Testing Graph Isomorphism *

Eldar Fischer † Arie Matsliah ‡

Abstract

Two graphs G and H on n vertices are ϵ-far from being isomorphic if at least $\epsilon\binom{n}{2}$ edges must be added or removed from $E(G)$ in order to make G and H isomorphic. In this paper we deal with the question of how many queries are required to distinguish between the case that two graphs are isomorphic, and the case that they are ϵ-far from being isomorphic. A query is defined as probing the adjacency matrix of any one of the two graphs, i.e. asking if a pair of vertices forms an edge of the graph or not.

We investigate both one-sided error and two-sided error testers under two possible settings: The first setting is where both graphs need to be queried; and the second setting is where one of the graphs is fully known to the algorithm in advance.

We prove that the query complexity of the best one-sided error testing algorithm is $\widetilde{\Theta}(n^{3/2})$ if both graphs need to be queried, and that it is $\widetilde{\Theta}(n)$ if one of the graphs is known in advance (where the $\widetilde{\Theta}$ notation hides polylogarithmic factors in the upper bounds). For the two-sided error testers we prove that the query complexity of the best tester is $\widetilde{\Theta}(\sqrt{n})$ when one of the graphs is known in advance, and we show that the query complexity lies between $\Omega(n)$ and $\widetilde{O}(n^{5/4})$ if both G and H need to be queried. All of our algorithms are additionally non-adaptive, while all of our lower bounds apply for adaptive testers as well as non-adaptive ones.

*Research supported in part by an Israel Science Foundation grant numbers 55/03 and 1101/06. A preliminary version appeared in Proc. of 17^{th} SODA, 2006

†Faculty of Computer Science, Technion – Israel institute of technology, Technion City, Haifa 32000, Israel. Email: {eldar,ariem}@cs.technion.ac.il.

1 Introduction

Combinatorial property testing deals with the following task: For a fixed $\epsilon > 0$ and a fixed property P, distinguish using as few queries as possible (and with probability at least $\frac{2}{3}$) between the case that an input of length m satisfies P, and the case that the input is ϵ-far (with respect to an appropriate metric) from satisfying P. The first time a question formulated in terms of property testing was considered is in the work of Blum, Luby and Rubinfeld [8]. The general notion of property testing was first formally defined by Rubinfeld and Sudan [17], mainly for the context of the algebraic properties (such as linearity) of functions over finite fields and vector spaces. The first investigation in the combinatorial context is that of Goldreich, Goldwasser and Ron [13], where testing of combinatorial graph properties is first formalized. The "dense" graph testing model that was defined in [13] is also the one that will serve us here. In recent years the field of property testing has enjoyed rapid growth, as witnessed in the surveys [16] and [9].

Formally, our inputs are two functions $g : \{1, 2, \ldots, \binom{n}{2}\} \to \{0, 1\}$ and $h : \{1, 2, \ldots, \binom{n}{2}\} \to \{0, 1\}$, which represent the edge sets of two corresponding graphs G and H over the vertex set $V = \{1, \ldots, n\}$. The *distance* of a graph from a property P is measured by the minimum number of bits that have to be modified in the input in order to make it satisfy P, divided by the input length m, which in our case is taken to be $\binom{n}{2}$. For the question of testing graphs with a constant number of queries there are many recent advances, such as [4], [11], [3] and [2]. For the properties that we consider here the number of required queries is of the form n^α for some $\alpha > 0$, and our interest will be to find bounds as tight as possible on α. We consider the following questions:

1. Given two input graphs G and H, how many queries to G and H are required to test that the two graphs are isomorphic? This property was already used in [1] for proving lower bounds on property testing, and a lower bound of the form n^α was known for quite a while (see e.g. [9]).
2. Given a graph G_k, which is known in advance (and for which any amount of preprocessing is allowed), and an input graph G_u, how many queries to G_u are required to test that G_u is isomorphic to G_k? Some motivation for this question comes from [10], where upper and lower bounds that correlate this question with the "inherent complexity" of the provided G_k are proven. In this paper, our interest is in finding the bounds for the "worst possible" G_k.

For the case where the testers must have one-sided error, our results show tight (up to logarithmic factors) upper and lower bounds, of $\widetilde{\Theta}(n^{3/2})$ for the setting where both graphs need to be queried, and $\widetilde{\Theta}(n)$ for the setting where one graph is given in advance. The upper bounds are achieved by trivial algorithms of edge sampling and exhaustive search. As we are interested in the number of queries we make no attempt to optimize the running time. The main work here lies in proving a matching lower bound for the first setting where both graphs need to be queried, as the lower bound for the second setting is nearly trivial.

Unusually for graph properties that involve no explicit counting in their definition, we can do significantly better if we allow our algorithms to have two-sided error. When one graph is given in advance, we show $\widetilde{\Theta}(n^{1/2})$ upper and lower bounds. The upper bound algorithm uses a technique that allows us to greatly reduce the number of candidate bijections that need to be checked, while assuring that for isomorphic graphs one of them will still be close to an isomorphism. For this to work we need to combine it with a distribution testing algorithm from [7], whose lower bound is in some sense the true cause of the matching lower bound here.

For two-sided error testers where the two graphs need to be queried, a gap in the bounds remains. We present here a lower bound proof of $\Omega(n)$ on the query complexity – it is in fact the lower bound proof already known from the literature, only here we analyze it to its fullest potential. The upper bound of $\widetilde{O}(n^{5/4})$ uses the ideas of the algorithm above for the setting where one of the graphs is known, with an additional mechanism to compensate for having to query from both graphs to find matching vertices.

To our knowledge, the best known algorithm for deciding this promise problem in the classical sense (i.e., given two graphs distinguish whether they are isomorphic or ϵ-far from being isomorphic) requires quasi-polynomial running time [6]. Both our two-sided error testers have the additional property of a quasi-polynomial running time (similarly to the algorithm in [6]) even with the restriction on the number of queries.

The following is the summary of our results for the query complexity in various settings. We made no effort to optimize the logarithmic factors in the upper bounds, as well as the exact dependance on ϵ (which is at most polynomial).

	Upper bound	Lower bound
One sided error, one graph known	$\widetilde{O}(n)$	$\Omega(n)$
One sided error, both graphs unknown	$\widetilde{O}(n^{3/2})$	$\Omega(n^{3/2})$
Two sided error, one graph known	$\widetilde{O}(n^{1/2})$	$\Omega(n^{1/2})$
Two sided error, both graphs unknown	$\widetilde{O}(n^{5/4})$	$\Omega(n)$

The rest of the paper is organized as follows. We provide some preliminaries and definitions in Section 2. Upper and lower bounds for the one-sided algorithms are proven in Section 3, and the upper and lower bounds for the two-sided algorithms are proven in Section 4. The final Section 5 contains some discussion and concluding comments.

2 Notations and preliminaries

All graphs considered here are undirected and with neither loops nor parallel edges. We also assume (even where not explicitly stated) that the number of vertices of the input graph is large enough, as a function of the other parameters. We denote by $[n]$ the set $\{1, 2, \ldots, n\}$. For a vertex

v, $N(v)$ denotes the set of v's neighbors. For a pair of vertices u, v we denote by $N(u) \triangle N(v)$ the symmetric difference between $N(u)$ and $N(v)$. Given a permutation $\sigma : [n] \to [n]$, and a subset U of $[n]$, we denote by $\sigma(U)$ the set $\{\sigma(i) : i \in U\}$. Given a subset U of the vertices of a graph G, we denote by $G(U)$ the induced subgraph of G on U. We denote by $G(n, p)$ the random graph where each pair of vertices forms an edge with probability p, independently of each other.

Definition 1. *Given two labeled graphs G and H on the same vertex set V, the* distance *between G and H is the size of the symmetric difference between the edge sets of G and H, divided by $\binom{|V|}{2}$.*

Given a graph G and a graph H on the same vertex set V, we say that H and G are ϵ-far*, if the distance between G and any permutation of H is at least ϵ.*

Given a graph G and a graph property (a set of graphs that is closed under graph isomorphisms) P, we say that G is ϵ-far *from satisfying the property P, if G is ϵ-far from any graph H on the same vertex set which satisfies P.*

Using this definition of the distance, we give a formal definition of a graph testing algorithm.

