertex pairs on which the graphs differ either in the existence of an edge or in the color of an edge that appears in both graphs.*[33]

Actually, the special case in which all spots are colored cliques suffices for our application (as stated next).

Claim 2.8.3 (testing H-freeness and spot-freeness): *Let the C_v and $G' = (V', E')$ be as above, and let $G'' = (V', E'')$ be a 2-colored graph such that $E'' = \cup_{u \neq v}(C_v \times C_u)$ and an edge of E'' is colored 1 if the edge is in E'* (and colored 2 otherwise). *Let $\mathcal{S}$ consists of the set of all 2-colored h-cliques that contain a subgraph colored 1 that is isomorphic to H. Then, the distance of G' to the set of graphs that contain no copy of $H = ([h], F)$ that intersects h different C_v's equals the distance of 2-colored G'' from being $\mathcal{S}$-free, where both distances refer to the uniform distribution on V'* (equiv., to the fraction of vertex pairs that should be modified).

Claim 2.8.3 follows by observing that a set of h vertices in G' contains a copy of H that intersects h different C_v's if and only if this set of vertices in G'' contains a spot that belongs to $\mathcal{S}$. Observing that the proof of Claim 2.8.1 actually reduces testing testing H-freeness under distribution $\mathcal{D}$ to the testing problem regarding copies of H that intersect h of the C_v's, we reduce the testing H-freeness under $\mathcal{D}$ to testing spot-freeness. Turning to the problem of testing spot-freeness (under the uniform distribution $\mathcal{U} = \mathcal{U}(V')$), we show that the ideas used to test subgraph freeness (under $\mathcal{U}$) extend and suffice.

Claim 2.8.4 (testing spot-freeness): *For any finite set of spots $\mathcal{S}$, the set of $\mathcal{S}$-freeness colored graphs has a proximity oblivious testers with a detection probability function $\texttt{dpf}(\delta) = 1/\texttt{T}(\text{poly}(1/\delta))$.*

Proof: We show that the ideas underlying the analysis of the natural tester for subgraph-freeness (which inspects the subgraph induced by h random vertices) extend to testing spot-freeness. The original analysis is pivoted at Szemerédi's regularity lemma [33], and here we use the generalization of the regularity lemma to r-colored graphs (presented in [26, Thm. 1.18], see also [2, Lem. 2.4]). This generalization asserts a partition of the graph (as in the original version) that is regular with respect to each of the r graphs that arise from the r sets of colored edges; that is, for graph $G = (V, E)$ and an r-coloring $\chi : E \to [r]$, we get a partition that is regular for each of the graphs $G_i = (V, \{e \in E : \chi(e) = i\})$.

[32] Alon *et. al.* [2] considered the case in which the spots are (the r) monochromatic (h-vertex) cliques. They studied the extremal question regarding r-colored graphs and these spots (i.e., the maximum number of r-colorings of some n-vertex graph such that this coloring contains none of the foregoing spots).

[33] Actually, the distance of a colored graph X to being $\mathcal{S}$-free equals its distance to a colored graph X' that is $\mathcal{S}$-free and is obtained by omitting the minimum number of edges from X; that is, the minimum distance to being $\mathcal{S}$-free is obtained by only omitting edges.

The next step is a generalization of the rest of the analysis of the subgraph-freeness tester (cf., e.g., [14, pp. 190-194]). Specifically, letting δ denote the distance of the input graph from being H-free, the original analysis uses a poly(δ)-regular partition of the input graph, and demonstrates the existence of many copies of H in the graph. The argument proceeds in a few simple steps, which we mimic here.

Assuming that the colored graph $G = (V, E)$ is δ-far from being $\mathcal{S}$-free, we focus on some $S \in \mathcal{S}$ such that the colored graph is $\delta/|\mathcal{S}|$-far from being S-free. (Note that, w.l.o.g., the cheapest way to make the graph satisfy these properties is obtained by omitting edges from the graph.) Now, using a poly(δ)-regular partition, denoted $(V_1, ..., V_T)$, of the colored graph G, we

1. omit all edges that are internal to some V_i;

2. omit all edges that connect pairs of V_i's that violate the regularity condition with respect to any of the r colors; and

3. omit all edges colored $c \in [r]$ among pairs (V_i, V_j) such that the density of the edges connecting V_i and V_j that are colored c is small.

   That is, for each pair (V_i, V_j), we only omit edges that (connect V_i and V_j and) have a specific color such that the number of edges with that color (that connect V_i and V_j) is small.

As in the original argument, since the total number of omitted edges is smaller than $\delta \cdot |V|^2$, we obtain a (colored) graph R that contains the spot S and is a (colored) subgraph of the input graph. Without loss of generality (see [14, p. 191]), we may assume that this spot intersects h different parts[34], denoted $V_1, .., V_h$. Viewing this spot as determining a subgraph over $U = \bigcup_{i \in [h]} V_i$ of the (colored) input graph, we argue exactly as in the original analysis, except that the subgraph we consider is different here. Whereas in the original analysis this subgraph consists of the subgraph of R induced by U, here it contains only the edges that agree with the colors of the spot S. That is, for every $i \neq j$ (in $[h]$), if the edge $\{i, j\}$ is colored $\chi_{i,j}$ in S, then we keep the edges between V_i and V_j that are colored $\chi_{i,j}$ in R. At this point we have h independent sets (i.e., the V_i's) such that each pair (V_i, V_j) is regular and contains many edges if $\{i, j\}$ is colored $\chi_{i,j}$ in S. This yields many subgraphs with the same colors (by following the original argument). ■

Conclusion and the case of bipartite H. Combining Claims 2.8.1 and 2.8.3 with Claim 2.8.4, we establish the main claim of the theorem. Turning to the furthermore claim (and forgetting about the foregoing colored graphs and spots), we consider the case that $H = ([h], F)$ is bipartite. Specifically, suppose that $H = ([h], F)$ is a subgraph of $K_{t,h-t}$, for some $t \in [h-1]$, and let $\delta = \delta_{\mathcal{D}}^{\Pi}(G) > 0$.

We warn that, unlike in the standard setting (cf. [1, Lem. 2.1]), here *it is not the case that the fact that the set of all edges has weight at least δ* (w.r.t $\mathcal{D}$) *implies that the graph contains a copy of* $K_{h,h}$, let alone $\Omega(\delta^{h^2})$ such copies (assuming, of course, that $h > 1$).[35] Indeed, our argument uses the value of $\delta = \delta_{\mathcal{D}}^{\Pi}(G)$, and not merely the (implied) lower bound on the weight (w.r.t $\mathcal{D}$) of all edges in $G = (V, E)$.

Loosely speaking, we distinguish the case in which much of $\delta_{\mathcal{D}}^{\Pi}(G)$ is due to edges connecting light vertices from the complimentary case. In the former case we can argue as in the standard setting (where $\mathcal{D} = \mathcal{U}(V)$), whereas in the latter case we can find the heavy vertices by sampling $\mathcal{D}$ and detect a copy of H in a natural manner. The latter case is easy when a copy of H resides on heavy vertices only, but in general the copies of H may contain both heavy and light vertices. The key observation is that a noticeable fraction of the distance to being H-free is due to copies of H that reside on $s \leq h$ of the heavy vertices. Furthermore, clustering the light vertices according to their heavy neighbors, a

[34] Actually, in case the spots are colored cliques (as in our application), the spot must intersect h different parts.

[35] Consider, for example the case that $\mathcal{D}(v) = 1/2$ for two vertices v and the graph consisting of a single edge between these two vertices. There are indeed many copies of $K_{h,h}$ in the corresponding blow-up graph G', but all these copies intersect two clouds of G' (and so yield no copy of $K_{h,h}$ in G).

noticeable fraction of the latter distance is due to a sequence of $h-s$ such clusters. Assuming the latter clusters are distinct, we conclude by observing that a sample of h vertices hits these s heavy vertices and $h-s$ clusters with noticeable probability. Extra care is required for handling the case that the $h-s$ clusters are not distinct. Details follow.

For $\gamma = \Theta(\delta^{h^2/4})$, we let $L = \{v \in V : \mathcal{D}(v) \leq \gamma\}$ denote the set of light vertices, and consider two cases regarding the weight of the edges that connect light vertices; that is, the edges in $E_L = \{\{u,v\} \in E : u,v \in L\}$.

Case 1: E_L carries much weight (i.e., $\sum_{\{u,v\}\in E_L} \mathcal{D}(u) \cdot \mathcal{D}(v) \geq \delta/4$). In this case, we consider the graph $G_1 = (V, E_L)$ and a corresponding blow-up graph $G' = (V', E')$ as in the main claim; that is, the vertices of V are replaced by clouds C_v's of size proportional to $\mathcal{D}(v)$, and the edges of E_L are replaced by complete bipartite graphs between the corresponding clouds. We shall only use the fact that G' has $\Omega(\delta \cdot |V'|^2)$ edges, which follows from the hypothesis of the current case, and the fact that $\sum_{v\in L} \mathcal{D}(v)^2 \leq \gamma$.

As shown by Alon [1, Lem. 2.1], the edge density of G' implies that the graph G' contains at least $\Omega(\delta^{t\cdot(h-t)}) \cdot |V'|^h$ copies of $K_{t,h-t}$. Hence, with probability $p = \Omega(\delta^{h^2/4})$, the subgraph of G' induced by h random vertices in V' contains a copy of H, whereas the probability that two of these vertices reside in the same cloud C_v for $v \in L$ is at most $\binom{h}{2} \cdot \sum_{v\in L} \mathcal{D}(v)^2$, which is upper-bounded by $\binom{h}{2} \cdot \max_{v\in L}\{\mathcal{D}(v)\} < h^2 \cdot \gamma$. By an appropriate choice of the hidden constant in the definition of γ, it follows that with probability at least $p/2$, a random choice of h vertices in V' contains a copy of H that intersects h different clouds.

Hence, when drawing h samples from $\mathcal{D}$, with probability at least $p/2 = \Omega(\delta^{h^2/4})$, these vertices induce a copy of H in G_1 (and so a copy of H in G).

Case 2: E_L carries little weight (i.e., $\sum_{\{u,v\}\in E_L} \mathcal{D}(u) \cdot \mathcal{D}(v) < \delta/4$). In this case, we consider the graph $G_2 = (V, E \setminus E_L)$, and note that $\delta^{\Pi}_{\mathcal{D}}(G_2) \geq \delta/2$ and that (in G_2) vertices in L only neighbor vertices in $V \setminus L$. Noting that each copy of H (in G_2) uses at most h vertices in $V \setminus L$, we identify a subset of $V \setminus L$ of size at most h that "contributes" most to $\delta^{\Pi}_{\mathcal{D}}(G_2)$, and use the fact that $|V \setminus L| \leq 1/\gamma = \mathrm{poly}(1/\epsilon)$. Only the copies of H that hit this subset of $V \setminus L$ will be used to establish the claim (that a random set of h vertices hits a copy of H with probability $\mathrm{poly}(\epsilon)$).

Letting $h' = \min(h, |V \setminus L|) \ll 1/\gamma$ (where "typically" $h' = h$), for every h'-subset $S \subseteq V \setminus L$, we consider the subgraph of G_2, denoted $G_{S,L}$, that contains only edges of $E \setminus E_L$ with both endpoints in $S \cup L$. Then, there exists an h'-subset $S \subseteq V \setminus L$ such that

$$\delta^{\Pi}_{\mathcal{D}}(G_{S,L}) \geq \frac{\delta/2}{\binom{1/\gamma}{h'}} > \delta \cdot \gamma^h, \tag{5}$$

since a copy of H uses at most h' vertices of $V \setminus L$.

Next, we *cluster* the vertices in L according to their adjacency relation with S such that the cluster $L_{S'}$ contains all vertices in L that neighbor each vertex in S' but neighbor no vertex in $S \setminus S'$; that is, $v \in L_{S'}$ iff $v \in L$ and the neighbor-set of v in $G_{S,L}$ equals S'. (Jumping ahead, we mention that in the graph $G_{S,L}$ vertices in L may only neighbor vertices in S, and so the vertices in $L_{S'}$ are effectively identical.) Letting $\beta = \delta \cdot \gamma^h/2^{h+1}$, we discard all vertices that resides in clusters having weight smaller than β; that is, we let $F_S \stackrel{\text{def}}{=} \{S' \subseteq S : \mathcal{D}(L_{S'}) > \beta\}$, where $\mathcal{D}(X) = \sum_{x\in X} \mathcal{D}(x)$, and consider $L' \stackrel{\text{def}}{=} \bigcup_{S'\in F_S} L_{S'}$. Given that the edges that are incident at $L \setminus L'$ have weight at most $\mathcal{D}(L \setminus L') \leq 2^{h'} \cdot \beta$ and using Eq. (5), it follows that

$$\delta^{\Pi}_{\mathcal{D}}(G_{S,L'}) > 2^{h+1} \cdot \beta - 2^{h'} \cdot \beta \geq 2^{h'} \cdot \beta. \tag{6}$$

(Indeed, $G_{S,L'}$ denotes the subgraph of G_2 that contains only edges of $E \setminus E_L$ with both endpoints in $S \cup L'$.)

Foreseeing the need to partition each cluster in F_S to parts of sufficient weight, we "trim" the clusters so to enable such a partition. Specifically, for every $S' \in F_S$, we denote by $L'_{S'} = \{v \in L_{S'} : \mathcal{D}(v) < \beta/4h\}$ the set of vertices of very low weight. We consider *residual clusters* obtained by discarding vertices of very low weight (i.e., vertices in $L'_{S'}$) in case their total weight is small (i.e., $\mathcal{D}(L'_{S'}) \leq \beta/2$). Specifically, if $\mathcal{D}(L'_{S'}) \leq \beta/2$, then we let $L''_{S'} = L_{S'} \setminus L'_{S'}$ and otherwise $L''_{S'} = L_{S'}$. Either way, $\mathcal{D}(L''_{S'}) > \beta/2$, since $\mathcal{D}(L_{S'}) > \beta$. Using Eq. (6), we have

$$\delta^{\Pi}_{\mathcal{D}}(G_{S,L''}) > 2^{h'} \cdot \beta - 2^{h'} \cdot \frac{\beta}{2} > \frac{\beta}{2} > 0, \tag{7}$$

where $L'' \stackrel{\rm def}{=} \bigcup_{S' \in F_S} L''_{S'}$ (and, again, $G_{S,L''}$ denotes the subgraph of G_2 that contains only edges of $E \setminus E_L$ with both endpoints in $S \cup L''$).

We use the fact that $G_{S,L''}$ contains a copy of H, and denote the (h different) vertices on which this copy resides by $v_1, ..., v_h$. Suppose that $v_1, ..., v_s \in S$ and $v_{s+1}, ..., v_h \in L''$, and assume first (for simplicity) that $v_{s+1}, ..., v_h$ appear in different residual clusters denoted $L''_{s+1}, ..., L''_h$. Then, for every $(u_{s+1}, ..., u_h) \in L''_{s+1} \times \cdots \times L''_h$, the subgraph of G_2 induced by $v_1, ..., v_s, u_{s+1}, ..., u_s$ contains a copy of H, since the subgraph (of G_2) induced by $\{v_1, .., v_h\}$ is isomorphic to the subgraph (of G_2) induced by $\{v_1, .., v_s, u_{s+1}, ..., u_h\}$. This is the case because, for each $i \in \{s+1, ..., h\}$, *by virtue of being in the same cluster* L''_i*, the vertex* u_i *neighbors the same vertices of* S *as* v_i, whereas (in G_2) *neither* u_i *nor* v_i *neighbors any vertex in* $L \supseteq L''$. It follows that h vertices drawn from $\mathcal{D}$ contain a copy of H with probability at least

$$\left(\prod_{i=1}^{s} \mathcal{D}(v_i)\right) \cdot \left(\prod_{i=s+1}^{h} \mathcal{D}(L''_i)\right) > \gamma^s \cdot (\beta/2)^{h-s}. \tag{8}$$

Recalling that $\gamma = \Theta(\delta^{h^2/4})$ and $\beta = \delta \cdot \gamma^h / 2^{h+1}$, we lower-bound Eq. (8) by $\Omega((\delta\gamma^h)^h) = \Omega(\delta^{h^4})$.

The foregoing argument presumes that $v_{s+1}, ..., v_h$ appear in different residual clusters. To handle the case that several v_i's occur in the same residual cluster, we partition these clusters to an appropriate number of parts such that each part has weight at least $\beta/4h$ and contains at most one v_i. This is possible since, for each $S' \in F_S$, either $L''_{S'} = L_{S'} \setminus L'_{S'}$, which implies $\mathcal{D}(v) > \beta/4h$ for every $v \in L''_{S'}$, or $\mathcal{D}(L'_{S'}) \geq \beta/2$, which implies that $L'_{S'}$ can be h-partitioned so that each part has weight at least $\beta/4h$ (by distributing the vertices of $L'_{S'}$, which have each weight smaller than $\beta/4h$, among the h parts). Considering all u_i's in the parts in which $v_i \in L''_i$ resides yields a probability lower bound of $\gamma^s \cdot (\beta/4h)^{h-s}$ (rather than $\gamma^s \cdot (\beta/2)^{h-s}$), which is also $\Omega(\delta^{h^4})$.

Having completed the treatment of both cases, the (furthermore) claim follows. ■

2.4 On two classes that are easy to test in the standard model

In this section, we consider two classes of properties that are very easy to test in the standard dense graph model (cf. [14, Sec. 8.2.2]). We shall see that one class, which is trivial to test with one-sided error in the standard model, is hard to test in the current VDF model, whereas for the other class the situation is mixed.

2.4.1 On properties that are trivial in the standard model

Recall that we have already seen, in Section 2.1, that *only* properties that have a strong tester of one-sided error in the standard model may be strongly testable in the current model. Hence, only properties of the former class are of interest to us here. One natural question is whether this necessary condition is a sufficient one.

We answer this question negatively by considering a class of properties that are trivial to test (with one-sided error) in the standard dense graph model, and showing that they are not strongly testable in the VDF version.

