Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 98 (2018)

Hierarchy Theorems for Testing Properties in Size-Oblivious Query Complexity

Oded Goldreich*

May 2, 2019

Abstract

Focusing on property testing tasks that have query complexity that is independent of the size of the tested object (i.e., depends on the proximity parameter only), we prove the existence of a rich hierarchy of the corresponding complexity classes. That is, for essentially any function $q : (0, 1] \to \mathbb{N}$, we prove the existence of properties for which ϵ-testing has query complexity $\Theta(q(\Theta(\epsilon)))$. Such results are proved in three standard domains that are often considered in property testing: generic functions, adjacency predicates describing (dense) graphs, and incidence functions describing bounded-degree graphs.

These results complement hierarchy theorems of Goldreich, Krivelevich, Newman, and Rozenberg (*Computational Complexity*, 2012), which refer to the dependence of the query complexity on the size of the tested object, and focus on the case that the proximity parameter is set to some small positive constant. We actually combine both flavors and get tight results on the query complexity of testing when allowing the query complexity to depend on both the size of the object and the proximity parameter.

Contents

*Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, ISRAEL. Email: oded.goldreich@weizmann.ac.il. This research was partially supported by the Israel Science Foundation (grants No. 671/13 and 1146/18).

ISSN 1433-8092

1 Introduction

In the last couple of decades, the area of property testing has attracted much attention (see, e.g., a recent textbook [9]). 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 length of the object).

Following most work in the area, we focus on the query complexity of property testing, measured as a function of the size of the object as well as the desired proximity (parameter), denoted ϵ. Interestingly, many natural properties can be tested in complexity that only depends on the proximity parameter; examples include linearity testing [6] and testing various graph properties in two natural models (e.g., [12, 2] and [14, 5], respectively). Focusing on such properties and on the known testers, we note that the specific dependency of the query complexity on the proximity parameter varies from linear in $1/\epsilon$ (e.g., [6]) to polynomial in $1/\epsilon$ (e.g., [12, 14]) and to tower-like (and even larger) functions in $1/\epsilon$ (e.g., [1, 2]). As for lower bounds, till recently, the only known super-polynomial lower bounds (of [1, 3]) referred to testing problems for which the corresponding known upper bounds are tower-like functions, leaving a huge gap between the *known* lower and upper bounds.

The foregoing gap has been recently addressed by Gishboliner and Shapira [8], who considered the special case of *one-sided error* testers for graph properties in the dense graph model.[1] In that context, they presented a general hierarchy theorem asserting, essentially for any $q : (0, 1] \to \mathbb{N}$, the existence of graph property of query complexity $\Theta(q(\Theta(\epsilon)))$. In this paper we present analogous results for the general definition of property testing, which allows *two-sided error* probability.

1.1 Our results

The hierarchy theorem is easiest to state and prove in the generic case (treated in Section 2). Loosely speaking, it asserts that *for essentially every function $q : (0, 1] \to \mathbb{N}$, there exists a property of Boolean functions that is testable using $O(q(\Omega(\epsilon)))$ queries but is not testable using $o(q(O(\epsilon)))$ queries*, where ϵ denotes the proximity parameter.[2] In other words, *the query complexity of testing this property is* $\Theta(q(\Theta(\epsilon)))$. In particular, this implies the existence of property testing problems of complexities such as $\exp(\Theta(1/\epsilon^c))$ for any constant $c > 0$.

Similar hierarchy theorems are proved also for two standard models of testing graph properties: the adjacency representation model (a.k.a. the dense graph model of [12]) and the incidence representation model (a.k.a. the bounded-degree graph model of [14]). These results are rigorously stated and proved in Sections 3 and 4.

The foregoing results complement hierarchy theorems of Goldreich, Krivelevich, Newman, and Rozenberg [13], which refer to the dependence of the query complexity on the size of the tested object, and focus on the case that the proximity parameter is set to some small positive constant. We can actually combine both flavors and get tight results on the query complexity of testing when allowing the query complexity to depend quite arbitrarily on both the size of the object and the

[1] We mention that a weaker hierarchy theorem (for one-sided testers in the dense graph model) was proved more than a decade ago by Alon and Shapira [4]. Specifically, [4, Thm. 4] asserts that for essentially every function q there exists a function Q and a graph property that is testable in $Q(\epsilon)$ queries but is *not* testable in $q(\epsilon)$ queries. We note that while Q depends only on q, the dependence proved in [4, Thm. 4] is quite weak (i.e., Q is lower bounded by a non-constant number of compositions of q); hence, the hierarchy obtained by setting $q_i = Q_{i-1}$ for $i \in \mathbb{N}$, is very sparse.

[2] The lower bounds holds provided the object is large enough (i.e., larger than $O(q(\epsilon)/\epsilon)$).

proximity parameter. These general results are stated in Section 5.

1.2 Our techniques

Following is a very rough sketch of our proof strategy. Our starting point is the hierarchy theorems of Goldreich, Krivelevich, Newman, and Rozenberg [13], which we re-interpret in concrete (rather than asymptotic) terms. Essentially, for some universal constant c and every fixed natural numbers $q' \leq n'$, these theorems state the existence of properties $\Pi'_{n',q'}$ of objects of size n' for which the query complexity of testing with proximity $1/c$ is between q'/c and q'. Furthermore, the upper bound extends to $\text{poly}(1/\epsilon) \cdot q'$, when testing with proximity parameter ϵ.

Now, suppose that we want to present a property such that testing it with proximity ϵ has complexity between q' and $\text{poly}(1/\epsilon) \cdot q'$. Then, we define objects of size n that consist of a "base" object from $\Pi'_{c\epsilon \cdot n,q'}$ padded to size $n \geq q'/c\epsilon$ (so that $q' \leq c\epsilon n$). Next, we reduce testing $\Pi'_{c\epsilon n,q'}$ with proximity $1/c$ to testing the new property with proximity ϵ (i.e., ϵ-testing the new property), which establishes the desired lower bound. Lastly, we construct an ϵ-tester of the desired complexity by letting it check that the amount of padding in the object is $(1 - c\epsilon) \cdot n$, and testing the base object (using the $1/c$-tester for $\Pi'_{c\epsilon n,q'}$). Indeed, this requires using a padding that is easily recognized (e.g., elements of the base object should be easy to distinguish from elements of the padding). In such a case, the query complexity of the resulting ϵ-tester will be $q' + O(1/\epsilon)$.

The foregoing construction is tailored for a fixed value of the proximity parameter, whereas we seek properties that exhibit the designated complexity for any value of the proximity parameter. This is achieved by creating properties that are the union of properties defined as above for a geometric sequence of values of the proximity parameter. Specifically, when seeking to establish query complexity $q : (0, 1] \to \mathbb{N}$, we use the base properties $\Pi'_{c^{-i} \cdot n,q(c^{-i})}$ for $i = 1, ..., \log_c n$. That is, we take the union of these properties after padding each of them to size n.

Establishing the lower and upper bounds in this case requires more care. Specifically, for the lower bound, we show that the properties introduced for handling the other values of the proximity parameter do not interfere with the argument that refers to the value of the proximity parameter that is of interest to us. For the upper bound, we first determine the size of the base object that seems to underlie the tested object (rejecting if none fits). Next, we emulate testing the base object, while capitalizing on the fact that the testers of [13] work for any value of the proximity parameter rather than only for a fixed constant value.

The proofs for the different testing models differ both in the result of of Goldreich *et. al.* [13] that is used as a starting point and in the notion of padding that is used in each case. Things are easy in the case of properties of generic Boolean functions, where we just use padding by a special symbol (and later encode the three-valued functions by Boolean ones). In the case of graph properties, we pad the graphs by a suitable number of easily identified vertices (i.e., vertices of higher degree).[3] Dealing with the padding when establishing the upper bound raises difficulties in the case of the graph testing models. These difficulties arise from the fact that (unlike in the case of generic functions) the padding does not appear in fixed locations in the (labeled) graph. In particular, the amount of padding can only be approximated and so we may need to test a base graph with a number of vertices that does not fit any $c^{-i} \cdot n$ (but is rather close to one of these values).

We present two ways of dealing with the latter problem. The simpler way, which yields weaker results, is to use very good approximations of the number of "padding" vertices. These approximation are so good that one can neglect the fact that they are not accurate, but obtaining such approximations has a cost in terms of squaring the query complexity of the tester used in the upper

[3] Indeed, it is tempting to pad the graphs with isolated vertices, but this raises some technical difficulaties, which can be solved but are avoided by the less natural choice we use.

bound. The alternative way relies on the fact that the testers provided by [13] fit a revised model of testing graph properties. In this model, which is of independent interest, the tester is given oracle access to independently sampled vertices of the graph, but is not given the size of the graph (i.e., its vertex set) as auxiliary input.[4]

1.3 Organization

In Section 2 we recall the definition of testing properties of generic functions and prove a hierarchy theorem for that setting. This section is most detailed one, because the other results are obtained by mimicking the basic strategy that is presented in Section 2. Later, in Section 3 we prove a hierarchy theorem for testing graph properties in the bounded-degree graph model, and in Section 4 we obtain the same for testing in the dense graph model.

In Section 5, we generalize the results of the aforementioned sections so to obtain tight results on the query complexity of testing when allowing the query complexity to depend quite arbitrarily on both the size of the object and the proximity parameter.

2 Properties of Generic Functions

In the generic function model, the tester is given oracle access to a function over $[n]$, and the distance between such functions is defined as the fraction of (the number of) arguments on which these functions differ. In addition to the input oracle, the tester is explicitly given two parameters: a size parameter, denoted n, and a proximity parameter, denoted ϵ.

Definition 2.1 (property testing, the case of Boolean functions): *Let $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$, where Π_n contains Boolean functions defined over the domain $[n] \stackrel{\rm def}{=} \{1,...,n\}$. A* tester for a property Π *is a probabilistic oracle machine T that satisfies the following two conditions:*

1. *The tester accepts each $f \in \Pi$ with probability at least $2/3$; that is, for every $n \in \mathbb{N}$ and* $f \in \Pi_n$ (and every $\epsilon > 0$), *it holds that* $\Pr[T^f(n,\epsilon)=1] \geq 2/3$.
2. *Given $\epsilon > 0$ and oracle access to any f that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, for every $\epsilon > 0$ and $n \in \mathbb{N}$, if $f : [n] \to \{0,1\}$ is ϵ-far from Π_n, then $\Pr[T^f(n,\epsilon) = 0] \geq 2/3$, where f is* ϵ-far from Π_n *if, for every $g \in \Pi_n$, it holds that $|\{i \in [n] : f(i) \neq g(i)\}| > \epsilon \cdot n$.*

We say that the tester has one-sided error *if it accepts each $f \in \Pi$ with probability 1; that is, for every $f \in \Pi$ and every $\epsilon > 0$, it holds that* $\Pr[T^f(n,\epsilon)=1] = 1$.

When $\epsilon > 0$ is fixed, we refer to the residual oracle machine $T(\cdot,\epsilon)$ by the term ϵ-tester. We also use the corresponding term ϵ-testing Π. Likewise, we may fix n (and possiblly other parameters), and consider the task of testing Π_n (resp., Π_n further restricted by the other parameters). We stress that even when we fix ϵ and n, we view them as generic.