Definition 2. *An* ϵ-testing algorithm with q queries *for a property P is a probabilistic algorithm, that for any input graph G makes up to q queries (a query consisting of finding whether two vertices u, v of G form an edge of G or not), and satisfies the following.*

- *If G satisfies P then the algorithm accepts G with probability at least $\frac{2}{3}$.*
- *If G is ϵ-far from P, then the algorithm rejects G with probability at least $\frac{2}{3}$.*

A property testing algorithm has one-sided error probability *if it accepts inputs that satisfy the property with probability* 1. *We also call such testers* one-sided error testers.

A property testing algorithm is non-adaptive *if the outcomes of its queries do not affect the choice of the following queries, but only the decision of whether to reject or accept the input in the end.*

The following is just an extension of the above definition to properties of pairs of graphs. In our case, we will be interested in the property of two graphs being isomorphic.

Definition 3. *An* ϵ-testing algorithm with q queries *for a property P of pairs of graphs is a probabilistic algorithm, that for any input pair G,H makes up to q queries in G and H (a query consisting of finding whether two vertices u, v of G (H) form an edge of G (H) or not), and satisfies the following.*

- *If the pair G,H satisfies P then the algorithm accepts with probability at least $\frac{2}{3}$.*
- *If the pair G,H is ϵ-far from P, then the algorithm rejects with probability at least $\frac{2}{3}$.*

To simplify the arguments when discussing the properties of the query sets, we define *knowledge charts*.

Definition 4. *Given a query set Q to the adjacency matrix A of the graph $G = (V, E)$ on n vertices, we define the* knowledge chart $I_{G,Q}$ *of G as the subgraph of G known after making the set Q of queries to A. We partition the pairs of vertices of $I_{G,Q}$ into three classes: Q^1, Q^0 and Q^*. The pairs in Q^1 are the ones known to be edges of G, the pairs in Q^0 are those that are known not to be edges of G, and all unknown (unqueried) pairs are in Q^*. In other words, $Q^1 = E(G) \cap Q$, $Q^0 = Q \setminus E(G)$, and $Q^* = [V(G)]^2 \setminus Q$. For a fixed q, $0 \leq q \leq n$, and G, we define $I_{G,q}$ as the set of knowledge charts $\{I_{G,Q} : |Q| = q\}$. For example, note that $|I_{G,0}| = |I_{G,\binom{n}{2}}| = 1$.*

We will ask the question of whether two query sets are consistent, i.e. they do not provide an evidence for the two graphs being non-isomorphic. We say that the knowledge charts are *knowledge-packable* if the query sets that they represent are consistent. Formally,

Definition 5. *A* knowledge-packing *of two knowledge charts I_{G_1,Q_1}, I_{G_2,Q_2}, where G_1 and G_2 are graphs with n vertices, is a bijection π of the vertices of G_1 into the vertices of G_2 such that for all $v, u \in V(G_1)$, if $\{v, u\} \in E(G_1) \cap Q_1$ then $\{\pi(v), \pi(u)\} \notin Q_2 \setminus E(G_2)$, and if $\{v, u\} \in Q_1 \setminus E(G_1)$ then $\{\pi(v), \pi(u)\} \notin E(G_2) \cap Q_2$.*

In particular, if G_1 is isomorphic to G_2, then for all $0 \leq q_1, q_2 \leq \binom{n}{2}$, every member of I_{G_1,q_1} is knowledge-packable with every member of I_{G_2,q_2}. In other words, if G_1 is isomorphic to G_2, then there is a knowledge-packing of I_{G_1,Q_1} and I_{G_2,Q_2} for any possible query sets Q_1 and Q_2.

Lemma 2.1. *Any one-sided error isomorphism tester, after completing its queries Q_1, Q_2, must always accept G_1 and G_2 if the corresponding knowledge charts I_{G_1,Q_1}, I_{G_2,Q_2} on which the decision is based are knowledge-packable. In particular, if for some G_1, G_2 and $0 \leq q \leq \binom{n}{2}$, any $I_{G_1,Q_1} \in I_{G_1,q}$ and $I_{G_2,Q_2} \in I_{G_2,q}$ are knowledge-packable, then every one-sided error isomorphism tester which is allowed to ask at most q queries must always accept G_1 and G_2.*

Proof. This is true, since if the knowledge charts I_{G_1,Q_1} and I_{G_2,Q_2} are packable, it means that there is an extension G_1' of G_1's restriction to Q_1 to a graph that is isomorphic to G_2. In other words, given G_1' and G_2 as inputs, there is a positive probability that the isomorphism tester obtained $I_{G_1',Q_1} = I_{G_1,Q_1}$ and I_{G_2,Q_2} after completing its queries, and hence, a one-sided error tester must always accept in this case. ∎

Proving lower bounds for the two-sided error testers involves Yao's method [18], which for our context informally says that if there is a small enough statistical distance between the distributions of q query results, from two distributions over inputs that satisfy the property and inputs that are far from satisfying the property, then there is no tester for that property which makes at most q queries. We start with definitions that are adapted to property testing lower bounds.

Definition 6 (restriction, variation distance)**.** *For a distribution D over inputs, where each input is a function $f : \mathcal{D} \to \{0, 1\}$, and for a subset $\mathcal{Q}$ of the domain $\mathcal{D}$, we define the* restriction $D|_{\mathcal{Q}}$ *of D to $\mathcal{Q}$ to be the distribution over functions of the type $g : \mathcal{Q} \to \{0, 1\}$, that results from choosing a* *random function $f : \mathcal{D} \to \{0,1\}$ according to the distribution D, and then setting g to be $f|_{\mathcal{Q}}$, the restriction of f to $\mathcal{Q}$.*

Given two distributions D_1 and D_2 of binary functions from $\mathcal{Q}$, we define the variation distance *between D_1 and D_2 as follows: $d(D_1, D_2) = \frac{1}{2}\sum_{g:\mathcal{Q}\to\{0,1\}} |\Pr_{D_1}[g] - \Pr_{D_2}[g]|$, where $\Pr_D[g]$ denotes the probability that a random function chosen according to D is identical to g.*

The next lemma follows from [18] (see e.g. [9]):

Lemma 2.2 (see [9])**.** *Suppose that there exists a distribution D_P on inputs over $\mathcal{D}$ that satisfy a given property P, and a distribution D_N on inputs that are ϵ-far from satisfying the property, and suppose further that for any $\mathcal{Q} \subset \mathcal{D}$ of size q, the variation distance between $D_P|_{\mathcal{Q}}$ and $D_N|_{\mathcal{Q}}$ is less than $\frac{1}{3}$. Then it is not possible for a non-adaptive algorithm making q (or less) queries to ϵ-test for P.*

An additional lemma for adaptive testers is proven implicitly in [12], and a detailed proof appears in [9]. Here we strengthen it somewhat, but still exactly the same proof works in our case too.

Lemma 2.3 ([12], see [9])**.** *Suppose that there exists a distribution D_P on inputs over $\mathcal{D}$ that satisfy a given property P, and a distribution D_N on inputs that are ϵ-far from satisfying the property. Suppose further that for any $\mathcal{Q} \subset \mathcal{D}$ of size q, and any $g : \mathcal{Q} \to \{0,1\}$, we have $\Pr_{D_P|_{\mathcal{Q}}}[g] < \frac{3}{2}\Pr_{D_N|_{\mathcal{Q}}}[g]$. Then it is not possible for any algorithm making q (or less) queries to ϵ-test for P. The conclusion also holds if instead of the above, for any $\mathcal{Q} \subset \mathcal{D}$ of size q and any $g : \mathcal{Q} \to \{0,1\}$, we have $\Pr_{D_N|_{\mathcal{Q}}}[g] < \frac{3}{2}\Pr_{D_P|_{\mathcal{Q}}}[g]$.*

Often, given two isomorphic graphs G, H on n vertices, we want to estimate how many vertices from both graphs need to be randomly chosen in order to get an intersection set of size k with high probability.