Proposition 2.9 (triviality in the standard model does not imply strong testability in the VDF model): *Suppose that the graph property* Π *satisfies the following two conditions:*

1. *For every* n*, the property* Π *does not contain the empty* n*-vertex graph* (i.e., a graph with no edges).
2. *There exists a constant* $c \in (0,2)$ *such that for all sufficiently large* $n \in \mathbb{N}$ *and every* n*-vertex graph* $G = (V, E)$ *there exists an* n*-vertex graph* $G' = (V, E')$ *in* Π *such that the symmetric difference between* E *and* E' *is at most* n^c*.*

Then, Π *is not strongly testable in the VDF dense graph model.*

The foregoing class of properties includes Connectivity, Hamiltonicity, and being Eulerian (but not empty). Recall that properties covered by Proposition 2.9 are trivial to test (with one-sided error) in the standard model by accepting without making any queries if $\epsilon > n^{-(2-c)}$ (and querying all $n^2 \leq (1/\epsilon)^{2/(2-c)}$ vertex pairs otherwise; see [14, Prop. 8.3]).[36] Hence, Proposition 2.9 establishes the "dense graph model" part of Theorem 1.2. The following proof relies on the fact that in the VDF model the tester does not obtain the size of the tested graph. An alternative proof of the dense graph model part of Theorem 1.2, which does not rely on the testers' obliviousness of the size of the tested graph, is used in establishing Proposition 2.11.

Proof: Using Theorem 2.3, it suffices to show that Π is not strongly testable *with one-sided error* in the VDF model. Actually, the only aspect of the VDF model that we use here is the fact that the tester does not obtain the size of the tested graph. (Hence, the proof extends to a corresponding version of the standard model considered in [15], which may be viewed as a restriction of the VDF model to the case that the distribution is uniform over the vertex-set.)

Fixing a sufficiently small $\epsilon > 0$, we let $n = (1/\epsilon)^{1/(2-c)}$, and consider (one of) the sparsest n-vertex graph in Π, which by Condition 2 has at most n^c edges. Denoting this graph by $G = (V, E)$, observe that G contains an independent set S of size $n' = n^{(2-c)/2}$, since the expected number of edges in a subgraph induced by n' random vertices is at most $\binom{n'}{2} \cdot n^{-(2-c)} < 1$.

Now, assume that Π has a one-sided error strong tester, denoted T. Then, on the one hand, T (having one-sided error) must always accept G (under any vertex-distribution $\mathcal{D}$ and any value of $\epsilon > 0$), which implies that T accepts G even when all sampled vertices fall in S. The latter event occurs with positive probability, provided $\mathcal{D}(S) > 0$, and in particular when the vertex-distribution $\mathcal{D}$ is uniform over V. Note that, in this case, T sees a subgraph that is an independent set. But, this implies that T always accepts the empty n'-graph, (i.e., the n'-vertex graph that contains no edges), although (by Condition 1) this graph is $(1/n')^2$-far from Π under the uniform distribution on its vertex-set. Recall that this holds for any value of ϵ, and in particular for $\epsilon = n^{-(2-c)} = (1/n')^2$ (as set above). This contradicts the hypothesis that T is a tester of Π. ■

2.4.2 On properties of sparse graphs

Although any property of sparse graphs is easy to test in the standard (dense graph) model, many of these properties are not strongly testable in the current VDF model. The source of trouble is that the

[36] The point is that in the standard model every n-vertex graph is $n^{-(2-c)}$-close to the property, and so we may accept any graph if $\epsilon > n^{-(2-c)}$, which is the typical case. This justifies the view that these properties are trivial to test.

relevant testers in the standard model are typically not of the one-sided error type. In fact, we shall show that for many of these properties strong testing (even in the standard model) requires two-sided error.

Proposition 2.10 (properties of sparse graphs that are hard to test in the VDF model): *For any unbounded* $f : \mathbb{N} \to \mathbb{N}$ *such that* $f(n) < n/2$ *for all* $n > 3$*, suppose that* Π *is a graph property that satisfies the following two conditions for infinitely many* n*'s.*

1. Π *contains an* n*-vertex graph that contains a cliques of size* $f(n)$*.*
2. Π *contains no* n*-vertex graph that contains more than* $0.5 \cdot \binom{n}{2}$ *edges.*

Then Π *is not strongly testable in the VDF model.*

Note that, when Condition 2 is made more stringent (e.g., n-vertex graphs in Π contain at most $n^{2-\Omega(1)}$ edges), Π is strongly testable (with two-sided error) in the standard model (cf. [14, Prop. 8.4]). This holds also if Condition 1 is dropped.

Proof: Using Corollary 2.4, it suffices to show that Π is not strongly testable with one-sided error in the standard model. This is the case, since when $f(n)/2$ exceeds the query complexity of the tester, which happens when n is large enough (whereas the query complexity only depends on the proximity parameter), the tester sees less than $f(n)$ vertices. Being a one-sided error tester, it must accept if the induced subgraph that it sees is a clique (since, by Condition 1, such an $f(n)$-clique may be part of an n-vertex graph in Π). But this implies that the tester always accepts the complete n-vertex graph, although this graph is 0.4-far from Π (by Condition 2). ■

Yet another negative result. Proposition 2.10 refers to properties that are strongly testable in the standard model *but not with one-sided error*, and the negative result regarding testing in the VDF model follows (by Corollary 2.4). A more interesting negative result (for the VDF model) refers to properties of sparse graphs that *are strongly testable with one-sided error in the standard model*, but are not strongly testable in the VDF model. One such example is presented next.

Proposition 2.11 (on the "dense graph model" part of Theorem 1.2): *Let* Π *be the set of graphs consisting of two stars, each holding at least one third of the vertices* (i.e., an n-vertex graph is in Π if it consists of two connected components of size at least $n/3$, each being a tree with a single internal vertex)*. Then,* Π *is strongly testable with one-sided error in the standard dense graph model, but is not strongly testable in the VDF model. Furthermore, the lower bound holds even if the tester is given the size of the graph as auxiliary input.*

Proof: We first show that Π is strongly testable with one-sided error in the standard model. On input parameters n and $\epsilon > 0$ and oracle access to the graph $G = ([n], E)$, the proposed tester behaves as follow: If $\epsilon \leq 5/n$, then the tester queries all $\binom{n}{2} = O(\epsilon^{-2})$ vertex-pairs, and decides accordingly. Otherwise (i.e., $\epsilon > 5/n$), the tester selects $m = O(1/\epsilon)$ random vertices and rejects if and only if the induced subgraph contains more than $2m$ edges. Observe that an m-vertex graph that contains more than $2m$ edges, must contain at least three vertices of degree greater than one, which means that this tester never rejects graphs in Π. Now, suppose that $G = ([n], E)$ is ϵ-far from Π and $\epsilon > 5/n$. Then, G has more than $\epsilon \cdot \binom{n}{2} - n > 0.5\epsilon \cdot \binom{n}{2}$ edges, because otherwise obtain a graph in Π by omitting all edges of G and then inserting the two stars (at an extra cost of $n-2$ edges). Using $n < 0.5\epsilon \cdot \binom{n}{2}$ (which holds if $\epsilon > 5/n$ for $n > 5$), it follows that the expected number of edges in a subgraph induced by m random vertices is at least $0.5\epsilon \cdot \binom{m}{2} > 2.3m$, provided that $m \geq 10/\epsilon$. Hence, with high probability, a random set of $m \geq 10/\epsilon$ vertices induces a subgraph with more than $2m$ edges.

Using Theorem 2.3, it suffices to show that Π is not strongly testable with one-sided error in the VDF model. We first observe that a one-sided error tester must always accept when it sees a subgraph consisting of a single star, because such a view may occur when accessing a graph in Π (e.g., when the vertex distribution is uniform over $[n]$ and we sample m vertices, this occurs with probability $\exp(-\Omega(m)) > 0$). But this means that such an alleged tester will accept (wvhp) an n-vertex graph that consists of a single n-vertex star when coupled with a vertex distribution $\mathcal{D}$ that assigns the center of the star probability $1/2$ (and uniform distribution on the other $n-1$ vertices). However, this n-vertex graph is $\Omega(1)$-far from Π under the vertex distribution $\mathcal{D}$, which means that this alleged tester fails (although n was fixed, and so is "known" to the tester). ■

A positive result. While a global bound on sparsity does not guarantee strong testability in the current model (cf. Propositions 2.10 and 2.11), a local bound – in the form of bounded degree – does guarantee it, when augmented with two natural conditions.

Theorem 2.12 (testing properties of bounded-degree graphs (in the VDF dense graph model)): *Suppose that Π is a graph property that satisfies the following conditions.*

bounded degree: *The exists a constant d such that each graph in Π has vertices of degree at most d.*

hereditary: *For every $G \in \Pi$, every induced subgraph of G is also in Π.*

paddability: *For every $G \in \Pi$, the graph obtained by augmenting G with an isolated vertex is also in Π.*

Then, Π is testable in the VDF dense graph model with query complexity $\widetilde{O}(1/\epsilon^2)$.

In particular, for any $d \geq 1$, the set of all graphs of maximal degree at most d is testable in the VDF dense graph model with query complexity $\widetilde{O}(1/\epsilon^2)$.

Proof: The key observation is that, in this case (i.e., for graphs in Π), the total probability weight of edges that have at least one endpoint of small weight is small. More generally, for any graph $G = (V, E)$ in Π, any set $V' \subseteq V$, and any distribution $\mathcal{D}$ on V, it holds that $\sum_{\{u,v\}\in E: u\in V'} \mathcal{D}(v) \cdot \mathcal{D}(u)$ is upper-bounded by $d \cdot \max_{u\in V'}\{\mathcal{D}(u)\}$, where d is the *degree bound of* Π. This claim is proved by letting $\Gamma(v)$ denote the set of neighbors of vertex v in G, and observing that

$$\begin{aligned}
\sum_{\{u,v\}\in E: u\in V'} \mathcal{D}(v) \cdot \mathcal{D}(u) \quad &\leq \quad \sum_{v\in V} \mathcal{D}(v) \cdot \sum_{u\in V'\cap\Gamma(v)} \mathcal{D}(u) && (9) \\
&\leq \quad \sum_{v\in V} \mathcal{D}(v) \cdot |\Gamma(v)| \cdot \max_{u\in V'}\{\mathcal{D}(u)\} \\
&\leq \quad d \cdot \max_{u\in V'}\{\mathcal{D}(u)\} && (10)
\end{aligned}$$

where the first inequality holds since each edge in the l.h.s of Eq. (9) is counted at least once in the r.h.s of Eq. (9). Hence, in our analysis, we can ignore edges with a light endpoint (e.g., edges that are incident at $\{u \in V : \mathcal{D}(u) \leq \epsilon/2d\}$). This observation is only used in the analysis; the tester itself proceeds as follows: *On input parameter ϵ, and access to the graph G and the sampling device $\mathcal{D}$, the tester obtains a sample of $m = \widetilde{O}(1/\epsilon)$ vertices from $\mathcal{D}$, and accepts if and only if the induced subgraph is in* Π.

We first observe that, by the hypothesis that Π *is hereditary*, any induced subgraph of a graph in Π is itself in Π, and so the foregoing tester always accepts graphs in Π. Turning to the case that $G = (V, E)$ is ϵ-far from Π under the distribution $\mathcal{D}$, we shall show that the tester rejects with high probability. Letting $H = \{v \in V : \mathcal{D}(v) > \epsilon/2d\}$ denote the set of heavy vertices, we consider two cases regarding the subgraph of G induced by H, denoted G_H.

Case 1: $G_H \notin \Pi$. In this case, with very high probability, the subgraph induced by the sample is not in Π (and the tester rejects). This is the case, because, with very high probability, the sample hits all vertices of H, which implies that subgraph induced by the sample is not in Π (since otherwise $G_H \in \Pi$ by the hypothesis that Π *is hereditary*).

Case 2: $G_H \in \Pi$. In this case, we start by considering the graph $G' = (V, E')$ such that $E' = \{\{u,v\} \in E : u,v \in H\}$; that is, G' consists of G_H and $|V \setminus H|$ isolated vertices. Hence, by the *paddability of* Π, it follows that $G' \in \Pi$, which implies that G is ϵ-far from G'. Since the difference between these two graphs is due to edges that have at least one end-point in $V' = V \setminus H$, it holds that

$$\sum_{\{u,v\} \in E : u \in V'} \mathcal{D}(v) \cdot \mathcal{D}(u) > \epsilon. \tag{11}$$

Indeed, Eq. (11) does contradict Eq. (9)–(10); it only implies that $G \notin \Pi$, which we know anyhow (since this is implied by our starting hypothesis that G is ϵ-far from Π). Our aim is showing that the tester will observe this contradiction; indeed, we shall show that (whp) the subgraph seen by the tester coupled with the empirical vertex-distribution induced by the sample (i.e., the distribution $\mathcal{D}'$ defined below) violates Eq. (9)–(10), and it follows that this subgraph is not in Π (and so the tester rejects).

Denoting the vertices that the tester's samples by $z_1, ..., z_m$, while noting that the z_i's are not necessarily distinct, we observe that for every $i \neq j$ it holds that the random variable that indicates whether or not the event $\{z_i, z_j\} \in E$ & $u \in V'$ holds is an unbiased estimator of the l.h.s of Eq. (11); that is, letting $\zeta_{i,j} = 1$ if $\{z_i, z_j\} \in E$ & $u \in V'$ and $\zeta_{i,j} = 0$ otherwise, we have $\text{Exp}[\zeta_{i,j}] = \sum_{\{u,v\} \in E : u \in V'} \mathcal{D}(v) \cdot \mathcal{D}(u)$. Hence, with high probability over the choice of the sample,

$$\sum_{i \in [m]} \sum_{j \in [m] \setminus \{i\}} \zeta_{i,j} > 0.9\epsilon \cdot m \cdot (m-1). \tag{12}$$

Likewise, with high probability over the choice of the sample, each vertex in V' occurs at most $0.6\epsilon \cdot m/d$ times in the sample, since $V' = \{v \in V : \mathcal{D}(v) \leq 0.5\epsilon/d\}$. Hence, Eq. (12) conflicts with Eq. (9)–(10), indicating that the subgraph induced by the sample is not in Π. Specifically, defining $\mathcal{D}'$ to be uniform over the sample (i.e., $\mathcal{D}'(v) = |\{i \in [m] : z_i = v\}|/m$ denotes the frequency of occurrences of the vertex v in the sample), we get

$$\sum_{\{u,v\} \in E : u \in V'} \mathcal{D}'(v) \cdot \mathcal{D}'(u) > 0.9\epsilon. \tag{13}$$

We conclude that the subgraph of G induced by the sample is not in Π, since otherwise we reach contradiction (because, by Eq. (9)–(10), if this subgraph were in Π, then $\sum_{\{u,v\} \in E : u \in V'} \mathcal{D}'(v) \cdot \mathcal{D}'(u) \leq d \cdot \max_{u \in V'}\{\mathcal{D}'(u)\} \leq 0.6\epsilon$ would hold, where the last inequality is due to $V' \subseteq \{v \in V : \mathcal{D}'(v) \leq 0.6\epsilon/d\}$). It follows that the tester rejects (w.h.p.) also in this case.

This completes the analysis of the foregoing tester, and establishes the theorem. ∎

Another positive result. We next show that, in some cases, the degree bound (condition of Theorem 2.12) can be relaxed to an arboricity bound (i.e., being covered by a bounded number of forests); in particular, this condition holds for minor-free properties (which always have bounded arboricity [27]). Recall that a graph property Π is called minor-free if there exists a finite set of graphs $\mathcal{M}$ such that G is in Π if any only if it does not contain a minor in $\mathcal{M}$, where a minor of G is a graph obtained from G by edge and vertex deletions as well as edge-contractions. Note that, in light of Proposition 2.11, assuming only bounded arboricity does not suffice (since even properties of forests may not be strongly testable in the VDF model). (On the other hand, minor-free graphs do satisfy the other two conditions of Theorem 2.12.)

Theorem 2.13 (testing minor-free properties (in the VDF dense graph model)): *Every minor-free property* Π *is testable in the VDF dense graph model with query complexity* $\text{poly}(1/\epsilon)$.

In particular, planarity is testable in the VDF dense graph model with query complexity $\text{poly}(1/\epsilon)$. The degree of the polynomial in Theorem 2.13 is determined by the property Π and essentially equals $2d+2$, where $d \geq 1$ is the corresponding arboricity bound. (Recall that the complexity bound established in Theorem 2.13 is $\widetilde{O}(1/\epsilon^2)$.)

Proof: Analogously to the proof of Theorem 2.12, we use the fact that (for graphs in Π) the total probability weight of edges that connect vertices of small weight is small. This is related but incomparable to the fact established in the proof of Theorem 2.12: On the one hand, here we only know that the graph is minor-free, which implies that it has bounded arboricity (but not necessarily bounded degree). On the other hand, here we upper-bound the total weight of edges with both endpoints in the set of light vertices (rather than edges with at least one endpoint in this set). More generally, we claim that, for any graph $G = (V, E)$ in Π, any set $V' \subseteq V$, and any distribution $\mathcal{D}$ on V, it holds that $\sum_{\{u,v\}\in E:u,v\in V'} \mathcal{D}(v) \cdot \mathcal{D}(u)$ is upper-bounded by $d \cdot \max_{u\in V'}\{\mathcal{D}(u)\}$, where d is the *arboricity bound of* Π. This claim is proved by letting $P_i(v)$ denote the parent of v in the i^{th} forest in a decomposition of G (while fictitiously defining $P_i(v) = \perp \notin V$ if v is a root in the i^{th} forest), and observing that

$$\begin{aligned}
\sum_{\{u,v\}\in E:u,v\in V'} \mathcal{D}(v) \cdot \mathcal{D}(u) &= \sum_{v\in V'} \mathcal{D}(v) \cdot \sum_{u\in V'\cap\{P_i(v):i\in[d]\}} \mathcal{D}(u) &(14)\\
&\leq \sum_{v\in V} \mathcal{D}(v) \cdot d \cdot \max_{u\in V'}\{\mathcal{D}(u)\}\\
&= d \cdot \max_{u\in V'}\{\mathcal{D}(u)\}, &(15)
\end{aligned}$$

where the equality holds since each edge with both endpoints in V' is counted once on both sides on Eq. (14). Hence, in our analysis, we can ignore edges that connect light vertices. Again, this observation is only used in the analysis; the tester itself proceeds as follows: *On input parameter* ϵ*, and access to the graph* G *and sampling device* $\mathcal{D}$*, the tester obtains a sample of* $s = \text{poly}(1/\epsilon)$ *vertices from* $\mathcal{D}$*, and accepts if and only if the induced subgraph is in* Π. The degree of the aforementioned polynomial will be determined by the property Π (as detailed in the analysis).