Definition 2.1 does not specify the query complexity of the tester, and indeed an oracle machine that queries the entire domain of the function qualifies as a tester (and may indeed have zero error probability). Needless to say, we are interested in testers that have significantly lower query complexity. Recall that [12] asserts that in some cases such testers do not exist; that is, there exist properties that require linear query complexity. Building on this result, Goldreich *et. al.* [13] showed the following hierarchy result that refers to $\Theta(1)$-testing:

[4] This model is an extreme case of a flexible framework that has appeared in our subsequent work [10].

Theorem 2.2 ([13, Thm. 2], revised): *There exists a universal constant $c > 2$ such that, for every $q', n' \in \mathbb{N}$ that satisfy $q' \leq n'$, there exists a property $\Pi'_{n',q'}$ of Boolean functions over $[n']$ such that the following holds:*

1. *There exists an oracle machine that, on input n', q' and ϵ', uses $q' + \frac{c}{\epsilon'}$ queries and constitutes an ϵ'-tester of $\Pi'_{n',q'}$* (with one-sided error).
2. *For every $q' \leq n'$, any $1/c$-tester of $\Pi'_{n',q'}$ requires at least q'/c queries* (even when allowing two-sided error).

The original statement of [13, Thm. 2] uses asymptotic notation and does not specify the dependence of the upper bound on ϵ. Nevertheless, the original proof explicitly establishes the result stated here (i.e., Theorem 2.2). Using Theorem 2.2, we prove our first result.

Theorem 2.3 (hierarchy theorem for size-oblivious query complexity, generic functions model): *For every monotonically non-increasing $q : (0,1] \to \mathbb{N}$, there exists a property $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$, where Π_n is a set of Boolean functions over $[n]$, such that Π is ϵ-testable* (with one-sided error) *in $q(\Omega(\epsilon)) + O(1/\epsilon)$ queries, but is not ϵ-testable in $o(\min(q(O(\epsilon)), \epsilon n))$ queries* (even when allowing two-sided error).

The fact that the lower bound may decrease with ϵ when n is small (e.g., when $n < q(\epsilon)/\epsilon$) is an artifact of the proof, which can be avoided (see Remark 2.4). In any case, for sufficiently large n (e.g., $n > q(\epsilon)/\epsilon$) we get a lower bound of $\Omega(q(O(\epsilon)))$. On the other hand, typically (i.e., for $q(\epsilon) = \Omega(1/\epsilon)$), the upper bound simplifies to in $O(q(\Omega(\epsilon)))$.

Proof: Using the family of properties asserted in Theorem 2.2 (and letting c be the corresponding universal constant), we let $\Pi_n = \bigcup_{i\in[1+\log_c n]} \Pi_n^{(i)}$ such that $f \in \Pi_n^{(i)}$ if there exists $f' \in \Pi'_{c^{-i}\cdot n, q(c^{-i})}$ such that $f(j) = f'(j)+1$ if $j \in [c^{-i} \cdot n]$ and $f(j) = 0$ otherwise.[5] Hence, the functions in $\Pi_n^{(i)}$ range over $\{0, 1, 2\}$ and assume the value 0 on points in $\{c^{-i} \cdot n + 1, ..., n\}$. Also note that the all-zero function is in Π_n, since it is in $\Pi_n^{(1+\log_c n)}$.

Claim 2.3.1 (upper bound): *The property Π_n is ϵ-testable* (with one-sided error) *in $q(\epsilon/c)+O(1/\epsilon)$ queries.*

Proof: Assuming that the function $f : [n] \to \{0, 1, 2\}$ is in Π_n, the tester tries to determine $i \in [1 + \log_c n]$ such that $f \in \Pi_n^{(i)}$. Actually, letting $\ell = \lceil \log_c(1/\epsilon)\rceil$, the tester tries to either find $i \in [\ell]$ such that $f \in \Pi_n^{(i)}$ or indicate that no such i exists, which may mean that $f \in \Pi_n \setminus \bigcup_{i\in[\ell]} \Pi_n^{(i)}$ (e.g., indeed, $f \in \Pi_n^{(i)}$ may hold for some $i \in \{\ell + 1, .., \log_c n + 1\}$). In the former case (where such an $i \in [\ell]$ was found), the tester checks that indeed $f \in \Pi_n^{(i)}$, by invoking the tester for $\Pi'_{c^{-i}\cdot n,q(c^{-i})}$ as well as testing that f assumes the value 0 on $\{c^{-i} \cdot n + 1, ..., n\}$. In the latter case (where no such an $i \in [\ell]$ was found), the tester merely checks that f assumes the value 0 on $\{c^{-(\ell+1)} \cdot n + 1, ..., n\}$, which implies that it is ϵ-close to the all-zero function.[6] Specifically, on input $n \in \mathbb{N}$ and $\epsilon > 0$, and oracle access to $f : [n] \to \{0, 1, 2\}$, the tester proceeds as follows.

1. For $i = 1, ..., \ell \stackrel{\text{def}}{=} \lceil \log_c(1/\epsilon)\rceil$, the tester queries f at the point $c^{-i} \cdot n$, and lets i^* be the smallest $i \in [\ell]$ such that $f(c^{-i} \cdot n) \in \{1, 2\}$, and $i^* = \ell + 1$ if no such i exists.

[5] We assume, for simplicity, that n is a power of c; otherwise, one may replace $c^{-i} \cdot n$ by $\lfloor c^{-i} \cdot n\rfloor$. Either way, if $c^{-i} \cdot n < 1$, then $\Pi_n^{(i)}$ cosists of the all-zero function. Also, while $\Pi'_{n',q'}$ was defined only for $q' \leq n'$, we let $\Pi'_{n',q'} = \emptyset$ if $q' > n'$.

[6] We don't check that f is the all-zero function in order to avoid a small error probability in case that $f \in \Pi_n^{(i)}$ for some $i \in \{\ell + 1, ..., \log_c n\}$.

2. The tester selects uniformly and independently $O(1/\epsilon)$ random points in $\{c^{-i^*} \cdot n+1,..,n\}$, queries f on each of these points, and *rejects* if any non-zero answer is obtained.

3. If $i^* \in [\ell]$, then the tester invokes the $(c^{i^*} \cdot \epsilon/2)$-tester for $\Pi'_{c^{-i^*} \cdot n, q(c^{-i^*})}$, and *outputs its verdict*. Specifically, the $(c^{i^*}\epsilon/2)$-tester is invoked while emulating a function $f' : [c^{-i^*} \cdot n] \to \{0,1\}$ defined by $f'(j) = f(j) - 1$, where if during the emulation the tester enconters a point on which f evaluates to 0, then it halts and *rejects*.

   Otherwise (i.e., $i^* = \ell + 1$), the tester just *accepts*.

The query complexity of the purported ϵ-tester is upper-bounded by $O(1/\epsilon) + (q(c^{-\ell}) + 2/\epsilon)$, where the first term is due to Step 2 (whose complexity dominates the complxity of Step 1), and the second term is due to the complexity of $(c^{i^*}\epsilon/2)$-testing $\Pi'_{c^{-i^*} \cdot n, q(c^{-i^*})}$, when $i^* \in [\ell]$. (Recall that the query complexity of ϵ'-testing $\Pi'_{n',q'}$ is upper-bounded by $q' + \frac{c}{\epsilon'}$, and that $q(c^{-i^*}) + \frac{c}{c^{i^*}\epsilon/2} \le q(c^{-\ell}) + 2/\epsilon$, when $i^* \in [\ell]$.) Using $c^{-\ell} > \epsilon/c$, which is due to $\ell < \log_c(1/\epsilon) + 1$, we derive the claimed query complexity bound (of $O(1/\epsilon) + q(\epsilon/c)$).

Next, we verify that in the case that $f \in \Pi_n$ the tester always accepts. If $f \in \Pi_n^{(i)}$ for some $i \in [\ell]$, then i^* is set to i (in Step 1), rejection does not happen in Step 2 (since f evaluates to 0 on $\{c^{-i^*} \cdot n+1,...,n\}$), and in Step 3 the tester for $\Pi'_{c^{-i} \cdot n, q(c^{-i})}$ is invokes by providing it with access to $f' \in \Pi'_{c^{-i} \cdot n, q(c^{-i})}$ (derived from f), which implies that our tester always accepts. If $f \in \Pi_n \setminus \bigcup_{i \in [\ell]} \Pi_n^{(i)}$, then i^* is set to $\ell+1$ (in Step 1), rejection does not happen in Step 2 (since f evaluates to 0 on $\{c^{-i^*} \cdot n+1,...,n\}$), and in Step 3 the tester accepts (without any farther checking).

We now turn to the case that f is ϵ-far from Π_n. Suppose that, in Step 1, i^* is set to $\ell+1$. In this case, the tester rejects with high probability, since f is ϵ-far from the all-zero function (which is in Π_n), whereas $[c^{-(\ell+1)} \cdot n]$ contains at most $(\epsilon/c) \cdot n$ points, which implies that $\{c^{-i^*} \cdot n+1,..,n\}$ contains more than $\epsilon \cdot n - (\epsilon/c) \cdot n > \epsilon n/2 > \frac{\epsilon}{2} \cdot (n - c^{-i^*} \cdot n)$ points on which f evalutes to non-zero. In this case, the tester rejects with high probability in Step 2.

Hence, we focus on the case that $i^* \in [\ell]$ and consider two cases. If $\{c^{-i^*} \cdot n+1,...,n\}$ contains $\epsilon n/2$ points on which f evaluates to non-zero, then the tester rejects with high probability in Step 2. Otherwise (as shown next), the function f' (as defined as in Step 3) is $(c^{i^*}\epsilon/2)$-far from $\Pi'_{c^{-i^*} \cdot n, q(c^{-i^*})}$, which cause the tester to reject with probability at least 2/3 in Step 3. Suppose, towards the contradiction, that f' is $(c^{i^*}\epsilon/2)$-close to some $g' \in \Pi'_{c^{-i^*} \cdot n, q(c^{-i^*})}$ (and recall that by the case's hypothesis $\{c^{-i^*} \cdot n+1,...,n\}$ contains less than $\epsilon n/2$ points on which f evaluates to non-zero). Then, we may obtain $g \in \Pi_n^{(i^*)}$ by defining $g(j) = g'(j) + 1$ if $j \in [c^{-i^*} \cdot n]$ and $g(j) = 0$ otherwise. But this implies that g differs from f on less than $(c^{i^*}\epsilon/2) \cdot (c^{-i^*} \cdot n) + \epsilon n/2 = \epsilon n$ points, which contradicts our hypothesis that f is ϵ-far from Π_n. ■

Claim 2.3.2 (lower bound): *The property* Π_n *is not* ϵ*-testable in less than* $\min(q(c^2 \cdot \epsilon), \epsilon n)/c$ *queries* (even when allowing two-sided error).