Lemma 2.4. *Given two graphs G, H on n vertices, a bijection σ of their vertices, and two uniformly random subsets $C_G \subset V(G), C_H \subset V(H)$, the following holds: for any $0 < \alpha < 1$ and any positive integers c, k, if $|C_G| = kn^\alpha \log^c n$ and $|C_H| = n^{1-\alpha}\log^c n$, then with probability $1 - o(2^{-\log^c n})$ the size of $C_G \cap \sigma(C_H)$ is greater than k.*

Proof sketch. By the linearity of expectation, the expected size of the intersection set is $\frac{|C_G||C_H|}{n} = k\log^{2c} n$. Using large deviation inequalities, $C_G \cap \sigma(C_H) > k$ with probability $1 - o(2^{-\log^c n})$. ∎

3 One-sided Testers

By Lemma 2.1, one-sided testers for isomorphism look at some query set Q of the input, and accept if and only if the restriction of the input to Q is extensible to some input satisfying the property. The main idea is to prove that if the input is far from satisfying the property, then with high probability its restriction Q will provide the evidence for it. To prove lower bounds for one-sided testers, it is sufficient to find an input that is ϵ-far from satisfying the property, but for which the restriction of the input to any possible set Q is extensible to some alternative input that satisfies the property. In this section we prove the following:

Theorem 3.1. *The query complexity of the best one-sided isomorphism tester is $\widetilde{\Theta}(n^{3/2})$ (up to coefficients depending only on the distance parameter ϵ) if both graphs are unknown, and it is $\widetilde{\Theta}(n)$ if one of the graphs is known in advance.*

We first prove Theorem 3.1 for the case where both graphs are unknown, and then move to the proof of the simpler second case where one of the graphs is known in advance.

3.1 One-sided testing of two unknown graphs

The upper bound

Algorithm 1.

1. For both graphs G_1, G_2 construct the query sets Q_1, Q_2 respectively by choosing every possible query with probability $\sqrt{\frac{\ln n}{\epsilon n}}$, independently of other queries.
2. If $|Q_1|$ or $|Q_2|$ is larger than $1000n^{3/2}\sqrt{\frac{\ln n}{\epsilon}}$, accept without making the queries. Otherwise make the chosen queries.
3. If there is a knowledge-packing of I_{G_1,Q_1} and I_{G_2,Q_2}, accept. Otherwise reject.

Clearly, the query complexity of Algorithm 1 is $O(n^{3/2}\sqrt{\log n})$ for every fixed ϵ.

Lemma 3.2. *Algorithm 1 accepts with probability 1 if G_1 and G_2 are isomorphic, and if G_1 and G_2 are ϵ-far from being isomorphic, Algorithm 1 rejects with probability* $1 - o(1)$.

Proof. Assume first that G_1 and G_2 are isomorphic, and let π be an isomorphism between them. Obviously π is also a knowledge-packing for any pair of knowledge charts of G_1 and G_2. Hence, if the algorithm did not accept in the second stage, then it will accept in the third stage.

Now we turn to the case where G_1 and G_2 are ϵ-far from being isomorphic. Due to large deviation inequalities, the probability that Algorithm 1 terminates in Step 2 is $o(1)$, and therefore we can assume in the proof that it reaches Step 3 without harming the correctness. Since G_1 and G_2 are ϵ-far from being isomorphic, every possible bijection π of their vertices has a set E_π of at least ϵn^2 pairs of G_1's vertices such that for every $\{u, v\} \in E_\pi$, either $\{u, v\}$ is an edge in G_1 or $\{\pi(u), \pi(v)\}$ is an edge in G_2 but not both. Now we fix π and let $\{u, v\} \in E_\pi$ be one such pair. The probability that $\{u, v\}$ was not queried in G_1 or $\{\pi(u), \pi(v)\}$ was not queried in G_2 is $1 - \frac{\ln n}{\epsilon n}$. Since the queries where chosen independently, the probability that for all $\{u, v\} \in E_\pi$ either $\{u, v\}$

was not queried in G_1 or $\{\pi(u), \pi(v)\}$ was not queried in G_2 is at most $(1 - \frac{\ln n}{\epsilon n})^{\epsilon n^2}$. Using the union bound, we bound the probability of not revealing at least one such pair in both graphs for all possible bijections by $n!(1 - \frac{\ln n}{\epsilon n})^{\epsilon n^2}$. This bound satisfies

$$n!(1 - \frac{\ln n}{\epsilon n})^{\epsilon n^2} \leq n!(e^{-\frac{\ln n}{\epsilon n}})^{\epsilon n^2} = n!\frac{1}{n^n} = o(1)$$

thus the algorithm rejects graphs that are ϵ-far from being isomorphic with probability $1 - o(1)$. ∎

The lower bound

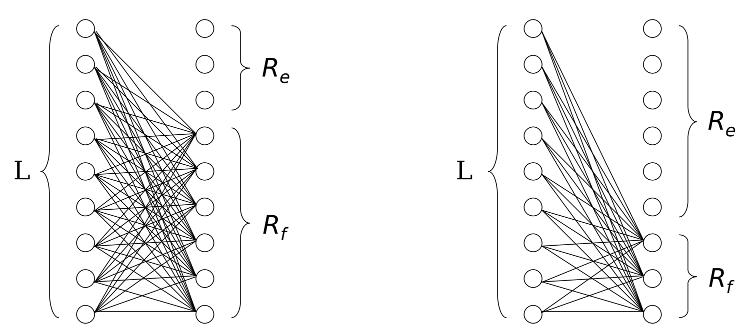
Here we construct a pair G, H of $1/100$-far graphs on n vertices, such that every knowledge chart from $I_{G,n^{3/2}/200}$ can be packed with every knowledge chart from $I_{H,n^{3/2}/200}$, and hence by Lemma 2.1, any one-sided algorithm which is allowed to use at most $n^{3/2}/200$ queries must always accept G and H. Note that this holds for non-adaptive as well as adaptive algorithms, since we actually prove that there is no certificate of size $n^{3/2}/200$ for the non-isomorphism of these graphs.

Lemma 3.3. *For every large enough n, there are two graphs G and H on n vertices, such that:*

1. *G is $1/100$-far from being isomorphic to H*
2. *Every knowledge chart from $I_{G,n^{3/2}/200}$ can be knowledge-packed with any knowledge chart from $I_{H,n^{3/2}/200}$*

Proof. We set both G and H to be the union of a complete bipartite graph with a set of isolated vertices. Formally, G has three vertex sets L, R_f, R_e, where $|L| = n/2, |R_f| = 26n/100$ and $|R_e| = 24n/100$, and it has the following edges: $\{\{u,v\} : u \in L \wedge v \in R_f\}$. H has the same structure, but with $|R_f| = 24n/100$ and $|R_e| = 26n/100$, as illustrated in Figure 1. Clearly, just by the difference in the edge count, G is $1/100$-far from being isomorphic to H, so G and H satisfy the first part of Lemma 3.3.

Figure 1: The graphs G and H (with the difference between them exaggerated)

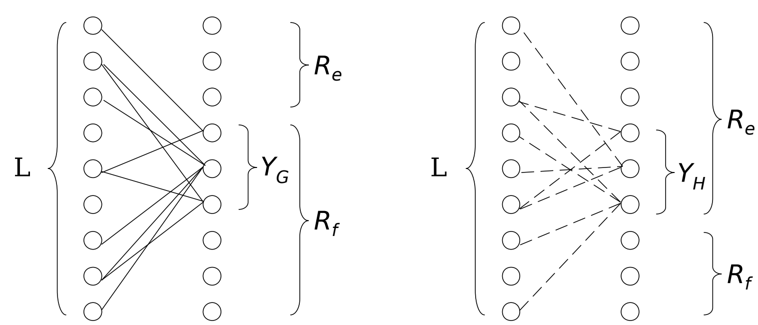


To prove that the second condition of Lemma 3.3 holds, we will show that for all possible query sets Q_G, Q_H of size $n^{3/2}/200$ there exist sets $Y_G \in R_f(G)$ and $Y_H \in R_e(H)$ that satisfy the following.