We first observe that, since Π *is hereditary*, any induced subgraph of a graph in Π is itself in Π, and so the foregoing tester always accepts graphs in Π. Turning to the case that $G = (V, E)$ is ϵ-far from Π under the distribution $\mathcal{D}$, we shall show that the tester rejects with high probability. Letting $H = \{v \in V : \mathcal{D}(v) > \epsilon/4d\}$, we consider the graphs $G' = (V, E')$ and $G'' = (V, E \setminus E')$ such that $E' = E \setminus \{\{u, v\} : u, v \in V \setminus H\}$ is the set of edges incident at H. That is, G' contains the edges of G that have at least one heavy endpoint (i.e., are incident at H), whereas G'' contains the edges of G that have no heavy endpoints (i.e., have both endpoints in $V \setminus H$). We first prove the following claim.

Claim 2.13.1 *Either the subgraph* G' *is* $\epsilon/2$*-far from* Π *or, with high probability, a random sample of* $\widetilde{O}(1/\epsilon)$ *vertices induces a subgraph of* G'' *that is not in* Π.

Note that in the first case it follows that $H \neq \emptyset$, whereas in the second case the foregoing tester rejects G with high probability.

Proof: The key observation is that if G' is $\epsilon/2$-close to Π, then G'' must be $\epsilon/2$-far from the empty graph (i.e., the total weight of the $E \setminus E'$ exceeds $\epsilon/2$), which implies that (whp) a random sample of vertices induces a subgraph of G'' that violaltes Eq. (14)–(15). The latter implication is prove analogously to the argument used in Case 2 of the proof of Theorem 2.12. Specifically, we consider the subgraph of G'' that is induced by a random sample of vertices along with the empirical distribution defined by this sample, and show that (whp) this subgraph along with this distribution violaltes Eq. (14)–(15).

(Even more specifically, we define V' as the set of all vertices that occur in the sample with frequency at most $\epsilon/3$, observe that (whp) $V' \subseteq V \setminus H$ and that (whp) the total weight of edges in the subgraph exceeds $\epsilon/3$, and note that this combination violaltes Eq. (14)–(15).) ■

Using Claim 2.13.1, we may focus on the case that G' is $\epsilon/2$-far from Π (and use $H \neq \emptyset$). Following is an outline of the rest of the proof. As when establishing (Case 2 of) the furthermore claim of Theorem 2.8, we cluster the vertices in $V' \stackrel{\rm def}{=} V \setminus H$ according to their adjacency to vertices in H. Here we have no degree bound, but we can easily treat vertices that have more than d neighbors in H (where d is the arboricity bound). Specifically, a subgraph of G' that contains all vertices of H and sufficiently many vertices of V' of degree exceeding d violates the arboricity bound. We cluster the vertices of V' that have degree at most d in G' according to their neighborhood and use the fact that these vertices are interchangeable. Using am upper bound on the number of vertices in V' that are essential towards forming a forbidden minor in G', we shall show that a sample of ${\rm poly}(1/\epsilon)$ vertices does contract to a forbidden minor. With this outline in mind, we turn to the actual proof.

We start by treating vertices that have more than d neighbors in H. For every $v \in V'$, we denote by $\Gamma_{G'}(v) \subseteq H$ the set of neighbors of v in G'. Letting $S' = \{v \in V' : |\Gamma_{G'}(v)| > d\}$ and $\epsilon' = \frac{\epsilon}{8(d+1)\cdot|H|}$, we distinguish $S'' \stackrel{\rm def}{=} \{v \in S' : \mathcal{D}(v) \leq \epsilon'\}$ from $S' \setminus S''$, and consider two cases:

Claim 2.13.2 *Either $\mathcal{D}(S'') \leq \epsilon/4$ or, with high probability, a random sample of $\widetilde{O}(1/\epsilon')$ vertices induces a subgraph of G' that is not in Π.*

In the first case we can afford to omit all edges adjacent to S'', whereas in the second case the foregoing tester rejects G with high probability.

Proof: Suppose that $\mathcal{D}(S'') > \epsilon/4$. Then, we can partition the vertices of S'' into $t = \frac{\epsilon/4}{2\epsilon'} = (d+1)\cdot|H|$ sets $S''_1, ..., S''_t$ such that $\mathcal{D}(S''_i) > \epsilon'$ for each $i \in [t]$. With high probability, a random sample of $\widetilde{O}(1/\epsilon')$ vertices hits each vertex in H and each set S''_i, and it follows that this samples contains a set with $|H| + t = (d+2)\cdot|H|$ vertices and at least $(d+1)\cdot t = (d+1)^2 \cdot |H| > d\cdot(d+2)\cdot|H|$ edges, since each vertex in $S'' \subseteq S'$ has degree at least $d+1$. But this induced subgraph violates the arboricity bound of Π. Hence, the sample contains an induced subgraph that is not in Π, which (by the hereditary feature of Π) implies that the subgraph induced by the sample is not in Π. ■

Using Claim 2.13.2, we may focus on the case that $\mathcal{D}(S'') \leq \epsilon/4$. In this case, we just ignore the set S'' (or rather omit the edges incident at it), and obtain a graph G'' that is $\epsilon/4$-far from Π. For sake of simplicity, from this point onwards, we assume that $G'' = G'$ and $S'' = \emptyset$, and so that all vertices $v \in S'$ satisfy $\mathcal{D}(v) > \epsilon'$. In this case, with high probability, the sample taken by the foregoing tester hits each vertex in $H \cup S'$, whereas the residual graph G' being tested is $\epsilon/4$-far from Π. We may assume that the subgraph induced by $H \cup S'$ does not have a forbidden minor, since otherwise we are done (i.e., the tester rejects).

Next, we cluster the vertices of $V' \setminus S'$ according to their neighborhood in H; that is, for each $H' \subseteq H$ of size at most d, we let $S_{H'} = \{v \in V' : \Gamma_{G'}(v) = H'\}$ denote the set of vertices in $V' = V \setminus H$ that neighbor each vertex in H' but no vertex in $H \setminus H'$. We get rid of clusters that have small weight (or rather omit the edges incident at them); that is, we redefine $S_{H'} \leftarrow \emptyset$ if $\mathcal{D}(S_{H'}) \leq \frac{\epsilon/8}{|H|^d}$, which means that we gave-up on a total weight of at most $\epsilon/8$ edges. Denoting the resulting subgraph of G' by G'', we note that G'' is not minor-free. (Actually, G'' is $\epsilon/8$-far from being minor-free.)

Foreseeing the case that the minor is obtained by contracting several vertices from some cluster, we upper-bound the number of such vertices. Specifically, we show that, without loss of generality, the number of vertices of a cluster $S_{H'}$ that appear in a contracted subgraph that yields a forbidden minor is at most $\binom{d}{2} + O(1) = O(d)^2 = O(1)$. This is the case since each of these vertices either remains uncontracted (as one of the $O(1)$ vertices of the minor) or is contracted into a vertex of H. (Recall that vertices in V' neighbor only vertices in H.) But the latter contraction makes sense only if it adds an edge between two vertices of H, whereas the number of possible edges that can be formed by contracting vertices in $S_{H'}$ is at most $\binom{d}{2}$ (or rather $\binom{d'}{2}$, where d' is the degree of a generic vertex in $S_{H'}$).

The key obervation is that the subgraph of G'' (and of G) that is contracted to a forbidden minor contains vertices of $H \cup S'$ as well as $B \stackrel{\rm def}{=} O(1)$ vertices of each $S_{H'}$, where $H' \subseteq H$ and $|H'| \leq d$. Recalling that $\mathcal{D}(v) \geq \epsilon'$ for each $v \in H \cup S'$ and $\mathcal{D}(S_{H'}) \geq \epsilon/8|H|^d$ for each such H', we are almost done. Specifically, we would be done if the foregoing subgraph (which yields a forbidden minor) had at most one vertex from each of the foregoing $S_{H'}$'s. In that case, with high probability, a sample of $\widetilde{O}(\epsilon/|H|^d)$ vertices would contain all vertices of $H \cup S'$ as well as a vertex from each of the relevant $S_{H'}$'s, where the vertices of $S_{H'}$ are interchangeable.

The foregoing assumption is removed by further partitioning the sets $S_{H'}$, analogously to the way this was done when establishing (Case 2 of) the furthermore claim of Theorem 2.8. Specifically, for each relevant set $H' \subseteq H$, we let $S''_{H'} \stackrel{\rm def}{=} \{v \in S' : \mathcal{D}(v) \leq \epsilon''\}$, where $\epsilon'' = \frac{\epsilon}{32B \cdot |H|^d}$. Next, let $S'_{H'} = S_{H'} \setminus S''_{H'}$ if $\mathcal{D}(S''_{H'}) < \epsilon/16|H|^d$, and $S'_{H'} = S_{H'}$ otherwise. Note that in both cases $\mathcal{D}(S'_{H'}) > \epsilon/16|H|^d$ and that $S'_{H'}$ either contains only vertices of weight at least ϵ'' or can be partitioned to B parts such each part has weight at least ϵ''. We conclude that, with high probability, a sample of $\widetilde{O}(\epsilon/|H|^d)$ vertices contains all vertices of $H \cup S'$ as well as an adequate number of vertices from each of $S_{H'}$'s (i.e., a number that at least equals the number of vertices of $S_{H'}$ in the subgraph that yields the forbidden minor). Hence, the foregoing tester rejects, since the sample it takes induces a subgraph that yields a forbidden minor. The theorem follows, where the size of the sample is $s = \widetilde{O}(1/\epsilon'') = \widetilde{O}(|H|^d/\epsilon) = \widetilde{O}(\epsilon^{-(d+1)})$. ■

3 The Bounded-Degree Graph Model

In this section, we generalize the notion of property testing in the bounded-degree graph model (a.k.a. the bounded incidence lists model, which was introduced in [19] and is reviewed in [14, Chap. 9]).

The bounded-degree graph model refers to a fixed (constant) degree bound, denoted $d \geq 2$. In this model, a graph $G = (V, E)$ of maximum degree d is represented by the incidence function $g : V \times [d] \to V \cup \{\perp\}$ such that $g(v, j) = u \in V$ if u is the $j^{\rm th}$ neighbor of v and $g(v, j) = \perp \notin V$ if v has less than j neighbors.[37]

As in the dense graph model, the tester is given oracle access to the representation of the input graph (i.e., to the incidence function g) as well as to a device, denoted $\mathcal{D}$, that returns identically and independently distributed elements in the graph's vertex-set. This distribution is also denoted $\mathcal{D}$. Following [15], we consider the case that the tester does not obtain any information about V as explicit input.

Distance between graphs is measured in terms of their foregoing representation and with reference to the distribution $\mathcal{D}$; that is, the distance between the graphs that are represented by the incidence functions $g : V \times [d] \to V \cup \{\perp\}$ and $g' : V \times [d] \to V \cup \{\perp\}$ is defined as

$$\begin{aligned} \delta_{\mathcal{D}}(g, g') &\stackrel{\rm def}{=} \Pr_{v \leftarrow \mathcal{D}, i \in [d]}[g(v, i) \neq g'(v, i)] \\ &= \sum_{v \in V} \mathcal{D}(v) \cdot \frac{|\{i \in [d] : g(v, i) \neq g'(v, i)\}|}{d} \end{aligned} \tag{16}$$

where $\mathcal{D}(v)$ denotes the probability that an element drawn from $\mathcal{D}$ equals v. For a graph property Π and a graph represented by the incidence function $g : V \times [d] \to V \cup \{\perp\}$, we let $\delta^{\Pi}_{\mathcal{D}}(g)$ denote the minimum of $\delta_{\mathcal{D}}(g, g')$ taken over all incidence functions $g' : V \times [d] \to V \cup \{\perp\}$ that represent graphs

[37]For simplicity, we adopt the standard convention by which the neighbors of v appear in arbitrary order in the sequence $g(v, 1), ..., g(v, \deg(v))$, where $\deg(v) \stackrel{\rm def}{=} |\{j \in [d] : g(v, j) \neq \perp\}|$ denotes the degree of v in G.

in Π. (We assume for simplicity that Π contains some graphs with vertex-set V; otherwise, one may define $\delta_{\mathcal{D}}^{\Pi}(g) > 1$.) When G is the graph represented by g, we may write $\delta_{\mathcal{D}}^{\Pi}(G)$ instead of $\delta_{\mathcal{D}}^{\Pi}(g)$. When the property Π is clear from the context, we may omit it from the notation and write $\delta_{\mathcal{D}}(\cdot)$ instead of $\delta_{\mathcal{D}}^{\Pi}(\cdot)$.

Definition 3.1 (VDF testing in the bounded-degree graph model): *For a fixed $d \in \mathbb{N}$, let Π be a property of graphs of degree at most d. A* VDF tester for the graph property Π (in the bounded-degree graph model) *is a probabilistic oracle machine T that is given access to two oracles, an incidence function $g : V \times [d] \to V \cup \{\perp\}$ and a device* (denoted $\mathcal{D}$) *that samples in V according to an arbitrary distribution $\mathcal{D}$, and satisfies the following two conditions* (for all sufficiently large V)*:*[38]

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* $\Pr[T^{g,\mathcal{D}}(\epsilon)=1] \geq 2/3$.
2. *Given $\epsilon > 0$ and oracle access to any $G = (V, E)$ and $\mathcal{D}$ such that $\delta_{\mathcal{D}}^{\Pi}(G) > \epsilon$, 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 $\delta_{\mathcal{D}}^{\Pi}(g) > \epsilon$, then it holds that* $\Pr[T^{g,\mathcal{D}}(\epsilon)=0] \geq 2/3$.

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* $\Pr[T^{g,\mathcal{D}}(\epsilon)=1] = 1$.

The definition of a proximity oblivious tester [21] is revised analogously. Specifically, a proximity oblivious tester is not given a proximity parameter (i.e., ϵ), it always accepts graphs in Π, and it rejects graphs G not in Π with probability that is related to $\delta_{\mathcal{D}}^{\Pi}(G)$. Specifically, the tester rejects with probability at least $\mathtt{dpf}(\delta_{\mathcal{D}}^{\Pi}(G))$, where $\mathtt{dpf} : (0, 1] \to (0, 1]$ is called the detection probability function.

As in the dense graph model, we may assume, without loss of generality (see Proposition 3.2), that the tester only makes queries that refer to vertices that appeared as answers to prior queries (i.e., either as samples provided by the sampling device or as answers to incidence queries). Note, however, that (unlike in the dense graph model) this does not mean that the number of queries can be upper-bounded in terms of the number of sample requests. On the other hand, whenever we only mention the query complexity, it is to be understood that the number of samples is similarly upper-bounded (since we may assume, with little loss of generality, that the tester makes some queries about the incidence relation of each sampled vertex).[39]

Proposition 3.2 (avoiding illegal queries in the bounded-degree graph model): *Suppose that Π can be tested by making at most s sampling requests and at most q queries to the graph, where both complexities depend on the proximity parameter and on label-invariant parameters of the distribution. Then, Π can be tested by making at most $3s$ sampling requests and at most $3q$ queries, while making queries only to vertices that occurred as answers to previous queries.*

An analogous statement holds with respect to proximity oblivious tester, where one can establish the asymptotic preservation of the detection probability function.

Proof: Mimicking the proof of Proposition 2.2, we let T be a tester as in the hypothesis. We consider a tester T' that, on input $g' : V' \times [d] \to V' \cup \{\perp\}$ and $\mathcal{D}'$, invokes T on related inputs $g : V \times [d] \to V \cup \{\perp\}$

[38] This relaxation, which allows the tester to fail on some tiny graphs, was not made in the first posting of this work [16]. See Footnote 20 for further discussion.

[39] The distinction between the sample and query complexity is reminiscent of [6]. However, in the current context, we envision the tester as making more queries than the number of samples, whereas the model in [6] envisions making less queries than the number of samples. Indeed, the possibility of making queries regarding elements that did not appear in the sample arises naturally in the bounded-degree graph model, but is not that natural in the settings envisioned in [6] (let alone in the dense graph model).

and $\mathcal{D}$, while constructing on-the-fly a random bijection π of $V' \uplus [100q \cdot |V'|]$ to $[(1+100q) \cdot |V'|]$, and setting $V = \pi(V')$. Specifically, when T asks for a sampled vertex, we obtain $v' \leftarrow \mathcal{D}'$ and answer with $\pi(v')$, where if π is not defined on v' then it is set at random (among the unused values). When T makes the query (v,i), algorithm T' proceeds as follows.

- If π^{-1} is not define on v, then it return a special error symbol (indicating that (v,i) is not in the domain of g).
- Otherwise, if $g'(\pi^{-1}(v),i) = \perp$, then it returns $\perp$.
- Lastly, letting $w \stackrel{\rm def}{=} g'(\pi^{-1}(v),i) \neq \perp$, the algorithm returns $\pi(w)$, where if π is undefined on w (indicating that w did not occur as a previous answer), it is set at random (among the unused values).

The key observation is that, on input g' and $\mathcal{D}'$, with very high probability (over the choice of π), algorithm T' emulates an execution of T on input g and $\mathcal{D}$ defined via a random bijection π such that $\mathcal{D}(v) = \mathcal{D}'(\pi^{-1}(v))$ and $g(v,i) = \pi(g'(\pi^{-1}(v),i))$. The deviation is due to the case that a query (v,i) was made although v was not obtained as an answer to a previous query and $\pi^{-1}(v) \in V'$ (although we behaved as if $v \notin \pi(V')$). The rest of the argument proceeds as in the proof of Proposition 2.2. ■

Organization of the rest this section. We first show that, as in the case of the dense graph model, it is possible to transform testers of the VDF version of the bounded-degree graph model into ones that have one-sided error, while incurring an overhead that is much lower than in the standard model. Again, this yields lower bounds on the testability in the current VDF model. Next, focusing on properties that are strongly testable with one-sided error in the standard (bounded-degree graph) model, we try to extend these testers to the VDF model. Specifically, building on the presentation in [14, Sec. 9.2], we present relatively simple testers for subgraph freeness, degree regularity, and being Eulerian. These poly$(1/\epsilon)$-query testers are based on conducting a small number of *very local* searches, but the parameters of these searches and their goals vary from one case to another. Likewise, building on results of [9], we study the testability of minor-freeness, focusing on cycle-free minors. Lastly, in Section 3.5, we present the t-removed VDF model (briefly discussed in Section 1.4), and extend our positive results regarding the VDF bounded-degree graph model to the t-removed VDF model, while incurring an overhead that is exponential in t.

3.1 One-sided error in the VDF model

In contrast to the situation in the standard model (see, e.g., the gap between the complexities of general versus one-sided error testing of cycle-freeness [19]), one-sided error comes at a low cost in the current model. This is the case since any strong tester in the VDF model can be transformed into a one-sided error tester (for this model) while maintaining query complexity that only depends on ϵ.

