Testing Isomorphism in the Bounded-Degree Graph Model

Oded Goldreich*

February 8, 2025

Abstract

We consider two versions of the problem of testing graph isomorphism in the bounded-degree graph model: A version in which one graph is fixed, and a version in which the input consists of two graphs. We essentially determine the query complexity of these testing problems *in the special case of n-vertex graphs with connected components of* $\mathrm{poly}(\log n)$ *size.* This is done by showing that these problems are computationally equivalent (up to polylogarithmic factors) to corresponding problems regarding isomorphism between sequences (over a large alphabet). Ignoring the dependence on the proximity parameter, our main results are:

1. The query complexity of testing isomorphism to a fixed object (i.e., an n-vertex graph or an n-long sequence) is $\widetilde{\Theta}(n^{1/2})$.
2. The query complexity of testing isomorphism between two input objects is $\widetilde{\Theta}(n^{2/3})$.

Testing isomorphism between two sequences is shown to be related to testing that two distributions are identical, and this relation yields reductions in three of the four relevant cases. Failing to reduce the problem of testing the equality of two input distributions to the problem of testing isomorphism between two input sequences, we adapt the proof of the lower bound on the complexity of the first problem to the second problem. This adaptation constitutes the main technical contribution of the current work.

We stress that determining the complexity of testing graph isomorphism (in the bounded-degree graph model), in the general case (e.g., for expander graphs), is left open.

A preliminary version of this work was posted as TR19-102 of *ECCC*. The current version eliminates some inaccuracies and hand-wavings that appeared in the preliminary version. In general, the exposition is more detailed.

1 Introduction

We consider the problem of testing graph isomorphism in the bounded-degree graph model (introduced in [11] and reviewed in [9, Chap. 9])). We actually considered two versions of the problem: In one version, called the fixed graph version, the input is a single graph and the task is testing whether this graph is isomorphic to a fixed graph (which "massively parametrized" the property); in the other version, one is given two input graphs and the task is testing whether they are isomorphic.[1] (Both versions of the problem were considered before, but mostly in different testing models (see Section 1.3).)

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

[1] The graphs $G_1 = ([n], E_1)$ and $G_2 = ([n], E_2)$ are isomorphic if there exists a bijection $\pi : [n] \to [n]$ (called an isomorphism) such that $\{\pi(u), \pi(v)\} \in E_2$ if and only if $\{u, v\} \in E_1$.

The bounded-degree graph model. Recall that, in the bounded-degree graph model, graphs are represented by their incidence functions and distances between graphs are measured accordingly. Specifically, for a fixed degree bound d, a graph $G = ([n], E)$ of maximum degree at most d is represented by a function $g : [n] \times [d] \to [[n]]$, where $[[n]] = \{0, 1, ..., n\} = [n] \cup \{0\}$, such that $g(v, i) = u \in [n]$ if u is the i^{th} neighbor of v (in G), and $g(v, i) = 0$ if v has less than i neighbors.

The graph $G = ([n], E)$ is said to be ϵ-far from the graph $G' = ([n], E')$ if the symmetric difference between E and E' is larger than $\epsilon dn/2$ (equiv., if any representations $g : [n] \times [d] \to [[n]]$ and $g' : [n] \times [d] \to [[n]]$ of G and G' differ on more than ϵdn entries (i.e., $|\{(v, i) : g(v, i) \neq g'(v, i)\}| > \epsilon dn$)). Otherwise, the graphs are ϵ-close. The graph $G = ([n], E)$ is said to be ϵ-far from a graph property Π (i.e., a set of graphs that is closed under isomorphism) if G is ϵ-far from any graph in Π.

We say that an oracle machine is an ϵ-tester of Π if, when given oracle access to an incidence function of the graph, it distinguishes between the case that the graph is in Π and the case that the graph is ϵ-far from Π (i.e., it accepts with probability at least $2/3$ in the first case and rejects with probability at least $2/3$ in the second case). Indeed, testing isomorphism to a fixed graph H is the task of testing the property that consists of the set of graphs that are isomorphic to H (i.e., H is a massive parameter that specifies the property).[2]

The complexity of testing a graph property Π (in the bounded degree graph model) is measured in terms of the degree bound d, the number of vertices n, and the proximity parameter ϵ. In this work we ignore the dependency on the degree bound, which is assumed to be a constant, and when presenting lower bounds we assume that $\epsilon > 0$ is an arbitrary small constant. That is, saying "testing Π requires Q queries" means that *for some $\epsilon > 0$, any ϵ-tester of Π requires Q queries.*

1.1 Testing Isomorphism Between Graphs with Small Connected Components

With the foregoing preliminaries in place, we can state our first main result.

Theorem 1.1 (testing isomorphism to a fixed graph (in the bounded-degree graph model)): *Let Φ_n denote the set of bounded-degree n-vertex graphs that consists of connected components that are each of size at most* $\text{poly}(\log n)$*. Then, for every $H \in \Phi_n$, the query complexity of ϵ-testing isomorphism to H is at most $\widetilde{O}(n^{1/2}/\epsilon^2)$, whereas for almost all $H \in \Phi_n$ the query complexity is at least $\widetilde{\Omega}(n^{1/2})$.*

Indeed, we leave open the question of what is the query complexity of testing isomorphism to H, in the general case. The upper bound on the size of the connected components implies that the query complexity of a potential tester is in the same ball-park as the number of connected components that this tester visits. A logarithmic lower bound on the size (of the connected components in H) will be used to guarantee that these connected components may be pairwise non-isomorphic (and even pairwise far from being isomorphic, see Lemma 2.3). This fact will be used when lower-bounding the query complexity of testing isomorphism.

Turning to the problem of testing isomorphism between two given graphs, we extend the testing framework to the case in which the potential tester is given a pair of input oracles. Specifically, we say that an oracle machine is an ϵ-tester for isomorphism between two graphs if, when given oracle access to incidence functions of the two graphs, it distinguishes between the case that the graphs

[2] See [9, Sec. 12.7.2] for a brief discussion of "massively parametrized" properties.

are isomorphic and the case that the first graph is ϵ-far from any graph that is isomorphic to the second graph.

Theorem 1.2 (testing isomorphism between two input graphs (in the bounded-degree graph model)): *Let Φ_n be the set of n-vertex graphs that consist of connected components that are each of size at most* $\text{poly}(\log n)$*. The query complexity of ϵ-testing isomorphism between two graphs that are promised to be in Φ_n is at most $\widetilde{O}(n^{2/3}/\epsilon^2)$ and at least $\widetilde{\Omega}(n^{2/3})$. Equivalently, the query complexity of ϵ-testing that two graphs are isomorphism to one another and are in Φ_n is at most $\widetilde{O}(n^{2/3}/\epsilon^2)$ and at least $\widetilde{\Omega}(n^{2/3})$. Furthermore, the time complexity is also $\widetilde{O}(n^{2/3}/\epsilon^2)$.*

The stated equivalence is due to the fact that membership in Φ_n can be ϵ-tested in time $O(\epsilon^{-1} \cdot \text{poly}(\log n))$. Again, we leave open the question of what is the query complexity of testing isomorphism between two graphs, in the general case. We comment that in the context of *one-sided error testing*, even the fixed-graph version requires linear query complexity (see Theorem 2.6).

1.2 Techniques: Testing Isomorphism Between Sequences

Theorems 1.1 and 1.2 are proved by showing the computational equivalence of these two graph-testing problems to corresponding problems of testing isomorphism between sequences (over a large alphabet), and obtaining bounds for the latter problems.

We say that two sequences, $\overline{\sigma} = (\sigma_1, ..., \sigma_n)$ and $\overline{\tau} = (\tau_1, ..., \tau_n)$, are isomorphic if there exists a bijection $\pi : [n] \to [n]$ such that $\tau_{\pi(j)} = \sigma_j$ for every $j \in [n]$. One corresponding problem refers to testing isomorphism to a fixed sequence, and the other problem refers to testing isomorphism between two sequences (equiv., two parts of a single sequence), where we say that two n-long sequences are ϵ-far if they differ on more than ϵn symbols; hence, $(\sigma_1, ..., \sigma_n)$ and $(\tau_1, ..., \tau_n)$ are ϵ-far from being isomorphic if $|\{j \in [n] : \sigma_j \neq \tau_{\pi(j)}\}| > \epsilon \cdot n$ for every bijection $\pi : [n] \to [n]$. When the studying the testing of such sequences, we consider the natural model in which the sequence $\overline{\sigma} \in \Sigma^n$ is viewed as a function $\overline{\sigma} : [n] \to \Sigma$, and query $i \in [n]$ to $\overline{\sigma}$ is answered by $\overline{\sigma}(i)$.

As stated above, Theorems 1.1 and 1.2 follow by presenting reductions between the graph-testing tasks and the corresponding sequence-testing tasks, and establishing (rather tight) results regarding the complexity of the sequence-testing tasks. Intuitively, we associated the (relatively small) connected components of the graphs with symbols in the sequences; that is, each possible symbol encodes a different unlabelled graph of relatively small size. The key observation is that graph isomorphism reduces to counting the number of occurrences of each unlabeled graph (as a connected component) in each of the two graphs, whereas sequence isomorphism reduces to counting the number of occurrences of each symbol in each of the two sequences. Needless to say, we need the connected components to be small enough so that ignoring their size in the complexity bounds is reasonable, while, on the other hand, we need them to be big enough so that we can have sufficiently many different (actually, far apart) unlabeled graphs. For details see Section 2.

Turning to the sequence testing problems, we essentially determine the query complexity of both problems. Specifically, ignoring the dependence on ϵ, the complexity of testing isomorphism to a fixed n-long sequence is $\Theta(n^{1/2})$, whereas the complexity of testing isomorphism between two n-long sequences is $\Theta(n^{2/3})$. More accurately, we prove the following two results.

Theorem 1.3 (testing isomorphism to a fixed sequence): *Fixing any Σ and n, for every $\overline{\sigma} \in \Sigma^n$ and $\epsilon > 0$, the query complexity of ϵ-testing isomorphism to $\overline{\sigma}$ is $O(n^{1/2}/\epsilon^2)$. On the other hand,* *if $|\Sigma| = \Omega(n)$, then for almost all $\overline{\sigma} \in \Sigma^n$, including $\overline{\sigma} = (1, ..., n)$, the query complexity of testing isomorphism to $\overline{\sigma}$ is $\Omega(n^{1/2})$.*

Note that the lower bound requires a large alphabet (i.e., $|\Sigma| = \Omega(n)$), whereas ϵ-testing isomorphism to a fixed sequence over a *constant-sized* alphabet has query complexity $O(1/\epsilon^2)$. The same holds with respect to the following result.

Theorem 1.4 (testing isomorphism between two input sequences): *Fixing any Σ and n, the time complexity of ϵ-testing isomorphism between n-long sequences over Σ is $O(n^{2/3}/\epsilon^2)$, provided that symbols in Σ can be compared in unit time. On the other hand, if $|\Sigma| = \Omega(n)$, then the query complexity of testing isomorphism between n-long sequences over Σ is $\Omega(n^{2/3})$.*

The proof of the lower bound of Theorem 1.4 is the main technical contribution of this work, but before discussing it we briefly sketch the other three proofs.

On proving Theorem 1.3. Both bounds are proved by observing that the sequence-testing problem is computationally equivalent to testing the identity of a given distribution to a fixed distribution, where the distribution-tester is given samples of the tested distribution. The fixed distribution is assigned the value v with probability $|\{i \in [n] : \sigma_i = v\}|/n$, and samples from the tested distribution correspond to the contents of the tested sequence at random location. (Note that, without loss of generality, we may assume that the sequence-tester queries the sequence at random locations, but without repetitions.) The gap between sampling with repetitions, which is available to the distribution-tester, and sampling without repetitions, which is available to the sequence-tester, can be ignored when establishing the $\Omega(n^{1/2})$ lower bound. (Given these reductions, Theorem 1.3 follows from the results surveyed in [9, Sec. 11.2].)[3]

On proving Theorem 1.4. While the upper bound follows easily by reducing the sequence-testing task to the distribution-testing task (very much as in the case of testing isomorphism to a fixed sequence), we failed to find a reduction in the opposite direction. The source of difficulty is the gap between sampling with and without repetitions, where here we cannot ignore this gap because we are considering $\omega(n^{1/2})$ random samples taken in the probability space $[n]$. The main technical contribution of this work is overcoming this difficulty.