- $|Y_G| = |Y_H| = n/50$
- the knowledge charts I_{G,Q_G} and I_{H,Q_H} restricted to $L(G) \cup Y_G$ and $L(H) \cup Y_H$ can be packed in a way that pairs vertices from $L(G)$ with vertices from $L(H)$

In Figure 2 we illustrate these restricted knowledge charts, where plain lines are known (queried) edges, and the dashed lines are known (queried) "non-edges". The existence of such Y_G and Y_H implies the desired knowledge-packing, since we can complete the partial packing from the second item by arbitrarily pairing vertices from $R_f(G) \setminus Y_G$ with vertices from $R_f(H)$, and pairing vertices from $R_e(G)$ with vertices from $R_e(H) \setminus Y_H$.

Figure 2: Finding Y_G and Y_H



Proving the existence of Y_G and Y_H

For every vertex $v \in V(G)$, we define its query degree as

$$d_Q(v) = |\{\{v, u\} : u \in V(G) \wedge \{v, u\} \in Q_G\}|$$

We also denote by $N_Q(v)$ the set $\{u : \{v, u\} \in E(G) \cap Q_G\}$ and we denote by $\overline{N}_Q(v)$ the set $\{u : \{v, u\} \in Q_G \setminus E(G)\}$. In other words, $N_Q(v)$ is the set of known neighbors of v and $\overline{N}_Q(v)$ is the set of known non-neighbors of v, and $d_Q(v) = |\overline{N}_Q(v)| + |N_Q(v)|$. We define $d_Q(v)$, $N_Q(v)$ and $\overline{N}_Q(v)$ for H's vertices similarly.

Since $|Q_G|, |Q_H| \leq n^{3/2}/200$, there must be two sets of vertices $D_G \in R_f(G)$ and $D_H \in R_e(H)$, both of size $n/10$, such that $\forall_{v \in D_G} : d_Q(v) \leq n^{1/2}/2$ and $\forall_{v \in D_H} : d_Q(v) \leq n^{1/2}/2$.

Now we prove the existence of Y_G and Y_H (as defined above) using a simple probabilistic argument. First we set an arbitrary pairing $B_D = \{\{v_G^1, u_H^1\}, \{v_G^2, u_H^2\}, \ldots, \{v_G^{n/10}, u_H^{n/10}\}\}$ of D_G's and D_H's elements. Then we choose a bijection $B_L : L(G) \to L(H)$ uniformly at random, and show that with some positive probability, there are at least $n/50$ consistent (packable) pairs in B_D. Formally, we define

$$Y = \{\{v_G, u_H\} \in B_D : B_L(N_Q(v_G)) \cap \overline{N}_Q(u_H) = \emptyset\}$$

as the set of consistent pairs, and show that $\Pr[|Y| \geq n/50] > 0$.

For a specific pair $\{v \in D_G, u \in D_H\}$, we have

$$\Pr_{B_L}[B_L(N_Q(v)) \cap \overline{N}_Q(u) = \emptyset] \geq \prod_{i=0}^{n^{1/2}/2-1} (1 - \frac{n^{1/2}/2}{n/2 - i})$$

$$\geq (1 - \frac{2n^{1/2}}{n})^{n^{1/2}/2} \geq (e + 0.001)^{-1} \geq 1/3$$

and by the linearity of expectation, $\mathrm{E}[|Y|] \geq |D_G|/3 > n/50$. Therefore, there is at least one bijection B_L for which the size of Y is no less than its expectation. We can now set

$$Y_G = \{u : \exists v \in V(H) \text{ such that } \{u, v\} \in Y\}$$

and

$$Y_H = \{v : \exists u \in V(G) \text{ such that } \{u, v\} \in Y\}$$

concluding the proof. ■

3.2 One-sided testing where one of the graphs is known in advance

The algorithm for testing isomorphism between an unknown graph and a graph that is known in advance is similar to Algorithm 1 above. In this case the algorithm makes a quasi-linear number of queries, to accept with probability 1 if the graphs are isomorphic and reject with probability $1 - o(1)$ if they are ϵ-far from being isomorphic. We also prove an almost matching nearly trivial lower bound for this problem.

The upper bound

Denote by G_k and G_u the known and the unknown graphs respectively.

Algorithm 2.

1. Construct a query set Q by choosing every possible query from G_u with probability $\frac{\ln n}{\epsilon n}$, independently at random.
2. If $|Q|$ is larger than $\frac{10 n \ln n}{\epsilon}$, accept without making the queries. Otherwise make the chosen queries.
3. If there is a knowledge-packing of $I_{G_u,Q}$ and $I_{G_k,[V(G_k)]^2}$, accept. Otherwise reject.

Clearly the query complexity of Algorithm 2 is $O(n \log n)$, and it rejects in Step 2 with probability $o(1)$.

Lemma 3.4. *Algorithm 2 always accepts isomorphic graphs, and it rejects ϵ-far graphs with probability* $1 - o(1)$.

Proof. The proof is almost identical to that of Lemma 3.2. It is clear that isomorphic graphs are always accepted by Algorithm 2. Now we assume that the graphs G_k and G_u are ϵ-far and that the algorithm reached Step 3 (as it stops at Step 2 with probability $o(1)$) . Given a bijection π, the probability that no violating pair $\{u, v\} \in E_\pi$ was queried is at most $(1 - \frac{\ln n}{\epsilon n})^{\epsilon n^2} \le e^{-n \ln n} = n^{-n}$. Applying the union bound over all $n!$ possible bijections, the acceptance probability is bounded by $n!/n^n = o(1)$ ■

The lower bound

As before, to give a lower bound on one-sided error algorithms it is sufficient to show that for some G_k and G_u that are far, no "proof" of their non-isomorphism can be provided with $\Omega(n)$ queries. First we formulate the second part of Lemma 2.1 for the special case where one of the graphs is known in advance.

Lemma 3.5. *If for some G_k, G_u, where G_k is known in advance, and some fixed $0 \le q \le \binom{n}{2}$, $I_{G_k,[v(G_k)]^2}$ is knowledge-packable with every $I_{G_u,Q} \in I_{G_u,q}$, then every one-sided error isomorphism tester which is allowed to ask at most q queries must always accept G_k and G_u.* ■

We set G_k to be a disjoint union of $K_{n/2}$ and $n/2$ isolated vertices, and set G_u to be a completely edgeless graph.

Observation 3.6. *G_k and G_u are $1/4$-far, and every $I_{G_u,Q} \in I_{G_u,\frac{n}{4}}$ is knowledge-packable with $I_{G_k,[V(G_k)]^2}$.*

Proof. Clearly, just by the difference in the edge count, G_k is $1/4$ far from being isomorphic to G_u. But since $n/4$ queries cannot involve more than $n/2$ vertices from G_u (all isolated), and G_k has $n/2$ isolated vertices, the knowledge charts are packable. ■

Together with Lemma 3.5, we get the desired lower bound. This concludes the proof of the last part of Theorem 3.1.

4 Two-sided testers

In the context of graph properties, two-sided error testers are usually not known to achieve significantly lower query complexity than the one-sided error testers, apart from the properties that explicitly involve counting, such as *Max-Cut* and *Max-Clique* [13]. However, in our case two-sided error isomorphism testers have substantially lower query complexity than their one-sided error counterparts.

4.1 Two-sided testing where one of the graphs is known in advance

Theorem 4.1. *The query complexity of two-sided error isomorphism testers is* $\widetilde{\Theta}(\sqrt{n})$ *if one of the graphs is known in advance, and the other needs to be queried.*

We prove the lower bound first. This way it will be easier to understand why certain stages of the upper bound testing algorithm are necessary.

The lower bound

Lemma 4.2. *Any isomorphism tester that makes at most* $\frac{\sqrt{n}}{4}$ *queries to* G_u *cannot distinguish between the case that* G_k *and* G_u *are isomorphic and the case that they are* $1/32$*-far from being isomorphic, where* G_k *is known in advance.*

We begin with a few definitions.