Theorem 3.3 (one-sided error testing reduces to general testing): *Let Π be a graph property that can be tested using $q(\epsilon)$ queries in the VDF bounded-degree graph model, where ϵ denotes the proximity parameter.*[40] *Then, Π has a one-sided error tester of query complexity $\exp(O(q(\epsilon)))$ in the current model.*

Theorem 3.3 implies that properties that do not have strong testers of one-sided error in the standard model cannot be strongly tested in the VDF model (see Corollary 3.4).

Proof: We follow the strategy used in the proof of Theorem 3.3. Let T be a (general) VDF tester of query complexity $q(\epsilon)$ for Π. Recall that by Proposition 3.2, we may assume, without loss of generality,

[40] Recall that by our convention, the query complexity also accounts for invocations of the sampling device.

that T does not query the graph on vertices that did not appear as answers to prior queries (or sample requests). We present a one-sided error tester for Π in the current model. On input parameter $\epsilon > 0$, and oracle access to a graph $G = (V, E)$ and a sampling device $\mathcal{D}$, the claimed tester operates as follows.

1. The tester takes $t = O(q(\epsilon)^2)$ samples, denoted $v_1, ..., v_t$, from the distribution $\mathcal{D}$. Note that the v_i's need not be distinct; that is, we may have $v_i = v_j$ for some $i \neq j$.
2. Letting $s = q(\epsilon)$, for every sequence $(i_1, ..., i_s)$ over $[t]$ and every possible random-pad r of T, the algorithm invokes $T(\epsilon)$ on randomness r and oracle access to G, while providing v_{i_j} as the j^{th} sampled vertex (i.e., as an answer to the j^{th} sampling request). That is, T is invoked on input ϵ, and provided access to G, but its randomness is set to r and the s samples it expects to receive from the sampling device are set to $v_{i_1}, ..., v_{i_s}$.
3. The algorithm accepts if and only if a majority of the invocations performed in Step 2 accept.

Since T only queries vertices that are provided as answers to prior queries, our algorithm queries the graph on vertices that are at distance at most $q(\epsilon) - 1$ from one of the sampled vertices.[41] Hence, the sample complexity of our algorithm is $t = O(q(\epsilon)^2)$ and its query complexity is smaller than $t \cdot d^{q(\epsilon)} = \exp(O(q(\epsilon)))$. The rest of the proof (i.e., showing that this algorithm constitutes a one-sided error tester for Π in the VDF model) is as in the proof of Theorem 3.3. ■

Using Theorem 3.3 towards establishing lower bounds in the current model. As noted above, Theorem 3.3 implies that properties that do not have strong testers of one-sided error in the standard (bounded-degree graph) model cannot be strongly tested in the current (VDF bounded-degree graph) model.

Corollary 3.4 (lower bounds via reduction from one-sided error testing): *Let Π be a graph property that can be tested using $q(\epsilon)$ queries in the VDF bounded-degree graph model, where ϵ denotes the proximity parameter. Then, Π has a one-sided error tester of query complexity $\exp(O(q(\epsilon)))$ in the standard bounded-degree graph model. The claim holds even if the VDF model tester is given the size of the graph as auxiliary input.*

Hence, lower bounds on the complexity of one-sided error testers in the standard bounded-degree graph model yield lower bounds on testers in the current model. In particular, this implies that testing cycle-freeness in the current model cannot be performed within complexity that only depends on the proximity parameter ϵ. Actually, the same holds for testing H-minor freeness for any H that contains a cycle (cf. [9]).

Beyond Corollary 3.4. As in the case of the dense graph model, also in the bounded-degree graph model strong one-sided error testing in the standard model does not suffice for strong testability in the VDF version.

Proposition 3.5 (on properties that are trivial in the standard model): *Suppose that the graph property Π of bounded-degree graphs satisfies the following two conditions:*

1. *For every n, the property Π does not contain the empty n-vertex graph* (i.e., a graph with no edges).

[41] Here, unlike in Steps 1 and 2 of the foregoing algorithm, we use $q(\epsilon)$ as an upper bound on the number of queries to the graph. Indeed, in Step 2, $s = s(\epsilon)$ equals the sample complexity of the original tester, whereas in Step 1 $t = O(s^2)$. Hence, the sample complexity of the resulting one-sided error tester is $O(s^2)$, whereas its actual query complexity is $O(s^2) \cdot d^{q(\epsilon)} = \exp(O(q(\epsilon) + \log s))$.

2. *There exists a constant $c \in (0,1)$ such that for all sufficiently large $n \in \mathbb{N}$ and every n-vertex graph $G = (V,E)$ there exists an n-vertex graph $G' = (V,E')$ in Π such that the symmetric difference between E and E' is at most n^c.*

Then, Π is not strongly testable in the current VDF model.

Note that properties covered by Proposition 3.5 are trivial to test (with one-sided error) in the standard (bounded-degree graph) model by accepting without making any queries if $\epsilon > n^{-(1-c)}$ (and querying the neighborhoods of all $n \leq (1/\epsilon)^{1/(1-c)}$ vertices otherwise). The foregoing (contrived) class contains the set of bounded-degree n-vertex graphs that contain a connected component of size at least $\sqrt{n}$. Anyhow, Proposition 3.5 establishes the "bounded-degree graph model" part of Theorem 1.2.

Proof Sketch: We follow the strategy used in the proof of Proposition 2.9, while setting $n = (1/d\epsilon)^{1/(1-c)}$ and using a bounded-degree graph $G \in \Pi$ that has n vertices and at most n^c edges. Fixing a set S of $n' = n^{1-c} < n - 2n^c$ isolated vertices in G, we observe that a potential one-sided error tester of Π must accept when seeing n' isolated vertices, since this view may occur when testing $G \in \Pi$. But this means that this alleged tester (which is not given the size of the graph) must accept the empty n'-vertex graph, although this graph is $1/dn'$-far from Π (under the uniform vertex-distribution). Contradiction follows, since $\epsilon = 1/n^{1-c}d = 1/n'd$. ■

The foregoing proof capitalizes on the tester's obliviousness of the size of the tested graph. The same holds with respect to the following proof, which uses a natural property for establishing the "bounded-degree graph model" part of Theorem 1.2. However, unlike Proposition 3.5 which becomes invalid when the tester obtains a rough estimate of the size of the graph (since such an estimation allows to distinguish large and small values of ϵ), the following result remains valid even if the tester obtains a very good estimate of this number (but not the exact value).[42]

Proposition 3.6 (on the difficulty of `Connectivity` in the VDF model): `Connectivity` *is not strongly testable in the VDF bounded-degree graph model.*

Proof: By Theorem 3.3 it suffices to show that `Connectivity` cannot be strongly tested with one-sided error in the VDF model. This is shown by observing that an alleged (strong) tester (with one-sided error for the VDF model) must always accepts when seeing a single connected component, since this connected component may be the entire graph. On the other hand, consider a graph consisting of such a connected component along with additional vertices (on top of such a connected component), when the distribution is concentrated uniformly on the former component (and assigns negligible probability to all other vertices). Specifically, consider an n-vertex graph with a connected component of size $k < n$ such that, for positive $\eta \ll 1/k$, each of the vertices of this component is assigned weight $(1-\eta)/k$ (and the rest of the weight (i.e., η) is partitioned arbitrarily among the other $n-k$ vertices). Then, this graph is $1/2dk$-far from being connected, where d is the degree bound, but an alleged tester that makes $q = q(1/2dk)$ queries accepts it with probability at least $1 - q \cdot \eta \approx 1$, where the approximation holds provided that η is sufficiently small as a function of k. (Indeed, we capitalized on the fact that the contribution of each edge to the distance is proportional to the weight of each of its endpoints, and so it is large even if only one of the endpoints has large weight. We note that the argument holds both for $n = k+1$ and $n \gg k$.) ■

[42]As shown in Section 4.2.1, the exact value is used to distinguish the case that the tester has visited all the vertices of the graph $G = (V,E)$ from the case that it has not visited some of the vertices. Typically, this matters only when $\min_{v \in V}\{\mathcal{D}(v)\} < \epsilon < 1/|V|$.

3.2 Testing subgraph freeness

Testing subgraph freeness (e.g., triangle-freeness), when the subgraph is not bipartite, is quite a challenge in the dense graph model. Recall that even testing triangle-freeness (in that model) involves the invocation of the Regularity Lemma. In contrast, we will present a relatively simple tester for the same properties in the current model (i.e., the VDF bounded-degree graph model). Recall that, for a fixed graph H, a graph G is called H-free if G contains no subgraph that is isomorphic to H.

We shall focus on the case that H is connected, although the general case can be handled similarly (yielding similar, but not identical results).[43] Let $\mathrm{rd}(H)$ denote the radius of H; that is, $\mathrm{rd}(H)$ is the smallest integer r such that there exists a vertex v in H such that all vertices in H are at distance at most r from v. Such a vertex v is called a center of H, and indeed H may have several centers (e.g., consider the case that H is a clique).

Theorem 3.7 (testing subgraph freeness (in the VDF bounded-degree graph model)): *Let $H = ([t], F)$ be a fixed* (connected) *graph of radius $r = \mathrm{rd}(H)$. Then, H-freeness has a* (one-sided error) *proximity-oblivious tester of query complexity $2d^{r+1}$ and linear detection probability. Furthermore, the time complexity of this tester is at most $(2d)^{rt}$.*

Proof: The tester presented in [14, Sec. 9.2.1] selects a start vertex, explores is r-neighborhood, and accept if and only if this subgraph is H-free. In the analysis it is shown that a center of a copy of H is selected with probability that is linearly related to the distance of the graph from being H-free, but this analysis is based on the fact that the cost of omitting an edge is the same for all edges. The argument could be extended to the case that the cost of omitting an edge is related to the minimum of the weights of its endpoints, but in our model the cost is related to the sum of these weights. This leads to the following revision of the original tester.

Algorithm 3.7.1 (testing H-freeness): *On input parameter d, given oracle access to the incidence function of a graph $G = (V, E)$, which has maximum degree d, and to a vertex-sampling device $\mathcal{D}$, the algorithm proceeds as follows.*

1. *Selects a vertex $s \in V$ according to $\mathcal{D}$* (i.e., $s \leftarrow \mathcal{D}$). *If s is an isolated vertex, then accept. Otherwise, set $v = s$ with probability $1/2$, and let v be a random neighbor of s otherwise.*
2. *Conducts a BFS of depth at most r starting from v.*
3. *Accept if and only if the explored subgraph is H-free.*

Step 2 is implemented by querying the incidence function, and so the query complexity of this algorithm is upper-bounded by $d + \sum_{i=0}^{r} d^i \cdot d < 2d^{r+1}$. Step 3 can be implemented by checking all possible mappings of H to the explored graph, and so the time complexity of Algorithm 3.7.1 is upper-bounded by $\binom{2d^r}{t} \cdot (t!) < (2d)^{rt}$.

Algorithm 3.7.1 never rejects a graph that is H-free, since H-freeness is preserved by subgraphs of the original graph. (Algorithm 3.7.1 can be modified to check induced subgraph freeness, while noting that this property is preserved by induced subgraphs of the original graph.) It is left to analyze the detection probability of Algorithm 3.7.1.

Claim 3.7.2 (the detection probability of Algorithm 3.7.1): *Algorithm 3.7.1 rejects the graph G with probability at least $\delta_{\mathcal{D}}(G)/2$, where $\delta_{\mathcal{D}}(G)$ denotes the distance* (according to $\mathcal{D}$) *of G from being H-free.*

[43] Cf. [14, Exer. 9.3], which refers to the case that $\mathcal{D}$ is uniform over V.

Proof: A vertex $v \in V$ is called detecting if it is a center of a copy of H that resides in G. Letting $\Gamma(v)$ denote the set of neighbours of vertex v, we observe that

$$\delta_{\mathcal{D}}(G) \leq \sum_{v \text{ is detecting}} \sum_{u \in \Gamma(v)} \left(\frac{\mathcal{D}(v)}{d} + \frac{\mathcal{D}(u)}{d} \right) \tag{17}$$

$$\leq \sum_{v \text{ is detecting}} \left(\mathcal{D}(v) + \sum_{u \in \Gamma(v)} \frac{\mathcal{D}(u)}{d} \right) \tag{18}$$

since omitting all edges incident at detecting vertices makes the graph H-free. On the other hand, in our algorithm, a BFS is started at (a detecting) vertex v with probability at least

$$\frac{\mathcal{D}(v)}{2} + \sum_{s \in \Gamma(v)} \frac{\mathcal{D}(s)}{2d}$$

where the first term is due to selecting v is Step 1 (i.e., $v = s$) and the second term is due to selecting a neighbor of v (i.e., selecting s and moving to v). Hence, the probability that the algorithm rejects is at least

$$\sum_{v \text{ is detecting}} \left(\frac{\mathcal{D}(v)}{2} + \sum_{s \in \Gamma(v)} \frac{\mathcal{D}(s)}{2d} \right) \geq \delta_{\mathcal{D}}(G)/2,$$

where the inequality is due to Eq. (17)–(18). The claim follows. ■

This completes the proof of the theorem. ■

3.3 Testing degree regularity

Degree-regularity cannot be strongly tested in the current model, since the tester (which is not given the number of vertices in the graph) cannot distinguish a graph $G = (V, E)$ with an even number of vertices that are each of the same odd degree (i.e., $|V| \equiv 0 \pmod 2$ and $|\Gamma(v)| = |\Gamma(u)| \equiv 1 \pmod 2$ for all $u, v \in V$) from a graph that is obtained from G by adding an isolated vertex, whereas the latter graph is $1/2d$-far from regular, where in both cases the distribution is uniform over V. Furthermore, even when guaranteed that the number of vertices is even, the tester cannot distinguish the graph G from a graph obtained from G by adding at most d isolated vertices.[44] (The argument mimics the proof of Proposition 3.6, and involves referring to $\epsilon = 1/2d|V|$ and to a distribution that is uniform over V and assigns the isolate vertices very small weight.)[45]

In light of the foregoing, we relax the definition of testing degree regularity so to allow for arbitrary verdict in case the graph has few exceptional vertices. That is, for every $d' \leq d$, we let $\Pi'_{d'}$ denote the set of graphs in which at most d' vertices have degree different from d', and let $\Pi' = \bigcup_{d' \in \{0,1,\ldots,d\}} \Pi'_{d'}$. A relaxed tester of degree regularity is required to accept every regular graph and reject (w.h.p) every graph that is far from Π'. The definition of a relaxed POT is adapted accordingly (see next).

Theorem 3.8 (relaxed testing of degree regularity (in the VDF bounded-degree graph model)):[46] *In the VDF bounded-degree graph model, with degree bound d, degree regularity has a relaxed* (one-sided error) *proximity-oblivious tester of* (query and) *time complexity $O(\log(d+1))$ and linear detection probability; that is:*

[44]Indeed, this stands in contrast to the statement of Theorem 3.8 in the first posting of this work [16].

[45]Note that G is regular, whereas the augmented graph is ϵ-far from regular (because some isolated vertex must be connected to some vertex in V). The argument extends to the case that the exceptional vertices are inter-connected in an arbitrary manner.

[46]This corrects the statement of Theorem 3.8 in the first posting of this work [16].

1. *The relaxed tester accepts every regular graph with probability 1.*

2. *The relaxed tester rejects every graph that is at distance δ from Π' with probability $\Omega(\delta)$.*

Proof: The tester presented in [14, Sec. 9.2.2] selects uniformly two vertices and compares their degrees. In the analysis it is shown that if the tester rejects a graph with small probability, then the graph is close to being regular. Specifically, this is shown by relating the fraction of incidences (i.e., arguments to the incidence function) that need to be modified to the rejection probability. Noting that, in the current context, difference incidences have different weight, we need to relate the weight of modified incidences to the probability that (the degree of) the corresponding vertices were checked. Hence, in the following algorithm, the vertex v is not selected according to $\mathcal{D}$, but rather according to a related probability that enables the aforementioned analysis.

Algorithm 3.8.1 (testing degree regularity): *On input parameter d, oracle access to the incidence function of a graph $G = (V, E)$, which has maximum degree d, and to a vertex-sampling device $\mathcal{D}$, the algorithm proceeds as follows.*

1. *Selects $u \in V$ arbitrarily* (e.g., $u \leftarrow \mathcal{D}$) *and select v by drawing $s \leftarrow \mathcal{D}$ and taking a lazy random walk of length one from s* (i.e., let $v = s$ with probability $1/2$ and let v be a random neighbor of s otherwise).

2. *Determines the degrees of u and v.*

3. *If the degrees of u and v are different, then the algorithm rejects. Otherwise, the algorithm accepts.*

Step 2 is implemented by a binary search on the incidence list of each selected vertex, and so the query (and time) complexity of this algorithm is $2 \cdot \lceil \log_2(d+1) \rceil$. Evidently, Algorithm 3.7.1 never rejects a regular graph.

The analysis of Algorithm 3.7.1 is based on generalizing the claim that the "distance" of G to Π' is upper bounded in terms of the difference between its own degrees, where in the standard formulation (see [14, Clm. 9.5.1]) the distance and the difference refer to the case that $\mathcal{D}$ is uniform on V. Here we generalize this local-vs-global claim to arbitrary $\mathcal{D}$, except that we refer to Π' rather than to degree regularity proper. (Consequently, our argument is simpler than the proof of [14, Clm. 9.5.1], which is merely a reformulation of [14, Clm. 8.5.1] (although the two claims are employed in different settings, which is the reason for the reformulation).)

Claim 3.8.2 (local-vs-global distance to degree regularity): *Let $d_G(v) \le d$ denote the degree of vertex v in the graph $G = (V, E)$. Let $p(v)$ denote the probability that a lazy random walk of length one that starts at $s \leftarrow \mathcal{D}$ reaches v. For any $d' \in \{0, 1, ..., d\}$, if $\sum_{v \in V: d_G(v) \neq d'} p(v) \le B$, then $\delta_{\mathcal{D}}^{\Pi'_{d'}}(G) \le (4d+2) \cdot B$.*

Proof: We modify G in two stages, while keeping track of the "cost" of these modifications.[47] In the first stage we reduce all vertex degrees to at most d', by scanning all vertices and omitting at most $d_G(v) - d'$ edges incident at each vertex $v \in H \stackrel{\text{def}}{=} \{u : d_G(u) > d'\}$. The cost of this modification is at most

$$\Pr_{v \leftarrow \mathcal{D}}[v \in H] \cdot (d - d')/d + \mathrm{Exp}_{v \leftarrow \mathcal{D}}[|\{u \in H : \{u, v\} \in E\}|]/d, \tag{19}$$

where the first term in Eq. (19) is due to removing edges from the incidence list of vertices in H, and the second term is due to effect of these removals at the other endpoint. Letting $H' = \Gamma(H)$ denote the set of neighbors of H in G, we upper bound Eq. (19) by $\Pr_{v \leftarrow \mathcal{D}}[v \in H] + \Pr_{v \leftarrow \mathcal{D}}[v \in H']$. Hence,

[47] The proof of [14, Clm. 9.5.1] (or rather [14, Clm. 8.5.1]) uses three stages, where the third stage (omitted here) is most complex.