Instead of reducing the distribution-testing problem to the sequence-testing problem, we adapt Valiant's [21] proof of an $\Omega(n^{2/3})$ lower bound for the distribution testing problem to the sequence-testing setting. The easy part is showing that it suffices to consider a "canonical" sequence-tester that rules according to the pattern of collisions among the oracle answers, while ignoring both the locations and values of the collisions. (This replaces an analogous statement regarding collisions in the samples given to the distribution-tester.) Next, we show that, for some YES and NO-instances, the pattern of collisions seen by a $o(n^{2/3})$-query tester for the sequence problem are statistically close. This is shown by "reducing" the *analysis* of the collision patterns seems by the sequence-tester to those analyzed (for distribution-testing) by Valiant [21].

Specifically, we transform the probability space that underlies the samples that are considered by Valiant (where random samples are drawn with repetitions) to a probability space that fits the sequence-testing setting (where random samples are drawn without repetitions). This transformation retains a tiny fraction of the original probability space and the resulting distribution of the

[3] For a list of credits, which starts with [12, 4], see [9, Sec. 11.5.1].

collision patterns is different from the original distribution of the collision patterns. Still, we show that, with high probability, the difference does not occur in places that matter.

More concretely, Valiant [21] (following Batu *et al.* [3]) considers pairs of probability distributions with heavy and light elements such that in the NO-case the two distributions agree on heavy elements but disagree on light elements.[4] He shows that the collision pattern of the light elements (which is significantly different in the two cases (i.e., the cases of YES-instances and NO-instances)) is "masked" by the collision pattern of the heavy elements. Our transformation preserves this masking effect: It *does change the collision pattern of heavy elements*, but does so *in an identically manner in the two cases* (and in a way that is oblivious of the light elements); furthermore (with high probability), the transformation *does not affect the collision pattern that involve light elements.* Hence, our transformation does not (significantly) increase the statistical difference between the collision patterns seen (by the potential tester) in the two cases (whereas in one case the tester has to accept but in the other case it has to reject).

1.3 Related Work

The two versions of the graph isomorphism testing problem were considered before [8, 16, 18], but mostly in different models. One exception that we are aware of is the work of Newman and Sohler [18], who considered testing properties of hyperfinite graphs in the bounded-degree graph model. Towards that end, they proved that testing isomorphism between two such graphs has complexity that only depends on the proximity parameter (i.e., ϵ); see [18, Thm. 3.2].

A few years earlier, Fischer and Matsliah [8] studied the query complexity of both versions of the graph isomorphism testing problem *in the dense graph model* (introduced in [10] and reviewed in [9, Chap. 8]). Interestingly, in all cases they considered, the complexity is sublinear (in the number of vertex-pairs, but polynomially related to that number). In particular, isomorphism between two n-vertex input graphs can be tested with one-sided error using $\widetilde{O}(n^{3/2})$ queries, and (two-sided error) testing of isomorphism to some fixed n-vertex graph requires $\widetilde{\Omega}(n^{1/2})$ queries.

More recently, Kusumoto and Yoshida [16] studied these testing problems in the "adjacency list model" (actually, in the general graph model (introduced in [19, 15] and reviewed in [9, Chap. 10])). They considered the case that the graphs are promised to be forests (or, alternatively, are required to be forests as part of the property). They showed that in this case the query complexity is polylogarithmic in the size of the graph.

The issue of *sampling with versus without repetitions* (a.k.a *with versus without replacement*) arose also in the work of Raskhodnikova *et al.* [20]. The context there was approximating the number of distinct elements in a sequence, and in that context they presented a reduction of $O(\alpha)$-factor approximation based on $O(s)$ samples with repetitions to α-factor approximation based on s samples without repetitions.

1.4 Subsequent Work

Considering the problem of testing isomorphism to a fixed graph, Goldreich and Tauber [14] showed that, for almost all d-regular n-vertex graphs H, testing isomorphism to H can be done using $\widetilde{O}(\sqrt{n})$ queries. This result is shown to be optimal (up to a polylog factor) by a matching lower bound, which also holds for almost all graphs H.

[4] Needless to say, the YES-instances consist of pairs of identical distributions.

The performance of their tester depends on natural graph parameters of the fixed (n-vertex) graph H such as its diameter and the minimum radius of "distinguishing neighborhoods" (i.e., the minimum $r = r(n)$ such that the "r-neighborhoods" of the n different vertices are pairwise non-isomorphic).

1.5 Organization and Notation

As stated in Section 1.2, our main results are proved by showing the computational equivalence of the graph isomorphism testing problems and the corresponding problems of testing isomorphism between sequences (over a large alphabet). This equivalence is shown in Section 2, and Section 3 focuses on the complexity of the sequence isomorphism testing problems. In particular, the lower bound on the complexity of testing isomorphism between two input sequences is proved in Section 3.2.

Notation. For a graph $G = ([n], E)$ and a bijection $\pi : [n] \to [n]$, we let $\pi(G)$ denotes the graph obtained from G by relabeling the vertices according to π; that is, $\{\pi(u), \pi(v)\}$ is an edge of $\pi(G_1)$ if and only if $\{u, v\}$ is an edge of G. Indeed, $G = ([n], E)$ and $G' = ([n], E')$ are isomorphic if there exists a bijection $\pi : [n] \to [n]$ such that $G' = \pi(G)$ (i.e., $E' = \{\{\pi(u), \pi(v)\} : \{u, v\} \in E\}$).

2 Graph Isomorphism Versus Sequence Isomorphism

In this section we show the computational equivalence of the graph isomorphism testing problems to corresponding problems of testing isomorphism between sequences (over a large alphabet). Specifically, we shall focus on (n-vertex) graphs that have small connected components (i.e., each of poly$(\log n)$-size) and on n-long sequences over the alphabet $[n]$. The fact that the connected components are small will be used when reducing the graph-testing problems to the corresponding sequence-testing problems. Specifically, the overhead of the reduction (presented in Section 2.1) is linearly related to the size of the connected components. When reducing in the opposite direction, we shall use a bijection between $[n]$ and a collection of n 3-regular $O(\log n)$-vertex expander graphs that are pairwise far from being isomorphic to one another. (Straightforward constructions of such a collection of gadgets are spelled out in Section 2.2.) Using this bijection we reduce the sequence isomorphism testing problems to the graph isomorphism testing problems (see Section 2.3). Lastly, in Section 2.4, we prove a linear lower bound on the query complexity of *one-sided error testers* for the various isomorphism problems.

2.1 Reducing the Graph Problems to the Sequence Problems

Our aim in reducing the graph problems to the sequence problems is making a step towards obtaining algorithms (i.e., testers) for the graph problems. Actually, we present two such reductions: The first reduction refers to graph problems in which all connected components are of equal size, and the second reduction refers to the case that all connected components are of bounded size.

Reductions in the context of property testing should preserve sublinear complexities as well as distances between the objects and the corresponding properties. We refrain from presenting here an adequate notion of a reduction, while regretting that the treatment in [9, Sec. 7.4] is not general enough for the current application.[5] Instead, we use the most generic notion possible, which merely asserts that *if one testing problem is solvable within some complexity then the other is solvable within related complexity.*

Proposition 2.1 (obtaining testers for the graph problems, equal size version): *Let Φ_n be the set of n-vertex graphs that consist of connected components that are each of size $s = s(n)$, and let d be the degree bound used in the bounded-degree graph model. Then, for every $\epsilon > 0$, the following holds.*

1. *If ϵ-testing isomorphism to any fixed (n/s)-long sequence has query complexity is at most q, then, for every n-vertex graph H in Φ_n, the query complexity of ϵ-testing isomorphism to H is at most $ds \cdot q$.*

2. *If ϵ-testing isomorphism between two (n/s)-long sequences has query complexity is at most q, then ϵ-testing isomorphism between two n-vertex graphs that are promised to be in Φ_n has query complexity $ds \cdot q$. Furthermore, if the time complexity of the first problem is T, then the time complexity of the second problem is $\text{poly}(s) \cdot T$.*

The graph-testing problems refer to the bounded-degree model, and the sequence-testing problems can be restricted to the alphabet $[2n/s]$.

Regarding the conclusion (in Part 2), since testing Φ_n is easy, we can remove the promise (in the conclusion) and test the property that consists of pairs of isomorphic graphs that are in Φ_n. On the hypothesis side, we can easily reduce testing m-long sequences over $[2m]$ to testing $2m$-long sequences over $[2m]$.

Proof: The fixed object case (i.e., Part 1) follows as a special case of the two-object case (i.e., Part 2), when allowing free oracle access to one of the objects. Focusing on the two-object case, we first assume that the sequences-tester works for sequences over any alphabet.

The basic idea is viewing the (s-vertex) connected components of the two input (n-vertex) graphs as symbols in two corresponding (n/s-long) sequences, while noting that the locations of the symbols in a sequence are immaterial (for the sequence isomorphism problem), just as the labels of vertices are immaterial for the graph-testing problem. Hence, our graph-tester invokes the guaranteed sequence-tester and answers its queries by finding and describing connected component that have not been used for that purpose before. Specifically, whenever the sequence-tester makes a new query, we select uniformly at random a vertex that was not visited so far, explore the connected component in which it resides, and answer the query with a description of the corresponding graph (either as an unlabeled graph or as a canonically labeled graph with vertex set $[s]$). All vertices in this explored connected component are marked as *visited*, and will not be selected when answering subsequent queries. We stress that isomorphic copies of the same s-vertex graph (potentially appearing as connected components in the tested graphs) are mapped to the same symbol.

Construction 2.1.1 (the reduction): *Considering an auxiliary sequence-tester that queries all locations after the actual tester has terminated, we obtain the following mapping of graphs in Φ_n to n/s-long sequences over unlabelled s-vertex graphs: A graph that, for every i, has t_i connected*

[5] The main difficulty arises in the reduction that is presented at the end of the proof of Proposition 2.1 (i.e., the (alphabet) reduction from n-long sequences over $[s^{ds}]$ to n-long sequences over $[2n]$). This reduction generates the reduced instance "on the fly" in a way that depends on the history of the execution (of the reduction) so far.

components that are isomorphic to the unlabeled graph H_i is mapped to a random n/s-long sequence that contains t_i occurrences of the symbol H_i.

The furthermore clause (of Part 2), referring to the running time of the resulting tester, is based on the fact that canonical labeling of bounded-degree graphs can be found in polynomial-time [2]. That is, the mapping of s-vertex connected components to symbols representing the set of all isomorphic copies (of the corresponding graph) can be implemented in poly(s)-time.

The analysis boils down to observing that both problems reduce to testing equality between the number of elements of each type that occur in the tested objects. In the case of graphs, these elements are the n/s connected components of the n-vertex graph and the types are the isomorphism classes, whereas in the case of sequences the elements are the n/s different locations and the types are the symbols. (We note that the distance between (unlabeled) graphs is upper-bounded by the distance between the (multi-sets of symbols that appear in the) corresponding sequences.)[6] Formally we prove the following.

Claim 2.1.2 (distance preservation): *Suppose that $G_1 \in \Phi$ is ϵ-far from being isomorphic to $G_2 \in \Phi_n$, and let S_1 and S_2 denote the corresponding sequences generated by the reduction of Construction 2.1.1. Then, S_1 is ϵ-far from being isomorphic to S_2.*

Needless to say, isomorphic graphs are mapped to isomorphic sequences.