Proof: Let $F_n^{(i)}$ denote the set of all function $f : [n] \to \{0,1,2\}$ such that $f(j) = 0$ if and only if $j \in \{c^{-i} \cdot n+1,...,n\}$, and let $\Gamma_\delta(\Pi)$ denote the set of all functions that are δ-far from Π. Then, for every $\epsilon > 0$, letting $i = \lfloor \log_c(1/c\epsilon) \rfloor$, we observe that an ϵ-tester for Π_n must distinguish $\Pi^{(i)}$ from $\Gamma_\epsilon(\Pi_n^{(i)}) \cap F_N^{(i)}$, since $\Pi_n^{(i)} \subseteq \Pi_n$ and (as shown below) $\Gamma_\epsilon(\Pi_n^{(i)}) \cap F_n^{(i)} \subseteq \Gamma_\epsilon(\Pi_n)$. But the latter distinguisher is essentially a $c^i \cdot \epsilon$-tester for $\Pi'_{c^{-i} \cdot n, q(c^{-i})}$, since functions in $\Pi_n^{(i)}$ and $F_n^{(i)}$ differ only on $[c^{-i} \cdot n]$, whereas $\Pi_n^{(i)}$ (resp., $\Gamma_\epsilon(\Pi_n^{(i)}) \cap F_n^{(i)}$) emulates $\Pi'_{c^{-i}\cdot n, q(c^{-i})}$ (resp., $\Gamma_{c^i\cdot\epsilon}(\Pi'_{c^{-i}\cdot n, q(c^{-i})})$).[7] Recalling that $c^i \cdot \epsilon \leq 1/c$, we have reduced $1/c$-testing $\Pi'_{c^{-i}\cdot n, q(c^{-i})}$ to ϵ-testing Π_n, which means that the latter task has complexity at least $q(c^{-i})/c \geq q(c^2 \cdot \epsilon)/c$, where the last inequality uses $c^{-i} \leq c^2\epsilon$.

We now turn to show that $\Gamma_\epsilon(\Pi_n^{(i)}) \cap F_n^{(i)} \subseteq \Gamma_\epsilon(\Pi_n)$. Suppose, towards the contradiction, that $f \in \Gamma_\epsilon(\Pi_n) \cap F_n^{(i)}$ is ϵ-close to $g \in \Pi_n^{(j)}$ for some $j \in [1 + \log_c n]$. Clearly, $j \neq i$ (since otherwise f is ϵ-close to $\Pi_n^{(i)}$). Hence, $f \in F_n^{(i)}$ is ϵ-close to $g \in F_n^{(j)}$, for some $j \neq i \in \mathbb{N}$, but we shall show next that this is impossible. The key observation is that functions in $F_n^{(i)}$ are at (relative) distance at least $|c^{-i} - c^{-j}|$ from functions in $F_n^{(j)}$, where the distance is due to the mismatch between zero and non-zero values. But we shall show that in both cases $|c^{-i} - c^{-j}| > \epsilon$, we contradicts the claim that f is ϵ-close to g. Specifically, the case of $j > i$ is impossible, because then the two functions differ on $c^{-i} \cdot n - c^{-j} \cdot n$ of the non-zeros of f, whereas $c^{-i} - c^{-j} > c^{-i-1} \geq \epsilon$. (Here we use $c > 2$ as well as $i \leq \log(1/c\epsilon)$, which implies $c^{-i-1} \geq c^{-\log_c(1/c\epsilon)-1} = \epsilon$.) Similarly, $j < i$ implies that the two functions differ on $c^{-j} \cdot n - c^{-i} \cdot n$ of the non-zeros of g, whereas $c^{-j} - c^{-i} > c^{-i} > \epsilon$. This completes the proof. ■

From tri-valued functions to Boolean ones. The foregoing argument established the theorem, except that the properties used are of tri-valued functions rather than of Boolean ones. This is easily fixed by encoding each trit by two bits, which means that distances and query complexity change by a factor of two. These effects can be covered by increasing some constants by a factor of two (in comparison to their values in Claims 2.3.1 and 2.3.2). ■

Remark 2.4 (forcing a lower bound of $\Omega(1/\epsilon)$): *As noted above, Theorem 2.3 asserts an unintuitive query complexity lower bound* (of the form $\min(q(\epsilon), \epsilon n)$)*; that is, this bound decreases with ϵ when n is small, whereas we expect the lower bound to be at least* $\Omega(1/\epsilon)$ (or rather $\Omega(\min(1/\epsilon, n))$). *We comment that such an unintuitive behavior can be eliminated by a generic transformation. More generally, let $q, Q : \mathbb{N} \times (0,1] \to \mathbb{N}$ be monotonically non-increasing and let Π_n be a property of Boolean functions over $[n]$ that is ϵ-testable in $q(n,\epsilon)$ queries, but is not ϵ-testable in less than $Q(n,\epsilon)$ queries. Then, there exists a property of Boolean functions over $[3n]$ that is ϵ-testable in $O(q(n,\epsilon)) + O(1/\epsilon)$ queries, but is not $\epsilon/3$-testable in less than $\max(Q(n,\epsilon), \min(1/\epsilon, n))$ queries. Specifically, for each $f \in \Pi_n$, we introduce the function $f' : [3n] \to \{0,1\}$ such that $f'(j) = f(j)$ if $j \in [n]$ and $f'(j) = 0$ otherwise. That is, we pad each function by $2n$ zeros and observe that testing the new property requires testing the padding, which has query complexity $\Theta(\min(1/\epsilon, n))$.*

The foregoing construction is adaptable also to the testing models considered in the following sections.

3 Graph Properties in the Bounded-Degree Model

The bounded-degree graph model refers to a fixed (constant) degree bound, denoted $d \geq 2$. An n-vertex graph $G = ([n], E)$ (of maximum degree d) is represented in this model by a function $g : [n] \times [d] \to \{0, 1, ..., n\}$ such that $g(v, i) = u \in [n]$ if u is the i^{th} neighbor of v and $g(v, i) = 0$ if v has less than i neighbors.[8] Distance between graphs is measured in terms of their aforementioned representation; that is, as the fraction of (the number of) different array entries (over dn). Graph

[7] That is, $f \in \Pi_n^{(i)}$ is obtained by padding some $f' \in \Pi'_{c^{-i}\cdot n, q(c^{-i})}$, and $f \in \Gamma_\epsilon(\Pi_n^{(i)}) \cap F_n^{(i)}$ corresponds to a padding of some $f' \in \Gamma_{c^i\cdot\epsilon}(\Pi'_{c^{-i}\cdot n, q(c^{-i})})$. In both cases, $f(j) = f'(j) + 1$ if $j \in [c^{-i} \cdot n]$ and $f(j) = 0$ otherwise.

[8] 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 $\deg(v) \stackrel{\text{def}}{=} |\{i : g(v,i) \neq 0\}|$.

properties are properties that are invariant under renaming of the vertices (i.e., they are actually properties of the underlying unlabeled graphs).

Recall that [7] proved that, in this model, testing 3-Colorability requires a linear number of queries (even when allowing two-sided error). Building on this result, Goldreich *et. al.* [13] showed:

Theorem 3.1 ([13, Thm. 3], revised):[9] *There exists a universal constant $c > 2$ such that, for every $d, q', n' \in \mathbb{N}$ that satisfy $d \geq c$ and $q' \leq n'$, there exists a graph property $\Pi'_{d,n',q'}$ of n'-vertex graphs such that the following holds in the bounded-degree graph model with degree bound d:*

1. *There exists an oracle machine that, on input d, n', q' and ϵ', makes $c \cdot dq'/\epsilon'$ queries and constitutes an ϵ'-tester of $\Pi'_{d,n',q'}$* (with one-sided error)*.*
2. *Any $1/c$-tester of $\Pi'_{d,n',q'}$ requires at least q'/c queries* (even when allowing two-sided error)*.*

Furthermore, $\Pi'_{d,n',q'}$ is the set of n'-vertex graphs of maximum degree $d/2$ that are 3-colorable and consist of connected components of size at most q'.

(Recall that d denotes the degree parameter used in the bound-degree graph model, whereas the degree of the graphs having the property of interest may be lower. We have set the later value to $d/2$ in order to facilitate the presentation in the rest of this section.) Using Theorem 3.1, we first prove a weak version of the hierarchy theorem that has been eluded to in the abstract and introduction.

Theorem 3.2 (hierarchy theorem for size-oblivious query complexity, a weak version for the bounded-degree graph model): *For all sufficiently large $d \in \mathbb{N}$ and every monotonically non-increasing $q : (0, 1] \to \mathbb{N}$, there exists a graph property $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$, where Π_n consists of n-vertex graphs of degree at most d, such that Π is ϵ-testable in $O(q(\Omega(\epsilon))^2/\epsilon^2)$ queries, but is not ϵ-testable in $o(\min(q(O(\epsilon)), \epsilon n))$ queries.*

Both the lower and upper bounds refer to two-sided error testers (in the bounded-degree graph model). A stronger version that asserts an upper bound of $O(q(\Omega(\epsilon))/\epsilon) + O(1/\epsilon^2)$ queries is presented later (in Theorem 3.5).

Proof: Following the strategy of the proof of Theorem 2.3, our starting point is the family of properties asserted in Theorem 3.1, for $d = c$. We wish to pad the n'-vertex graphs (provided by Theorem 3.1) to n-vertex graphs such that the added $n - n'$ vertices are easy to identify. The first idea that comes to mind is to augment the "(n'-vertex) base graph" by $n - n'$ isolated vertices, but this presumes that the base graph has no isolated vertices. One possible solution is to modify the properties asserted in Theorem 3.1 so that they contain only graphs that contain no isolated vertices, while showing that the asserted upper and lower bounds still holds. Instead, we choose the alternative solution of using a different type of padding. Specifically, we augment the n'-vertex base graph with $\frac{n-n'}{d+1}$ isolated $(d + 1)$-cliques, as detailed next.

For every $n \in \mathbb{N}$, we let $\Pi_n = \bigcup_{i\in[\log_c n]} \Pi_n^{(i)}$ such that an n-vertex graph of maximum degree d is in $\Pi_n^{(i)}$ if it consists of a graph in $\Pi'_{d,c^{-i}\cdot n,q(c^{-i})}$, hereafter referred to as the base graph, and $(1-c^{-i})\cdot n$ vertices of degree d that reside in isolated $(d + 1)$-vertex cliques.[10] (Actually, to avoid integrality problems, one may postulate that the vertices of degree d are arranged in connected components that are each d-regular graphs of size at most $2d$ (which implies that they have diameter at most 2);

[9] Again, the original statement of [13, Thm. 3] uses asymptotic notation and does not specify the dependence of the upper bound on ϵ. Nevertheless, the original proof explicitly establishes the result stated here. Also, the statement of [13, Thm. 3] asserts the existence of a constant d, but the argument extends to all sufficiently large d.

[10] If $c^{-i} \cdot n < 1$, then $\Pi'_{d,c^{-i}\cdot n,q(c^{-i})}$ consists of the d-regular graph than contains only $(d + 1)$-cliques.

but, for sake of simplicity, we mandate that each of these connected components is of size $d+1$ (i.e., is a clique).) We stress that the vertices of the base graph have degree at most $d/2$, which makes them easy to distinguish from the "padding" vertices, which have degree d. Hence, we relate to the latter vertices as high-degree vertices, whereas the vertices of the base graph will be referred to as low-degree vertices.