$\delta_{\mathcal{D}}(G,G') \leq \Pr_{v\leftarrow\mathcal{D}}[v \in H] + \Pr_{v\leftarrow\mathcal{D}}[v \in H']$. Furthermore, denoting the resulting graph by G' and the degree of v in G' by $d_{G'}(v) \leq d'$, we have

$$\Pr_{v\leftarrow\mathcal{D}}[d_{G'}(v) < d'] \leq \Pr_{v\leftarrow\mathcal{D}}[d_G(v) < d'] + \Pr_{v\leftarrow\mathcal{D}}[v \in H'], \tag{20}$$

since $d_{G'}(v) < d' \leq d_G(v)$ only if v neighbors a vertex of H (which means that $v \in H'$).

In the second stage, we insert an edge between each pair of vertices that are currently non-adjacent and have both degree smaller than d'. Thus, we obtain a graph G'' such that $\{v : d_{G''}(v) < d'\}$ is a clique in G'' (and $d_{G''}(v) \leq d'$ for all v). The cost of this modification is at most $\Pr_{v\leftarrow\mathcal{D}}[d_{G'}(v) < d']$. Note that $G'' \in \Pi'_{d'}$ (although it may not be regular).[48] The total cost of all modifications is $\delta_{\mathcal{D}}(G,G') + \delta_{\mathcal{D}}(G',G'')$, and so

$$\begin{aligned}
\delta_{\mathcal{D}}(G,G'') &\leq \delta_{\mathcal{D}}(G,G') + \delta_{\mathcal{D}}(G',G'') \\
&\leq \Pr_{v\leftarrow\mathcal{D}}[v \in H] + \Pr_{v\leftarrow\mathcal{D}}[v \in H'] + \Pr_{v\leftarrow\mathcal{D}}[d_{G'}(v) < d'] \\
&\leq \Pr_{v\leftarrow\mathcal{D}}[d_G(v) > d'] + \Pr_{v\leftarrow\mathcal{D}}[d_G(v) < d'] + 2 \cdot \Pr_{v\leftarrow\mathcal{D}}[v \in H'] \\
&= \sum_{v:d_G(v)\neq d'} \mathcal{D}(v) + 2 \cdot \sum_{v\in\Gamma(H)} \mathcal{D}(v)
\end{aligned}$$

where the third inequality uses Eq. (20). Observing that $p(v) \geq \mathcal{D}(v)/2$ and $p(v) \geq \mathcal{D}(s)/2d$ if $v \in \Gamma(s)$, we conclude that $\delta_{\mathcal{D}}(G,G'') \leq 2 \cdot \sum_{v:d_G(v)\neq d'} p(v) + 2 \cdot 2d \cdot \sum_{s\in H} p(s)$, and the claim follows. ∎

The lower bound on the rejection probability of Algorithm 3.7.1 follows from Claim 3.8.2. Specifically, let d' denote the degree of the first vertex (i.e., u) selected in Step 1 of the algorithm. Then, the probability that the second vertex selected in Step 1 (i.e., v) has degree that differs from d' is at least $\delta_{\mathcal{D}}^{\Pi'}(G)/6d$. The theorem follows. ∎

Testing whether a graph is Eulerian. Recall that a graph is called Eulerian if all its vertices have even degree. (Note that we do not require here that the graph be connected.) We can easily test if a graph is Eulerian by sampling a random vertex (according to $\mathcal{D}$) and determining its degree, but again the analysis is not trivial because we need to preserve the degree bound (and the simplicity) of the graph. Actually, this is a problem only for odd d, since otherwise we can just pair all vertices of odd degree and flip the adjacency relation of each of these pairs. However, in case d is odd, we may have odd vertices of degree d, and these vertices may be nonadjacent (and so we cannot connect them while maintaining the degree bound). Actually, by proceeding as in the case of even d, we may reach a situation in which all vertices of odd degree are nonadjacent (and have degree d). Considering two such vertices, u and v, we observe that their neighbors must have even degree (or else we can proceed with the aforementioned process). If u neighbors u' and v neighbors v', then we omit the edges $\{u,u'\}$ and $\{v,v'\}$ and flip the adjacency relation of the pair (u',v') (assuming that $u' \neq v'$, and do nothing otherwise). Observing that we have also modified the incidences of vertices of even degree that neighbor vertices of odd degree, we need to modify the tester so that it checks not only the degree of the sampled vertex but also the degree of a random neighbor of the sampled vertex.

3.4 Testing minor-freeness

We first recall that strongly testing cycle-freeness in the VDF (bounded-degree graph) model is not possible. The same holds for strongly testing H-minor freeness for any H that contains a cycle.[49] This follows by combining Corollary 3.4 with the fact that in the standard VDF (bounded-degree graph) model these properties have no strong tester with one-sided error tester (see [9, Thm. 6.1]).

[48] Hence, the third stage in the proof of [14, Clm. 9.5.1] (or rather [14, Clm. 8.5.1]) is omitted here.

[49] Recall that C_3-minor freeness is a special case, which is equivalent to cycle-freeness, where C_3 denotes the 3-vertex cycle.

Recalling that, for every H that is cycle-free, strongly testing H-minor freeness is possible in the standard (bounded-degree graph) model with one-sided error tester (see [9, Sec. 7]), we ask whether this is also possible in the current model. We answer this question positively in two natural extreme cases: the cases of H of maximal and of minimal diameter (i.e., a path and a star, respectively). Given that the general case is complex enough in the standard model, we refrain from addressing it here, leaving it to future research.

Testing that the graph contains no simple k-long path. The existence of a simple k-long path is equivalent to having a P_k-minor, where P_k denotes the k-long path. A natural proximity oblivious tester, analyzed in [9, Sec. 7.2], consists of searching for such a path at random. Specifically, in the standard setting, one may select uniformly a random start vertex, take a random k-step walk from it, and reject if and only if the walk corresponds to a simple path. This simple solution does not quite work in the current setting, since the vertex distribution $\mathcal{D}$ may be concentrated on vertices that are at the internal vertices of k-length paths; indeed, consider the case that the graph consists of a collection of isolated k-long paths.

The source of the problem is a possible imbalance between the (probability) weights of the endpoints of "violating" edges. Specifically, in the foregoing example all edges are heavy, due to the weight of at least one of their endpoints, but the k-path can only be detected when starting the search at a leaf, whereas the leaves may have tiny weight. Nevertheless, a simple variation on the foregoing algorithm does work.

Algorithm 3.9 (a P_k-minor freeness tester): *On oracle access to a graph $G = (V, E)$ and a vertex sampling device $\mathcal{D}$, the tester proceeds as follows.*

1. *Obtains a sample from $\mathcal{D}$; that is, $v \leftarrow \mathcal{D}$.*
2. *Select uniformly $i \in [k]$;*
3. *Take two random walks from v, one of length i and the other of length $k - i$.*
4. *Reject if and only if joining these walks yields a simple k-long path.*

Clearly, this tester never rejects a P_k-minor free graph, and so it is left to analyze the probability that this tester rejects graphs that are far from being P_k-minor free.

Claim 3.10 (analysis of Algorithm 3.9): *Suppose that $G = (V, E)$ is at distance δ from being P_k-minor free with respect to the distribution $\mathcal{D}$. Then, Algorithm 3.9 rejects with probability at least $\delta/\exp(O(k))$, where the O-notation hides a dependence on the degree bound d.*

Proof: We follow the strategy used in the proof of [9, Clm. 7.3], but our actual analysis is different.[50] We call an edge $\{u, v\}$ `bad` if it resides on a simple path of length k. Let ρ denote the total probability, under $\mathcal{D}$, of bad edges in G, where the probability of an edge $\{u, v\}$ is $(\mathcal{D}(u) + \mathcal{D}(v))/d$, just as it is (implicitly) in Definition 3.1. Then, on the one hand, we reject G with probability at least $\rho/(kd^{k-1})$, since an endpoint of a bad edge $\{u, v\}$ is selected in Step 1 with probability $\mathcal{D}(u) + \mathcal{D}(v)$ and the corresponding path is viewed with probability at least $1/(k \cdot d^k)$. On the other hand, $\rho \geq \delta$, because omitting all bad edges from G results in a graph that has no simple k-long paths. ∎

[50]In particular, recall that the algorithm analyzed in [9, Clm. 7.3] actually fails. In addition, our notion of "badness" refers to edges rather than to vertices, since otherwise one cannot relate the weight of bad objects to the cost (in terms of distance) of omitting them from the graph.

Testing that the graph contains no tree with k leaves. The existence of a tree with (at least) k leaves is equivalent to having a k-star as a minor, where a k-star, denoted S_k, is a $(k+1)$-vertex tree that has k leaves and a single vertex of degree k, called its center.

A natural tester, used in [9, Sec. 7.3], consists of searching for such a tree at random. Specifically, in the standard setting, one may select uniformly a random start vertex, start a BFS from it and suspend the search if either a layer with at least k vertices is found or more than $O(k/\epsilon)$ vertices where encountered (where $\epsilon > 0$ is, as usual, the proximity parameter). In the first case the test rejects, and in the second case it (tentatively) accepts (where actual acceptance requires $O(1/\epsilon)$ trials). This simple solution does not quite work in the current setting, since the vertex distribution $\mathcal{D}$ may be concentrated on the leaves of isolated k-stars, where here we assume that $k \in \{2, ..., d\}$ (although the following argument can be extended to general k).

As in the case of simple k-long paths, the source of the problem is a possible imbalance between the weights of the endpoints of "violating" edges. Specifically, in the foregoing example, all edges of the k-star are heavy, due to the weight of their leaves, but a k-star may be detected only when starting the search at its center, which may have tiny weight. We fix the problem by starting the BFS either at the sampled vertex or at one of its neighbors, selected at random. Also, for $k = O(1)$, we use a slightly deeper BFS (i.e., $\widetilde{O}(1/\epsilon)$ rather than $O(1/\epsilon)$ layers).[51]

Following [9, Sec. 7.3], the key observation here is that a graph $G = (V, E)$ is S_k-minor free if and only if for every set S such that the subgraph induced by S is connected it holds that the set S has less than k neighbors (in $V \setminus S$). As shown in the proof of Claim 3.12, this implies that if for every connected set S of size at most $O(\epsilon^{-1} \log(k/\epsilon))$, the set S has less than k neighbors in $V \setminus S$, then the graph is ϵ-close to being S_k-minor free. (This is the case because the (small-cuts) hypothesis allows for removing a set of edges of total weight at most $\epsilon/2$ so to partition the graph into connected components that are each S_k-minor free.)

Algorithm 3.11 (a S_k-minor freeness tester): *On input $\epsilon > 0$, and oracle access to a graph $G = (V, E)$ and a sampling device $\mathcal{D}$, the tester performs the following steps $O(1/\epsilon)$ times.*

1. *Obtain a sample from $\mathcal{D}$; that is, $v \leftarrow \mathcal{D}$. With probability $1/2$, the tester sets $s = v$, and otherwise it lets s be a uniformly selected neighbor of v.*
2. *Perform a BFS starting at s and stopping as soon as either $t = 5\epsilon^{-1} \ln(k/\epsilon)$ layers were explored or a layer with at least k vertices was encountered.*

   Note that it may happen that the BFS terminates before either of these conditions hold; this can only happen if s resides in a connected component of size smaller than $t \cdot k$.
3. *If the explored subgraph contains a S_k-minor, then the test rejects.*

If none of the foregoing trials rejected, then the tester accepts.

Clearly, Algorithm 3.11 never rejects a S_k-minor free graph, and its query complexity is at most $O(1/\epsilon) \cdot t \cdot k \cdot d = \widetilde{O}(k/\epsilon^2)$. Thus, all that is left is to prove the following claim.

Claim 3.12 (analysis of Algorithm 3.11): *Suppose that $G = (V, E)$ is ϵ-far from being S_k-minor free with respect to the distribution $\mathcal{D}$. Then, each of the $O(1/\epsilon)$ trials performed by Algorithm 3.11 rejects with probability at least $\epsilon/4d$.*

Proof: We follow the strategy used in the proof of [9, Clm. 7.5], but our actual analysis is more complex because we have to deal with weights of edges and vertices rather than with their number.

[51] Actually, the hidden dependence of the depth on k is lesser (i.e., it is logarithmic rather than linear), due to our more refined analysis.

We call a vertex s bad if there exists a set S such that (i) in the subgraph of G induced by S all vertices are at distance smaller than $t = 5\epsilon^{-1}\ln(k/\epsilon)$ from s, and (ii) the set S has at least k neighbors in $V \setminus S$ (i.e., $|\{\{v,w\} \in E : (v,w) \in S \times (V \setminus S)\}| \geq k$). Letting G_S denote the subgraph of G induced by S and its neighbors, we observe that Condition (ii) implies that G_S contains a S_k-minor, whereas Condition (i) implies that a BFS of G_S started at v is completed after exploring at most t layers.[52] Hence, if a bad vertex is chosen in Step 1 (equiv., serves as a start vertex s for Step 2), then Algorithm 3.11 rejects in Step 3, because either a $(t-1)$-step BFS of G starting at s reaches a layer with at least k vertices or it reaches all vertices in the set S (whereas contracting S yields a vertex with at least k neighbors). Denoting the set of bad vertices by B, we infer that the probability that a trial (performed by Algorithm 3.11) rejects is at least

$$\rho \stackrel{\rm def}{=} \frac{1}{2} \cdot \Pr_{s \leftarrow \mathcal{D}}[s \in B] + \frac{1}{2} \cdot \frac{1}{d} \cdot \Pr_{v \leftarrow \mathcal{D}}[\exists s \in B : \{v,s\} \in E].$$

We next show that G must be $(2d \cdot \rho + (\epsilon/2))$-close to S_k-minor free, and so $\rho > \epsilon/4d$ follows.

Let $G^{(0)}$ denote the graph obtained from G by omitting all the edges that are incident at bad vertices. Then, the distance (wrt $\mathcal{D}$) of $G^{(0)}$ from G is at most

$$\begin{aligned}
\sum_{\{u,w\} \in E : u \in B} \frac{\mathcal{D}(u) + \mathcal{D}(w)}{d} &\leq \frac{1}{d} \cdot \left(\sum_{u \in B} |\{w : \{u,w\} \in E\}| \cdot \mathcal{D}(u) + \sum_{w \in V} |\{u \in B : \{u,w\} \in E\}| \cdot \mathcal{D}(w) \right) \\
&\leq \frac{1}{d} \cdot (d \cdot \Pr_{s \leftarrow \mathcal{D}}[s \in B] + d \cdot \Pr_{w \leftarrow \mathcal{D}}[\exists s \in B : \{w,s\} \in E]) \\
&\leq 2d \cdot \rho.
\end{aligned}$$

It remains to show that $G^{(0)}$ is $\epsilon/2$-close to being S_k-minor free. The rest of our analysis proceeds in iterations. We shall construct a sequence of graphs $G^{(0)}, G^{(1)}, G^{(2)}, ...$ such that each $G^{(i)}$ is $\epsilon/2$-close to $G^{(0)}$ and the last graph in the sequence is S_k-minor free.

If the current graph $G^{(i-1)}$ is S_k-minor free, then we are done. Otherwise, we pick an arbitrary connected component that contains a S_k-minor, and let $h^{(i)}$ be the heaviest vertex in this component (w.r.t probability weights assigned by $\mathcal{D}$). Since $h^{(i)}$ is not bad, *this S_k-minor of $G^{(i-1)}$ must contain vertices of distance greater than $t+1$ from $h^{(i)}$*. This is because, otherwise, contracting all edges in the subgraph that yields this S_k-minor, except for the k edges incident at the leaves of this S_k-minor, yields a set S that witnesses the badness of $h^{(i)}$.

Now, consider *as a mental experiment*, executing a BFS on $G^{(i-1)}$ starting at the vertex $h^{(i)}$, and suspending the execution once we reach a layer such that the total weight of all vertices in previous layers is at least $2/\epsilon$ times larger than the total weight of all vertices in the current and next layers (see more precise definition below). That is, for $j = 0, ..., t$, let L_j denote the set of all vertices that are at distance j from $h^{(i)}$ in $G^{(i-1)}$; indeed, $L_0 = \{h^{(i)}\}$ and $|L_j| < k$ (since otherwise contracting $L_0, ..., L_{j-1}$ yields a set that witnesses the badness of $h^{(i)}$). We suspend the BSF at layer $j \in \{0, 1, ..., t\}$ if $\mathcal{D}(\bigcup_{j' \in \{0,...,j\}} L_{j'}) \geq (2/\epsilon) \cdot \mathcal{D}(L_j \cup L_{j+1})$. Assuming that the BFS is not suspended in layer j (equiv., when encountering layer $j+1$), it holds that $\mathcal{D}(L_j \cup L_{j+1}) > (\epsilon/2) \cdot \mathcal{D}(\bigcup_{j' \in \{0,...,j\}} L_{j'})$. This implies that

$$\begin{aligned}
\mathcal{D}\left(\bigcup_{j' \in \{0,...,j+1\}} L_{j'} \right) &= \mathcal{D}\left(\bigcup_{j' \in \{0,...,j-1\}} L_{j'} \right) + \mathcal{D}(L_j \cup L_{j+1}) \\
&> \left(1 + \frac{\epsilon}{2}\right) \cdot \mathcal{D}\left(\bigcup_{j' \in \{0,...,j-1\}} L_{j'} \right) \\
&\geq \left(1 + \frac{\epsilon}{2}\right)^{\lfloor j/2 \rfloor} \cdot \mathcal{D}(L_0)
\end{aligned}$$

[52] Condition (i) also implies that G_S is connected and that $s \in S$.

and it follows that

$$\mathcal{D}(L_j \cup L_{j+1}) \;>\; (\epsilon/2) \cdot \mathcal{D}\left(\bigcup_{j' \in \{0,\ldots,j\}} L_{j'}\right) \;>\; (\epsilon/2) \cdot (1+(\epsilon/2))^{\lfloor (j-1)/2 \rfloor} \cdot \mathcal{D}(L_0).$$

Since $\mathcal{D}(L_0) = \max_{v \in \bigcup_{j \in \{0,\ldots,t\}} L_j}\{\mathcal{D}(v)\}$ and $|L_j| < k$, it follows that $2k \cdot \mathcal{D}(L_0) > (\epsilon/2) \cdot (1+(\epsilon/2))^{\lfloor (j-1)/2 \rfloor} \cdot \mathcal{D}(L_0)$, which implies that this imaginary BFS must be suspended before encountering layer $j+1$ such that $j < 2 + 2 \cdot \log_{1+(\epsilon/2)}(4k/\epsilon)$, whereas $\log_{1+(\epsilon/2)}(4k/\epsilon) =\approx 2\epsilon^{-1} \ln(4k/\epsilon) < 0.5t - 1$. Letting $S^{(i)} = \bigcup_{j' \in \{0,\ldots,j+1\}} L_{j'}$, note that $S^{(i)} \ni h^{(i)}$ is connected and contains vertices that are at distance at most t from $h^{(i)}$. We now obtain $G^{(i)}$ by omitting from $G^{(i-1)}$ the edges of the cut $(S^{(i)}, V \backslash S^{(i)})$, while observing that the weight of these edges is at most $\epsilon/2$ times the weight of $S^{(i)}$, since upon suspending the imaginary BFS at layer j we have $\mathcal{D}(L_j \cup L_{j+1}) \leq (\epsilon/2) \cdot \mathcal{D}(\bigcup_{j' \in \{0,\ldots,j\}} L_{j'}) = (\epsilon/2) \cdot \mathcal{D}(S^{(i)})$, whereas the cut edges are a subset of the edges that have both endpoints in $L_j \cup L_{j+1}$. Furthermore, $G^{(i)}_{S^{(i)}}$ is S_k-minor free (and $S^{(i)}$ will not intersect with any future $S^{(i')}$).