Definition 7. *Given a graph* G *and a set* W *of* $\frac{n}{2}$ *vertices of* G*, we define the* clone $G^{(W)}$ *of* G *in the following way:*

- *the vertex set of* $G^{(W)}$ *is defined as:* $V(G^{(W)}) = W \cup \{w' : w \in W\}$
- *the edge set of* $G^{(W)}$ *is defined as:* $E(G^{(W)}) =$

$$\Big\{\{v,u\} : \{v,u\} \in E(G)\Big\} \cup \Big\{\{v',u\} : \{v,u\} \in E(G)\Big\} \cup \Big\{\{v',u'\} : \{v,u\} \in E(G)\Big\}$$

In other words, $G^{(W)}$ *is the product of the subgraph of* G *induced on* W *with the graph* K_2*.*

For the two copies $v, v' \in V(G^{(W)})$ *of* $v \in W$*, we say that* v *is the* source *of both* v *and* v'*.*

Lemma 4.3. *Let* $G \sim G(n, 1/2)$ *be a random graph. With probability* $1 - o(1)$ *the graph* G *is such that for every subset* $W \subset V(G)$ *of size* $n/2$*, the clone* $G^{(W)}$ *of* G *is* $1/32$*-far from being isomorphic to* G*.*

Proof. Let G be a random graph according to $G(n, 1/2)$, and let $W \subset V(G)$ be an arbitrary subset of G's vertices of size $n/2$. First we show that for an arbitrary bijection $\sigma : V(G^{(W)}) \to V(G)$ the graphs $G^{(W)}$ and G are $1/32$-close under σ with probability at most $2^{-\Omega(n^2)}$, and then we apply the union bound on all bijections and every possible subset W.

We split the bijection $\sigma : V(G^{(W)}) \to V(G)$ into two injections $\sigma_1 : W \to V(G)$ and $\sigma_2 : V(G^{(W)}) \setminus W \to V(G) \setminus \sigma_1(W)$. Note that either $|W \setminus \sigma_1(W)| \geq n/4$ or $|W \setminus \sigma_2(W)| \geq n/4$.

Assume without loss of generality that the first case holds, and let U denote the set $W \setminus \sigma_1(W)$. Since every edge in G is chosen at random with probability $1/2$, the probability that for some pair $u, v \in U$ either $\{u, v\}$ is an edge in G and $\{\sigma(u), \sigma(v)\}$ is not an edge in G or $\{u, v\}$ is not an edge in G and $\{\sigma(u), \sigma(v)\}$ is an edge in G is exactly $1/2$. Therefore, using large deviation inequalities, the probability that in the set U there are less than $\binom{n}{2}/32$ such pairs is at most $2^{-\Omega(n^2)}$ (as these events are all independent). There are at most $n!$ possible bijections, and $\binom{n}{n/2}$ possible choices for W, so using the union bound, the probability that for some W the graph $G \sim G(n, 1/2)$ is not $1/32$-far from being isomorphic to $G^{(W)}$ is at most $2^{-\Omega(n^2)}\binom{n}{n/2}n! = o(1)$. ■

Given a graph G satisfying the assertion of Lemma 4.3, we set $G_k = G$ and define two distributions over graphs, from which we choose the unknown graph G_u:

- D_P: A permutation of G_k, chosen uniformly at random.
- D_N: A permutation of $G_k^{(W)}$, where both W and the permutation are chosen uniformly at random.

According to Lemma 4.3 and Lemma 2.3, it is sufficient to show that the distributions D_P and D_N restricted to a set of $\sqrt{n}/4$ queries are close. In particular, we intend to show that for any $\mathcal{Q} \subset \mathcal{D} = V^2$ of size $\sqrt{n}/4$, and any $Q : \mathcal{Q} \to \{0, 1\}$, we have $\Pr_{D_P|_\mathcal{Q}}[Q] < \frac{3}{2}\Pr_{D_N|_\mathcal{Q}}[Q]$. This will imply a lower bound for adaptive (as well as non-adaptive) testing algorithms.

Observation 4.4. *For a set U of $G^{(W)}$'s vertices, define the event E_U as the event that there is no pair of copies w, w' of any one of G's vertices in U. For a given set of pairs Q, let U_Q be the set of all vertices that are incident with a pair in Q. Then the distribution $D_N|_\mathcal{Q}$ conditioned on the event E_{U_Q} (defined above) and the unconditioned distribution $D_P|_\mathcal{Q}$ are identical.*

Proof. In D_N, if no two copies of any vertex were involved in the queries, then the source vertices of the queries to G_u are in fact a uniformly random sequence (with no repetition) of the vertices of G_k, and this (together with G_k) completely determines the distribution of the answers to the queries. This is the same as the unconditioned distribution induced by D_P. ■

Intuitively, the next lemma states that picking two copies of the same vertex in a randomly permuted $G^{(W)}$ requires many samples, as per the well known birthday problem.

Lemma 4.5. *For a fixed set Q of at most $\sqrt{n}/4$ queries and the corresponding set U of vertices, the probability that the event E_U did not happen is at most $1/4$.*

Proof. The bound on $|Q|$ implies that $|U| \leq \sqrt{n}/2$. Now we examine the vertices in U as if we add them one by one. The probability that a vertex v that is added to U is a copy (with respect to the original graph G) of some vertex u that was already inserted to U (or vice versa) is at most $\frac{\sqrt{n}}{2n}$. Hence, the probability that eventually (after $\sqrt{n}/2$ insertions) we have two copies of the same vertex in U is at most $\frac{\sqrt{n}}{2n} \cdot \sqrt{n}/2 = 1/4$. ■

¿From Observation 4.4, the distribution $D_N|_{\mathcal{Q}}$ conditioned on the event E_U and the unconditioned distribution $D_P|_{\mathcal{Q}}$ are identical. By Lemma 4.5 it follows that $\Pr[E_U] > 2/3$. Therefore, for any $g : \mathcal{Q} \to \{0, 1\}$ we have

$$\Pr_{D_P|_{\mathcal{Q}}}[g] < \frac{3}{2}\Pr_{D_N|_{\mathcal{Q}}}[g]$$

hence the distributions D_P and D_N satisfy the conditions of Lemma 2.3. The following corollary completes the proof of Lemma 4.2.

Corollary 4.6. *It is not possible for any algorithm (adaptive or not) making $\sqrt{n}/4$ (or less) queries to test for isomorphism between a known graph and a graph that needs to be queried.* ■

The upper bound

We start with a few definitions. Given a graph G and a subset C of $V(G)$, we define the C*-labeling* of G's vertices as follows: every vertex $v \in V(G)$ gets a label according to the set of its neighbors in C. Note that there are $2^{|C|}$ possible labels for a set C, but even if $2^{|C|} > n$ still at most n of the labels occur, since there are only n vertices in the graph. On the other hand, it is possible that several vertices will have the same label according to C. Such a labeling implies the following distribution over the vertices of G.

Definition 8. *Given a graph G and a C-labeling of its vertices (according to some $C \subset V(G)$), we denote by D_C the distribution over the actual labels of the C-labeling (at most n labels), in which the probability of a certain label γ is calculated from the number of vertices from $V(G)$ having the label γ under the C-labeling, divided by n.*

Given a graph G on n vertices and a graph C on $k < n$ vertices, we say that a one to one function $\eta : V(C) \to V(G)$ is an *embedding* of C in G . We also call $\eta(V(C))$ the *placement* of C in G. With a slight abuse of notation, from now on by a placement $\eta(V(C))$ we mean also the correspondence given by η, and not just the set.

Given graphs G, H on n vertices, a subset C_G of $V(G)$ and a placement C_H of C_G in H under an embedding η, we define the *distance* between the C_G-labeling of G and the C_H-labeling of H as

$$\frac{1}{2}\sum_{\gamma \in 2^{C_G}} \big| |\{u \in V(G) : N(u) \cap C_G = \gamma\}| - |\{v \in V(H) : N(v) \cap \eta(C_G) = \gamma\}| \big|$$

this distance measure is equal to the usual variation distance between D_{C_G} and D_{C_H} multiplied by n. We are now ready to prove the upper bound.

Lemma 4.7. *Given an input graph G_u and a known graph G_k (both of order n), there is a property tester A_{ku} that accepts with probability at least $2/3$ if G_u is isomorphic to G_k, and rejects with probability at least $2/3$ if G_u is ϵ-far from G_k. Furthermore, A_{ku} makes $\widetilde{O}(\sqrt{n})$ queries to G_u.*

We first outline the algorithm: The test is performed in two main phases. In Phase 1 we randomly choose a small subset C_u of G_u's vertices, and try all possible placements of C_u in the known graph G_k. The placements that imply a large distance between the labeling of G_u and G_k are discarded. After filtering the good placements of C_u in G_k, we move to Phase 2. In Phase 2 every one of the good placements is tested separately, by defining a random bijection $\pi : V(G_u) \to V(G_k)$ and testing whether π is close to being an isomorphism. Finally, if one of the placements passed both Phase 1 and Phase 2, the graphs are accepted. Otherwise they are rejected.