Proof: We prove the contrapositive; that is, assuming that the sequence S_1 is ϵ-close to being isomorphic to S_2, we show that G_1 is ϵ-close to being isomorphic to G_2. Let $\pi : [n/s] \to [n/s]$ denote the former isomorphism, and let $\phi_b : [n/s] \to [n/s]$ be a bijection such that the i^{th} connected component of G_b is mapped by the reduction to the $\phi_b(i)^{\text{th}}$ location in S_b. Then, the i^{th} connected component of G_1 is isomorphic to the $\phi_2^{-1}(\pi(\phi_1(i)))^{\text{th}}$ connected component of G_2 if the $\phi_1(i)^{\text{th}}$ location of S_1 contains the same symbol as the $\pi(\phi_1(i))^{\text{th}}$ location of S_2. It follows that at least a $1-\epsilon$ fraction of the connected components of G_1 are isomorphic to corresponding connected components of G_2, which implies that $\phi_2^{-1}(\pi(\phi_1(G_1)))$ is ϵ-close to G_2. □

Alphabet reduction. The foregoing description refers to an alphabet that consists of all (unlabelled) s-vertex graphs (of maximum degree d). We now show that the sequence testing problem for m-long sequences over arbitrary alphabets reduces to the corresponding problem for the alphabet $[2m]$. Essentially, we invoke the tester T for the special case and answer its queries by querying our own oracle and maintaining the list of symbols viewed so far. If our oracle answers with a symbol that was not viewed so far, then we answer with a random value in $[2m]$ that was not used by us so far, and record the symbol and the selected value. On the other hand, if our oracle answers with a symbol that was viewed before, then we answer with the same element of $[2m]$ that was used at that time. Hence, when given access to the two m-long sequences over Σ, where Σ is the set of symbols that *actually* occur in these sequences (i.e., $|\Sigma| \leq 2m$), we *effectively* invoke T on corresponding sequences that are obtained by applying a random 1-to-1 mapping $\psi : \Sigma \to [2m]$ to the two original sequences (i.e., the sequence $(\sigma_1, ..., \sigma_m)$ is mapped to the sequence $(\psi(\sigma_1), ..., \psi(\sigma_m))$). ■

The bounded size version. Recall that Proposition 2.1 referred to n-vertex graphs consisting of connected components that are each of size *exactly* $s = s(n)$. In contrast, Proposition 2.2 refers

[6] In fact, this upper bound is typically not tight, but this only works in our favor, because it means that we may reduce far apart graphs to sequences that are even more far apart.

to n-vertex graphs consisting of connected components that are each of size *at most* $s = s(n)$. This difference requires a somewhat different reduction. In particular, we now reduce the graph problems to problems regarding n-long sequences, rather than to problems regarding n/s-long sequences.

Proposition 2.2 (obtaining testers for the graph problems, bounded size version): *Let Φ_n be the set of n-vertex graphs that consist of connected components that are each of size at most $s = s(n)$, and let d be the degree bound used in the bounded-degree graph model. Then, for every $\epsilon > 0$, the following holds.*

1. *If ϵ-testing isomorphism to any fixed n-long sequence has query complexity is at most q, then, for every n-vertex graph H in Φ_n, the query complexity of ϵ-testing isomorphism to H is at most $ds \cdot q$.*
2. *If ϵ-testing isomorphism between two n-long sequences has query complexity is at most q, then ϵ-testing isomorphism between two n-vertex graphs that are promised to be in Φ_n has query complexity $ds \cdot q$. Furthermore, if the time complexity of the first problem is T, then the time complexity of the second problem is $\mathrm{poly}(s) \cdot T$.*

The graph-testing problems refer to the bounded-degree model, and the sequence-testing problems can be restricted to the alphabet $[2n]$.

Recall that we can remove the promise and test the property that consists of pairs of isomorphic graphs that are both in Π, and that we can easily reduce testing m-long sequences over $[2m]$ to testing $2m$-long sequences over $[2m]$.

Proof: The proof strategy is similar to the one employed in the proof of Proposition 2.1, but differs from it in its actual details. First, let us see why the reduction presented in Construction 2.1.1 should not be employed as is. For starters, note that, in Part 2 of the current setting, the number of connected components in the tested graphs is not known *a priori*; it may be anything between n/s and n. More importantly, connected components of different size have a different contribution to the distance between graphs, whereas when measuring the distance between sequences all symbols are equal.

Intuitively, both difficulties are resolved by associating n-vertex graphs (with connected components of size at most s) with n-long sequences (initially over $[\exp(\widetilde{O}(s))]$)[7] such that a connected component of size $s' \leq s$ is associated with s' locations in the sequence. Whenever the sequence-tester makes a new query, we answer it by selecting at random a vertex that was not *selected* before (rather than not *visited* before), exploring the connected component in which it resides, and returning the corresponding canonically labeled graph as well as the *canonical location* of vertex in this graph.

Indeed, for every $s' \in [s]$, we use a canonical labeling of s'-vertex graphs (cf. [2]). For such an s'-vertex graph H and for any graph G' that is isomorphic to it, we consider a canonical isomorphism that maps G' to H. (Note that if H is asymmetric, then there exists a unique isomorphism that maps G' to H, but otherwise there are several such isomorphisms; we fix a single one (based on G' and H), and call it canonical.) The canonical location of a vertex v (which resides in G') in H is the vertex to which the canonical isomorphism maps v.

[7] Recall that we latter reduce the problems regarding m-long sequences over arbitrary alphabet to the corresponding problems regarding m-long sequences over $[2m]$.

Construction 2.2.1 (the reduction): *Considering an auxiliary sequence-tester that queries all locations after the actual tester has terminated, we obtain the following mapping of graphs in Φ_n to n-long sequences over pairs consisting of labeled graphs and locations in them: The n-vertex graph $G = ([n], E)$ is mapped to a random n-long sequence such that for every $v \in [n]$ there exists a unique location in the sequence, denoted $\phi(v)$, that contains a symbol that encodes (C_v, i_v), where C_v is the canonical labeling of the connected component in which v resides and i_v is the canonical location of v in this labeled graph.*

Recall that the canonical labelling of bounded-degree graphs as well as canonical isomorphism can be found in polynomial time.[8] Analogously to Claim 2.1.2 , we have

Claim 2.2.2 (distance preservation): *Suppose that $G_1 \in \Phi$ is ϵ-far from being isomorphic to $G_2 \in \Phi_n$, and let S_1 and S_2 denote the corresponding sequences generated by the reduction of Construction 2.2.1. Then, S_1 is ϵ-far from being isomorphic to S_2.*

Again, isomorphic graphs are mapped to isomorphic sequences.

Proof: We prove the contrapositive; that is, assuming that the sequence S_1 is ϵ-close to being isomorphic to S_2, we show that G_1 is ϵ-close to being isomorphic to G_2. Let $\pi : [n] \to [n]$ denote the former isomorphism, and let $\phi_b : [n] \to [n]$ be a bijection such that the i^{th} vertex of G_b is mapped by the reduction to the $\phi_b(i)^{\text{th}}$ location in S_b. Then, the i^{th} vertex of G_1 reside in a connected component that is isomorphic to connected component that contains the $\phi_2^{-1}(\pi(\phi_1(i)))^{\text{th}}$ vertex of G_2 if the $\phi_1(i)^{\text{th}}$ location of S_1 contains the same symbol as the $\pi(\phi_1(i))^{\text{th}}$ location of S_2. Furthermore, in this case (i.e., vertices mapped to the same symbol), both vertices (i.e., the i^{th} vertex of G_1 and the $\phi_2^{-1}(\pi(\phi_1(i)))^{\text{th}}$ vertex of G_2) occupy the same canonical location in the corresponding connected components.

Note, however, that this does not mean that all vertices in the former connected component (of G_1) are mapped to the same connected component (of G_2), where here we refer to vertices that the reduction maps to locations that contain equal symbols (equiv., u and v such that $(C_u, i_u) = (C_v, i_v)$). We correct this state of affairs by relocating vertices among isomorphic connected components, while maintaining their canonical location. It follows that the subgraph induced by at least a $1 - \epsilon$ fraction of the vertices of G_1 is isomorphic to an induced subgraph of G_2, which implies that $\phi_2^{-1}(\pi(\phi_1(G_1)))$ is ϵ-close to G_2. □

Applying alphabet reduction, as in the proof of Proposition 2.1, the current proposition follows. ■

2.2 A Collection of Graphs that are Pairwise Far From Being Isomorphic

When reducing the sequence problems to the graph problems, we shall use a collection of small graphs that are pairwise far from being isomorphic. Specifically, these small graphs will serve as potential connected components in the graph problems and will encode symbols of the sequence problems. As one may expect, a random collection of $\exp(\Omega(k))$ 3-regular k-vertex graphs will do for our purposes. (In fact, we get an even larger collection, whereas a collection of $\exp(k^{\Omega(1)})$ gadgets would have sufficed too.)

[8] For the former we may use the algorithm of [2], whereas the latter can be found by a process of self-reduction (in which one may find the lexicographically-first isomorphism that maps a graph G' to a graph H).

Lemma 2.3 *For every sufficiently large even $k \in \mathbb{N}$, there exists a collection of $\exp(\Omega(k \log k))$ 3-regular k-vertex graphs that are pairwise $\Omega(1)$-far from being isomorphic. Furthermore, with overwhelmingly high probability, a random collection of $\exp(\Omega(k \log k))$ 3-regular k-vertex graphs satisfies the condition.*

Recall that, with probability $1 - o(1)$, a random 3-regular k-vertex graph is an expander; that is, every set of $k' < k/2$ vertices neighbors at least $\Omega(k')$ vertices outside it [7].

Proof: Our starting point is Bollobas's estimate for the number of labeled d-regular k-vertex graphs [6], which is

$$N_d(k) \stackrel{\rm def}{=} e^{-c-c^2} \cdot \frac{(dk)!}{(dk/2)! \cdot 2^{dk/2} \cdot (d!)^k} \tag{1}$$

where $c = (d-1)/2$ (and dk is even and $d = o(\log k)^{1/2}$). Using a rough approximation, we have

$$\begin{aligned} N_d(k) &\approx e^{-c-c^2} \cdot \frac{(dk/e)^{dk}}{(dk/2e)^{dk/2} \cdot 2^{dk/2} \cdot (d!)^k} && (2) \\ &= e^{-c-c^2} \cdot \frac{(dk/e)^{dk/2}}{(d!)^k} && (3) \end{aligned}$$

Hence, for any constant $d \geq 1$, we have $N_d(k) = \Omega(k)^{dk/2}$.

We prove the existence of the claimed collection by using a greedy algorithm. For some small constant $\epsilon > 0$, at each step, we select an arbitrary graph that is not ϵ-close to being isomorphic to any of the graph already selected. The point is that the number of graphs that are ϵ-close to being isomorphic to a fixed k-vertex graph is at most $M_d(k) \stackrel{\rm def}{=} k! \cdot \binom{dk}{\epsilon dk} \cdot k^{\epsilon dk} \ll k^{k+(\epsilon+o(1)) \cdot dk}$. Hence, we can select $N_d(k)/M_d(k)$ graphs, and the claim follows, since $N_3(k)/M_3(k) \gg k^{k/3}$ for $\epsilon < 1/18$ and sufficiently large k. In fact, with overwhelmingly high probability, selecting a random set of $k^{k/4}$ (d-regular k-vertex) graphs will do. ■

Remark 2.4 (large set expanders): *The fact that, with very high probability, almost all the graphs in the random collection that satisfies Lemma 2.3 are expanders suffices for our main results.*[9] *We get slightly more appealing results by observing that, with very high probability, a random collection of $\exp(\Omega(k))$ 3-regular k-vertex graphs contains only* large set expanders *in which the expansion condition holds for every set of $k' \in [\Omega(k), k/2]$ vertices* (rather than for every $k' \in [k/2]$). *The point is that a random 3-regular k-vertex graph is an expander with probability $1 - k^{-\Theta(1)}$, whereas it is a large-set-expander with probability $1 - \exp(-\Omega(k))$, where the hidden constant in the Ω-notation depends on the constant used in the definition of a large set.*[10]

[9] Specifically, it suffices for all results stated in the introduction with the exception that the lower bound of Theorem 1.1 holds only for exponentially many 3-regular graphs H rather than for almost all 3-regular graphs.

[10] Specifically, fixing $\alpha, \gamma \in (0, 0.5)$, consider the probability that a 3-regular k-vertex graph contains a set of $k' \in [\alpha \cdot k, 0.5 \cdot k]$ vertices that neighbors less than $\gamma \cdot k'$ vertices outside it. A crude upper bound for the number of such graphs is given by

$$\begin{aligned} B(k) &= \sum_{k' \in [\alpha k, 0.5k]} \binom{k}{k'} \cdot N_3(k') \cdot N_3(k-k') \cdot \sum_{k'' \in [[\gamma k']]} \binom{3k'}{3k''} \cdot \binom{3(k-k')}{3k''} \\ &\ll \sum_{k' \in [\alpha k, 0.5k]} \binom{k}{k'} \cdot N_3(k) \cdot \left((k'/k)^{3k'/2} \cdot ((k-k')/k)^{3(k-k')/2} \right) \cdot \binom{3k}{6\gamma k'} \end{aligned}$$

2.3 Reducing the Sequence Problems to the Graph Problems

Our aim in reducing the sequence problems to the graph problems is making a step towards obtaining lower bounds on the complexity of the graph-testing problems.