Claim 3.2.1 (upper bound): *The property* Π_n *is* ϵ*-testable in* $O(q(\epsilon/c)^2/\epsilon^2)$ *queries.*

Proof: Wishing to follow the strategy used in the proof of Claim 2.3.1, we note that in the current setting we cannot directly sample vertices in the base graph (nor can we directly sample the high-degree vertices used to augment it to an n-vertex graph). Nevertheless, when the graph is in $\Pi_n^{(i)}$, we can indirectly sample both sets at an (expected) cost of c^i actual samples per each desired sample of the base graph.[11] More importantly, the value of i^* cannot be determined by accessing few fixed locations in the graph (like in the proof of Claim 2.3.1). Instead, the value of i^* is determined by estimating the density of the low-degree vertices via sampling, which means that this determination is only approximate and carries an error probability (also in the case that the graph has the property). Specifically, we set $i^* \in [\ell]$ such that the estimated density of the low-degree vertices (in the input graph) is approximately c^{-i^*}, and otherwise we use $i^* = \ell + 1$ (as in the proof of Claim 2.3.1).[12] We highlight two issues here:

- As noted above, since the density is estimated by random sampling, it introduces an error probability also in the case that the input graph is in Π_n, and hence we obtain a two-sided error tester (although the upper bound of Theorem 3.1 is established using a one-sided error tester).

- Since the density is only approximated, we may end-up invoking the tester for $\Pi'_{d,c^{-i^*}\cdot n,q(c^{-i^*})}$ on a graph that has approximately but not exactly $c^{-i^*}\cdot n$ vertices. This may happen when testing an input graph not in Π_n, and it is relevant when the graph is far from Π_n and has approximately but not exactly $n' \stackrel{\rm def}{=} c^{-i^*}\cdot n$ low-degree vertices. The problem is that there is no guarantee as to how the tester for $\Pi'_{d,n',q(c^{-i^*})}$ behaves when inspecting an n''-vertex graph such that $n'' \neq n'$.

  Nevertheless, we observe that if n'' is extremely close to n' (say, $n'' = (1 \pm O(c^{i^*}\epsilon/q(c^{-i^*})))\cdot n'$) and the n-vertex graph is ϵ-far from Π_n (and the high-degree vertices reside in $(d+1)$-cliques), then the tester will reject with probability at least 0.6. This is the case since if $n'' = n' + k$ (resp., $n'' = n' - k$), for $k \in [0.01n'/q^*]$, then making $q^* \stackrel{\rm def}{=} O(q(c^{-i^*})/c^{i^*}\epsilon)$ queries to the n''-vertex graph G'' is almost the same as making these queries to an n'-vertex induced subgraph of G'' (resp., to an n'-vertex graph that contains G'' as an induced subgraph), whereas this n'-vertex graph is $c^{i^*}\epsilon/2$-far from $\Pi'_{d,n',q(c^{-i^*})}$.

For sake of clarity, we spell out the resulting tester. On input an n-vertex graph G and proximity parameter ϵ, the tester proceeds as follows (while setting $\ell = \lceil \log_c(1/\epsilon)\rceil$):

1. *Determining* i^*: The tester selects uniformly and independently $m = O(1/\epsilon)$ random vertices, and obtains a very rough estimate of the number of low-degree vertices (i.e., vertices of degree at most $d/2$) in G. If, for some $i^* \in [\ell]$, the number of low-degree vertices is $(1 \pm 0.1)\cdot c^{-i^*}\cdot m$, then i^* is set accordingly; if the number of low-degree vertices seen in this sample is smaller

[11] Recall that, without loss of generality, testers in this model query about the incidence of either a previously seen vertex or a randomly selected vertex.

[12] Recall that $\ell = \lceil \log_c(1/\epsilon)\rceil$. Actually, if the estimated density is above $\epsilon/2$ but is not close to a power of $1/c$ we may reject on the spot.

than $1.1 \cdot c^{-(\ell+1)} \cdot m$, then the tester sets $i^* = \ell+1$; otherwise (i.e., i^* was not set so far), the tester *rejects*.

2. *Testing that the high-degree vertices constitute a proper padding and obtaining a refined estimate of the number of low-degree vertices*: The tester checks that the vertices that have degree greater than $d/2$ are arranged in $(d+1)$-cliques and that the number of low-degree vertices is very close to $c^{-i^*} \cdot n$. Specifically:

   (a) The tester selects $O(1/\epsilon)$ random vertices in G and checks that each vertex of degree greater than $d/2$ resides in an isolated $(d+1)$-clique. If some vertex of degree greater than $d/2$ was found not to reside in a $(d+1)$-clique, then the tester *rejects*. If $i^* = \ell+1$ and the tester did not reject, then it *accepts*.

   (b) Letting $\delta = \min(0.1c^{i^*}\epsilon, 0.01/q^*))$, where $q^* \stackrel{\rm def}{=} O(q(c^{-i^*})/c^{i^*}\epsilon)$ is the query complexity of the $(c^{i^*} \cdot \epsilon/2)$-tester for $\Pi'_{d,c^{-i^*}\cdot n,q(c^{-i^*})}$, and using a sample of $O(c^{i^*}/\delta^2)$ random vertices, the tester estimates the number of low-degree vertices (i.e., vertices of degree at most $d/2$) up to $\pm\delta \cdot c^{-i^*} \cdot n$ (with high probability). If this estimate deviates from $c^{-i^*} \cdot n$ by more than $\delta \cdot c^{-i^*} \cdot n$, then the tester *rejects*.

   Note that the size of the sample is

$$\begin{array}{rcl} O(c^{i^*}/\delta^2) & = & \max(O(c^{i^*}/(c^{i^*}\epsilon)^2), O(c^{i^*} \cdot (q(c^{i^*})/c^{i*}\epsilon)^2)) \\ & = & \max(O(1/\epsilon^2), O(q(c^{i^*})^2/c^{i^*}\epsilon^2)) \end{array}$$

$$O(q(c^{i^*})^2/\epsilon^2).$$

   (Indeed, if $q(e') = \Omega(\epsilon')$, then we obtain an improved upper bound of $\max(O(1/\epsilon^2), O(q(c^{i^*})^2/\epsilon))$.)

3. *Testing the subgraph induced by low-degree vertices*: The tester invokes the $(c^{i^*} \cdot \epsilon/2)$-tester for $\Pi'_{d,c^{-i^*}\cdot n,q(c^{-i^*})}$ on the subgraph of G induced by its low-degree vertices, and *outputs the verdict of this tester*. Specifically, the $(c^{i^*}\epsilon/2)$-tester is invoked while emulating a graph with $(1 \pm \delta) \cdot c^{-i^*} \cdot n$ vertices that contains only low-degree vertices. (Recall that we emulate each vertex selection at an (expected) cost of c^{i^*} actual samples, hence the number of queries made in this step is $c^{i^*} \cdot O(q(c^{-i^*})/(c^{i^*}\epsilon/2)) = O(q(c^{-i^*})/\epsilon)$.)

   Formally, the tester for $\Pi'_{d,c^{-i^*}\cdot n,q(c^{-i^*})}$ expects the labels of the tested graph to reside in $[c^{-i^*} \cdot n]$, whereas the labels of the vertices of the induced subgraph are in $[n]$. This discrepancy is easy to bridge by maintaining a partial injection of $[n]$ to $[c^{-i^*} \cdot n]$. Specifically, a query regarding the incidence list of a vertex v in $[c^{-i^*} \cdot n]$, is emulated by using the matched vertex in $[n]$ if v has appeared before (either as a query or as an answer) and by a new random low-degree vertex of G (obtained by repeated sampling) otherwise.

The query complexity of the purported ϵ-tester is upper-bounded by $O(1/\epsilon) + O(q(\epsilon/c)^2/\epsilon^2) + O(q(\epsilon/c)/\epsilon)$, where the three terms correspond to the three foregoing steps (and the complexity of each of the last two steps is maximal when $i^* = \ell = \lceil \log_c(1/\epsilon) \rceil$). Indeed, the query complexity is dominated by the second term, which accounts for the cost of obtaining a good approximation (in Step 2b).

The analysis of the case that $G \in \Pi_n$ proceeds very much as in the proof of Claim 2.3.1, except that here Steps 1 and 2b carry a small probability of error (also in the case of an input in Π_n). Turning to the case that G is ϵ-far from Π_n, the case of improper padding (i.e., the high-degree vertices of G are 0.1ϵ-far from forming a collection of $(d+1)$-cliques) is handled by Step 2a (analogously to Step 2 in the proof of Claim 2.3.1). More care is required with the remaining case in which the distance of G from Π_n is due to its low-degree vertices. Specifically, we are concerned of the case that G, which is ϵ-far from Π_n, contains $n'' = (1 \pm 2\delta) \cdot c^{-i^*} \cdot n$ low-degree vertices, whereas n'' does not necessarily equal $c^{-i^*} \cdot n$. (The other cases are handeled by Steps 1 and 2b.)[13] Letting G'' denote the subgraph of G induced its low-degree vertices, we consider two cases.

- If $n'' \geq n' \stackrel{\rm def}{=} c^{i^*} \cdot n$, then *any n'-vertex induced subgraph of G'' is $c^{i^*} \cdot (\epsilon - 0.3\epsilon)$-far from $\Pi'_{d,n',q(c^{-i^*})}$*. This is so because G is $(2\delta \cdot c^{-i^*} + 0.1\epsilon)$-close to an n-vertex graph that consists of the latter n'-vertex subgraph and $n - n'$ high-degree vertices arranged in $(d+1)$-cliques, which implies that G is $(\epsilon - 0.3\epsilon)$-far from $\Pi^{(i^*)}$ (since $\delta \cdot c^{-i^*} \leq \epsilon$ and $n'/n = c^{-i^*}$).

  Now, let G' be an arbitrary n'-vertex induced subgraph of G''. Then, the probability that the tester (for $\Pi'_{d,n',q(c^{-i^*})}$) queries G'' on a vertex that is not in G' (or obtains such a vertex as an answer) is at most $q^* \cdot 2\delta \leq 0.02$, since $q^* \stackrel{\rm def}{=} O(q(c^{-i^*})/c^{i^*}\epsilon)$ is the query complexity of the $(c^{i^*} \cdot \epsilon/2)$-tester for $\Pi'_{d,n',q(c^{-i^*})}$. This implies that, when invoked on G'', this tester rejects with probability at least $2/3 - 0.02 > 0.6$.

- A similar argument holds when $n'' < n'$, since in this case any n'-vertex graph that contains G'' as an induced subgraph is $c^{i^*} \cdot (\epsilon - 0.3\epsilon)$-far from $\Pi'_{d,n',q(c^{-i^*})}$.

Hence, we obtain the desired tester, which has two-sided error, where the error on inputs in Π_n arises from a mistaken determination of i^* (which occurs with small probability). ■

Claim 3.2.2 (lower bound): *The property Π_n is not ϵ-testable in less than* $\min(q(c^2 \cdot \epsilon), \epsilon n)/c$ *queries* (even when allowing two-sided error).

Proof: Claim 3.2.2 follows by a quite straightforward adaptation of the proof of Claim 2.3.2; that is, for any $\epsilon > 0$, letting $i = \lfloor \log_c(1/c\epsilon) \rfloor$, we reduce $1/c$-testing $\Pi'_{d,c^{-i}\cdot n,q(c^{-i})}$ to ϵ-testing Π_n by emulating answers to the queries issued by the latter tester, which means that we emulate an n-vertex graph using queries to an n'-vertex graph such that $n' = c^{-i} \cdot n$. Specifically, when $1/c$-testing the n'-vertex graph G' we invoke the ϵ-tester on an imaginary n-vertex graph G that consists of G' and $(n - n')/(d+1)$ isolated $(d+1)$-cliques. Indeed, we can place the the vertices of G' at fixed locations of our choice (e.g., we may just assign these vertices labels in $[c^{-i} \cdot n]$).