Phase 1

In the first phase we choose at random a core set C_u of $\log^2 n$ vertices from G_u (the unknown graph). For every embedding η of C_u in G_k and the corresponding placement $C_k \in G_k$, we examine the distributions D_{C_u} and D_{C_k} as in Definition 8. Since the graph G_k is known in advance, we know exactly which are the actual labels according to C_k (in total no more than n labels), so from now on we will consider the restriction of both distributions to these actual labels only. Next we test for every embedding of C_u whether D_{C_u} is statistically close to D_{C_k}. Note that the distribution D_{C_k} is explicitly given, and the distribution D_{C_u} can be sampled by choosing a vertex v from $V(G_u)$ uniformly at random, and making all queries $\{v\} \times C_u$. If the label of some $v \in V(G_u)$ does not exist in the C_k-labeling of G_k, we immediately reject this placement and move to the next one. Now we use the following lemma from [7], which states that $\widetilde{O}(\sqrt{n})$ samples are sufficient for testing if the sampled distribution is close to the explicitly given distribution.

Lemma 4.8. *There is an algorithm that given two distributions D_K,D_U over n elements and a distance parameter ϵ, where D_K is given explicitly and D_U is given as a black box that allows sampling according to the distribution, satisfies the following: If the distributions D_K and D_U are identical, then the algorithm accepts with probability at least $1 - 2^{-\log^7 n}$; and if the variation distance between D_K and D_U is larger than $\epsilon/10$, then the algorithm accepts with probability at most $2^{-\log^7 n}$. For a fixed ϵ, the algorithm uses $\widetilde{O}(\sqrt{n})$ many samples.*

Actually, this is an amplified version of the lemma from [7], which can be achieved by independently repeating the algorithm provided there $\mathrm{polylog}(n)$ many times and taking the majority vote. This amplification allows us to reuse the same $\widetilde{O}(\sqrt{n})$ samples for all possible placements of the core set. As a conclusion of Phase 1, the algorithm rejects the placements of C_u that imply a large variation distance between the above distributions, and passes all other placements of C_u to Phase 2. Naturally, if Phase 1 rejects all placements of C_k due to distribution test failures or due to the existence of labels in G_u that do not exist in G_k, then G_u is rejected without moving to Phase 2 at all. First we observe the following.

Observation 4.9. *With probability $1 - o(1)$, all of the placements that passed Phase 1 imply $\epsilon/10$-close distributions, and all placements that imply identical distributions passed Phase 1. In other words, the distribution test did not err on any of the placements.*

Proof. There are at most $2^{\log^3 n}$ possible placements of C_u. Using the union bound with Lemma 4.8, we conclude that Phase 1 will not err with probability $1 - o(1)$. ■

Phase 2

Following Observation 4.9, we need to design a test such that given a placement C_k of C_u in G_k that implies close distributions, the test satisfies the following conditions:

1. If the graphs are isomorphic and the embedding of C_u is expandable to some isomorphism, then the test accepts with probability at least $3/4$
2. If the graphs G_u and G_k are ϵ-far, then the test accepts with probability at most $o(2^{-\log^3 n})$.

If our test in Phase 2 satisfies these conditions, then we get the desired isomorphism tester. From now on, when we refer to some placement of C_u we assume that it has passed Phase 1 and hence implies close distributions.

In Phase 2 we choose a set W_u of $\log^4 n$ vertices from $V(G_u)$, and retrieve their labels according to C_u by making the queries $W_u \times C_u$. Additionally, we split W_u into $\frac{1}{2}\log^4 n$ pairs $\{\{u_1, v_1\}, \ldots, \{u_{\frac{1}{2}\log^4 n}, v_{\frac{1}{2}\log^4 n}\}\}$ randomly, and make all $\frac{1}{2}\log^4 n$ queries according to these pairs. This is done once, and the same set W_u is used for all the placements of C_u that are tested in Phase 2. Then, for every placement C_k of C_u, we would like to define a random bijection $\pi_{C_u,C_k} : V(G_u) \to V(G_k)$ as follows. For every label γ, the bijection π_{C_u,C_k} pairs the vertices of G_u having label γ with the vertices of G_k having label γ uniformly at random. There might be labels for which one of the graphs has more vertices than the other. We call these remaining vertices *leftovers*. Note that the amount of leftovers from each graph is equal to the distance between the C_k-labeling and the C_u-labeling. Finally, after π_{C_u,C_k} pairs all matching vertices, the leftover vertices are paired arbitrarily. In practice, since we do not know the labels of G_u's vertices, we instead define a partial bijection $\widetilde{\pi}_{C_u,C_k}(W_u) \to V(G_k)$ as follows. Every vertex $v \in W_u$ that has the label γ_v is paired uniformly at random with one of the vertices of G_k which has the same label γ_v and was not paired yet. If this is impossible, we reject the current placement of C_u and move to the next one.

Denote by δ_{C_u,C_k} the fraction of the queried pairs from W_u for which exactly one of $\{u_i, v_i\}$ and $\{\widetilde{\pi}_{C_u,C_k}(u_i), \widetilde{\pi}_{C_u,C_k}(v_i)\}$ is an edge. If $\delta_{C_u,C_k} \leq \epsilon/2$, then G_u is accepted. Otherwise we move to the next placement of C_u. If none of the placements was accepted, G_u is rejected.

Correctness

A crucial observation in our proof is that with high probability, any two vertices that have many distinct neighbors in the whole graph will also have distinct neighbors within a "large enough" random core set.

Formally, given a graph G and a subset C of its vertices, we say that C is *β-separating* if for every pair of vertices $u, v \in V(G)$ such that $d_{uv} \triangleq \frac{1}{n}|\{N(u) \triangle N(v)\}| \geq \beta$ the vertices u and v have different labels under the C-labeling of G.

Claim 4.10. *Let $\beta > 0$ be fixed, let G be a graph of order n and let $C \subset V(G)$ be uniformly chosen random subset of size $\log^2 n$. Then C is β-separating with probability $1 - o(1)$.*

Proof. Fix a pair $u, v \in V(G)$. If u, v are such that $d_{uv} > \beta$, then the probability that they share exactly the same neighbors in C is bounded by $(1-\beta)^{\log^2 n} \leq e^{-\beta \log^2 n} = n^{-\beta \log n}$. Using the union bound, with probability $1-o(1)$ every pair u, v of vertices with $d_{uv} > \beta$ will not have exactly the same neighbors in C, i.e. the vertices will have different labels under the C-labeling. ■

Lemma 4.11 (completeness)**.** *Conditioned over the event that C_u is $\epsilon/8$-separating, if the graphs G_u and G_k are isomorphic and the placement C_k of C_u is expandable to some isomorphism, then $\Pr[\delta_{C_u,C_k} \leq \epsilon/2] = 1 - o(1)$, and hence C_k is accepted in Phase 2 with probability $1 - o(1)$.*

Proof. Let $\phi : V(G_u) \to V(G_k)$ be an isomorphism to which the placement of C_u is expandable. By definition, for every pair v_1, v_2 of G_u's vertices, $\{v_1, v_2\}$ is an edge in G_u if and only if $\{\phi(v_1), \phi(v_2)\}$ is an edge in G_k. In addition, for every vertex $v \in V(G_u)$, the vertices v and $\phi(v)$ have exactly the same labels. Let σ be the permutation, such that π_{C_u,C_k} is the composition of σ and the isomorphism ϕ. In the rest of this proof, by distance we mean the absolute distance between two labeled graphs (which is between 0 and $\binom{n}{2}$).

First we show that the distance from $\sigma(G_u)$ to G_k is almost the same as the distance from $\phi(G_u)$ to G_k (which is zero since ϕ is an isomorphism), and then we apply large deviation inequalities to conclude that $\Pr[\delta_{C_u,C_k} \leq \epsilon/2] = 1 - o(1)$.