In this case, the basic formalism of [9, Def. 7.13] suffices for capturing the relevant reductions. Loosely speaking, a q-local (ϵ, ϵ')-reduction of Π to Π' is a mapping of objects of the first type (i.e., the type of Π) to objects of the second type (i.e., the type of Π') that satisfies the following three conditions:

1. Locality (local reconstruction): The value of an object in the image of the reduction at any point is determined by the value of the preimage at q points; that is, if the reduction maps f to f', then the value of f' at any point is determined by the value of f at q points.

2. Preservation of the properties: The reduction maps objects in Π to objects in Π'.

3. Partial preservation of distance to the properties: An object that is ϵ-far from Π is mapped to an object that is ϵ'-far from Π'.

It follows that if Π' can be ϵ'-tested within query complexity Q', then Π can be ϵ-tested within query complexity $q \cdot Q'$ (see [9, Thm. 7.14]). Indeed, q is the overhead of the reduction. We shall actually use the contrapositive (i.e., if Π cannot be ϵ-tested within query complexity Q, then Π' cannot be ϵ'-tested within query complexity Q/q).

Proposition 2.5 (towards deriving lower bounds on the graph-testing problems): *Let Φ_n be the set of $n \cdot s$-vertex graphs that consist of connected components that are each of size $s = s(n) \geq \log_2 n$, and let $d \geq 3$ be the degree bound used in the bounded-degree graph model. Then, for every $\epsilon > 0$, the following holds.*

1. *Testing isomorphism to any fixed n-long sequence over $[n]$ is 1-locally $(\epsilon, \Omega(\epsilon))$-reducible to testing isomorphism to a fixed 3-regular $s \cdot n$-vertex graph in Φ_n.*

   Furthermore, for almost all 3-regular $s \cdot n$-vertex graph H in Φ_n, testing isomorphism to the sequence $(1, 2..., n)$ is 1-locally $(\epsilon, \Omega(\epsilon))$-reducible to testing isomorphism to H.

2. *Testing isomorphism between two n-long sequences over $[n]$ is 1-locally $(\epsilon, \Omega(\epsilon))$-reducible to testing isomorphism between two 3-regular $s \cdot n$-vertex graphs that are promised to be in Φ_n.*

The graph-testing problems refer to the bounded-degree model.

Proof: For some fixed constant $\delta > 0$, let C be a collection of n graphs, each being a 3-regular s-vertex expander[11], that are pairwise δ-far from being isomorphic to one another. (The existence

$$\approx \sum_{k' \in [\alpha k, 0.5k]} 2^{H_2(k'/k) \cdot k} \cdot N_3(k) \cdot 2^{-H_2(k'/k) \cdot 1.5k} \cdot 2^{H_2(6\gamma k'/3k) \cdot 3k}$$

$$= \sum_{k' \in [\alpha k, 0.5k]} 2^{-0.5 \cdot H_2(k'/k) \cdot k + 3 \cdot H_2(2\gamma k'/k) \cdot k} \cdot N_3(k),$$

where the sharp inequality uses $N_3(m) \approx (em/d)^{dm/2}$ and H_2 is the binary entropy function. Using a sufficiently small $\gamma > 0$, the claim holds (for any $\alpha \in (0, 0.5]$).

[11] Recall that an s-vertex graph is an expander if any set of $s' \leq s/2$ vertices in it neighbors at least $\Omega(s')$ vertices outside this set.

of such a collection is guaranteed by Lemma 2.3 and [6].) Fixing a bijection $\psi : [n] \to C$, we spell out the claimed reduction, while focusing on the case of two input-objects, and viewing the fixed-object case as a special case (in which oracle access to the first object is for free).

The reduction maps each n-long sequence over $[n]$, viewed as a function $\overline{\sigma} : [n] \to [n]$, to an $n \cdot s$-vertex graph G in Φ such that the $i^{\rm th}$ connected component in G is isomorphic to the s-vertex graph $\psi(\overline{\sigma}(i))$. The vertices of this connected component are labeled $(i-1) \cdot s+1, ..., i \cdot s$. Hence, a query for the neighbor of vertex $(i-1) \cdot s + j$, where $(i,j) \in [n] \times [s]$, is answered by querying the $i^{\rm th}$ location in $\overline{\sigma}$, and determining the corresponding neighbor of the $j^{\rm th}$ vertex in $\psi(\overline{\sigma}(i))$. That is, the sequence-tester obtains the value $\overline{\sigma}(i)$, applies the mapping ψ to this value, considers the canonical labeling of the resulting s-vertex graph, and answers according to the neighborhood of the $j^{\rm th}$ vertex.

Note that if the input sequences are isomorphic, then the corresponding graphs are isomorphic. On the other hand, as shown next, if the input sequences, denoted S_1 and S_2, are ϵ-far from being isomorphic, then the corresponding graphs, denoted G_1 and G_2, are $\Omega(\epsilon)$-far from being isomorphic.

We prove the contrapositive. Suppose that the first graph is ϵ'-close to an isomorphic copy of the second graph, and let $\pi' : [ns] \to [ns]$ denote the mapping that witnesses this fact (i.e., $\pi'(G_1)$ is ϵ'-close to G_2). Let us assume first, for simplicity, that π' *maps connected components of* G_1 *to connected components of* G_2. In such a case, we define a bijection $\pi : [n] \to [n]$ such that the $i^{\rm th}$ connected component of G_1 is mapped by π' to the $\pi(i)^{\rm th}$ connected component of G_2; that is,

$$\{(\pi(i)-1) \cdot s + j : j \in [s]\} = \{\pi'((i-1) \cdot s + j) : j \in [s]\}. \tag{4}$$

Recall that the garphs in C are pairwise δ-far from being isomorphic to one another.

Claim 2.5.1 (warm-up claim): *Let π be as in Eq.* (4)*. Then, for at least a $1-(\epsilon'/\delta)$ fraction of the i's in $[n]$, it holds that the $i^{\rm th}$ connected component of G_1 is isomorphic to the $\pi(i)^{\rm th}$ connected component of G_2.*

It follows that the sequence $S_1 \circ \pi$ (i.e., the sequence whose $i^{\rm th}$ symbol is $S_1(\pi(i))$) is (ϵ'/δ)-close to S_2, which implies that S_1 is (ϵ'/δ)-close to an isomorphic copy of S_2.

Proof: Letting δ_i denote the (relative) distance between the $i^{\rm th}$ connected component of G_1 and the set of graphs that are isomorphic to the $\pi(i)^{\rm th}$ connected component of G_2, we have $\frac{1}{n} \cdot \sum_{i \in [n]} \delta_i \leq \epsilon'$. Using an averaging argument, it follows that, for at least a $1-(\epsilon'/\delta)$ fraction of the i's in $[n]$, the $i^{\rm th}$ connected component of G_1 is δ-close to an isomorphic copy of the $\pi(i)^{\rm th}$ connected component of G_2. Recalling that (1) all the connected components of G_1 and G_2 are in C, and that (2) different graphs in C are δ-far from being isomorphic, it follows that *the connected components that are δ-close to being isomorphic are actually isomorphic copies of the same graph* (in C). The claim follows. □

Recall that the foregoing analysis was based on the simplifying assumption that π' maps connected components of G_1 to connected components of G_2. This is not necessarily the case, but the fact that the connected components are expanders can be used to show that this is essentially the case.

We say that a connected component of G_1 is effectively preserved by $\pi' : [sn] \to [sn]$ if at least a $1-0.5\delta$ fraction of its vertices are mapped by π' to the same connected component of G_2; that is, the $i^{\rm th}$ connected component of G_1 is effectively preserved by π' if for some $t \in [n]$ it holds that $|\{\pi'((i-1) \cdot s + j) : j \in [s]\} \cap \{(t-1) \cdot s + j : j \in [s]\}| \geq (1-0.5\delta) \cdot s$. Assume, for a moment, that at least a $1-\gamma\epsilon'$ fraction of the connected components of G_1 are effectively preserved by π', where the constant $\gamma > 1$ will be determined later.

Claim 2.5.2 (main claim): *Suppose that at least a $1-\gamma\epsilon'$ fraction of the connected components of G_1 are effectively preserved by π'. Then, there exists a bijection $\pi : [n] \to [n]$ such that for at least a $1-\gamma\epsilon' - (\epsilon'/0.5\delta)$ fraction of the i's in $[n]$ it holds that the i^{th} connected component in G_1 is δ-close to an isomorphic copy of the $\pi(i)^{\text{th}}$ connected component in G_2.*

Using Claim 2.5.2 (and recalling that graphs in C are δ-far apart), it follows that for at least a $1-(\gamma+(2/\delta))\cdot\epsilon'$ fraction of the i's in $[n]$ it holds that the i^{th} connected component in G_1 is isomorphic to the $\pi(i)^{\text{th}}$ connected component in G_1. Hence, S_1 is $(\gamma+(2/\delta))\cdot\epsilon'$-close to an isomorphic copy of S_2 (again, by virtue of the sequence-relocation mapping π).

Proof: Indeed, π is any bijection that fits the preservation condition of π'; that is, the connected components of G_1 that are effectively preserved by π' are mapped by π to the corresponding connected components of G_2. Hence, if the i^{th} connected component of G_1 is effectively preserved by π', then $|\{\pi'((i-1)\cdot s+j) : j \in [s]\} \cap \{(\pi(i)-1)\cdot s + j : j \in [s]\}| \geq (1-0.5\delta)\cdot s > s/2$. Now, essentially, the claim holds because otherwise the distance between G_1 and its image under π' is greater than $(\epsilon'/0.5\delta)\cdot(\delta-0.5\delta)$, where the first factor accounts for the fraction of effectively preserved components that are δ-far from the corresponding mapped components, and the second factor accounts for the fraction of mapped vertex-incidences in which these components differ. Details follow.

Let I denote the set of i's such that the i^{th} connected component in G_1 is effectively preserved and is δ-far from an isomorphic copy of the $\pi(i)^{\text{th}}$ connected component in G_2. Assuming towards the contradiction that $|I| > (\epsilon'/0.5\delta)\cdot n$, recall that for each $i \in I$ it holds that π' maps at least $1-0.5\delta$ fraction of the vertices of the i^{th} component of G_1 to the $\pi(i)^{\text{th}}$ component of G_2. Hence, for each $i \in I$, the incidences of vertices in the i^{th} component differ from the incidences of their image under π' in at least $\delta\cdot 3s - 3\cdot 0.5\delta\cdot s$ entries (i.e., the number of incidence-differences between the components minus the number of incidences that belong to vertices of the i^{th} component that were not mapped to the $\pi(i)^{\text{th}}$ component). It follows that, under π', the incidence functions of the two graphs differ in $|I|\cdot 3\delta s/2 > \frac{\epsilon'\cdot n}{0.5\delta}\cdot\frac{3\delta s}{2} = \epsilon'\cdot 3ns$ entries, which contradicts the hypothesis that π' witnesses a distance of at most ϵ' towards graph isomorphism (i.e., $\pi'(G_1)$ is ϵ-close to G_2). □

Getting rid of the assumption. The foregoing analysis relied on the assumption that at least a $1-\gamma\epsilon'$ fraction of the connected components of G_1 are effectively preserved by the witness mapping π'. However, this assumption is actually a fact (i.e., it must hold), *provided the constant γ is chosen to be sufficiently large* (i.e., $\gamma \geq 1/\Omega(\delta)$). This is the case because otherwise expansion (w.r.t the non-preserved components of G_1)[12] implies that $\pi'(G_1)$ is $\gamma\epsilon'\cdot\Omega(\delta)$-far from G_2, in contradiction to the hypothesis regarding π' (i.e., that $\pi'(G_1)$ is ϵ'-close to G_2). This establishes the main claims of the proposition and leaves us with the furthermore claim of Part 1, which refers to the case that the fixed graph is uniformly selected among the 3-regular graphs in Φ (i.e., that the fixed graph is uniformly selected among all 3-regular graphs that have n connected components that is each of size s).