The crucial fact is that, as in the proof of Claim 2.3.2, the elements of the base object are different from those used in the padding; specifically, here, the vertices of the base graph have low degree whereas the vertices used in the padding have high degree. Again, the key observation is that, for any $j \in [1 + \log_c n] \setminus \{i\}$, graphs in $\Pi_n^{(i)}$ are relatively far from graphs in $\Pi_n^{(j)}$; that is, the relative distance is at least $|c^{-i} - c^{-j}|/2 > \epsilon$, where the factor of two is due to the fact that changing a high-degree vertex into a low-degree vertex (or vice versa) requires modifying at least half of its incidences.[14] ■

Combining 3.5.1 and 3.2.2, the theorem follows. ■

A tighter hierarchy theorem. The quadratic gap between the upper and lower bounds in Theorem 3.2 is due to the way we coped with the problem of invoking a tester for n'-vertex graphs on an n''-vertex graph, where $n'' \approx n'$ (alas $n'' \neq n'$). Specifically, we used a very good approximation of the number of low-degree vertices in the input n-vertex graph, denoted n'', and invoked the tester only if n'' is very close to n' (while rejecting the graph otherwise). This was

[13] Assuming that G is ϵ-far from Π_n and that (almost all) its high degree vertices reside in $(d+1)$-cliques, the other (two) cases are the case that the number of low-degree vertices is at least $1.2 \cdot c^{-(\ell+1)}n$ but is not in $\bigcup_{i\in[\ell]}\{(1 \pm 0.2) \cdot c^{-i}n\}$, and the case that the number of low-degree vertices is not $(1 \pm 2\delta) \cdot c^{-i^*} \cdot n$ (where i^* is as step in Step 1). In the first case Step 1 rejects (w.h.p.), whereas in the second case Step 2b rejects (w.h.p.).

[14] The inequality $|c^{-i} - c^{-j}|/2 > \epsilon$ uses $|c^{-i} - c^{-j}|/2 > c^{-i-1} > \epsilon$, which in turn uses $c > 3$ and $i \leq \log_c(1/c\epsilon)$.

obtained by approximating n'' such that the relative deviation is smaller that the reciprocal of the query complexity of the tester. Unfortuantely, such a good approximation had a high cost (i.e., quadratic in the query complexity of the given tester).

Fortunately, the tester provided by the proof of Theorem 3.1 is actually oblivious of the number of vertices in the tested graph, denoted n'. It only uses this number in order to sample the vertex set, which is identified with $[n']$. (The same holds for many other known testers.) Hence, we may provide this tester with a device that samples the vertex set rather than with the size of this set, and subsequently avoid the need to obtain a very good approximation of the number of low-degree vertices in our input graph. This leads to the following definition (which was subsequently generalized in different ways in [10, Sec. 2] and [11, Sec. 3]).

Definition 3.3 (property testing in the bounded-degree graph model, revised): *For a fixed $d \in \mathbb{N}$, let Π be a property of graphs of maximum degree d such that each graph is represented by an incidence function of the form $g : V \times [d] \to V \cup \{0\}$, where V is an arbitrary subset of $\mathbb{N}$. A* tester for the graph property Π *is a probabilistic oracle machine T that is given access to two oracles, an incidence function $g : V \times [d] \to V \cup \{0\}$ and a device denoted $\mathtt{Samp}(g)$ that samples uniformly in V* (equiv., in the domain of g)*, and satisfies the following two conditions:*

1. *The tester accepts each $g \in \Pi$ with probability at least $2/3$; that is, for every $g \in \Pi$* (and every $\epsilon > 0$)*, it holds that* $\Pr[T^{g,\mathtt{Samp}(g)}(\epsilon)=1] \geq 2/3$.

2. *Given $\epsilon > 0$ and oracle access to any g that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, for every $\epsilon > 0$ and $g : V \times [d] \to V \cup \{0\}$ that is ϵ-far from Π_V, it holds that* $\Pr[T^{g,\mathtt{Samp}(g)}(\epsilon)=0] \geq 2/3$*, where Π_V consists of the graphs in Π that have vertex-set V.*

   (As usual, $g : V \times [d] \to V \cup \{0\}$ is ϵ-far from Π_V if, for every $g' \in \Pi_V$, it holds that $|\{(v,i) \in V \times [d] : g(v,i) \neq g'(v,i)\}| > \epsilon \cdot d|V|$.)

The notions of one-sided error *tester and* ϵ-tester *are defined analogously.*

It turns out that the upper bound established in [13, Thm. 3] (restated in Theorem 3.1) holds also in the model of Definition 3.3. This is the case since the corresponding tester takes a sample of vertices[15] and explores the connected component in which each of these vertices resides, while suspending the exploration once a predetermined number of vertices is encountered (where this predtermined number is $q' + 1$). Hence, we have:

Proposition 3.4 ([13, Thm. 3], further revised): *Let c and the $\Pi'_{d,n',q'}$'s be as in Theorem 3.1. Then, there exists an oracle machine that, on input d, q' and ϵ', makes $c \cdot dq'/\epsilon'$ queries and constitutes an ϵ'-tester of $\bigcup_{n' \in \mathbb{N}} \Pi'_{d,n',q'}$* (in the sense of Definition 3.3).

Using Proposition 3.4, we prove the following.

Theorem 3.5 (hierarchy theorem for size-oblivious query complexity, actual version for the bounded-degree graph model): *For all sufficiently large $d \in \mathbb{N}$ and every monotonically non-increasing $q : (0,1] \to \mathbb{N}$, there exists a graph property $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$, where Π_n consists of n-vertex graphs of degree at most d, such that Π is ϵ-testable in $O(q(\Omega(\epsilon))/\epsilon) + O(1/\epsilon^2)$ queries, but is not ϵ-testable in $o(\min(q(O(\epsilon)), \epsilon n))$ queries.*

[15] Actually, this ϵ'-tester repeats a basic test for $O(1/\epsilon')$, where the basic test calls for sampling a single vertex and exploring its connected component.

Both the lower and upper bounds refer to two-sided error testers (in the bounded-degree graph model).

Proof: We follow the proof of Theorem 3.2, while modifying the emulation performed there towards establishing the stronger upper bound. Specifically, we shall use less accurate approximation of the number of low-degree vertices and rely on Proposition 3.4. In particular, we use the same graph properties $\Pi'_{d,c^{-i}\cdot n,q(c^{-i})}$ and Π_n (as in the proof of Theorem 3.2), while recalling that, for each $i \in \mathbb{N}$, the corresponding $\bigcup_{n'\in\mathbb{N}} \Pi'_{d,n',q(c^{-i})}$ can be tested in the model of Definition 3.3. We thus focus on replacing Claim 3.2.1 by the stronger upper bound provided by the following claim.

Claim 3.5.1 (upper bound): *The property Π_n is ϵ-testable in $O(q(\epsilon/c)/\epsilon) + O(1/\epsilon^2)$ queries.*

Proof: We follow the strategy of the proof of Claim 3.2.1, while using less accurate approximation for the number of low-degree vertices and relying on Proposition 3.4. Specifically, we approximate the said number up to an additive deviation of $0.1\epsilon \cdot n$, and invoke the $(c^{i^*} \cdot \epsilon/2)$-tester for $\bigcup_{n'\in\mathbb{N}} \Pi'_{d,n',q(c^{-i^*})}$ on the subgraph induced by the low-degree vertices. For sake of clarity, we spell out the derived tester (as operating on input an n-vertex graph G and proximity parameter ϵ).

1. The tester selects uniformly and independently $m = O(1/\epsilon^2)$ random vertices, and obtains a rough estimate of the number of low-degree vertices in G; specifically, the density of low-degree vertices is estimated within an additive deviation of $\pm 0.1\epsilon$. If, for some $i^* \in [\log_c(1/\epsilon)]$, the number of low-degree vertices seen is $(c^{-i^*} \pm 0.1\epsilon) \cdot m$, then i^* is set accordingly; if the number of low-degree vertices is smaller than $1.1 \cdot c^{-(\ell+1)} \cdot m$, then the tester sets $i^* = \ell + 1$; otherwise (i.e., i^* was not set so far), the tester *rejects*.

   (This step replaces Steps 1 and 2b in the proof of Claim 3.2.1.)[16]

   In addition (as in Step 2a in the proof of Claim 3.2.1), the tester selects $O(1/\epsilon)$ random vertices and checks whether each vertex of degree greater than $d/2$ resides in a $(d+1)$-clique. If this check fails, then the tester *rejects*. If this check passes and $i^* = \ell + 1$, then the tester *accepts*.

2. The tester invokes the $(c^{i^*} \cdot \epsilon/2)$-tester for $\bigcup_{n'\in\mathbb{N}} \Pi'_{d,n',q(c^{-i^*})}$ on the subgraph of G induced by its low-degree vertices, and *outputs the verdict of this tester*.

   (Indeed, this step replaces the last step in the proof of Claim 3.2.1. The point is that we invoke the tester for $\bigcup_{n'\in\mathbb{N}} \Pi'_{n',q(c^{-i^*})}$ while emulating a sampling oracle to the said subgraph, denoted G'', but without providing the approximate size of G''. In contrast, in the proof of Claim 3.2.1, we invoked a tester for $\Pi'_{c^{-i^*}\cdot n,q(c^{-i^*})}$ on G'', while relying on the hypothesis that the size of G'' is $c^{-i^*} \cdot n \pm \Theta(\epsilon/q(c^{-i^*}) \cdot n$.)

The query complexity of the purported ϵ-tester is upper-bounded by $O(1/\epsilon^2) + O(q(\epsilon/c)/\epsilon)$, where the second term is due to the complexity of emulating the $(c^{i^*}\epsilon/2)$-testing $\bigcup_{n'\in\mathbb{N}} \Pi'_{d,n',q(c^{-i^*})}$, which is $c^{i^*} \cdot O(q(c^{-i^*})/(c^{i^*}\epsilon/2))$. We note that, unlike in the proof of Claim 3.2.1, the approximation to the number of low-degree vertices (obtained in Step 1) is not used in the analysis of Step 2 (but it is rather used in order to assert that G is closed to a graph that has $c^{i^*} \cdot n$ low-degree vertices). Hence, we obtain the desired tester, which has two-sided error, where the error on inputs in Π_n arises from a possibly mistaken determination of i^* (which occurs with small probability). ■

Noting that Claim 3.2.2 remains intact, the theorem follows. ■

[16] In contrast, in the proof of Claim 3.2.1, the approximation (in Step 2b) was aimed at an additive deviation of $\pm\Theta(\epsilon/q(c^{-i^*}))$, which is typically much smaller.

An alternative construction. We present an alternative proof of Theorem 3.5, which actually yields a stronger result. Most importantly, the upper bound is established using a tester of one-sided error probability, while the lower bound still holds for general testers (having two-sided error probability). (In addition, we eliminated the additive $O(1/\epsilon^2)$ term in the upper bound, which is quite insignificant, since typically $q(\epsilon) = \Omega(1/\epsilon)$.)