When the process ends, we have a S_k-minor free graph, G'. During this process, we omitted edges of total weight at most $\epsilon/2$, and so we conclude that $G^{(0)}$ is $\epsilon/2$-close to G', which establishes our claim (that $G^{(0)}$ is $\epsilon/2$-close to being S_k-minor free), and completes the entire proof. ■

On testing F-minor freeness, for general forests F: As stated in the beginning of this section, the strong one-sided error testers known for F-minor freeness in the standard bounded-degree graph model, when F is an arbitrary forest [9], beg the question of whether such results can be obtained in the vertex-distribution-free setting. Given the complexity of the original analyses, we refrain from addressing this question here.

3.5 The t-removed VDF model

In continuation to the discussion in Section 1.4, we now present the t-removed VDF model. Recall that we envision processes that, in addition to selecting vertices according to some distribution $\mathcal{D}$, may also take walks of length at most t (from any of these selected vertices). In this setting, the importance of a vertex v is defined as proportional to the sum of the probabilities of all vertices that are at distance at most t from v. Specifically, fixing a graph $G = (V, E)$ of degree bound d, we let $\Gamma_t^G(v)$ denote the set of vertices that are at distance at most t from v in the graph G. For a distribution $\mathcal{D}$ over V, we let $\mathcal{D}_t^G(v) = \sum_{w \in \Gamma_t^G(v)} \mathcal{D}(w)$, and $\overline{\mathcal{D}}_t^G(v) = \mathcal{D}_t^G(v) / \sum_{u \in V} \mathcal{D}_t^G(u)$. We define the ($t$-removed) distance of a graph G, represented by $g : V \times [d] \to V \cup \{\perp\}$ to a graph represented by $g' : V \times [d] \to V \cup \{\perp\}$ as

$$\delta_{t,\mathcal{D}}(g, g') \stackrel{\text{def}}{=} \sum_{v \in V} \overline{\mathcal{D}}_t^G(v) \cdot |\{i \in [d] : g(v,i) \neq g'(v,i)\}|/d. \tag{21}$$

Note that $\delta_{t,\mathcal{D}}(g, g')$ does not necessarily equal $\delta_{t,\mathcal{D}}(g', g)$, since $\mathcal{D}_t^G$ does not necessarily equal $\mathcal{D}_t^{G'}$, unless $t = 0$. Indeed, in the case of $t = 0$, it holds that $\overline{\mathcal{D}}_t^G \equiv \mathcal{D}_0^G \equiv \mathcal{D}$ and $\delta_{t,\mathcal{D}}(g, g') = \delta_{\mathcal{D}}(g, g')$ follows. Lastly, note that

$$1 \;\leq\; \sum_{v \in V} \mathcal{D}_t^G(v) \;\leq\; \max_{v \in V}\{|\Gamma_t^G(v)|\} \;\leq\; \sum_{i=0}^{t} d^i \;<\; 2 \cdot d^t.$$

For a graph property Π and a graph represented by the incidence function $g : V \times [d] \to V \cup \{\perp\}$, we let $\delta_{t,\mathcal{D}}^{\Pi}(g)$ denote the minimum of $\delta_{t,\mathcal{D}}(g, g')$ taken over all incidence functions $g' : V \times [d] \to V \cup \{\perp\}$ that represent graphs in Π. (We assume for simplicity that Π contains some graphs with vertex-set V;

otherwise, one may define $\delta_{t,D}^{\Pi}(g) > 1$.) The following definition is identical to Definition 3.1, except that in the second item $\delta_{\mathcal{D}}^{\Pi}(G)$ is replaced by $\delta_{t,\mathcal{D}}^{\Pi}(G)$.

Definition 3.13 (VDF testing in the t-removed model): *For fixed $d, t \in \mathbb{N}$, let Π be a property of graphs of degree at most d. A* VDF tester for the graph property Π in the t-removed model *is a probabilistic oracle machine T that is given access to two oracles, an incidence function $g : V \times [d] \to V \cup \{\perp\}$ and a device* (denoted $\mathcal{D}$) *that samples in V according to an arbitrary distribution $\mathcal{D}$, and satisfies the following two conditions* (for all sufficiently large V)*:*

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* $\Pr[T^{g,\mathcal{D}}(\epsilon) = 1] \geq 2/3$.
2. *Given $\epsilon > 0$ and oracle access to any $G = (V, E)$ and $\mathcal{D}$ such that $\delta_{t,\mathcal{D}}^{\Pi}(G) > \epsilon$, 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 $\delta_{t,\mathcal{D}}^{\Pi}(g) > \epsilon$, then it holds that* $\Pr[T^{g,\mathcal{D}}(\epsilon) = 0] \geq 2/3$.

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* $\Pr[T^{g,\mathcal{D}}(\epsilon) = 1] = 1$.

The definition of a proximity oblivious tester is revised analogously. Definition 3.1 (and its proximity oblivious version) are obtained when setting $t = 0$.

Theorem 3.14 (reducing the t-removed model to the VDF model): *For every $t \in \mathbb{N}$, if a graph property Π is strongly testable in the VDF bounded-degree graph model, then Π is strongly testable in the t-removed model. Specifically, if the VDF tester has sample complexity s and query complexity q, then the t-removed tester has sample complexity s' and query complexity q' such that $s'(\epsilon) = s(\exp(-t) \cdot \epsilon)$ and $q'(\epsilon) = q(\exp(-t) \cdot \epsilon) + t \cdot s'(\epsilon)$. Furthermore, one-sided error is preserved.*

A natural question is whether or not the overhead of the reduction can be decreased in general, and if not then which additional conditions allow for such a decrease.

Proof: Given a VDF tester T, we present the following tester for the t-removed model. On input ϵ and access to a graph $G = (V, E)$ and sampling device $\mathcal{D}$, we invoke T on input $\epsilon' = (d + 1)^{-t} \cdot \epsilon$ (and access to G), and emulate a sampling device to the following distribution $\mathcal{D}'$.

Distribution $\mathcal{D}'$: *A sample of $\mathcal{D}'$ is generated by selecting $v \leftarrow \mathcal{D}$, taking a t-step random walk on the graph G' obtained by adding self-loops to G, and outputting the last vertex reached in this walk.*

Hence, for every $v \in V$, it holds that

$$\sum_{w \in \Gamma_t^G(v)} (d + 1)^{-t} \cdot \mathcal{D}(w) \;\leq\; \mathcal{D}'(v) \;\leq\; \sum_{w \in \Gamma_t^G(v)} \mathcal{D}(w).$$

On the other hand, letting g represent G and g' represent any other bounded-degree graph over V, we have

$$\begin{aligned}
\delta_{t,\mathcal{D}}(g, g') &= \sum_{v \in V} \frac{\mathcal{D}_t^G(v)}{\sum_{u \in V} \mathcal{D}_t^G(u)} \cdot |\{i \in [d] : g(v, i) \neq g'(v, i)\}|/d \\
&\leq \sum_{v \in V} \sum_{w \in \Gamma_t^G(v)} \mathcal{D}(w) \cdot |\{i \in [d] : g(v, i) \neq g'(v, i)\}|/d \\
&\leq \sum_{v \in V} (d + 1)^t \cdot \mathcal{D}'(v) \cdot |\{i \in [d] : g(v, i) \neq g'(v, i)\}|/d \\
&= (d + 1)^t \cdot \delta_{\mathcal{D}'}(g, g'),
\end{aligned}$$

where the first inequality is due to $\sum_{u\in V}\mathcal{D}_t^G(u)\geq 1$ and $\mathcal{D}_t^G(v)=\sum_{w\in\Gamma_t^G(v)}\mathcal{D}(w)$, and the second inequality is due to $\sum_{w\in\Gamma_t^G(v)}(d+1)^{-t}\cdot\mathcal{D}(w)\leq\mathcal{D}'(v)$. Hence, $\delta_{t,\mathcal{D}}^{\Pi}(g)>\epsilon$ implies $\delta_{\mathcal{D}'}^{\Pi}(g)>(d+1)^{-t}\cdot\epsilon=\epsilon'$, and the theorem follows since T is emulated w.r.t vertex distribution $\mathcal{D}'$ (and with proximity parameter ϵ'). ■

Discussion. Note that the tester establishing Theorem 3.14 takes longer walks than the application that was envisioned in motivating the t-removed model. This is reminiscent of the fact that canonical derandomizers (introduced in [29] and reviewed in [13, Sec. 8.3]) are more powerful than the randomized algorithms that they fool.

4 Secondary models

In this section, we re-examine the reductions and negative results obtained in the prior sections with a focus on the question of whether or not they rely on the postulate that the tester only accesses the graph via the sampling device $\mathcal{D}$ and the corresponding representation of the graph (i.e., via either an adjacency predicate or an incidence function).

Throughout this section, for any finite set S, we denote by $\mathcal{U}(S)$ the uniform distribution over S. The testers we consider here are defined as in the previous sections, except that *in addition to oracle access to a graph $G=(V,E)$ and a (vertex) sampling device $\mathcal{D}$, the tester has access also to a device that samples $\mathcal{U}(V)$*. We stress that distances are still measured with respect to $\mathcal{D}$, as defined in the previous sections. At times, we may also consider providing the tester with $|V|$ as (an additional) explicit input. Hence, we actually consider three different types of secondary models –

The main secondary models: Here, when testing the graph $G=(V,E)$, the tester is also provided with a sampling device $\mathcal{U}(V)$.

The full-fledged secondary models: Here, when testing the graph $G=(V,E)$, the tester is provided both with a sampling device $\mathcal{U}(V)$ and with the (explicit) auxiliary input $|V|$.

The weak secondary models: Here, when testing the graph $G=(V,E)$, the tester is provided with the (explicit) auxiliary input $|V|$.

We first note that, in all three types of models (just as in the primary VDF models), we may assume, without loss of generality, that the tester never queries the graph on a vertex that did not appear in as an answer to a prior query.

More importantly, as stated explicitly in Corollaries 2.4 and 3.4, the reduction of one-sided error testing to general testing holds also in the weak secondary models. In contrast, this reduction does not hold in the other two types of the secondary models. In particular, *there exists properties that can be strongly tested in the main secondary models, although they are not strongly testable with one-sided error in the corresponding standard models.*

In light of our view of the secondary models, which is reflected in the term we chose to name them, we do not venture into an extensive study of their power. Still, we provide one negative result for the dense graph model and a couple of positive results for the bounded-degree graph model. In particular, we show that `Connectivity` is strongly testable in the weak secondary (bounded-degree graph) model, although it is not strongly testable in the corresponding VDF model (see Proposition 3.6).

4.1 The Dense Graph Model

In contrast to Theorem 2.3 and Corollary 2.4, we show that properties that are strongly tested in the main secondary (dense graph) model are not necessarily strongly testable with one-sided error in the corresponding standard (and VDF) models. We demonstrate this fact by considering the set of graphs having a clique of density ρ, which is *not* strongly testable with one-sided error in the standard (dense graph) model (cf. [17, Sec. 10.1.6]).

Theorem 4.1 (strongly testing ρ-Clique in the main secondary model) *For every $\rho \in (0,1)$, the set of graphs having a clique of density ρ is strongly testable* (with $\text{poly}(1/\epsilon)$ queries) *in the main secondary* (dense graph) *model.*

Needless to say, the asserted tester has two-sided error probability.

Proof Sketch: Recall that ρ-Clique is strongly testable in the standard dense graph model; specifically, the query complexity of this tester is polynomial in the reciprocal of the proximity parameter, but it has two-sided error probability. The basic idea is to just use this (standard model) tester while relying on the fact that if the graph is far from ρ-Clique with respect to an arbitrary vertex distribution, then it is also far from ρ-Clique under the uniform vertex distribution. Specifically, we claim that *for every $\epsilon > 0$ and all sufficiently large graphs, if $G = (V, E)$ is $((1-\rho)\cdot\epsilon^2/4)$-close to ρ-Clique under $\mathcal{U}(V)$, then it is ϵ-close to ρ-Clique under any distribution $\mathcal{D}$* (over V).

To prove this claim, let $\epsilon' = (1-\rho)\cdot\epsilon^2/4$ and consider a graph $G' = (V, E')$ that is ϵ'-close to G under the uniform distribution and has a clique of density ρ, denoted C. Fixing a distribution $\mathcal{D}$ over V, let $H \stackrel{\text{def}}{=} \{v \in V : \mathcal{D}(v) > \frac{2}{(1-\rho)\cdot\epsilon}\cdot|V|^{-1}\}$ denote the set of relatively heavy vertices, and note that $|H| < \frac{(1-\rho)\cdot\epsilon}{2}\cdot|V|$. Consider a graph $G'' = (V, E'')$ that is obtained by modifying G as follows:

- Make $C' \stackrel{\text{def}}{=} C \setminus H$ be a clique in G''.

  We claim that the cost of this modification under $\mathcal{D}$ is at most $\epsilon' \cdot \frac{2}{(1-\rho)\cdot\epsilon} = \epsilon/2$. Recalling that C' is a clique in G', we observe that the modifications of G applied in order to make C' a clique (in G') are counted in the distance between G and G', which is at most ϵ' under $\mathcal{U}(V)$. On the other hand, C' only contains vertices whose weight under $\mathcal{D}$ is at most $2/((1-\rho)\cdot\epsilon)$ times their weight under $\mathcal{U}(V)$. The claim follows.

- Pick the $|C \cap H|$ lightest vertices in $V \setminus C$ and connect each of these vertices to all other vertices. (Actually, it suffices to connect each of these light vertices to all other light vertices as well as to $C \setminus H = C'$.)

  The average weight of each of these vertices (under $\mathcal{D}$) is $\frac{\mathcal{D}(V\setminus C)}{|V\setminus C|} \leq \frac{1}{(1-\rho)\cdot|V|}$. Hence, the cost of this modification (under $\mathcal{D}$) is at most $|C\cap H| \cdot \frac{1}{(1-\rho)\cdot|V|} \leq \frac{|H|}{|V|}\cdot(1-\rho)^{-1} < \frac{(1-\rho)\cdot\epsilon}{2}\cdot(1-\rho)^{-1} = \epsilon/2$.

Hence, C is a clique in G'' (i.e., G'' has a ρ-clique), whereas G'' is ϵ-close to G under the distribution $\mathcal{D}$.

When given proximity parameter ϵ and a sampling device $\mathcal{D}$, the desired tester for the secondary model is obtained by invoking the tester of the standard model with a proximity parameter set to $(1-\rho)\cdot\epsilon^2/4$. Recall that the latter tester uses the size of the vertex-set only in order to generate uniformly distributed vertices, which we can provide it via $\mathcal{U}(V)$. ■

Digest. Note that the foregoing tester (establishing Theorem 4.1) does not use the sampling device $\mathcal{D}$. It uses only the sampling device $\mathcal{U}(V)$ (as well as oracle access to the tested graph $G = (V, E)$). The analysis is based on the observation that, for this specific property, distance from the property with respect to an arbitrary distribution can be upper-bounded in terms of the distance with respect to the uniform distribution. Needless to say, the latter feature is not generic; it is based on the fact that the graph can be made to satisfy the property without modifying the adjacency relation of any specific ("adversarially selected") set of few of its vertices (in particular, the heavy vertices).[53]

[53] The feature does not hold in case of k-Colorability: If the graph contains a k-clique of heavy vertices, then we need to omit at least one of the edges in this clique.

Recall that Corollary 2.4, which is based on Theorem 2.3, was used to obtain negative results regarding testing graph properties in the VDF model (i.e., the model of Definition 2.1). In light of Theorem 4.1, we can no longer take this road. Turning to the specific negative results in Section 2, we note that a variant of Proposition 2.10, which uses a smaller edge density bound, holds also in the (full-fledged) secondary model. Specifically, we have

Proposition 4.2 (on properties of sparse graphs): *Suppose that Π is a graph property that satisfies the following two conditions for some functions f and g such that $f(n) \in [\omega(1), g(n)]$ and $g(n) = o(f(n) \cdot n)^{1/2}$.*

1. *Π contains an n-vertex graph that contains a clique of size $f(n)$.*
2. *Π contains no n-vertex graph that contains more than $g(n)^2$ edges.*

Then, Π is not strongly testable in the (full-fledged) *secondary model.*

Recall that Π is strongly testable (with two-sided error) in the standard (dense graph) model.[54]

Proof Sketch: We consider the following two input-cases (when n is sufficiently large).

1. The n-vertex graph is a graph $G = ([n], E) \in \Pi$ containing an $f(n)$-vertex clique, and the distribution $\mathcal{D}$ is uniform on the clique vertices.
2. The n-vertex graph $G = ([n], E)$ consists a $2g(n)$-vertex clique and $n - 2g(n)$ isolated vertices, and the distribution $\mathcal{D}$ is uniform on the clique vertices.