[12] Recall that each non-preserved connected components of G_1 is mapped by π' to several connected components of G_2 such that no connected component of G_2 contains at least a $1-0.5\delta$ fraction of the vertices of the original component (of G_1). Hence, at least a $\Omega(\delta)$ fraction of the edges of the original component of G_1 are mapped to non-edges in G_2.

Proving the furthermore claim of Part 1. Recall that the furthermore claim (of Part 1) refers to reducing from the problem of testing isomorphism to the fixed sequence $(1, 2, ..., n)$, whereas the instance produced by reduction refers to a fixed $n \cdot s$-vertex graph H that consists of the connected components $\psi(1), ..., \psi(n)$, where $\psi : [n] \to C$ is a bijection and C is a fixed collection of s-vertex graphs. Specifically, C consists of n expander graphs that are pairwise far from being isomorphic, whereas by Lemma 2.3 a random set of n graphs satisfies the first condition. The problem is that a random collection C of n graphs contains $n - o(n)$ expanders, whereas we have assumed that all graphs in C are expanders.[13] However, a closer look at the foregoing argument reveals that it holds even if all graphs in C are only large-set-expanders (in the sense defined in Remark 2.4), because we only relied on the expansion of sets of density at least 0.5δ. Since random s-vertex graphs are large-set-expanders with high probability $1 - \exp(-\Omega(s))$, the furthermore claim follows. ■

2.4 Implications for Testing Graph Isomorphism

Using the first (resp., second) part of Propositions 2.2 and 2.5, the claims of Theorem 1.1 (resp., Theorem 1.2) follow from Theorem 1.3 (resp., Theorem 1.4). Indeed, this uses $s = \text{poly}(\log n)$. A more general statement is presented in Section 4.

One-sided error testing. Recall that a one-sided error tester is required to *always* accept any object that has the property (i.e., accept with probability 1 any such object), while rejecting with high probability (e.g., with probability at least 2/3) any object that is far from the property, as a usual tester. It is quite easy to see that all testing problems considered in this work have no one-sided error tester of sublinear query complexity. We prove this assertion for the problem of testing isomorphism to a fixed graph (with extremely small connected components).

Theorem 2.6 (a linear lower bound on the query complexity of one-sided testers for the fixed-graph version): *There exists an n-graph H of degree bound two and connected components of size three such that, in the bounded-degree graph model, any one-sided error tester of isomorphism to H requires $\Omega(n)$ queries.*

Proof: Let H be composed of $n/6$ isolated triangles and $n/6$ isolated 2-paths, and assume (w.l.o.g.) that the tester for isomorphism decides based on the number of isolated triangles and 2-paths that it sees.[14] Then, a one-sided error tester for isomorphism to H that makes at most $n/6$ queries must always accept when seeing any proportion of isolated triangles and 2-paths, since such a proportion may occur when querying an isomorphic copy of H. It follows that such a tester (always) accepts when inspecting a graph composed of $n/9$ isolated triangles and $2n/9$ isolated 2-paths, whereas such a graph is $\Omega(1)$-far from the property. ■

[13] Note that, with probability $1/\text{poly}(s)$, a random 3-regular s-vertex graph is not even connected, whereas we have $s = \text{poly}(\log n)$.

[14] This claim is analogous to [13, Clm. 5.5.2], which asserts that any q-query tester can be emulated by selecting q random vertices and inspecting their q-neighborhoods (i.e., the vertices that are at distance at most q from a selected vertex). Here, however, it suffices to consider 2-neighborhoods. That is, if the tester is guaranteed that the input is composed of connected components that are each of size 3, then we may assume that it merely samples random vertices and inspects their respective connected components.

3 On the Complexity of Testing Isomorphism Between Sequences

In this section we focus on the two versions of the sequence-testing problem, and establish Theorems 1.3 and 1.4. We shall often view n-long sequences over Σ as functions from $[n]$ to Σ.

3.1 Testing Isomorphism to a Fixed Sequence

We start with the problem of testing isomorphism to a fixed sequence, which serves as a good warm-up towards our study of the complexity of testing isomorphism between two input sequences.

Theorem 3.1 (the query complexity of testing isomorphism to a fixed sequence (Theorem 1.3, restated)):

1. *For every $\epsilon > 0$, the query complexity of ϵ-testing isomorphism to any fixed n-long sequence is $O(n^{1/2}/\epsilon^2)$. Furthermore, the time complexity is the same if the tester can determine the number of occurrences of a given symbol in the fixed sequence in constant time.*

2. *Testing isomorphism of an n-long sequence over $[n]$ to the sequence $(1, 2, ..., n)$ requires $\Omega(n^{1/2})$ queries. The same holds with respect to almost all n-long sequences over $[n]$.*

Of course, testing isomorphism to a fixed sequence may be easier in some cases (e.g., the all-1 sequence). In fact, the following reduction that establishes the upper bound yields such results via the results of [22] (see also [5]).

Proof: Both parts rely on the relation of testing isomorphism to a fixed and testing identity to fixed distributions that are grained, where a distribution is called n-grained if all elements in its support appear with probability that is a multiple of $1/n$ (cf. [9, Def. 11.7]). Specifically, although these testing problems refer to very different models, we show that each of them is easily reducible to the other.

Proving the upper bound (i.e., Part 1). Here we use a reduction of testing *isomorphism to a fixed n-long sequence* to testing *identity to a fixed n-grained distribution.* Specifically, given a fixed sequence $\overline{\sigma} : [n] \to \Sigma$ and access to an input sequence $\overline{\tau} : [n] \to \Sigma$, we define the fixed distribution D and the input distribution X such that $\Pr[D = v] = |\{i \in [n] : \overline{\sigma}(i) = v\}|/n$ and $\Pr[X = v] = |\{i \in [n] : \overline{\tau}(i) = v\}|/n$ for every $v \in \Sigma$. We then invoke the guaranteed distribution-tester (for the distribution D (which is determined by $\overline{\sigma}$)), and provide it with samples of X in the obvious matter; that is, if we need to provide s samples, then we select uniformly and independently $i_1, ..., i_s \in [n]$, and use the values $\overline{\tau}(i_1), ..., \overline{\tau}(i_s)$. (The straightforward analysis of this reduction is presented in the first paragraph of the proof of Theorem 3.2.)

Using the known distribution-testers (see [9, Thm. 11.11]), the claimed upper bound follows. The time bound (asserted in the furthermore clause) follows by observing that the reduction of *testing identity to a fixed n-grained distribution* to *testing that a distribution is uniform on* $[n]$ only requires an evaluation oracle to the fixed distribution, and makes one query to this oracle per each example obtained from the input distribution (see [9, Sec. 11.2.2.1]).

Proving the lower bound (i.e., Part 2). Here we use a reduction in the opposite direction; specifically, we reduce testing *whether an n-grained distribution is uniform over* $[n]$ to testing *isomorphism to the fixed sequence* $(1, 2, ..., n)$. We capitalized on the fact that we are proving a lower bound of the form $\Omega(n^{1/2})$, since in that regime the difference between sampling indices in $[n]$ *without repetitions* and sampling them *with repetitions* can be ignored.

The latter comment is crucial because a q-query sequence-tester can be assumed to query the input sequence at q *distinct* random locations; that is, the set of q queried locations is distributed uniformly among all q-subsets of $[n]$ (i.e., it samples without repetitions). In contrast, the distribution-tester obtains samples of some (n-grained) distribution X, which may be thought of as being generated by selecting uniformly $i \in [n]$ and outputting a value $G(i)$ for a suitable function $G : [n] \to [n]$; that is, these samples are generated independently of one another (i.e., they are generated with repetitions). Details follow.

We reduce testing whether an n-grained distribution X over $[n]$ is uniform to testing isomorphism to the fixed sequence $\overline{\sigma} = (1, 2, ..., n)$. The point is that the input distribution X can be viewed as generated by selecting at random $i \in [n]$ and outputting the value $G(i)$ for an adequate $G : [n] \to [n]$; that is, $\Pr[X = e] = |\{i \in [n] : G(i) = e\}|/n$. So testing whether X is uniform over $[n]$ corresponds to testing whether G viewed as a sequence is isomorphic to $\overline{\sigma}$. (Recall that this distribution-testing problem has complexity $q = \Omega(\sqrt{n})$, even when restricted to n-grained distributions [9, Cor. 11.13].) In the reduction, given a tester T for isomorphism to the fixed sequence $\overline{\sigma}$, we construct a distribution-tester that invokes T and answers T's (distinct, w.l.o.g.) queries by using the samples provided to it. That is, for every $j \in [q]$, the j^{th} query is answered by the j^{th} sample.

In the analysis, we think of the q samples given to our tester as being generated by selecting $i_1, ..., i_q \in [n]$ uniformly and independently (with repetitions), and being presented with $G(i_1), ..., G(i_q)$. In contrast, without loss of generality, we can think of T as selecting uniformly a sequence of q distinct elements in $[n]$ and querying its oracle for their value.[15] That is, whereas T should be given the answers $G(i_1), ..., G(i_q)$ such that $i_1, ..., i_q$ are selected uniformly in $[n]$ *without repetitions*, we gave it corresponding answers with respect to $i_1, ..., i_q$ that are selected uniformly in $[n]$ *with repetitions*. However, the statistical difference between these two q-long sequences of samples is at most $\binom{q}{2}/n$, and we can ignore this difference when establishing a lower bound of the form $q = \Omega(\sqrt{n})$.

The foregoing establishes the lower bound for the n-long sequence $(1, 2, ..., n)$; that is, it establishes the main claim of Part 2. The secondary claim (i.e., referring to almost all n-long sequences) follows by observing that all is used in [9, Cor. 11.13] is the hypothesis that the collision probability of the fixed distribution is $O(1/n)$. Note that the collision probability of the distribution associated with $\overline{\sigma} \in [n]^n$ is $\sum_{i\in[n]}(n_i/n)^2$, where $n_i = |\{j \in [n] : \sigma_j = i\}|$. Observing that, for a uniformly distributed $\overline{\sigma} \in [n]^n$, the expected value of $\sum_{i\in[n]} \binom{n_i}{2}$ equals $\binom{n}{2} \cdot 1/n$, it follows that for almost all $\overline{\sigma} \in [n]^n$ the corresponding n_i's satisfy $\sum_{i\in[n]} n_i^2 = O(n)$. The claim follows. $\blacksquare$

3.2 Testing Isomorphism Between Two Input Sequences

We now turn to the problem of testing isomorphism between two input sequences, while adopting slightly different notation than the one used so far. The first three paragraph recap ideas that were already presented in Section 3.1.

[15] Given an arbitrary tester T that tests $\overline{\tau} : [n] \to [n]$, suppose that we answer its j^{th} query with $\overline{\tau}(i_j)$, where $(i_1, ..., i_q)$ is uniformly distributed among all q-long sequences of *distinct* elements in $[n]$. Then, we actually emulate an execution of T in which it is given access to a random isomorphic copy of $\overline{\tau}$ (i.e., to the function $\overline{\tau} \circ \pi$, where $\pi : [n] \to [n]$ is a uniformly distributed bijection).

Given two sequences $S_1, S_2 \in \Sigma^n$, presented as functions $S_1, S_2 : [n] \to \Sigma$, the sequence isomorphism testing problem is to determine whether there exists a permutation $\pi : [n] \to [n]$ such that $S_1(j) = S_2(\pi(j))$ for every $j \in [n]$ (i.e., $S_1 = S_2 \circ \pi$) or S_1 is far from $S_2 \circ \pi$ for every permutation π. This can be captured as a property testing problem by considering $S : \{1, 2\} \times [n] \to \Sigma$ such that $S(i, j) = S_i(j)$.

Clearly, the sequence isomorphism testing problem is reducible the testing equality between (n-grained) distributions, by considering random variables X_1 and X_2 such that $\Pr[X_i = \sigma] = |\{j \in [n] : S_i(j) = \sigma\}|/n$ for every $\sigma \in \Sigma$. Hence, the complexity of testing sequence isomorphism is upper-bounded by the complexity of testing equality between distributions, which is $O(1/\epsilon^2) \cdot n^{2/3}$ (see [9, Sec. 11.3], presenting the best result known, which in turn improves over earlier work of Batu *et al.* [3]).