Theorem 3.6 (a stronger version of Theorem 3.5): *For all sufficiently large $d \in \mathbb{N}$ and every monotonically non-increasing $q : (0,1] \to \mathbb{N}$, there exists a graph property $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$, where Π_n consists of n-vertex graphs of degree at most d, such that Π is ϵ-testable* with one-sided error *in $O(q(\Omega(\epsilon))/\epsilon)$ queries, but is not ϵ-testable in $o(\min(q(O(\epsilon)), \epsilon n))$ queries* (even when allowing two-sided error).

Proof Sketch: Using the same graph properties $\Pi'_{d,c^{-i}\cdot n,q(c^{-i})}$ (as in the proof of Theorems 3.2 and 3.5), we describe a different construction of Π. Specifically, for $n' = c^{-i} \cdot n$, rather than padding the n'-vertex base graph with $n - n'$ vertices of high degree that reside in $(d+1)$-cliques, we pad it with $(d-1)$-ary trees of height $\log_{d-1}((n-n')/n')$, and connect the root of each of these n' trees to a different vertex of the base graph. Actually, to facilitate the argument, we also connect all pairs of leaves that share the same parent node, forming $(d-1)$-cliques, as detailed next.

For every $n \in \mathbb{N}$, we let $\Pi_n = \bigcup_{i\in[\log_c n]} \Pi_n^{(i)}$ such that an n-vertex graph of maximum degree d is in $\Pi_n^{(i)}$ if it consists of a (base) graph in $\Pi'_{d,c^{-i}\cdot n,q(c^{-i})}$, and $(1-c^{-i}) \cdot n$ vertices of degree at least $d-1$ that are connected as follows.

- The high-degree vertices reside in $c^{-i} \cdot n$ trees, each being a $(d-1)$-ary tree of size $c^i - 1$. Hence, each tree has depth $\log_{d-1}(c^i - 1)$.
- The root of each tree is connected (by an edge) to a different vertex of the base graph.
- Leaves that are siblings in the tree are connected by additional edges, forming a $(d-1)$-clique. In other word, if v is the parent of leaves $w_1, ..., w_{d-1}$, then the graph induced by $\{v, w_1, ..., w_{d-1}\}$ is a d-clique.

Hence, each vertex of the base graph has degree at most $(d/2) + 1$, each internal node of the foregoing $(d-1)$-ary trees has degree d, and each leave of each tree has degree $d-1$. Hence, the gap between the degrees of vertices of the base graph (a.k.a low-degree vertices) and the padded vertices (a.k.a high-degree vertices) is maintained (assuming, w.l.o.g., that $(d/2)+1 < d-1$).

The crucial fact about this construction is that it allows for an efficient determination of $i^* \in [\ell+1]$, where (as before) $\ell = \lceil \log_c(1/\epsilon) \rceil$. Specifically, given access to a graph $G \in \Pi_n^{(i)}$, for an unknown $i \in [\ell]$, we can determine the value of i by making $O(d \cdot c^i)$ queries. This can be done by selecting an arbitrary vertex in the graph and exploring its neighborhood by a search to depth $h = \log_{d-1} c^i$. The point is that, assuming the graph is in $\Pi_n^{(i)}$, a search of depth h suffices for exploring the tree to which the start vertex belongs (or the tree that is connected to it, in case it resides in the base graph). Actually, we select a single vertex in the graph, and conduct a search of depth $h = \log_{d-1} c^\ell \le \log_{d-1}(c/\epsilon)$ starting at it, setting $i^* = i$ if we observed a $(d-1)$-ary tree of depth $\log_{d-1}(c^i - 1)$ with a clique among its silbling leaves, and setting $i^* = \ell + 1$ otherwise.[17]

Once i^* is determined, we use a random sample of $O(c^{-i^*}/\epsilon)$ vertices to test that all low-degree vertices are connected to $(d-1)$-ary trees of depth $\log_{d-1}(c^{i^*} - 1)$ (with cliques connecting sibling leaves). This is done by conducting a search of depth $\log_{d-1}(c^{i^*} - 1)$ from each of thec sampled vertices. Hence, we use a one-sided error test of the "properness" of the padding. Note that the complexity of this "padding test" is $O(c^{-i^*}/\epsilon) \cdot O(c^{i^*}) = O(1/\epsilon)$.

[17]Indeed, if such trees were observed for different values of $i \in [\ell]$, then we can reject on the spot.

The foregoing one-sided error padding test, which indirectly also tests that density of the set of low-degree vertices equals c^{-i^*}, replaces the probabilistic estimation of the density of this set as well as the setting of i^* itself (as conducted in the proofs of Claims 3.2.1 and 3.5.1, and leading to the two-sided error probability there). (As before, if $i = \ell + 1$ and the padding test passed, then we accept.) Lastly, if $i \in [\ell]$, then we test the base graph as in the proof of Claim 3.5.1, and rely on the fact that this tester has one-sided error. As before, this test has query complexity $c^{i^*} \cdot O(q(c^{-i^*})/c^{i^*}\epsilon) = O(q(\epsilon/c)/\epsilon)$.

Having concluded that (the redefine property) Π can be ϵ-tested with one-sided error and $O(q(\epsilon/c)/\epsilon)$ queries, we note that the lower bound asserted in Claim 3.2.2 hold also for the redefined property Π. This is the case since the proof is quite oblivious of the specific form of padding used, as long as the degrees of the padded vertices are far from the degrees of the vertices of the base graph. ■

4 Graph Properties in the Adjacency Matrix Model

In the adjacency matrix model (a.k.a the dense graph model), an n-vertex graph $G = ([n], E)$ is represented by the Boolean function $g : [n] \times [n] \to \{0, 1\}$ such that $g(u, v) = 1$ if and only if u and v are adjacent in G (i.e., $\{u, v\} \in E$). Distance between graphs is measured in terms of their aforementioned representation; that is, as the fraction of (the number of) different matrix entries (over n^2). Again, we focus on graph properties (i.e., properties of labeled graphs that are invariant under renaming of the vertices).

Recall that [12] proved that, in this model, there exist graph properties for which testing requires a quadratic (in the number of vertices) query complexity (even when allowing two-sided error). Building on this result, Goldreich *et. al.* [13] showed:

Theorem 4.1 ([13, Thm. 4], revised):[18] *There exists a universal constant $c > 5$ and a fixed quadratic polynomial p such that, for every $q', n' \in \mathbb{N}$ that satisfy $q' \leq \binom{n'}{2}$, there exists a graph property $\Pi'_{n',q'}$ of n'-vertex graphs such that the following holds in the dense graph model:*

1. *There exists an algorithm that, on input n', q' and ϵ', makes $p(1/\epsilon') \cdot q'$ queries and constitutes an ϵ'-tester of $\Pi'_{n',q'}$.*
2. *For every $q' \leq n'$, any $1/c$-tester of $\Pi'_{n',q'}$ requires at least q'/c queries.*

Both the lower and upper bounds refer to two-sided error testers. As in Section 3, using Theorem 4.1, we first prove a weak version of the hierarchy theorem that has been eluded to in the abstract and introduction.

Theorem 4.2 (hierarchy theorem for size-oblivious query complexity, a weak version for the dense graph model): *For every $q : (0, 1] \to \mathbb{N}$ that is monotonically non-increasing, there exists a graph property $\Pi = \bigcup_{n\in\mathbb{N}} \Pi_n$, where Π_n consists of n-vertex graphs, that is ϵ-testable in $O(q(\Omega(\epsilon))/\epsilon^2)^2$ queries, but is not ϵ-testable in $o(\min(q(O(\epsilon)), \epsilon n^2))$ queries.*

Both the lower and upper bounds refer to two-sided error testers in the dense graph model. A stronger version that asserts an upper bound of $O(q(\Omega(\epsilon))/\epsilon^2)$ queries is presented later (in Theorem 4.5).

[18]Again, the original statement of [13, Thm. 4] uses asymptotic notation and does not specify the dependence of the upper bound on ϵ. Nevertheless, the original proof explicitly establishes the result stated here. We comment that [13, Thm. 6] provides a stronger result in which the upper bound holds for one-sided error tester, but our technique introduces an error probability also in case the graph has the property. We postulated that $c > 5$ (rather than $c > 2$) in order to simplify the rest of the exposition, while noting that one can always increase c.

Proof: Following the strategy of the proof of Theorem 3.2, our starting point is the family of properties asserted in Theorem 4.1. Again, we avoid the temptation to pad the base n'-vertex graph by $n - n'$ isolated vertices. Instead, we use a single $(n - n')$-clique as the padding. Note that here, unlike in the proof of Theorem 3.2, the description of the (n'-vertex) base graph occupies an $(n'/n)^2$ fraction of the description of the n-vertex graph (rather than an (n'/n) fraction of it).

Specifically, we let $\Pi_n = \bigcup_{i\in[\log_c n]} \Pi_n^{(i)}$ such that an n-vertex graph is in $\Pi_n^{(i)}$ if it consists of a $(1 - c^{-i}) \cdot n$-vertex clique and graph in $\Pi'_{c^{-i}\cdot n, q(c^{-2i})}$, which is called the base graph. Note that each graph in $\Pi_n^{(i)}$ is c^{-2i}-close to graph that consists of an $(1 - c^{-i}) \cdot n$-vertex clique and $c^{-i} \cdot n$ isolated vertices (whereas in the proofs of Theorems 2.3 and 3.2 the corresponding relative distance was c^{-i}).

We shall use the fact that the vertices of the base graph have low degree (i.e., degree smaller than $n/c < 0.2n$), whereas the remaining vertices have high degree (i.e., degree at least $n - (n/c) - 1 > 0.8n$).[19] It is tempting to say that these two cases can be easily distinguished, but we warn that the distinguishing procedure is randomized (i.e., it relies on sampling additional vertices and checking whether they are adjacent to the tested vertex). We shall address this issue shortly.

Claim 4.2.1 (upper bound): *The property Π_n is ϵ-testable in $O(q(\epsilon/c^2)^2/\epsilon^4)$ queries.*

Proof: As in the proof of Claim 3.2.1, we wish to follow the strategy used in the proof of Claim 2.3.1. As noted above, one crucial difference is in the way we identify the vertices of the base graph (of a graph in $\Pi_n^{(i)}$), which are chracterized by their low degree. Here, unlike in the proof of Claim 3.2.1, we cannot identify these vertices by few deterministicly determined queries, but rather rely on random sampling. The straighforward way amounts to samplying several random vertices and asserting that the tested vertex is of low degree if it neighbors less than half of them, but a characterization of the vertices according to their degrees requires fixing the sample.[20] We prefer an alterantive way that consists of finding one high degree vertex (by sampling), and characterizing all other vertices in the graph according to whether or not they neighbor this vertex. (Indeed, vertices in the base graph do not neighbor any high degree vertex, whereas all other vertices of the large clique do neighbor such a vertex.)

Another difference between the proofs of Claims 2.3.1 and 3.2.1 and the current proof is that here we set $\ell = \lceil 0.5 \log_c(1/\epsilon)\rceil$, rather than $\ell = \lceil \log_c(1/\epsilon)\rceil$ as in the previous proofs. (Likewise, we invoke the $c^{-2i^*}\epsilon/2$-tester (rather than a $c^{-i^*}\epsilon/2$-tester).)