   Note that $\delta_{\mathcal{D}}^{\Pi}(G) > 1/3$, since G contains at least $\binom{2g(n)}{2} > (2 - o(1)) \cdot g(n)^2$ edges that are each of weight $2 \cdot (2g(n))^{-2}$.

We complete the proof by letting $h(n) = \sqrt{f(n) \cdot n}/g(n) = \omega(1)$, where here we use $g(n) = o(f(n) \cdot n)^{1/2}$, and showing that *an algorithm that takes $s = o(\min(f(n)^{1/2}, h(n)))$ samples* (from both $\mathcal{D}$ and $\mathcal{U}([n])$) cannot distinguish the two foregoing cases. Specifically, we shall show that in both cases, the vertices sampled from $\mathcal{U}([n])$ are likely to look isolated (in the subgraph induced by both samples).

Denote the s samples obtained from $\mathcal{D}$ by $v_1, ..., v_s$, and the s samples obtained from $\mathcal{U}([n])$ by $u_1, ..., u_s$. We shall show that, whp, in both cases, the $2s$-vertex subgraph of G induced by these samples consists of an s-clique and s isolated vertices. First note that, in both cases, all v_i's reside in the clique, and, whp, the samples are distinct, since the collision probability of $\mathcal{D}$ (and $\mathcal{U}([n])$) is at most $1/f(n)$ and $\binom{s}{2} = o(f(n))$. Next, note that, in the first case, it holds that $\Pr_{v \leftarrow \mathcal{D}, u \leftarrow \mathcal{U}([n])}[\{u, v\} \in E]$ is at most $\frac{|E|}{f(n) \cdot n} \leq \frac{g(n)^2}{f(n) \cdot n} = h(n)^{-2} = o(1/s^2)$. Last, note that, in the second case, with probability at least $1 - (s/2g(n)) = 1 - o(1)$, all the u_i's are isolated in G. The claim follows. ■

4.2 The Bounded-Degree Graph Model

Again, in contrast to Theorem 3.3 and Corollary 3.4, we show that properties that are strongly tested in the main secondary (bounded-degree graph) model are not necessarily strongly testable with one-sided error in the corresponding standard (and VDF) models. We demonstrate this fact by considering the set of bounded-degree graphs that have relatively many edges, while noting that this set is *not* strongly testable with one-sided error in the standard (bounded-degree graph) model.

[54] On input parameters n and ϵ, if $\epsilon > (2g(n)/n)^2$, the tester estimates the edge density using $O(1/\epsilon^2)$ queries and decides accordingly (i.e., accepting if and only if the number of edges is estimated to be at most $g(n)^2$). Otherwise (i.e., when $\epsilon \leq (2g(n)/n)^2$, which implies that $n^2 \leq q(\epsilon)$ for some function $q : (0, 1] \to \mathbb{N}$), the tester explores the entire graph and decides accordingly.

Theorem 4.3 (strongly testing average degree in the main secondary model) *Let $d \in \mathbb{N}$ denote the degree bound of the model. For every $\rho \in (0,1)$, the set of graphs of maximal degree d and average degree at least $\rho \cdot d$ is strongly testable* (with $\mathrm{poly}(1/\epsilon)$ queries) *in the main secondary* (bounded-degree graph) *model.*

Needless to say, the asserted tester has two-sided error probability.

Proof Sketch: Denoting the foregoing set by Π, note that Π is strongly testable in the standard bounded-degree graph model; specifically, on proximity parameter ϵ, the tester just samples $O(1/\epsilon^2)$ vertices (uniformly at random), checks their degrees, and accepts if and only if their average exceeds $(\rho - 0.5\epsilon) \cdot d$. As in the proof of Theorem 4.1, the basic idea is to just use this tester while relying on the fact that if the graph is far from Π with respect to an arbitrary vertex distribution, then it is far from Π under the uniform vertex distribution. Specifically, we prove that *for every $\epsilon > 0$ and all sufficiently large graphs, if $G = (V,E)$ is $(\epsilon/(1-\rho))$-close to Π under $\mathcal{U}(V)$, then it is ϵ-close to Π under any distribution $\mathcal{D}$* (over V).

To prove this claim, let $\epsilon' = \epsilon/(1-\rho)$ and suppose that G is ϵ'-close to Π under $\mathcal{U}(V)$. That is, letting $d_G(v)$ denote the degree of $v \in V$ in G, we have $\sum_{v \in V} d_G(v) \geq (\rho - \epsilon') \cdot d|V|$. Fixing a distribution $\mathcal{D}$ over V, let $H \stackrel{\rm def}{=} \{v \in V : \mathcal{D}(v) > \frac{1}{1-\rho} \cdot |V|^{-1}\}$, and note that $|H| < (1-\rho) \cdot |V|$. Now, we add at most $\epsilon' \cdot d|V|/2$ edges to G with both endpoints in $V' \stackrel{\rm def}{=} V \setminus H$, till obtaining a graph with average degree at least $\rho \cdot d$. The cost of this modification is terms of $\mathcal{D}$ is at most $\frac{1}{1-\rho} \cdot \epsilon' = \epsilon$, and the question is whether this modification can be performed while maintaining the degree bound (of d).

As a sanity check towards a positive answer, note that the desired sum of degrees is $\rho \cdot d|V|$ and this quantity is upper-bounded by the degree allowance for V', which is $d|V'|$, where the upper bound hold since $|V'| > \rho \cdot |V|$. A full affirmative answer requires showing that we can increase the degrees of individual vertices in V' (upto d) and do so without using parallel edges. This can be achieved by a greedy strategy that matches non-adjacent vertices of degree at most $d-1$, while noting that this process may continue as long as there are at least $d+1$ vertices of degree at most $d-1$ (because a vertex of degree $\leq d-1$ may not neighbor all other d vertices).[55] ■

The rest of this section. In light of Theorem 4.3, we can not use negative results regarding one-sided error testing in the standard model towards deriving negative results for the secondary model. We also note that the other negative results of Section 3 do not seem to extend to the current model. In fact, in the rest of this section, we present strong testers (in the weak and main secondary models) for two natural properties that are not strongly testable in the VDF model (of Section 3). Furthermore, these testers have one-sided error.

4.2.1 Testing connectivity

In contrast to Proposition 3.6, which asserts that `Connectivity` is not strongly testable in the VDF bounded-degree graph model, we show that this property is strongly testable in either the weak or the main secondary model. Specifically, it suffices to augment the tester (of the VDF model) by either providing it with a device that samples uniformly the vertex-set of the tested graph or handing it the size of that set. Actually, these augmentations are needed only when the size of the vertex-set exceeds $O(1/\epsilon)$. We start by assuming that the vertex-set is of size $\omega(1/\epsilon)$ (and refer to the complementary case at the end of the proof of Theorem 4.5).

[55] Hence, we can actually reach a sum of degrees that is $(|V'| - (d+1)) \cdot d$ rather than $|V'| \cdot d$. The argument can be completed by using $1-\rho-o(1)$ instead of $1-\rho$ (in the definition of ϵ' and H), so that $|V'| > (\rho + o(1)) \cdot |V|$ (which implies $d|V'| - (d+1)d > \rho \cdot |V|$). Alternatively, we can afford first introducing $O(d^2)$ parallel edges, and later eliminating them as follows. To eliminate an edge between u and v, we indentify an edge $\{u', v'\}$ such that u' (resp., v') is not connected to u (resp., v), and omit the edges $\{u, v\}$ and $\{u', v'\}$, while adding the edges $\{u, u'\}$ and $\{v, v'\}$.

The tester for `Connectivity` in the standard bounded-degree graph model is based on the fact that graphs that have few connected components are close to being connected (see [14, Prop. 9.7]). Hence, if a graph is far from being connected, then it must have many small connected components. In the current setting numbers should be replaced by probability weights; that is, we shall show that if a graph is far from being connected, then the probability weight of its small connected components must be large, where the probability weight of a set of vertices is the sum of the probabilities assigned to the vertices in the set.

Proposition 4.4 (distance from connectivity versus number of connected components): *For a degree bound $d \geq 2$, let Π denote the set of connected graphs of maximum degree d. For any $k \in \mathbb{N}$, let G be a graph of maximum degree d and S denote the set of vertices of G that reside in connected components of size at most k. Then, $\delta_{\mathcal{D}}^{\Pi}(G) \leq 2 \cdot \mathcal{D}(S) + O(1/k)$, where $\mathcal{D}(S) = \Pr_{v \leftarrow \mathcal{D}}[v \in S]$.*

Proof: Suppose that G has $m > 1$ connected components and more than k vertices. Ignoring (for a moment) the fact that we should maintain the degree bound, the basic idea is to connect the m connected components by $m-1$ edges. If all connected components are large (i.e., larger than k), then we can pick the lightest vertices in each of them to serve as the "connection ports" (so that the cost of these modifications is at most $O(1/k)$). As for the small connected components, we bound the cost of modifying their incidences by their total weight. The degree bound can be maintained by observing that the problematic case is when all relatively light vertices in a connected component have degree d, but in this case we omit two of these edges and obtain a situation that allows the foregoing process. Details follow.

We say that a small connected component is unsaturated if it contains at least two vertices of degree at most $d-1$ (or a vertex of degree at most $d-2$). Note that connected components of size at most d are unsaturated. Furthermore, any other (small) connected component can be made unsaturated by omitting an edge that does not disconnect it (i.e., an edge that does not reside on a fixed spanning tree).[56] The cost of this omission is at most the weight of the connected component, and so we make all small component unsaturated at a cost of $\mathcal{D}(S)$.

Dealing with large (i.e., larger than k) connected components requires more care. We say that a vertex is (relatively) light if its probability (under $\mathcal{D}$) is at most ten times the average probability of vertices in this connected component. Note that at least 90% of the vertices in each connected component are light. We say that a connected component is unsaturated if it contains at least two light vertices of degree at most $d-1$. (Note that if $d = 2$, then a connected component can be made unsaturated by possibly omitting its lightest edge.)[57] Focusing on that case of $d \geq 3$ and fixing an arbitrary spanning tree of each connected component, we observe that if the component is saturated then it contains an non-tree edge that connects two light vertices.[58] Omitting this edge, we make the component unsaturated at a cost of $O(p/k)$, where p is the probability weight of the connected component. In total, we make all large component unsaturated at a cost of $O(1/k)$.

Now that all connected components are unsaturated, we can connect them by adding edges (while preserving the degree bound). Specifically, we connect these components by ordering them arbitrarily, and connecting each pair of consecutive components by a single edge (using relatively light vertices of degree lower than d). The cost of this modification is the sum of the probability-weight of the small

[56] Such an edge must exist in a saturated connected component of size $s > d$, since the average degree in such a component is at least $\frac{(s-1)\cdot d}{s} > d-1 \geq 2$ if $d \geq 3$, where the case of $d = 2$ is even more immediate.

[57] If this connected component is saturated, then it consists of a cycle of length $\ell > k$. In this case, the lightest edge carries at most $2/\ell$ fraction of the weight of the component.

[58] Assuming towards the contradiction that an s-vertex connected component is saturated but has no non-tree edges that connect light vertices, we reach contradiction by considering the number of non-tree edges that connect light and non-light vertices. On the one hand, looking at the light side, the number of such edges is at least $0.9s \cdot d - (s-1) > (0.9d-1) \cdot s > 0.5s \cdot d$. On the one hand, looking at the non-light side, the number of such edges is at most $0.1s \cdot d$.

components and $O(1/k)$, where the latter term accounts for weights of (light vertices in) the large components. ■

A straightforward tester and improving it. Proposition 4.4 implies that *if a graph is ϵ-far from being connected* (i.e., $\delta_{\mathcal{D}}(G) > \epsilon$), *then, with probability $\Omega(\epsilon)$, a vertex selected from $\mathcal{D}$ resides in a connected component of size $O(1/\epsilon)$.* Hence, selecting at random $O(1/\epsilon)$ vertices, and conducting a "truncated BFS" from each of them (so that the BFS is suspended once more than $O(1/\epsilon)$ vertices are encountered) yields a tester for `Connectivity`. The time (and query) complexity of this tester is $O(1/\epsilon^2)$, but using "Levin's economical work investment strategy" (see [14, Sec. 8.2.4]), we can do better.

Theorem 4.5 (testing connectivity (in the secondary bounded-degree graph model)): `Connectivity` *has a* (one-sided error) *tester of time* (and query) *complexity $\widetilde{O}(1/\epsilon)$ in the* (either weak or main) *secondary bounded-degree graph model.*

Proof: A closer look at the proof of Proposition 4.4, when applied to the graph $G - (V, E)$, reveals that the contribution of the connected component C to the distance $\delta_{\mathcal{D}}(G)$ is $O(\mathcal{D}(C)/|C|)$. (Indeed, this refers to the analysis of the large components, but the argument holds for $k = |C|$.) Towards a more refined analysis, for every $i = 1, ..., \ell \stackrel{\rm def}{=} \log(1/\epsilon) + O(1)$, we denote by $S_i \subseteq V$ the set of vertices that reside in connected components of size at least 2^{i-1} and at most $2^i - 1$. Letting $R = V \setminus \bigcup_{i\in[\ell]} S_i$, we have

$$\delta_{\mathcal{D}}(G) \ \leq \ \sum_{i\in[\ell]} O(\mathcal{D}(S_i)/2^{i-1}) \ + \ O(\mathcal{D}(R)/2^{\ell}). \tag{22}$$

Hence, if $\delta_{\mathcal{D}}(G) > \epsilon$, then there exists $i \in [\ell]$ such that $O(\mathcal{D}(S_i)/2^{i-1}) > \epsilon/2\ell$, and it follows that $\Pr_{s\leftarrow\mathcal{D}}[s \in S_i] = \Omega(2^i\epsilon/\ell)$. This leads to the following tester, where we assume that $|V| > 2^\ell$.

Algorithm 4.5.1 (tester for the case of $|V| > 2^\ell$): *On input $\epsilon > 0$, given oracle access to $G = (V, E)$ and $\mathcal{D}$ such that $|V| > 2^\ell$, the tester perform the following steps, for every $i = 1, ..., \ell$:*

1. *Sample $O(2^{-i}\ell/\epsilon)$ vertices from the distribution $\mathcal{D}$.*
2. *For each of these vertices, denoted v, perform a* (BFS or DFS) *search starting at v, suspending the execution if 2^i vertices were encountered in this search* (or if the search scanned the entire connected component).
3. *If any of these searches detected a connected component of size at most 2^i, then the tester rejects.* (Here we rely on $2^\ell < |V|$.)

If none of these searches caused rejection (i.e., none detected a connected component that is smaller than $|V|$)*, then the tester accepts.*

Note that any linear-time search can be used in Step 2, and in such a case the overall time complexity of the tester is $\sum_{i\in[\ell]} O(2^{-i}\ell/\epsilon) \cdot O(2^i) = O(\ell^2/\epsilon)$.

By its construction (and the assumption $2^\ell < |V|$), the foregoing tester always accepts a connected graph. On the other hand, any graph that is ϵ-far from being connected is rejected with high probability, because there exists an $i \in [\ell]$ such that $\mathcal{D}(S_i) = \Omega(2^i\epsilon/\ell)$, which implies that a vertex residing in a connected component of size at most 2^i is selected, w.h.p., in Step 1 (of iteration i), fully explore in Step 2, and causing rejection in Step 3.

We stress that the foregoing analysis presumes that $|V| > 2^\ell = O(1/\epsilon)$. To handle the case of $|V| = O(1/\epsilon)$, we use one of the two possible augmentations provided by the secondary models. First, assuming that the tester gets $|V|$ as an auxiliary input, we let it invoke the foregoing tester (i.e., Algorithm 4.5.1) if $|V| > 2^\ell$, and start a search from an arbitrary vertex (obtained by sampling $\mathcal{D}$) otherwise (i.e., if $|V| \leq 2^\ell = O(1/\epsilon)$). In the latter case, the tester accepts if and only if the search encountered $|V|$ vertices. Hence, we obtained a tester in the weak secondary model.

Next, we turn to the main secondary model (in which the tester is also provided with a device sampling $\mathcal{U}(V)$, but not with the value of $|V|$). In this case, the tester invokes Algorithm 4.5.1 (whose analysis presumes $V > 2^\ell$) in parallel to invoking the following algorithm:

Algorithm 4.5.2 (tester for the case of $|V| \leq 2^\ell = O(1/\epsilon)$): *On input $\epsilon > 0$, given oracle access to $G = (V, E)$ and $\mathcal{U}(V)$, the tester proceeds as follows.*

1. *Start a search at an arbitrary vertex* (obtained by sampling either $\mathcal{D}$ or $\mathcal{U}(V)$)*, suspending the execution if more than 2^ℓ vertices were encountered in this search* (or if the search scanned the entire connected component)*. If the search encountered more than 2^ℓ vertices, then halt and abstain from judgment* (i.e., let Algorithm 4.5.1 decide).

   (We proceed to Step 2 only when encountering a connected component with at most 2^ℓ vertices, which may or may not be the entire graph.)

2. *Select uniformly at random $m = O(1/\epsilon)$ vertices* (by sampling $\mathcal{U}(V)$)*. If all these vertices reside in the subgraph scanned in Step 1, then accept, else reject.*

Hence, the combined tester accepts when either Algorithm 4.5.2 accepts (in Step 2) or when Algorithm 4.5.2 abstains (in Step 1) and Algorithm 4.5.1 accepts. We break the analysis to four cases.

Case 1: G is connected and $|V| \leq 2^\ell$. In this case Algorithm 4.5.2 always reaches Step 2 and accepts (in Step 2). (Indeed, it has explored the entire graph in Step 1, and therefore encounter no new vertex in Step 2.)

Case 2: G is connected and $|V| > 2^\ell$. In this case Algorithm 4.5.2 always halts and abstains (in Step 1), and Algorithm 4.5.1 always accepts.

Case 3: G is ϵ-far from being connected and $|V| \leq 2^\ell$. In this case Algorithm 4.5.2 rejects (w.h.p.), since it always reaches Step 2 whereas G has several connected components (and Step 2 detects this fact w.h.p.). Specifically, Step 2 select a vertex that was not visited in Step 1 with probability at least $1 - (1 - |V|^{-1})^m \geq 1 - (1 - 2^{-\ell})^m > 2/3$, provided $m = O(2^\ell)$ is sufficiently large.

(Indeed, here we only used the hypothesis that G is not connected and $|V| \leq 2^\ell$.)