To prove that the distance from $\sigma(G_u)$ to G_k is close to zero we show a transformation of ϕ into π_{C_u,C_k} by performing "swaps" between vertices that have the same label. Namely, we define a sequence of permutations ϕ_i, starting from $\phi_0 = \phi$, and ending with $\phi_t = \pi_{C_u,C_k}$. In each step, if there is some vertex v_0 such that $\phi_i(v_0) = u_1$ while $\pi_{C_u,C_k}(v_0) = u_0$, then we find a vertex v_1 such that $\phi_i(v_1) = u_0$, and set $\phi_{i+1}(v_0) = u_0$ and $\phi_{i+1}(v_1) = u_1$. The rest of the vertices are mapped by ϕ_{i+1} as they were mapped by ϕ_i.

Since in each step we only swap between vertices with the same label, and since the core set C_u is $\epsilon/8$-separating, every such swap can increase the distance by at most $\epsilon n/8$, so eventually the distance between $\sigma(G_u)$ and G_k is at most $\epsilon n^2/8$. Therefore, by large deviation inequalities, δ_{C_u,C_k} as defined in Phase 2 is at most $\epsilon/2$ with probability $1 - o(1)$, and so the placement C_k is accepted. ■

We now turn to the case where G_u and G_k are ϵ-far. Note that until now we did not used the fact that C_u and C_k imply close distributions. To understand why this closeness is important, recall the pairs of graphs from the lower bound proof. If we give up the distribution test in Phase 1, then these graphs will be accepted with high probability, since the algorithm cannot reveal two copies of the same vertex when sampling $o(\sqrt{n})$ vertices (recall that $|W_u| = O(\log^4 n)$). Intuitively, the problem is that in these pairs of graphs, the partial random bijection $\widetilde{\pi}_{C_u,C_k}$ will not simulate a restriction of the random bijection π_{C_u,C_k} to a set of $\log^4 n$ vertices. In the lower bound example, $\widetilde{\pi}_{C_u,C_k}$ will have no leftovers with high probability, even though π_{C_u,C_k} will always have $\Omega(n)$ leftovers. The reason is that in the cloned graph G_u, for each of about half of the labels from C_k there are two times more vertices, while for the second half there are no vertices at all. The distribution test in Phase 1 actually checks whether the clustering of the vertices according to the labels is into subsets of almost equal sizes in both G_u and G_k. If it is so, then the partial random bijection $\widetilde{\pi}_{C_u,C_k}$ is indeed similar to the restriction of a bijection π_{C_u,C_k} to a set of $\log^4 n$ vertices.

Lemma 4.12 (soundness). *If the graphs G_u and G_k are ϵ-far, and the placement C_k implies $\epsilon/10$-close distributions, then $\Pr[\delta_{C_u,C_k} \leq \epsilon/2] \leq o(2^{-\log^3 n})$, and hence C_k is accepted in Phase 2 with probability at most $o(2^{-\log^3 n})$.*

Proof. Assume that for a fixed C_k the random bijection π_{C_u,C_k} is ϵ-far from isomorphism. We then need to show that δ_{C_u,C_k} as defined in Phase 2 is larger than $\epsilon/2$ with probability $1 - o(2^{-\log^3 n})$.

Since the variation distance between the distributions D_{C_u} and D_{C_k} is at most $\epsilon/10$, the amount of leftovers (which is exactly the distance between the C_u-labeling of G_u and the C_k-labeling of G_k) is at most $\epsilon n/10$. Therefore, even if we first remove those $\epsilon n/10$ (or less) leftovers, the fraction of pairs u, v for which exactly one of $\{u, v\}$ and $\{\widetilde{\pi}_{C_u,C_k}(u), \widetilde{\pi}_{C_u,C_k}(v)\}$ is an edge is not smaller by more than $4\epsilon/10$ from that of π_{C_u,C_k}.

Let $\widetilde{\pi}_{C_u,C_k}$ be the random partial bijection as defined above. The distribution test of Phase 1 guaranties that $\widetilde{\pi}_{C_u,C_k}$ is a random restriction of a function that is $\epsilon/10$-close to some bijection π_{C_u,C_k}. Since G_u is ϵ-far from G_k, the bijection π_{C_u,C_k} must be ϵ-far from being an isomorphism, and hence $\widetilde{\pi}_{C_u,C_k}$ must exhibit a $6\epsilon/10$-fraction of mismatching edges. Note that the acceptance probability of C_k given $\widetilde{\pi}_{C_u,C_k}$ is equal to the probability that δ_{C_u,C_k} as defined in Phase 2 is at most $\epsilon/2$. Large deviation inequalities show that this probability is at most $2^{-\Omega(\log^4 n)} = o(2^{-\log^3 n})$. ■

As a conclusion, if G_k and G_u are isomorphic, then the probability that C_u is not $\epsilon/8$-separating is at most $o(1)$, and for a correct (under some isomorphism) embedding of C_u in G_k, the probability that the distribution test will fail is also $o(1)$, so in summary algorithm A_{ku} accepts with probability greater than $2/3$. In the case that G_k and G_u are ϵ-far from being isomorphic, with probability $1 - o(1)$ all placements that are passed to Phase 2 imply close label distributions. Then each such placement is rejected in Phase 2 with probability $1 - o(2^{-\log^3 n})$, and by the union bound over all possible placements the graphs are accepted with probability less than $1/3$. Algorithm A_{ku} makes $\widetilde{O}(\sqrt{n})$ queries in Phase 1 and $\widetilde{O}(n^{1/4})$ queries in Phase 2. This completes the proof of Lemma 4.7 and so of Theorem 4.1.

4.2 Two-sided testing of two unknown graphs

Theorem 4.13. *The query complexity of two-sided error isomorphism testers is between $\Omega(n)$ and $\widetilde{O}(n^{5/4})$ if both graphs need to be queried.*

The upper bound

Lemma 4.14. *Given two unknown graphs G and H on n vertices, there is a property tester A_{uu} that accepts with probability at least $2/3$ if G is isomorphic to H, and rejects with probability at least $2/3$ if G is ϵ-far from H. Furthermore, A_{uu} makes $\widetilde{O}(n^{5/4})$ queries to G and H.*

We use here ideas similar to those used in the upper bound proof of Lemma 4.7, but with several modifications. The main difference between this case and the case where one of the graphs is known in advance is that here we cannot write all label distributions with all possible core sets in either one of the unknown graphs (because doing that would require $\Omega(n^2)$ queries). We overcome this difficulty by sampling from both graphs in a way that with high probability will make it possible to essentially simulate the test for isomorphism where one of the graphs is known in advance.

Phase 1

First we randomly pick a set U_G of $n^{1/4}\log^3(n)$ vertices from G, and a set U_H of $n^{3/4}\log^3(n)$ vertices from H. Then we make all $n^{5/4}\log^3(n)$ possible queries in $U_G \times V(G)$. Note that if G and H have an isomorphism σ, then according to Lemma 2.4 with probability $1-o(1)$ the size of $U_G \cap \sigma(U_H)$ will exceed $\log^2(n)$.

For all subsets C_G of U_G of size $\log^2 n$ we try every possible placement $C_H \subset U_H$ of C_G. There are at most $2^{\log^3 n}$ subsets C_G, and at most $2^{\log^3 n}$ possible ways to embed each C_G in U_H. Since we made all $n^{5/4}\log^3(n)$ possible queries in $U_G \times V(G)$, for every $C_G \subset U_G$ the corresponding distribution D_{C_G} is entirely known.

So now for every possible placement of C_G in U_H we test if the variation distance between the distributions D_{C_G} and D_{C_H} is at most $\epsilon/10$. Since we know the entire distributions D_{C_G}, we only need to sample the distribution D_{C_H}, therefore we can still use the amplified distribution test of Lemma 4.8. The test there requires $\widetilde{O}(\sqrt{n})$ samples, so similarly to the proof of Lemma 4.7 we take a random set S of $\widetilde{O}(\sqrt{n})$ vertices from H and make all $n^{5/4}\text{polylog}(n)$ queries in $S \times U_H$.