Again, the determination of i^* is done by sampling (see above), while here we rely on the fact that if $G \in \Pi_n^{(i)}$ then the vertices in its base graph do not neighbor any high degree vertex, whereas all high degree vertices neighbor each other (since they reside in an $(n - c^{-i^*}n)$-vertex clique). Hence, as in the proof of Claim 3.2.1, the determination of i^* carries an error probability (also in the case that the graph has the property).

Specifically, we set $i^* \in [\ell]$ such that the estimated density of the base graph (defined as the subgraph induced by the vertices that do not neighbor the selected high degree vertex) is approximately c^{-i^*}, and set $i^* = \ell + 1$ if the estimated density is below $1.1 \cdot c^{-(\ell+1)}$. Like in the proof of Claim 3.2.1, we approximate the size of the base graph up-to $\pm \min(0.1\epsilon, 0.01c^{-i^*}/q^*) \cdot n$, where $q^* = p(2/c^{2i^*}\epsilon) \cdot q(c^{-2i^*})$ is the query complexity of the $c^{-2i^*}\epsilon/2$-tester of $\Pi'_{c^{-i^*}\cdot n, q(c^{-2i^*})}$, and the complexity of obtaining this approximation (i.e., $\max_{i^*\in[\ell]}\{O(c^{-6i^*} \cdot q(c^{-2i^*})^2/\epsilon^4)\} = O(q(\epsilon/c^2)^2/\epsilon^4)$)[21]

[19] Indeed, in the case of graphs in $\Pi_n^{(i)}$, the vertices of the base graph have degree smaller than n/c^i, whereas the remaining vertices have degree $n - (n/c^i) - 1$.

[20] That is, a fixed partition of the vertices to high and low degree ones is obtained when fixing the coins used for the corresponding sampling process.

[21] The complexity of obtaining this approximation is upper-bounded by

$$\max(O(1/\epsilon^2), O(\max_{i^*\in[\ell]}\{O(c^{i^*} \cdot p(2/c^{2i^*}\epsilon) \cdot q(c^{-2i^*}))^2\}))$$

dominates the complexity of the entire tester. This very good approximation allows to invoke the $c^{-2i^*}\epsilon/2$-tester of $\Pi'_{c^{-i^*}\cdot n, q(c^{-2i^*})}$ on the subgraph of G induced by vertices that do not neighbor the selected high degree vertex, although the number of such vertices may slightly deviate from $c^{-i^*}\cdot n$ (where the deviation is upper-bounded by $0.01 \cdot c^{-i^*}n/q^*$).

Lastly, as in the proof of Claim 3.2.1, we cannot directly sample vertices in the base graph (nor can we directly sample the vertices of the large clique used to augment the base graph to an n-vertex graph), but we can indirectly sample both sets at the (expected) cost of c^{i^*} actual samples per each desired sample (of the base graph). For sake of clarity, we spell out the derived tester. On input an n-vertex graph G and proximity parameter ϵ, the tester proceeds as follows (while setting $\ell = \lceil 0.5 \log_c(1/\epsilon)\rceil$):

1. *Determining* i^*: The tester finds a vertex, denoted s, that seems to have degree at least $n/2$, by sampling $O(1)$ random vertices and very roughly approximating their degree (by considering their adjacency relation to $O(1)$ random vertices).

   Letting B denote the set of vertices that do not neighbor s in G, the tester obtains a rough estimate of the size of B; specifically, using a sample of $m = O(1/\epsilon^2)$ random vertices, with high probability, the relative deviation error is smaller than 0.1ϵ. If, for some $i^* \in [\ell]$, the number of sampled vertices in B is $(1 \pm 0.1)\cdot c^{-i^*}\cdot m$, then i^* is set accordingly; if the number of sampled vertices in B is smaller than $1.1 \cdot c^{-(\ell+1)} \cdot m$, then the tester sets $i^* = \ell + 1$; otherwise (i.e., if i^* was not set), the tester *rejects*.

2. *Testing the subgraph induced by* $[n] \setminus B$ *and obtaining a refined estimate of* $|B|$: The tester checks that the subgraph induced by $[n] \setminus B$ is a clique and that $|B|$ is very close to $c^{-i^*}\cdot n$. Specifically:

   (a) The tester selects $O(1/\epsilon)$ random vertices in G and checks that the vertices that neighbor s are adjacent to one another. If two vertices that neighbor s were found not to be adjacent, then the tester *rejects*.

   (b) Letting $\delta = \min(0.1c^{i^*}\epsilon, 0.01/p(2/c^{i^*}\epsilon)\cdot q(c^{-2i^*}))$ and using a sample of $O(c^{i^*}/\delta^2)$ random vertices, the tester estimates the size of B up to $\pm\delta \cdot c^{-i^*}\cdot n$ (with high probability). If this estimate deviates from $c^{-i^*}\cdot n$ by more than $\delta \cdot c^{-i^*}\cdot n$, then the tester *rejects*. (Note that $O(c^{i^*}/\delta^2) = O(q(c^{2i^*})^2/\epsilon^4))$.)

3. *Testing the subgraph induced by* B: The tester invokes the $(c^{2i^*}\cdot\epsilon/2)$-tester for $\Pi'_{c^{-i^*}\cdot n, q(c^{-2i^*})}$ on the subgraph of G induced by B, and *outputs the verdict of this tester*. Specifically, the $(c^{2i^*}\epsilon/2)$-tester is invoked while emulating a graph with $(1\pm\delta)\cdot c^{-i^*}\cdot n$ vertices. (Recall that we emulate each vertex selection at an (expected) cost of c^{i^*} actual samples.)[22]

   (The complexity of this emulation is $c^{i^*}\cdot O(q(c^{-2i^*})/(c^{2i^*}\cdot\epsilon)^2) = O(q(c^{-2i^*})/\epsilon^2)$.)

The analysis of the foregoing tester mimics the analysis of the tester presented in the proof of Claim 3.2.1. The claim follows, while using $q(c^{-2\ell}) \le q(\epsilon/c^2)$, which holds since $\ell < 0.5\log_c(1/\epsilon)+1$. ■

$$= \max(O(1/\epsilon^2), O(\max_{i^*\in[\ell]}\{O(c^{-6i^*}\cdot q(c^{-2i^*}))^2/\epsilon^4\})),$$

since $p(m) = O(m^2)$.

[22] As in the proof of Claim 3.2.1, the tester for $\Pi'_{c^{-i^*}\cdot n, q(c^{-2i^*})}$ expects the labels of the tested graph to reside in $[c^{-i^*}\cdot n]$, whereas the labels of the vertices of the induced subgraph are in $[n]$. Again, this discrepancy is easy to bridge by maintaining a partial injection of $[n]$ to $[c^{-i^*}\cdot n]$.

Claim 4.2.2 (lower bound): *The property Π_n is not ϵ-testable in less than* $\min(q(c^3 \cdot \epsilon), \epsilon n^2)/c$ *queries* (even when allowing two-sided error).

The proof of Claim 4.2.2 mimics the proofs of Claims 2.3.2 and 3.2.2, except that here i is set such that $c^{-2i} \in [c\epsilon, c^3\epsilon]$ (i.e., $i = \lfloor 0.5 \log_c(1/c\epsilon) \rfloor$). The theorem follows. ■

A tighter hierarchy theorem. As in the proof of Theorem 3.2, the quadratic gap between the upper and lower bounds in Theorem 4.2 is due to the way we coped with the problem of invoking a tester for n'-vertex graphs on an n''-vertex graph, where $n'' \approx n'$ (alas $n'' \neq n'$). Specifically, we used a very good approximation of the number of "high degree" vertices in the input n-vertex graph, but such a good approximation had a high cost (i.e., quadratic in the query complexity of the given tester).

Fortunately, the tester provided by the proof of Theorem 4.1 is actually oblivious of the number of vertices in the tested graph, denoted n'. Like in the case of Theorem 3.1, the tester only uses n' to sample the vertex set, which is identified with $[n']$. Hence, we may provide this tester with a device that samples the vertex set rather than with the size of this set, and avoid the need to obtain a very good approximation of the number of low-degree vertices in our input graph. This leads to the following definition (which was subsequently generalized in different ways in [10, Sec. 1] and [11, Sec. 2]).

Definition 4.3 (property testing in the dense graph model, revised): *Let Π be a property of graphs such that each graph is represented by an adjacency predicate of the form $g : V \times V \to \{0,1\}$, where V is an arbitrary subset of $\mathbb{N}$. A* tester for the graph property Π *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* $\texttt{Samp}(g)$ *that samples uniformly in the domain of g* (i.e., $V \times V$)*, and satisfies the following two conditions:*

1. *The tester accepts each $g \in \Pi$ with probability at least $2/3$; that is, for every $g \in \Pi$* (and every $\epsilon > 0$)*, it holds that* $\Pr[T^{g,\texttt{Samp}(g)}(\epsilon)=1] \geq 2/3$.
2. *Given $\epsilon > 0$ and oracle access to any g that is ϵ-far from Π, the tester rejects with probability at least $2/3$; that is, for every $\epsilon > 0$ and $g : V \times V \to \{0,1\}$ that is ϵ-far from Π_V, it holds that* $\Pr[T^{g,\texttt{Samp}(g)}(\epsilon)=0] \geq 2/3$*, where Π_V consists of the graphs in Π that have vertex-set V.*

   (As usual, $g : V \times V \times \{0,1\}$ is ϵ-far from Π_V if, for every $g' \in \Pi_V$, it holds that $|\{(u,v) \in V \times V : g(u,v) \neq g'(u,v)\}| > \epsilon \cdot |V|^2$.)

*The notion of ϵ-*tester *is defined analogously.*

It turns out that the upper bound established in [13, Thm. 4] (restated in Theorem 4.1) holds also in the model of Definition 4.3. This is the case since the corresponding tester (presented as [13, Alg. 4.5]) takes a sample of vertices, queries some of the corresponding vertex pairs, and decides based on some statistics of the sample (obliviously of the size of the tested graph).[23] Hence, we have:

[23]Specifically, for $s = \sqrt{q'}$, using a sample of $O(1/\epsilon')^2 \cdot s$ vertices in the input graph G, the tester first clusters the vertices in the sample according to their adjacency to $O(\log s)$ "signature vertices" (i.e., the $O(\log s)$-bit long string descibing the adjacency values); it continues if and only if exactly s clusters were found and these clusters are of approximately the same size (otherwise it rejects). It next selects (arbitrarily) a representative vertex in each cluster, and checks (making $\binom{s}{2}$ queries) that the subgraph induced by these representatives satisfies the "underlying property" (used to construct the property being tested, which is a "balanced blow-up" of the former). In addition, the tester check that random vertex pairs (in G) fit the adjacency relation of the representatives of the clusters to which they belong (per their own adjacency pattern). All operations are performed without reference to the number of vertices in G. (We stress that s is determined by q' only.)

Proposition 4.4 ([13, Thm. 3], further revised): *Let p and the $\Pi'_{n',q'}$'s be as in Theorem 4.1. Then, there exists an oracle machine that, on input q' and ϵ', makes $p(1/\epsilon') \cdot q'$ queries and constitutes an ϵ'-tester of $\bigcup_{n' \in \mathbb{N}} \Pi'_{n',q'}$* (in the sense of Definition 4.3).