Case 4: G is ϵ-far from being connected and $|V| > 2^\ell$. In this case Algorithm 4.5.1 rejects (w.h.p.), whereas Algorithm 4.5.2 either abstains or rejects (w.h.p., as in Case 3).

The theorem follows. ■

On testing k-connectivity: The strong one-sided error testers known for k-edge-connectivity and k-vertex-connectivity in the standard bounded-degree graph model (see [19] and [34], resp), beg the question of whether Theorem 4.5 can be extended to these properties. Given the complexity of the original analyses, we refrain from addressing this question here.

4.2.2 Testing whether a graph is connected and Eulerian

In the standard setting, one can reduce testing whether a graph is connected and Eulerian to testing that it has each of these properties separately. This relied on the fact that if a graph is ϵ-close to each of the properties, then it is $O(\epsilon)$-close to their intersection (cf., [14, Exer. 9.5]). Unfortunately, this fact does not hold in the current setting: A graph can be connected and very close to being Eulerian, but far from any connected Eulerian graph.[59] Fortunately, the specific testers we have used "interact" better that generic testers for the two properties. Specifically, the tester for Eulerian graphs outlined at the end of Section 3.3 is based on the observation that if odd degree vertices and their neighbors are assigned total probability ϵ, then the graph is $O(\epsilon)$-close to be Eulerian. Likewise, the tester asserted for `Connectivity` in Theorem 4.5 is based on the fact that if for every $i \in [\log(1/\epsilon) + O(1)]$ the total probability weight of vertices that reside in connected components of size $\approx 2^i$ is $O(2^i\epsilon/\log(1/\epsilon))$, then the graph is $O(\epsilon)$-close to a connected graph. We adapt the proof of the latter fact so that the transformation preserves the Eulerian property.

Theorem 4.6 (testing the set of connected Eulerian graphs (in the secondary bounded-degree graph model)): *The set of connected and Eulerian graphs has a* (one-sided error) *tester of time* (and query) *complexity $\widetilde{O}(1/\epsilon)$ in the* (either weak or main) *secondary bounded-degree graph model.*

We note that this property is not strongly testable in the VDF secondary bounded-degree graph model. This fact can be proved by mimicking the argument used in the proof of Proposition 3.6.

Proof Sketch: Again, we focus on the case of $|V| = \omega(1/\epsilon)$, handling the complementary case as in the proof of Theorem 4.5 (where here we also check that the connected component is Eulerian, in case this connected component was fully explored).

The tester (for the case of $|V| = \omega(1/\epsilon)$) consists of running both the aforementioned testers (on proximity parameter $\epsilon' = \epsilon/O(1)$) and accepting if both accept. Clearly, this tester always accepts connected graphs that are Eulerian, and so we focus on showing that graphs that are accepted with high probability are close to having this property.

Indeed, suppose that G is accepted with high probability. Then, its odd degree vertices and their neighbors are assigned total probability at most ϵ', since the degrees of these vertices are checked by the Eulerian tester (which selects $O(1/\epsilon')$ vertices according to $\mathcal{D}$ and checks their degrees and the degrees of their neighbors). In that case, by the analysis outlined at the end of Section 3.3, G is $O(\epsilon')$-close to an Eulerian graph G', which satisfies the same degree bound as G. Looking at the connected components of G', we pick in each component an edge with the lightest weight (equiv., weight of its endpoints) and omit it from G', observing that this omission does not disconnect the component (because this edge resides on an Eulerian cycle). The cost of each such modification is charged to the connected component (of G') if it was also a connected component in G, and to a vertex in this component that had odd degree in G otherwise. As in the proof of Proposition 4.4, the resulting graph G'' is $O(\epsilon')$-close to G'. Note that, in the resulting graph G'', the endpoints of these light edges are of odd degree (and they are the only vertices of odd degree in G''). Finally, we make G'' connected by adding edges between the light vertices as in the proof of Proposition 4.4, while observing that these vertices now

[59] For example, let d be odd, and consider the n-vertex graph G that consists of $n' \stackrel{\text{def}}{=} n/d$ cliques, each of size d, that are connected by a cycle of n' *vertex-disjoint edges* such that the endpoints of these edges have weight $\eta = o(1/n)$ each whereas each other vertex in the graph has weight $(1 - 2n'\eta)/n$. That is, we designate two distinct vertices of weight η in each d-clique, and connect the first designated vertex of the i^{th} clique to the second designated vertex of the $i - 1^{\text{st}}$ clique. Then, G is connected and $(2\eta n/d^2)$-close (i.e., $o(1/d^2)$-close) to being Eulerian, since we can make it Eulerian by omitting the n' light edges. On the other hand, G is $\Omega(1/d^2)$-far from the intersection of both properties, because it is necessary to change the incidence relation of at least one heavy vertex in each clique in order to make G Eulerian and connected. (This is the case because vertices in the modified graph must have degree at most $d - 1$, since d is odd, whereas keeping all the edges of any d-clique leaves no "vacancy" for connecting this clique to any other vertex.)

become of even degree. Hence, G'' is $O(\epsilon')$-close to being connected and Eulerian, and the same holds for G' and G. ■

5 Subsequent work and future directions

Section 5.1 has been revised in light of a subsequent work of Gishboliner and Shapira [12].

5.1 More about strong testability

Propositions 2.9 and 3.5, which establish the two parts of Theorem 1.2, are proved by using properties that are trivial to test in the standard models. The impossibility of strongly testing these properties in the VDF models relies on the fact that the tester does not know the size of the graph. This raises the question of what happens if we either provide the VDF tester with the size of the graph (which is not compatible with the settings that we envision) or restrict the standard-model testers in a similar manner (as done in [5]). Recall that the first option was adopted in Proposition 2.11, which provides an alternative proof of the "dense graph" model part of Theorem 1.2. Here we focus on the second option.

We recall that Alon and Shapira [5] defined "oblivious testers" as strong testers (operating in the standard dense graph model) that obtain a uniformly distributed sample of vertices, query the corresponding vertex pairs, and decide according to the (unlabeled) subgraph that they see. The point is that *both the size of the sample and the final decision are independent of the size of the graph.* We call such testers size-oblivious and observe that this definition is a special case of Definition 2.1 in which the sample complexity is a predetermined function of ϵ (i.e., strong tester) and the sampling device samples the vertex-set uniformly (i.e., $\mathcal{D} = \mathcal{U}(V)$). We stress that this notion of size-oblivious tester is also applicable to the bounded-degree graph model, by restricting Definition 3.1 in an analogous manner.

Alon and Shapira provided a characterization of the class of graph properties that have size-oblivious one-sided error testers (in the standard dense graph model) [5, Thm. 2]. A natural question that arises is whether the same class is also strongly testable in the VDF dense graph model.

Open Problem 5.1 (size-oblivious testing in the standard model vs strong testing in the VDF model): *Is it the case that any graph property that has a size-oblivious one-sided error tester in the standard dense graph model* (resp., bounded-degree graph model) *is also strongly testable in the VDF version of the dense graph model* (resp., bounded-degree graph model)*?*

Note that the converse is obvious; that is, any strong tester in the VDF model can be converted into one with one-sided error (see Theorem 1.1), whereas the latter tester is size-oblivious (and has one-sided error in the standard model). On the other hand, a positive resolution of Problem 5.1 would have established the main results of Theorems 1.3 and 1.4, with the exception that the complexity bounds may be worse, since the classes of properties listed in these theorems have *size-oblivious* one-sided error testers in the corresponding standard models.

5.1.1 In the dense graph model

Following the first posting of this work [16], Gishboliner and Shapira [12] provided a negative answer to the case of the dense graph model by characterizing the class of properties that are strongly testable in the VDF version (and using the aforementioned characterization of [5, Thm. 2]). Although their main result refers to the variant of Definition 2.1 that appeared in our first posting of this work [16][60], its

[60] See Footnote 20.

ideas can be used to provide a negative answer also to the current variant (Asaf Shapira, priv. comm.). In light of the foregoing, one may seek an augmentation of the hypothesis (i.e., strong testability in the standard model) that makes the conclusion hold (i.e., yields strong testability in the VDF model). Furthermore, it will be nice to develop transformations of large classes of strong testers with one-sided error for the standard model to testers for the VDF model.

We note that the testers used to establish Theorems 2.7, 2.8, and 2.12 are fundamentally different: The testers establishing Theorem 2.7 invoke $\exp(\text{poly}(1/\epsilon))$ copies of the standard model tester, the testers establishing the main part of Theorem 2.8 use a generalization of the standard model testers (from subgraph-freeness to "colored subgraph freeness"), whereas the testers establishing Theorem 2.12 (as well as Theorem 2.13) check that a random $\text{poly}(1/\epsilon)$-sized induced subgraph has the property. We mention that the work of Gishboliner and Shapira [12] implies different testers for all these cases: These alternative testers are derived uniformly from the mere fact that all these properties are hereditary and "extendable" (a generalization of the notion of paddable used in Theorem 2.12).

5.1.2 In the bounded-degree graph model

Turning to the bounded-degree graph model, we note that a positive resolution of Problem 5.1 for that case would also resolve the following problem, which appears easier to resolve.[61]

Open Problem 5.2 (additional strong testers in the VDF bounded-degree graph model): *Obtain strong testers in the VDF bounded-degree graph model for the following classes of sets.*

(general tree-minor-freedom): *For every fixed forest F, the set of F-minor free graphs.*

(high connectivity of connected components): *For any fixed $k > 1$, the set of graphs such that each of their connected components is k-edge connected* (resp., k-vertex connected).

Alternatively, prove that some of these properties do not have strong testers in the VDF bounded-degree graph model.

Recall that the cases of F being either a simple path or a star were established in Theorem 1.4. On the other hand, the negative result of Proposition 3.6 does not seem to extend to k-connectivity of connected graphs, since the argument is based on the need and infeasibility of deciding whether or not the explored subgraph is the entire graph. Also recall that the sets listed in Problem 5.2 are known to have size-oblivious testers with one-sided error in the standard bounded-degree graph model (see [19, 34] and [9], resp).

5.2 Beyond strong testability

In this work, we have focused on (VDF) testers having query complexity that only depends on the proximity parameter. Nevertheless, it makes sense to also study (VDF) testers that have complexity that depends on the vertex-distribution.

In the standard graph testing models, the complexity of the tester is measured in terms of the proximity parameter $\epsilon > 0$ and the size of the graph (e.g., the number of vertices). In the current (VDF) models, it makes little sense to refer to the size of the graph as a yardstick of complexity, since the vertex-distribution $\mathcal{D}$ may be concentrated on a relatively small part of the graph. Instead, here it is natural to relate the complexity to label-invariant parameters of the distribution $\mathcal{D}$, were a parameter is called label-invariant if it remains intact when relabeling the vertices (cf. [14, Sec. 11.1.3]); that is, if, for every distribution $\mathcal{D}$ over $\{0,1\}^*$ and every bijection π of $\{0,1\}^*$ to itself, it holds that the value

[61] Needless to say, the more refined question raised in Section 5.1.1 is relevant here too, but it feels premature to raise it now.

of the parameter on $\mathcal{D}$ equals its value on $\pi \circ \mathcal{D}$ (i.e., the distribution obtained by sampling $v \leftarrow \mathcal{D}$ and outputting $\pi(v)$). In other words, a label-invariant parameter of distributions is a function of the histogram of the distribution, where the histogram of $\mathcal{D}$ is the set of pairs $(p,i) \in (0,1] \times \mathbb{N}$ such that $\mathcal{D}$ assigns probability $p > 0$ to $i > 0$ different elements. Examples of label-invariant parameters (of distributions) include the support-size of $\mathcal{D}$ (i.e., $|\{v : \mathcal{D}(v) > 0\}|$), the "effective support size" (i.e., being "close" to a distribution with the specified support-size (cf., [8])[62]), the collision probability of $\mathcal{D}$ (i.e., $\sum_{v:\mathcal{D}(v)>0} \mathcal{D}(v)^2$), the entropy of $\mathcal{D}$ (i.e., $\mathrm{Exp}_{v \leftarrow \mathcal{D}}[\log_2(1/\mathcal{D}(v))]$), and the min-entropy of $\mathcal{D}$ (i.e., $\min_{v:\mathcal{D}(v)>0}\{\log_2(1/\mathcal{D}(v)\})$).

While the study of the standard dense graph model has focused on strong testers, the study of the standard bounded-degree graph model also yielded appealing testers that are not strong. An archetypical example is the tester of Bipartiteness, which has complexity $\mathrm{poly}(1/\epsilon) \cdot \sqrt{n}$, where n denotes the number of vertices. An interesting open problem is to present a Bipartite tester for the VDF bounded-degree graph model. One may hope for complexity $\mathrm{poly}(1/\epsilon) \cdot \sqrt{s}$, where s is the size of the support of $\mathcal{D}$. Actually, some notion of "effective support size" may be more adequate here.

5.3 Providing the VDF tester with the approximate size of the graph

Recall that we have expressed the opinion that providing the VDF tester with the size of the tested graph is not compatible with our motivation (as detailed in Section 1.4). Nevertheless, providing the VDF tester with an approximation of this size is not incompatible with this motivation, especially if this approximation is rather rough. In particular, suppose that for some function f (e.g., $f(n) = O(n)$ or $f(n) = \mathrm{poly}(n)$), we provide the tester with an auxilay input N such that the number of vertices in the tested graph is between N and $f(N)$. We encourage a study of this relaxation of the VDF model, while noting that the proof of Theorem 4.5 extends to this case, except that the complexity is $\widetilde{O}(f(1/\epsilon))$.

Acknowledgements

I am grateful to Dana Ron for several helpful discussion concerning both the conceptual and technical aspects of this work. I also wish to thank Asaf Shapira for claricatiions regarding [12].

[62] Specifically, we may say that $\mathcal{D}$ has *f-effective support-size n* if $\mathcal{D}$ is $f(n)$-close to a distribution of support size n. Natural choices of the function $f : \mathbb{N} \to (0,1)$ include negligible functions (i.e., $f < 1/p$ for any positive polynomial p) and functions f satisfying only $f(n) = o(1/n)$.

References

[1] N. Alon. Testing subgraphs of large graphs. *Random Structures and Algorithms*, Vol. 21, pages 359–370, 2002.

[2] N. Alon, J. Balogh, P. Keevash, and B. Sudakov. The Number of Edge Colorings with No Monochromatic Cliques. *Journal of the London Mathematical Society*, Vol. 70 (2), pages 273–288, 2006.

[3] N. Alon, E. Fischer, M. Krivelevich and M. Szegedy. Efficient Testing of Large Graphs. *Combinatorica*, Vol. 20, pages 451–476, 2000.

[4] N. Alon and J. Fox. Easily testable graph properties. *Combinatorics, Probability and Computing*, Vol. 24 (4), pages 646–657, 2015.

[5] N. Alon and A. Shapira. A Characterization of the (natural) Graph Properties Testable with One-Sided Error. *SIAM Journal on Computing*, Vol. 37 (6), pages 1703–1727, 2008.

[6] M. Balcan, E. Blais, A. Blum, and L. Yang. Active property testing. In *53rd FOCS*, pages 21–30, 2012.

[7] I. Benjamini, O. Schramm, and A. Shapira. Every Minor-Closed Property of Sparse Graphs is Testable. In *40th STOC*, pages 393–402, 2008.

[8] 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.

[9] 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.

[10] I. Dinur, O. Goldreich, and T. Gur. Every set in P is strongly testable under a suitable encoding. *ECCC*, TR18-050, 2018.

[11] J. Fox. A new proof of the graph removal lemma. *Ann. of Math*, Vol. 174 (1), pages 561–579, 2011.

[12] L. Gishboliner and A. Shapira. Which Properties are Testable in the Vertex-Distribution-Free Model. Preprint, to appear in *51st STOC*, 2019.

[13] O. Goldreich. *Computational Complexity: A Conceptual Perspective.* Cambridge University Press, 2008.

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

[15] O. Goldreich. Flexible models for testing graph properties. *ECCC*, TR18-104, 2018.

[16] O. Goldreich. Testing Graphs in Vertex-Distribution-Free Models. *ECCC*, TR18-171, 2018.

[17] 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.

[18] O. Goldreich, M. Krivelevich, I. Newman, and E. Rozenberg. Hierarchy Theorems for Property Testing. *Computational Complexity*, Vol. 21 (1), pages 129–192, 2012.

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

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

[21] O. Goldreich and D. Ron. On Proximity Oblivious Testing. *SIAM J. on Comp.*, Vol. 40, No. 2, pages 534–566, 2011. Extended abstract in *41st STOC*, 2009.

[22] O. Goldreich and D. Ron. On Sample-Based Testers. *TOCT*, Vol. 8 (2), 2016.

[23] O. Goldreich and L. Trevisan. Three theorems regarding testing graph properties. *Random Structures and Algorithms*, Vol. 23 (1), pages 23–57, August 2003.

[24] O. Goldreich and L. Trevisan. Errata to [23]. Manuscript, August 2005. Available from `http://www.wisdom.weizmann.ac.il/~oded/p_ttt.html`

[25] 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.

[26] J. Komlos and M. Simonovits. Szemeredi's regularity lemma and its applications in graph theory. *Paul Erdos is 80*, Proceedings of Colloquia of the Bolyai Mathematical Society 2, pages 295–352, 1993. See also DIMACS Tech. Rep. 96-10, 1996.

[27] W. Mader. Homomorphiesätze für graphen. *Mathematische Annalen*, Vol. 178, pages 154–168, 1968.

[28] Y. Nakar and D. Ron. On the Testability of Graph Partition Properties. In *Proc. of RANDOM'18*, pages 53:1–53:13, 2018.

[29] N. Nisan and A. Wigderson. Hardness vs Randomness. *JCSS*, Vol. 49, No. 2, pages 149–167, 1994. Preliminary version in *29th FOCS*, 1988.

[30] M. Parnas and D. Ron. Testing the diameter of graphs. *Random Structures and Algorithms*, Vol. 20 (2), pages 165–183, 2002.

[31] O. Pikhurko. An Analytic Approach to Stability. *Discrete Math.*, Vol. 310, pages 2951–2964, 2010.

[32] R. Rubinfeld and M. Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2), pages 252–271, 1996.

[33] E. Szemeŕedi. Regular partitions of graphs. In *Proceedings, Colloque Inter. CNRS*, pages 399–401, 1978.

[34] Y. Yoshida and H. Ito. Property Testing on k-Vertex-Connectivity of Graphs. *Algorithmica*, Vol. 62 (3), pages 701–712, 2012.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il