It is tempting to hope that a reduction in the opposite direction holds if we restrict the distributions to be n-grained, where a distribution is n-grained if each element in its support is assigned a probability mass that is a multiple of $1/n$. Indeed, one can show that, without loss of generality, a tester of sequence-isomorphism queries the sequences at a random set of location, but the distribution-tester obtains a sequence of samples that corresponds to a random multi-set of locations. That is, we face a gap between sampling with and without repetitions, and this gap matters because we are interested in the case that the number of samples is larger than the square root of the support size.

We were unable to show a reduction of the distribution-testing problem to the sequence-testing problem, which would have allowed for inferring a lower bound on the sequence-testing problem from a lower bound on the distribution-testing problem; but we were able to adapt the proof of the known $\Omega(n^{2/3})$ lower bound (of Valiant [21]) for the distribution-testing problem to the sequence-testing problem.

Theorem 3.2 (on the complexity of sequence isomorphism (Theorem 1.4, restated)):

1. *For every $\epsilon > 0$, the time complexity of ϵ-testing sequence isomorphism* (for n-long sequences) *is $O(n^{2/3}/\epsilon^2)$, provided that symbols can be compared in unit time.*

2. *Testing sequence isomorphism for n-long sequences over $[n]$ requires $\Omega(n^{2/3})$ queries.*

Proof: The upper bound follows by the reduction outlined above; that is, given sequences $S_1, S_2 : [n] \to \Sigma$, consider the random variables X_1 and X_2 such that $\Pr[X_i = \sigma] = |\{j \in [n] : S_i(j) = \sigma\}|/n$. If S_1 is isomorphic to S_2, then $X_1 \equiv X_2$. On the other hand, if S_1 is ϵ-far from being isomorphic to S_2 (i.e., $\min_{\pi \in \mathrm{Sym}_n}\{|\{j \in [n] : S_1(j) \neq S_2(\pi(j))\}|\} > \epsilon \cdot n$), then $\sum_\sigma |\#_\sigma(S_1) - \#_\sigma(S_2)| > 2 \cdot \epsilon n$, where $\#_\sigma(S)$ denotes the number occurrences of σ in the sequence S (i.e., $\#_\sigma(S) = |\{j \in [n] : S(j) = \sigma\}|$).[16] Hence, ϵ-testing of sequence isomorphism reduces to ϵ-testing of identity of distributions, by virtue of invoking the distribution-tester and providing it with samples of X_1 (resp., X_2) by querying S_1 (resp., S_2) at random locations. Using the distribution-tester that works in time $O(1/\epsilon^2) \cdot n^{2/3}$ (see [9, Sec. 11.3]), the theorem's upper bound follows.

Turning to the lower bound, we adapt Valiant's proof [21] of the lower bound for the distribution-testing problem into a lower bound for the sequence-testing problem. We first mimic the relatively easy argument that shows that all that matters is the distribution of "histograms" seen by the tester. We then turn to the actual challenge, which is showing that the relevant histograms (of

[16] It may be easier to see that the distance of S_1 from being isomorphic to S_2 equals $\sum_\sigma \max(0, \#_\sigma(S_1) - \#_\sigma(S_2))$.

some YES and NO-instances) seen by a $o(n^{2/3})$-query tester for the sequence problem are statistically close. This is done by "reducing" the *analysis* to the case analyzed by Valiant [21], and constitutes the core of the proof.

Histograms. The histograms that we define next refer to a pair of sequences; these sequences are not the input sequences (or distributions) but rather sequences of samples that the tester obtains from these sequences (or distributions). (Indeed, this generalizes the more basic notion of a histogram of a single sequence.)[17]

Definition 3.2.1 (the relevant histograms): *For a pair of m-long sequences $((s_1,...,s_m),(s'_1,...,s'_m)) \in \Sigma^{m+m}$, the corresponding* histogram *is an $(m+1)$-by-$(m+1)$ matrix $H=(h_{t,t'})_{t,t'\in[[m]]}$ such that $h_{t,t'}$ is the number of symbols that occur exactly t times in the first sequence and t' times in the second sequence; that is,*

$$h_{t,t'} \;=\; \left|\left\{\sigma\in\Sigma:\ \#_\sigma(s_1,...,s_m)=t\ \ \&\ \ \#_\sigma(s'_1,...,s'_m)=t'\right\}\right|, \tag{5}$$

$$\text{where } \#_\sigma(\sigma_1,...,\sigma_m)=|\{j\in[m]:\sigma_j=\sigma\}|. \tag{6}$$

(Indeed, we use the notation $[[m]] \overset{\rm def}{=} \{0,1,...,m\}$.)

Note that $\sum_{t,t'} h_{t,t'} = |\Sigma|$, since each $\sigma\in\Sigma$ contributes (exactly one unit) to a single $h_{t,t'}$, and $\sum_{t,t'} h_{t,t'}\cdot(t+t') = 2m$, since each location $j\in[m+m]$ in one of the two sequences (resp., occurrence of a symbol in one of the two sequences) is counted once in the sum.

Claim 3.2.2 (histograms are all that matters): *If the isomorphism of n-long sequences over Σ can be tested within query complexity $q=q(n,\Sigma,\epsilon)$, then it can be tested by a* canonical tester *that obtains the values of each of the two sequences in q random positions and rules according to the corresponding histogram. In other words, when testing the sequences $S_1,S_2:[n]\to\Sigma$, the canonical tester rules according to the histogram of $((S_1(j_1),...,S_1(j_q)),(S_2(k_1),...,S_2(k_q)))$, where $(j_1,...,j_q)$ and $(k_1,...,k_q)$ are distributed uniformly and independently in the set of q-long sequences of* distinct *elements in $[n]$.*[18]

Proof: For sake of clarity, we proceed in two steps (where the first step details an argument already used before (see Footnote 15)).

Given an arbitrary tester T as in the hypothesis, and fixing (n,Σ,ϵ) and $q=q(n,\Sigma,\epsilon)$, we first construct an algorithm T' that obtains, as input, a pair of q-long sequences, denoted $((s_1,...,s_q),(s'_1,...,s'_q))$, invokes T, while answering its $j^{\rm th}$ query to the first (resp., second) sequence with s_j (resp., s'_j), and outputs the verdict of T. When analysing T', we consider, for any two sequences $S_1,S_2:[n]\to\Sigma$, what happens when selecting $\overline{j}=(j_1,...,j_q)$ and $\overline{k}=(k_1,...,k_q)$ uniformly and independently among all possible q-long sequences of distinct elements in $[n]$, and feeding T' with $((S_1(j_1),...,S_1(j_q)),(S_2(k_1),...,S_2(k_q)))$. We observe that in this case the output of T' is distributed identically to the output of T when given oracle access to random isomorphic copies of S_1 and S_2 (i.e., to the oracles $S_1\circ\pi_1$ and $S_2\circ\pi_2$, where π_1 and π_2 are uniformly and independently distributed permutations of $[n]$); that is,

$$\Pr_{\overline{j},\overline{k}}\left[T'((S_1(j_1),...,S_1(j_q)),(S_2(k_1),...,S_2(k_q)))=1\right] = \Pr_{\pi_1,\pi_2}\left[T^{S_1\circ\pi_1,S_2\circ\pi_2}=1\right]. \tag{7}$$

[17] A histogram of a sequence $\overline{s}=(s_1,...,s_m)$ is a sequence $(h_0,h_1,...,h_m)$ such that h_t equals the number of elements that occur exactly t times in $\overline{s}$ (i.e., $h_t=|\{\sigma\in\Sigma:\#_\sigma(\overline{s})=t\}|$, where $\#_\sigma(s_1,...,s_m)=|\{j\in[m]:s_j=\sigma\}|$).

[18] That is, the set $\{(i_1,...,i_q)\in[n]^q:|\{i_1,...,i_q\}|=q\}$.

Hence, when fed with random samples (without repetitions) of the two sequences, T' distinguishes between the case that S_1 is isomorphic to S_2 and the case that S_1 is ϵ-far from being isomorphic to S_2.

Next, we present the desired canonical tester, denoted T''. On input a (valid) histogram, denoted $H = (h_{t,t'})_{t,t' \in [[q]]}$, algorithm T'' selects at random a pair of q-long sequences that fits the histogram H, feeds it to T', and outputs its verdict. That is, T'' selects uniformly at random a pair of q-long sequences $((s_1, ..., s_q), (s'_1, ..., s'_q)) \in \Sigma^{q+q}$ such that

$$|\{\sigma \in \Sigma : \#_\sigma(s_1, ..., s_q) = t \ \& \ \#_\sigma(s'_1, ..., s'_q) = t'\}| \ = \ h_{t,t'}$$

holds for every $t, t' \in [[q]]$, and outputs $T'((s_1, ..., s_q), (s'_1, ..., s'_q))$. Then, for any two sequences $S_1, S_2 : [n] \to \Sigma$, the output of T'' when given a *histogram* of a sequence of q distinct random values in S_1 and q distinct random values in S_2 equals the output of T' when given the corresponding samples (*themselves*!) from $\psi(S_1)$ and $\psi(S_2)$, where ψ is a random permutation of Σ; that is, letting $H_q^{S_1,S_2}$ represent the distribution of the histogram of q distinct samples from S_1 and q distict samples from S_2, we have

$$\Pr\left[T''(H_q^{S_1,S_2}) = 1\right] = \Pr_{\overline{j},\overline{k},\psi}\left[T'((\psi(S_1(j_1)), ..., \psi(S_1(j_q))), (\psi(S_2(k_1)), ..., \psi(S_2(k_q)))) = 1\right]. \quad (8)$$

Hence, $T''(H^{S_1,S_2})$ distinguishes between the case that S_1 is isomorphic to S_2 and the case that S_1 is ϵ-far from being isomorphic to S_2.

(Indeed, our argument only relies on the fact that T distinguishes between the case that $\psi \circ S_1 \circ \pi_1$ is isomorphic to $\psi \circ S_2 \circ \pi_2$ and the case that $\psi \circ S_1 \circ \pi_1$ is ϵ-far from being isomorphic to $\psi \circ S_2 \circ \pi_2$, where $\pi_1, \pi_2 : [n] \to [n]$ and $\psi : \Sigma \to \Sigma$ are random permutations (as above).)[19] □

The core of the proof. We wish to show that a canonical sequence-tester must make $\Omega(n^{2/3})$ queries in order to distinguish between the case that it is provided with a histogram that corresponds to a YES-instance (i.e., isomorphic sequences) and the case that it is provided with a histogram that corresponds to a NO-instance (i.e., sequences that are far from being isomorphic). Specifically, for carefully chosen sequences S_1 and S_2, the YES-instance will consists two copies of the sequence S_1, whereas the NO-instance will consist of the sequences S_1 and S_2. The foregoing sequences S_1 and S_2 will correspond to the n-grained distributions analyzed by Valiant [21], which we will denote X_1 and X_2.

Each of the two distributions used in Valiant's proof [21] has $n/4$ elements of individual probability weight $2/n$ (called light), and $n^{2/3}$ elements each of weight $n^{-2/3}/2$ (called heavy). Furthermore, the proof refers to NO-instances that are pairs of distributions that agree on the heavy elements, but are disjoint on their light elements. We de-construct these *n-grained* distributions by specifying the underlying probability space, which equals $[n]$, and the way this space is mapped to values, which are either heavy or light. In the following description, the set of heavy (resp., light) elements is denoted H (resp., L).

- Fixing two disjoint sets H and L such that $|H| = n^{2/3}$ and $|L| = n/4$, we consider a mapping of the probability space $[n]$ to these sets such that each element in H is assigned $n^{1/3}/2$ elements (in the probability space) and each element in L is assigned two elements.

[19] Specifically, combining Eq. (7) and Eq. (8), we have

$$\Pr\left[T''(H_q^{S_1,S_2}) = 1\right] = \Pr_{\pi_1,\pi_2,\psi}\left[T^{\psi \circ S_1 \circ \pi_1, \psi \circ S_2 \circ \pi_2} = 1\right].$$

We denote this mapping by $G : [n] \to H \cup L$; that is, for each $h \in H$ (resp., $\ell \in L$) it holds that $|\{j \in [n] : G(j) = h\}| = n^{1/3}/2$ (resp., $|\{j \in [n] : G(j) = \ell\}| = 2$). Indeed, for each $h \in H$ (resp., $\ell \in L$) it holds that $\Pr_{j\in[n]}[G(j) = h] = n^{-2/3}/2$ (resp., $\Pr_{j\in[n]}[G(j) = \ell] = 2/n$).