Using Proposition 4.4, we prove the following.

Theorem 4.5 (hierarchy theorem for size-oblivious query complexity, actual version for the dense graph model): *For every $q : (0,1] \to \mathbb{N}$ that is monotonically non-increasing, there exists a graph property $\Pi = \bigcup_{n \in \mathbb{N}} \Pi_n$, where Π_n consists of n-vertex graphs, that is ϵ-testable in $O(q(\Omega(\epsilon))/\epsilon^2)$ queries, but is not ϵ-testable in $o(\min(q(O(\epsilon)), \epsilon n^2))$ queries.*

Both the lower and upper bounds refer to two-sided error testers in the dense graph model.

Proof: We follow the proof of Theorem 4.2, while modifying the emulation performed towards establishing the upper bound. (The modification is analogous to the modification of the proof of Theorem 3.2 employed when proving Theorem 3.5.) Specifically, we shall use less accurate approximation of the number of "low degree" vertices and rely on Proposition 4.4. In particular, we use the same graph properties $\Pi'_{c^{-i} \cdot n, q(c^{-2i})}$ and Π_n (as in the proof of Theorem 4.2), while observing that the $\bigcup_{n' \in \mathbb{N}} \Pi'_{n',q(c^{-2i})}$'s while recalling that, for each $i \in \mathbb{N}$, the corresponding $\bigcup_{n' \in \mathbb{N}} \Pi'_{n',q(c^{-i})}$ can be tested in the model of Definition 4.3. We thus focus on replacing Claim 4.2.1 by the stronger upper bound provided by the following claim.

Claim 4.5.1 (upper bound): *The property Π_n is ϵ-testable in $O(q(\epsilon/c)/\epsilon^2)$ queries.*

Proof: We follow the strategy of the proof of Claim 4.2.1, while using less accurate approximation for the number of "low degree" vertices and rely on Proposition 4.4. Specifically, we approximate the said number up to an additive deviation of $0.1\epsilon \cdot n$, and invoke the $(c^{2i^*} \cdot \epsilon/2)$-tester for $\bigcup_{n' \in \mathbb{N}} \Pi'_{n',q(c^{-2i^*})}$ on the subgraph induced by the "low degree" vertices (or rather by the non-neighbors of the selected high-degree vertex). For sake of clarity, we spell out the derived tester (as operating on input an n-vertex graph G and proximity parameter ϵ, while letting $\ell = \lceil 0.5 \log_c(1/\epsilon) \rceil$).

1. Determining s, B and i^* is done exactly as in Step 1 of the tester presented in the proof of Claim 4.2.1.
2. The tester checks that the subgraph induced by $[n] \setminus B$ is a clique. This is done exactly as in Step 2a of the tester presented in the proof of Claim 4.2.1.

   (Indeed, we avoid Step 2b in the foregoing tester.)
3. *Testing the subgraph induced by B*: The tester invokes the $(c^{2i^*} \cdot \epsilon/2)$-tester for $\cup_{n' \in \mathbb{N}} \Pi'_{n',q(c^{-2i^*})}$ on the subgraph of G induced by B, and *outputs the verdict of this tester*.

   (Indeed, this step replaces the last step of the tester presented in the proof of Claim 3.2.1. The point is that we invoke the tester for $\bigcup_{n' \in \mathbb{N}} \Pi'_{n',q(c^{-2i^*})}$ while emulating a sampling oracle to the said subgraph, but without providing the approximate size of this subgraph. In contrast, in the proof of Claim 3.2.1, we invoked a tester for $\Pi'_{c^{-i^*} \cdot n, q(c^{-2i^*})}$ on this subgraph, while relying on the hypothesis that its size was $(1 \pm 0.01/q^*) \cdot c^{-i^*} \cdot n$, where $q^* = p(2/c^{2i^*}\epsilon) \cdot q(c^{-2i^*})$.)

The query complexity of the purported ϵ-tester is upper-bounded by $O(1/\epsilon^2) + O(q(\epsilon/c)/\epsilon^2)$, where the second term is due to the complexity of emulating the $(c^{2i^*}\epsilon/2)$-testing $\bigcup_{n' \in \mathbb{N}} \Pi'_{n',q(c^{-2i^*})}$, which is $c^{i^*} \cdot O(q(c^{-i^*})/(c^{2i^*}\epsilon/2)^2)$. We note that, unlike in the proof of Claim 3.2.1, the approximation to the number of "low degree" vertices (obtained in Step 1) is not used in the analysis of Step 3 (but it is rather used in order to assert that G is closed to a graph that has $c^{i^*} \cdot n$ "low degree" isolated vertices). Hence, we obtain the desired tester, which has two-sided error, where the error on inputs in Π_n arises from a mistaken determination of i^* (which occurs with small probability). ■

Noting that Claim 3.2.2 remains intact, the theorem follows. ■

5 On Query Complexity that Depends on Both Parameters

As stated in the introduction, the results presented in Sections 2–4 complement hierachy theorems of Goldreich, Krivelevich, Newman, and Rozenberg [13]. While our results refer to query complexity that depends only on the proximity parameter, the results in [13] refer to the dependence of the query complexity on the size of the tested object (and focus on the case that the proximity parameter is set to some small positive constant). We can actually combine both flavors and get quite tight results on the query complexity of testing when allowing the query complexity to depend quite arbitrarily on both the size of the object and the proximity parameter. The proof are identical to those presented in Sections 2–4, except that the parameter q' will be determine as a function of the desired query complexity, which is now a function of both parameters.

Properties of generic functions. Generalizing the proof of Theorem 2.3, we obtain the following

Theorem 5.1 (Theorem 2.3, generalized): *For every $q : \mathbb{N} \times (0,1] \to \mathbb{N}$ that is monotonically non-decreasing in the first parameter and monotonically non-increasing in the second parameter, there exists a property Π_n of Boolean functions over $[n]$ that is ϵ-testable* (with one-sided error) *in $q(n,\Omega(\epsilon)) + O(1/\epsilon)$ queries, but is not ϵ-testable in $o(\min(q(n,O(\epsilon)),\epsilon n))$ queries* (even when allowing two-sided error).

Recall that by Remark 2.4, we may obtain a property of Boolean functions over $[3n]$ that is ϵ-testable in $O(q(n,\Omega(\epsilon))) + O(1/\epsilon)$ queries, but is not ϵ-testable in $o(\max(\min(q(n,O(\epsilon)),\epsilon n),\min(1/\epsilon,n)))$ queries. Note that when $\epsilon > 1/\sqrt{n}$, the lower bound simplifies to $\Omega(\min(q(n,O(\epsilon)),\epsilon n))$.

Proof Sketch: As stated above, we proceed as in the proof of Theorem 2.3, except that we place f in $\Pi_n^{(i)}$ if there exists $f' \in \Pi'_{c^{-i}\cdot n, q(n,c^{-i})}$ such that $f(j) = f'(j) + 1$ if $j \in [c^{-i} \cdot n]$ and $f(j) = 0$ otherwise. (Here, letting $\Pi'_{n',q'} = \emptyset$ if $q' > n'$ means that $\Pi_n^{(i)} = \emptyset$ if $q(n,c^{-i}) > c^{-i}n$.) The rest of the arguement proceeds exactly as in the case of Theorem 2.3. ■

Graph properties in the bounded-degree model. Generalizing the proof of Theorem 3.5, we obtain the following

Theorem 5.2 (Theorem 3.5, generalized): *For all sufficiently large $d \in \mathbb{N}$ and every $q : \mathbb{N}\times(0,1] \to \mathbb{N}$ that is monotonically non-decreasing in the first parameter and monotonically non-increasing in the second parameter, there exists a graph property Π_n of n-vertex graphs of degree at most d that is ϵ-testable in $O(q(n,\Omega(\epsilon))/\epsilon) + O(1/\epsilon^2)$ queries, but is not ϵ-testable in $o(\min(q(n,O(\epsilon)),\epsilon n))$ queries.*

Both the lower and upper bounds refer to two-sided error testers (in the bounded-degree graph model). Generalizing Theorem 3.6, we can show ϵ-testability *with one-sided error* using $O(q(\Omega(\epsilon))/\epsilon)$ queries, while maintained the $\Omega(\min(q(O(\epsilon)),\epsilon n))$ lower bound for general ϵ-testers (of two-sided error).

Graph properties in the adjacency matrix model. Generalizing the proof of Theorem 4.5, we obtain the following

Theorem 5.3 (Theorem 4.5, generalized): *For every $q : \mathbb{N} \times (0,1] \to \mathbb{N}$ that is monotonically non-decreasing in the first parameter and monotonically non-increasing in the second parameter, there exists a graph property Π_n of n-vertex graphs that is ϵ-testable in $\widetilde{O}(q(n,\Omega(\epsilon))/\epsilon^2)$ queries, but is not ϵ-testable in $o(\min(q(n,O(\epsilon)),\epsilon n^2))$ queries.*

Both the lower and upper bounds refer to two-sided error testers in the dense graph model.

6 Open Problems

While our query complexity bounds for the generic model are of the form $\Theta(q(n,\Theta(\epsilon)))$, in the two graph testing models we have a slackness of the form ϵ^c, where $c \in \{1,2\}$ depending on the model. This is most significant in the case of $q(n,\epsilon) = \mathrm{poly}(1/\epsilon)$, and a well-defined open problem is to get rid of this slackness.

Another problem left open refers to the dense graph model and consists of obtaining a hierarchy theorem in which the upper bound is established via one-sided error testers, whereas the lower bound holds also for general testers (with two-sided error). Note that such results were obtained for the other two models (see Theorems 2.3 and 3.6, resp.).

A much more vague open problem is to avoid the unnatural flavor of the properties used in establishing our results (as well as those of [13]). One cannot expect natural problems for arbitrary complexity bounds (i.e., for results stated for essentially any function q), but one may ask for it in cases such as $q(n,\epsilon) = \exp(1/\epsilon)$ and $q(n,\epsilon) = \epsilon^{-3}$. For sake of concreteness, we adopt the definition of Noam Livne for natural problems saying that a *problem is natural, beyond its use in a specific context, if it was considered before in a different context.*

References

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

[2] N. Alon, E. Fischer, I. Newman, and A. Shapira. A Combinatorial Characterization of the Testable Graph Properties: It's All About Regularity. In *38th STOC*, pages 251–260, 2006.

[3] N. Alon and A. Shapira. Testing subgraphs in directed graphs. *JCSS*, Vol. 69, pages 354–482, 2004.

[4] N. Alon and A. Shapira. Every Monotone Graph Property is Testable. *SIAM Journal on Computing*, Vol. 38, pages 505–522, 2008.

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

[6] M. Blum, M. Luby and R. Rubinfeld. Self-Testing/Correcting with Applications to Numerical Problems. *JCSS*, Vol. 47, No. 3, pages 549–595, 1993.

[7] A. Bogdanov, K. Obata, and L. Trevisan. A lower bound for testing 3-colorability in bounded-degree graphs. In *43rd FOCS*, pages 93–102, 2002.

[8] L. Gishboliner and A. Shapira. A Generalized Turan Problem and its Applications. *ECCC*, TR18-007, 2018.

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

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

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

[12] 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.

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

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

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

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il