- Fixing a set L' of size $|L|$ that is disjoint of $H \cup L$, we consider a bijection $\phi : H \cup L \to H \cup L'$ that is invariant on H (i.e., $\phi(h) = h$ for every $h \in H$ (and $\phi(L) = L'$)). We shall consider the random variables (or distributions) X_1 and X_2 such that $X_1(j) = G(j)$ and $X_2(j) = \phi(G(j))$, where j is uniformly distributed in $[n]$.[20]

  (We shall later view X_1 and X_2 as values sampled from corresponding n-long sequences S_1 and S_2, where $S_1(j) = G(j)$ and $S_2(j) = \phi(G(j))$. Note that X_i represents the value of a uniformly distributed location in S_i, and that the statistical distance between X_1 and X_2 equals the relative Hamming distance between S_1 and S_2, which in turn equals $1/2$ (since $S_1(j) \neq S_2(j)$ if and only if $G(j) \in L$, and $|\{j \in [n] : G(j) \in L\}| = n/2$).)

Next, we define random variables (or distributions) that represent pairs of sequences of samples drawn either from two copies of X_1 or from X_1 and X_2.

- For a fixed $m = \Theta(n^{2/3})$, we denote by Y a pair of m-long sequences, each consisting of m *independent* samples of X_1; that is, using $[n]^{2m}$ as an undelying probability space, we have

  $$Y(j_1, ..., j_m, k_1, ..., k_m) = ((X_1(j_1), ..., X_1(j_m)), (X_1(k_1), ..., X_1(k_m))). \qquad (9)$$

  (Indeed, here $j_1, ..., j_m$ are distributed independently in $[m]$, and ditto for $k_1, ..., k_m$. This is the setting that is suitable for distribution-testing, and it was analyzed in [21]. Our challenge would be to move from this setting to the one in which $j_1, ..., j_m$ are distinct and ditto for $k_1, ..., k_m$ (as is suitable for sequence-testing).)

  Likewise, we denote by Z a pair of m-long sequences such that the first sequence consists of m independent samples of X_1 and the second sequence consists of m independent samples of X_2; that is,

  $$Z(j_1, ..., j_m, k_1, ..., k_m) = ((X_1(j_1), ..., X_1(j_m)), (X_2(k_1), ..., X_2(k_m))). \qquad (10)$$

  Hence, Y and Z differ only in the values assigned to light samples that occur in the second m-long sequence; that is, the $m+i^{\text{th}}$ element of $Y(j_1, ..., j_m, k_1, ..., k_m)$ differs from the $m+i^{\text{th}}$ element of $Z(j_1, ..., j_m, k_1, ..., k_m)$ if and only if $G(k_i) \in L$.

- The histograms of $Y = Y(j_1, ..., j_m, k_1, ..., k_m)$ and $Z = Z(j_1, ..., j_m, k_1, ..., k_m)$ are denoted $\mathtt{h}(Y)$ and $\mathtt{h}(Z)$, respectively.

  Recall that the $(t, t')^{\text{th}}$ entry in $\mathtt{h}(Y)$ (resp., $\mathtt{h}(Z)$) is the number of values σ that occur t times in $((X_1(j_1), ..., X_1(j_m))$ and t' times in $(X_1(k_1), ..., X_1(k_m))$ (resp., in $(X_2(k_1), ..., X_2(k_m))$).

The crucial fact, proved in [21], is that, for a sufficiently small $m = \Omega(n^{2/3})$, the (random variables representing the) histograms $\mathtt{h}(Y)$ and $\mathtt{h}(Z)$ are statistically close (say, are at total variation distance at most 0.1).

[20] Following Batu *et al.* [3], Valiant [21] uses $H = [n^{2/3}]$, $L = [0.5n + 1, ..., 0.75n]$, and $L' = [0.75n + 1, ..., n]$. This choice is, of course, immaterial. Also note that, for simplicity, we use the elements of the intermediate sets H and L also as final value (as reflected in the fact that ϕ is only applied in the generation of X_2 and that $\phi(H) = H$).

Our aim is to show that the foregoing fact continues to hold when the probability space (underlying these random variables) is restricted to pairs of m-long sequences of *distict elements*. Recall that the underling probability space for Y (and likewise for Z) is $[n]^{2m}$, whereas here we wish the underlying space to include only pairs of m-long sequences of distinct elements in $[n]$ (i.e., $(i_1, ..., i_m)$ is eligible if and only if $i_p \neq i_q$ for every $p \neq q$). Note that this restriction retains only a tiny portion of the original probability space, since $m = \Omega(n^{2/3}) \gg O(n^{1/2})$; nevertheless, we shall show that the corresponding histograms remain statistically close.

Indeed, the purpose of the foregoing restriction is to yield samples that are distributed as postulated for the sequence-tester. We shall extensively use the assumption that $m = c \cdot n^{2/3}$, for a sufficiently small constant $c > 0$.

Starting from two uniformly and identically distributed sequences $(j_1, ..., j_m), (k_1, ..., k_m) \in [n]^m$, we first observe that, with probability at least 0.99, there are no three-way collisions in $(j_1, ..., j_m)$, because the expected number of three-way collisions is $\binom{m}{3} \cdot (1/n)^2 < c^3 < 0.01$. Furthermore, with probability at least 0.99, there are at most $100 \cdot \binom{m}{2} \cdot (1/n) \ll n^{1/3}$ pairwise collisions in $(j_1, ..., j_m)$, because the expected number of pairwise collisions is $\binom{m}{2} \cdot (1/n)$. Ditto for $(k_1, ..., k_m)$. Conditioned on the foregoing case, for each collision $\{j_p, j_q\}$ (i.e., $j_p = j_q$ although $p \neq q$), we re-select at random (i.e., re-randomize) one of the colliding indices (where the index to be re-randomized is selected obliviously of the sequence $(k_1, ..., k_m)$; e.g., we re-randomize j_p if $p < q$). Note that the number of *potential* new collisions (between the re-randomized indices and all other indices) is upper-bounded by $\frac{m^2}{n} \cdot m < 0.01 \cdot n$, whereas each collision *actually* occurs with probability $1/n$. Hence, with probability 0.99, no new collision actually occurs, which means that the re-randomization yields an m-long sequence of distinct elements (which is uniformly distributed among all such sequences). We do the same for the sequence $(k_1, ..., k_m)$. Let us denote the resulting pair of m-long sequences by $((j'_1, ..., j'_m), (k'_1, ..., k'_m))$.

Construction 3.2.3 (re-randomization): *For any $(i_1, ..., i_m) \in [n]^m$, we consider a random process that outputs $(i'_1, ..., i'_m)$ such that i'_p is uniformly distributed in $[n]$ if $i_p \in \{i_{p+1}, ..., i_m\}$, and $i'_p = i_p$ otherwise* (i.e., if $i_p \notin \{i_{p+1}, ..., i_m\}$).

We stress that this re-randomization is applied separately to $(j_1, ..., j_m)$ and to $(k_1, ..., k_m)$. Recall that in the likely case (in which there are no three-way collisions in $(j_1, ..., j_m)$ (resp., in $(k_1, ..., k_m)$)), with high probability, the resulting $(j'_1, ..., j'_m)$ (resp., $(k'_1, ..., k'_m)$) is uniformly distributed among all m-long sequence of distinct elements in $[n]$, and in any case $(j'_1, ..., j'_m)$ and $(k'_1, ..., k'_m)$ are distributed independently of one another. Hence, *with high probability, the indices $((j'_1, ..., j'_m), (k'_1, ..., k'_m))$ are distributed as expected by the sequence-tester.*

The crucial *fact* is that the re-randomization of indices has a small effect on the statistical difference between the histograms of the modified Y and Z. Essentially, this is the case because the foregoing statistical difference changes only when the collision pattern among the values in L changes, which happens rarely (since less than $m' = 50m^2/n$ indices get re-randomized, whereas each of the $m' \cdot m$ potential value-collisions in L actually occurs with probability $2/n$).[21] The foregoing fact will be proved next, where $\Delta(.,.)$ denotes the statistical difference (a.k.a. total variation difference) between distributions.

Claim 3.2.4 (the effect of re-randomization): *For $(j_1, ..., j_m, k_1, ..., k_m)$ that is uniformly distributed in $[n]^{2m}$ and $(j'_1, ..., j'_m, k'_1, ..., k'_m)$ that is generated as in Construction 3.2.3, consider*

[21] Recall that $m^3/n^2 = c^3$ can be made arbitrarily small by a suitable choice of the constant $c > 0$.

the random variables

- $Y = Y(j_1, ..., j_m, k_1, ..., k_m)$,
- $Z = Z(j_1, ..., j_m, k_1, ..., k_m)$,
- $Y' = Y(j'_1, ..., j'_m, k'_1, ..., k'_m)$, *and*
- $Z' = Z(j'_1, ..., j'_m, k'_1, ..., k'_m)$.

Suppose that $m = c \cdot n^{2/3}$*, for a sufficiently small constant* $c > 0$ (e.g., $c = 1/50$ will do)*. Then,* $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z')) \leq \Delta(\mathtt{h}(Y), \mathtt{h}(Z)) + 0.05$*, where* $\mathtt{h}(.)$ *denotes the histogram of the distribution.*

Recalling that, for a sufficiently small constant $c > 0$, it holds that $\Delta(\mathtt{h}(Y), \mathtt{h}(Z)) < 0.05$ (cf. [21]), we get $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z')) < 0.1$.

Proof: Throughout the proof, we consider collisions among the *values* of the variables X_i, while recalling that collisions among the underlying indices are likely to be eliminated by the re-randomization.

The key observation is that $\mathtt{h}(Y)$ and $\mathtt{h}(Z)$ (and likewise $\mathtt{h}(Y')$ and $\mathtt{h}(Z')$) reflect the pattern of collisions among X_i-values, where Y (resp., Y') contains only X_1-values and Z (resp., Z') contains also X_2-values in its second part. Hence, only collisions between the X_1-values of indices of the first part (i.e., $j_1, ..., j_m$) and the X_i-values of indices of the second part (i.e., $k_1, ..., k_m$) contribute to $\Delta(\mathtt{h}(Y), \mathtt{h}(Z))$.[22] Furthermore, such a contribution (which arises from the difference between X_2 and X_1) occurs only for indices in $G^{-1}(L)$, since indices in $G^{-1}(H)$ are assigned the value of X_1 in both Y and Z. The same consideration applies to $j'_1, ..., j'_m$ and $k'_1, ..., k'_m$ regarding their contribution to $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z'))$. Hence, the difference between $\Delta(\mathtt{h}(Y), \mathtt{h}(Z))$ and $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z'))$ is due to re-randomization of indices j_p (resp., k_p) that reside in $G^{-1}(L)$ either initially or after re-randomization, and to the collision of their X_1-value (resp., X_2-value) with the X_2-value (resp., X_1-value) of some k_q or k'_q (resp., j_q or j'_q). Details follow.

Recalling that, with probability at least 0.98, there are no three-way collisions in $(j_1, ..., j_m)$ and there are at most $50m^2/n$ pairwise collisions in it, we consider the re-randomization applied to one index in each pair $(j_p, j_q) \in [m^2]$ such that $j_p = j_q$ (and $p \neq q$). The same analysis is applied to $(k_1, ..., k_m)$. Suppose that we re-randomized j_p, replacing it by a uniformly distributed j'_p. We consider four cases.

1. If $G(j_p) \in H$ and $G(j'_p) \in H$, then the replacing of j_p by j'_p does not contribute the difference between $\Delta(\mathtt{h}(Y), \mathtt{h}(Z))$ and $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z'))$, because these two statistical differences are only due to light indices. Details follow.

   Recall that $X_2(k) = X_1(k)$ if $G(k) \in H$, whereas $Y = (X_1(j_1), ..., X_1(j_m), X_1(k_1), ..., X_1(k_m))$ and $Z = (X_1(j_1), ..., X_1(j_m), X_2(k_1), ..., X_2(k_m))$. Therefore replacing $j_p \in G^{-1}(H)$ by $j'_p \in G^{-1}(H)$ may change the pattern of X_1-value collisions within the Y sequence, but the same change will occur in the Z sequence (since $G(j_p)$ and $G(j'_p)$ have values in H, which implies $X_2(j) = X_1(j)$ for every $j \in G^{-1}(H)$). In other words, $\mathtt{h}(Y')$ may differ from $\mathtt{h}(Y)$ due to the replacement of j_p by j'_p, but exactly the same effect occurs between $\mathtt{h}(Z')$ and $\mathtt{h}(Z)$, because in all cases we refer to collisions of X_1-values.

[22] We stress that collisions inside each part do not contribute to $\Delta(\mathtt{h}(Y), \mathtt{h}(Z))$, regardless if they are in the first part (i.e., among $X_1(j_1), ..., X_1(j_m)$) or in the second part (i.e., among $X_i(k_1), ..., X_i(k_m)$).

2. If $G(j_p) \in H$ and $G(j'_p) \in L$, then the probability that $G(j'_p)$ hits $\{G(k'_q) : q \in [m]\}$ is at most $m \cdot \frac{2}{n} = 2m/n$ (because in that case $G(j'_p)$ must hits $\{G(k'_q) : q \in [m] \ \& \ G(k'_q) \in L\}$) and otherwise this re-randomization does not affect the difference between $\Delta(\mathtt{h}(Y), \mathtt{h}(Z))$ and $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z'))$. Details follow.

   If $G(j_p) \in H$ and $G(j'_p) \in L$ does not hit $\{G(k'_q) : q \in [m] \ \& \ G(k'_q) \in L\}$, then the replacement of j_p by j'_p does not affect the foregoing statistical difference, since a possible collision of $G(j'_p)$ with a value in $\{G(j'_q) : q \in [m] \setminus \{p\}\}$ does not contribute to $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z'))$. Hence, a re-randomized $G(j'_p) \in L$ contributes to the difference between $\Delta(\mathtt{h}(Y), \mathtt{h}(Z))$ and $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z'))$ only if $G(j'_p)$ hits $\{G(k'_q) : q \in [m] \ \& \ G(k'_q) \in L\}$, which happens with probability at most $2m/n$. Recalling that there are at most $50m^2/n$ index-collisions among the j_p's (i.e., pairs (j_p, j_q) such that $j_p = j_q$ and $p \neq q$), it follows that the total contribution of this case is at most $\frac{50m^2}{n} \cdot \frac{2m}{n} = \frac{100m^3}{n^2} < 0.001$, by an appropriate choice of the constant $c > 0$ (in $m = c \cdot n^{2/3}$).

3. If $G(j_p) \in L$ and $G(j'_p) \in H$, then the probability that $G(j_p)$ hits $\{G(k_q) : q \in [m]\}$ is at most $2m/n$, and otherwise this re-randomization does not affect the difference between $\Delta(\mathtt{h}(Y), \mathtt{h}(Z))$ and $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z'))$. So, again, the total contribution of this case is at most $100m^3/n^2 < 0.001$.

4. If $G(j_p) \in L$ and $G(j'_p) \in L$, then the probability that either $G(j_p)$ hits $\{G(k_q) : q \in [m]\}$ or $G(j'_p)$ hits $\{G(k'_q) : q \in [m]\}$ is at most $2 \cdot 2m/n$, and otherwise this re-randomization does not affect the difference between $\Delta(\mathtt{h}(Y), \mathtt{h}(Z))$ and $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z'))$. So the total contribution of this case is is at most $200m^3/n^2 < 0.002$.

To summarize: The only contribution of the re-randomization of the indices that form collisions in $(j_1, ..., j_m)$ to the difference between $\Delta(\mathtt{h}(Y), \mathtt{h}(Z))$ and $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z'))$ arises from indices j_p that are re-randomized to j'_p such that either j_p hits $\{G(k_q) : q \in [m] \ \& \ G(k_q) \in L\}$ or j'_p hits $\{G(k'_q) : q \in [m] \ \& \ G(k'_q) \in L\}$. But the probability of this event is small (i.e., smaller than $O(m^3/n^2) < 0.005$), since the number of re-randomized indices is relatively small (i.e., smaller than $50m^2/n$) and the probability of each hit is small (i.e., at most $2m/n$). Applying an analogous analysis to $(k_1, ..., k_m)$, where here we consider collisions with either $\{G(j_q) : q \in [m] \ \& \ G(j_q) \in L\}$ or $\{G(j'_q) : q \in [m] \ \& \ G(j'_q) \in L\}$, the claim follows (because we were considering an event that occurs with probability at least 0.96 and showed that in that case the difference is smaller than $2 \cdot 0.005$). □

Conclusion. The theorem follows by combining Claims 3.2.2 and 3.2.4, while recalling that (with probability at least 0.98) the sequences $(j'_1, ..., j'_m)$ and $(k'_1, ..., k'_m)$ are uniformly and independently distributed among the m-long sequences of distinct elements in $[n]$. Specifically, with high probability, the canonical tester is presented with histograms (of either Y' or Z') that are statistically close (i.e., $\Delta(\mathtt{h}(Y'), \mathtt{h}(Z')) < 0.1$), and so it cannot distinguish them. On the other hand, Y' represents answers to m distinct random queries made to each of the two copies of the sequence S_1, whereas Z' represents answers to m distinct random queries made to the sequences S_1 and S_2, which are at distance 1/2 of one another.[23] That is, Y' (resp., Z') represents answers from a pair of sequences that should be accepted (resp., rejected) by a tester with probability at least 2/3,

[23] Recall that S_1 and S_2 are n-long sequences (viewed as functions defined over $[n]$) such that $S_1(j) = G(j)$ and $S_2(j) = \phi(G(j))$ for every $j \in [n]$, whereas X_1 and X_2 are defined in the same manner but viewed as random variables

whereas $\Delta(\mathtt{h}(Y'),\mathtt{h}(Z')) < 0.1$. It follows that for a sufficiently small constant $c > 0$, isomorphism between sequences cannot be tested using $m = c \cdot n^{2/3}$ queries. ■

4 Concluding Comments

The results in this paper determine the query complexity of both version of the problem of testing graph isomorphism, in the bounded-degree graph model, up to a factor that is polynomial in the size of the largest connected components. In fact, Theorems 1.1 and 1.2 are special cases of the following general results (which are stated only for constant $\epsilon > 0$ and $s > \log_2 n$):[24]

1. For every sufficiently small constant $\epsilon > 0$ and almost all bounded-degree n-vertex graphs H that consists of connected components of size at most s, the query complexity of ϵ-testing isomorphism to H lies between $\Omega(n^{1/2}/s^{1/2})$ and $O(s \cdot n^{1/2})$.

2. For every sufficiently small constant $\epsilon > 0$, the query complexity of ϵ-testing isomorphism between a pair of n-vertex graphs that consist of connected components of size at most s lies between $\Omega(n^{2/3}/s^{2/3})$ and $O(s \cdot n^{2/3})$.

The proofs of these results employ graph theoretic arguments (i.e., the notion of an expander) only in order to reduce the analysis to (sequence-isomorphism) problems that ignore the graph theoretic origin. That is, the core of the aforementioned proofs has nothing to do with graphs. In contrast, we believe that determining the query complexity of testing graph isomorphism in the general case (e.g., for connected graphs, let alone for expanders) may require some graph theoretic insights. This belief was recently confirmed by [14] (see brief discussion in Section 1.4).

Turning back to the general bounds that we known, recall that testing isomorphism to some fixed n-vertex graphs has query complexity $\widetilde{\Theta}(n^{1/2})$. In fact, by [14], this bound holds for almost all n-vertex graphs. Obviously, for some n-vertex graphs the query complexity (of testing isomorphism to them) may be smaller, but are there graphs for which it is larger? And it that case, how large may it be?

Open Problem 4.1 (sublinear complexity for testing isomorphism to a fixed graph): *Is it possible to* 0.01*-test isomorphism to any fixed 3-regular graph in query complexity that is sublinear in the number of vertices?*

Note that such a tester must have two-sided error probability (cf. Theorem 2.6). As for testing isomorphism between two input graphs, we believe that the complexity of this problem is larger than the known lower bound. Is this the case?

over $[n]$. Also recall that

$$\Delta(X_1, X_2) \;=\; \Pr_{j\in[n]}[S_1(j) \neq S_2(j)] \;=\; \Pr_{j\in[n]}[G(j) \in L] \;=\; 1/2.$$

[24] Actually, the following bounds hold also for $s > \frac{3\log_2 n}{\log_2 \log_2 n}$, but for smaller s the query complexity is exponential in $\Theta(s \log s)$. The latter claim holds since the proofs of Propositions 2.2 and 2.5 actually establish the computational equivalence of the n-vertex graph isomorphism problems to the sequence-isomorphism problems regarding sequences (of length either n/s or n) over an alphabet of size $N(s) = \exp(\Theta(s \log s))$. Recall that these sequence-isomorphism testing problems have query complexity between $\Theta(N(s)^{1/2})$ and $O(N(s))$.

Open Problem 4.2 (higher lower bounds for testing isomorphism between two input graphs): *Does* 0.01*-testing isomorphism between two* n*-vertex graphs* (of bounded degree) *require* $\omega(n^{2/3})$ *queries?*

As hinted above, the focus should be on graphs that cannot be partitioned into small parts with few edges among them. In contrast, we mention that ϵ-testing isomorphism between "hyperfinite" graphs (i.e., graphs that can be partitioned to $P(\epsilon)$-sized parts with at most $\epsilon dn/6$ edges among them) is doable using $Q(\epsilon)$ queries [18, Thm. 3.2], where $P, Q : (0, 1] \to \mathbb{N}$.

Acknowledgements

I am grateful to Noga Alon and Reut Levi for helpful discussions. I also wish to thank Yoav Ben-Dov for pointing out significant typos in the original version.

References

[1] N. Alon, E. Blais, S. Chakraborty, D. Garcia-Soriano, and A. Matsliah. Nearly Tight Bounds for Testing Function Isomorphism. *SIAM Journal on Computing*, Vol. 42 (2), pages 459–493, 2013.

[2] L. Babai and E.M. Luks. Canonical Labeling of Graphs. In *15th ACM Symposium on the Theory of Computing*, pages 171–183, 1983.

[3] T. Batu, L. Fortnow, R. Rubinfeld, W.D. Smith, P. White. Testing that Distributions are Close. In *41st IEEE Symposium on Foundations of Computer Science*, pages 259–269, 2000.

[4] T. Batu, E. Fischer, L. Fortnow, R. Kumar, R. Rubinfeld, and P. White. Testing Random Variables for Independence and Identity. In *42nd IEEE Symposium on Foundations of Computer Science*, pages 442–451, 2001.

[5] 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.

[6] B. Bollobas. A Probabilistic Proof of an Asymptotic Formula for the Number of Labelled Regular Graphs. *European Journal of Combinatorics*, Vol. 1, 311–316, 1980.

[7] B. Bollobas. The Isoperimetric Number of Random Regular Graphs. *European Journal of Combinatorics*, Vol. 9, 241–244, 1988.

[8] E. Fischer and A. Matsliah. Testing Graph Isomorphism. *SIAM Journal on Computing*, Vol. 38 (1), pages 207–225, 2008.

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

[10] 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.

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

[12] O. Goldreich and D. Ron. On Testing Expansion in Bounded-Degree Graphs. *ECCC*, TR00-020, March 2000.

[13] O. Goldreich and D. Ron. On Proximity Oblivious Testing. *SIAM Journal on Computing*, Vol. 40 (2), pages 534–566, 2011.

[14] O. Goldreich and L. Tauber. On Testing Isomorphism to a Fixed Graph in the Bounded-Degree Graph Model. *ECCC*, TR23-146, 2023.

[15] 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.

[16] M. Kusumoto and Y. Yoshida. Testing Forest-Isomorphism in the Adjacency List Model. In *Int. Colloquium on Automata, Languages and Programming*, pages 763–774, LNCS 8572, 2014.

[17] E.M. Luks. Isomorphism of Graphs of Bounded Valence can be Tested in Polynomial Time. *Journal of Computer and System Science*, Vol. 25 (1), pages 42–65, 1982.

[18] I. Newman and C. Sohler. Every Property of Hyperfinite Graphs Is Testable. *SIAM Journal on Computing*, Vol. 42 (3), pages 1095–1112, 2013.

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

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

[21] P. Valiant. Testing Symmetric Properties of Distributions, PhD Thesis, MIT, 2012.

[22] G. Valiant and P. Valiant. Instance-by-instance optimal identity testing. *ECCC*, TR13-111, 2013.