We reject the pairs of a set C_G and a placement C_H that were rejected by the distribution test for D_{C_G} and D_{C_H}, and pass all other pairs to Phase 2. If Phase 1 rejects all possible pairs, then the graphs G and H are rejected without moving to Phase 2. The following observation is similar to the one we used in the case where one of the graphs is known in advance.

Observation 4.15. *With probability* $1 - o(1)$*, all of the placements that passed Phase 1 imply* $\epsilon/10$*-close distributions, and all placements that imply identical distributions passed Phase 1. In other words, the distribution test did not err on any of the placements.* ∎

Phase 2

As in Lemma 4.7, we need to design a test which given a placement C_H of C_G in H that implies close distributions, satisfies the following conditions:

1. If the graphs are isomorphic and the embedding of C_H is expandable to some isomorphism, then the test accepts with probability at least $3/4$
2. If the graphs G and H are ϵ-far, then the test accepts with probability at most $o(2^{-2\log^3 n})$.

In Phase 2 we choose at random a set W_G of $n^{1/2}\log^{13} n$ vertices from $V(G)$, and a set W_H of $n^{1/2}\log^6 n$ vertices from $V(H)$. We retrieve the labels in W_H according to any C_H by making the queries $W_H \times U_H$. Additionally, we make all queries inside W_H and all queries inside W_G. This is done once, and the same sets W_G, W_H are used for all of the pairs C_G, C_H that are tested in Phase 2. According to Lemma 2.4, if the graphs are isomorphic under some isomorphism σ, then $|W_H \cap \sigma(W_G)| > \log^7 n$ with probability $1 - o(1)$.

Then, similarly to what is done in Lemma 4.7, for every pair C_G, C_H, we would like to define a random bijection $\pi_{C_G,C_H} : V(G) \to V(H)$ as follows. For every label γ, π_{C_G,C_H} pairs the vertices of G having label γ with the vertices of H having label γ uniformly at random. After π_{C_G,C_H} pairs all matching vertices, the leftover vertices are paired arbitrarily. Then again, since we do not know the labels of H's vertices, we define a partial bijection $\widetilde{\pi}_{C_G,C_H}(W_H) \to V(G)$ instead, in which every vertex $v \in W_H$ that has the label γ_v is paired uniformly at random with one of the vertices of G which has the same label γ_v and was not paired yet. If this is impossible, we reject the current pair C_G, C_H and move to the next one.

Denote by I_H the set $\widetilde{\pi}_{C_G,C_H}(W_H) \cap W_G$, and denote by S_H the set $\widetilde{\pi}^{-1}_{C_G,C_H}(I_H)$. According to Lemma 2.4, $|I_H| > \log^7 n$ with probability $1 - o(2^{-\log^6 n})$, that is, with probability $1 - o(1)$ we have $|I_H| > \log^7 n$ for every pair C_G, C_H (if this is not the case, we terminate the algorithm and answer arbitrarily). Next we take $\frac{1}{2}\log^7 n$ pairs $\{\{u_1, v_1\}, \ldots, \{u_{\frac{1}{2}\log^7 n}, v_{\frac{1}{2}\log^7 n}\}\}$ randomly from S_H, and denote by δ_{C_G,C_H} the fraction of S_H's pairs for which exactly one of $\{u_i, v_i\}$ and $\{\widetilde{\pi}_{C_G,C_H}(u_i), \widetilde{\pi}_{C_G,C_H}(v_i)\}$ is an edge. If $\delta_{C_G,C_H} \leq \epsilon/2$, then the graphs are accepted. Otherwise we move to the next pair C_G, C_H. If none of the pairs accepted, then the graphs are rejected.

As noted above, if G and H are isomorphic, then according to Lemma 2.4 with probability $1 - o(1)$ the size of $U_G \cap \sigma(U_H)$ is at least $\log^2(n)$. Therefore with probability $1 - o(1)$ for some pair C_H, C_G the placement C_H of C_G is expandable to an isomorphism. We now need to show that in this case the pair C_H, C_G is accepted with sufficient probability.

Lemma 4.16 (completeness)**.** *If the graphs G and H are isomorphic and σ is an isomorphism between them, then with probability at least $3/4$ there exists $C_G \subset U_G$ with a placement $C_H \subset U_H$ which is expandable to σ, and for which $\delta_{C_G,C_H} \leq \epsilon/2$.*

Proof sketch. First we look at the set $\Delta = U_G \cap \sigma^{-1}(U_H)$. By Lemma 2.4 the size of Δ is at least $\log^2 n$ with probability $1 - o(1)$. Conditioned on this event, we pick $C_G \subseteq \Delta \subseteq U_G$ uniformly from all subsets of Δ with size $\log^2 n$, and set $C_H = \sigma(C_G)$ to be its placement in U_H. We now prove that conditioned on the event that Δ is large enough, C_G and C_H will be as required with probability $1 - o(1)$.

Our main observation is that if we condition only on the event that Δ is large enough, then C_G is distributed uniformly among all subsets with this size of $V(G)$, so we proceed similarly to the case where one of the graphs is known in advance. We observe that if two vertices have many distinct neighbors, then with high probability they will not share exactly the same neighbors within a random core set of size $\log^2 n$ (see Lemma 4.10), so C_G has a separating property. When this happens, it is possible to switch between the vertices with identical labels and still retain a small enough bound on δ_{C_G,C_H}. ■

Lemma 4.17 (soundness)**.** *If the graphs G and H are ϵ-far, and the pair C_G, C_H implies close distributions, then $\Pr[\delta_{C_G,C_H} \leq \epsilon/2] \leq o(2^{-\log^6 n})$, and hence the pair C_G, C_H is accepted in Phase 2 with probability at most $o(2^{-\log^6 n})$.*

Proof sketch. As before, assume that for a fixed pair C_G, C_H the random bijection π_{C_G,C_H} is ϵ-far from isomorphism. We then need to show that δ_{C_G,C_H} as defined in Phase 2 is at most $\epsilon/2$ with probability only $o(2^{-\log^6 n})$.

Since the variation distance between the distributions D_{C_G} and D_{C_H} is at most $\epsilon/10$, the amount of leftovers (which is exactly the distance between the C_G-labeling and the C_H-labeling) is at most $\epsilon n/10$. After removing those $\epsilon n/10$ (or less) leftovers, the fraction of pairs u, v for which exactly one of $\{u, v\}$ and $\{\widetilde{\pi}_{C_G,C_H}(u), \widetilde{\pi}_{C_G,C_H}(v)\}$ is an edge is still not smaller than that of π_{C_G,C_H} by more than $4\epsilon/10$. Now the distribution test of Phase 1 guaranties that $\widetilde{\pi}_{C_G,C_H}$ is $\epsilon/10$-close to the restriction of some random bijection π_{C_G,C_H}. Since the graph G is ϵ-far from being isomorphic to the graph H, the bijection π_{C_G,C_H} must be ϵ-far from an isomorphism, and hence $\widetilde{\pi}_{C_G,C_H}$ must exhibit a $6\epsilon/10$-fraction of incompatible edges, and the acceptance probability of the pair C_G, C_H given $\widetilde{\pi}_{C_G,C_H}$ is equal to the probability that δ_{C_G,C_H} as defined in Phase 2 is at most $\epsilon/2$. Applying large deviation inequalities shows that this probability is at most $2^{-\Omega(\log^7 n)} = o(2^{-\log^6 n})$. ■

The isomorphism testing algorithm A_{uu} makes $\widetilde{O}(n^{5/4})$ queries in total, completing the proof of Theorem 4.13.

The lower bound

A lower bound of $\Omega(n)$ queries is implicitly stated in [9] following [1]. Here we provide the detailed proof for completeness.

Lemma 4.18. *Any adaptive (as well as non-adaptive) testing algorithm that makes at most $\frac{n}{4}$ queries cannot distinguish between the case that the unknown input graphs G and H are isomorphic, and the case that they are $\frac{1}{8}$-far from being isomorphic.*

Proof. We construct two distributions over pairs of graphs. The distribution D_P is constructed by letting the pair of graphs consist of a random graph $G \sim G(n, 1/2)$ and a graph H that is a random permutation of G. The distribution D_N is constructed by letting the pair of graphs consist of two independently chosen random graphs $G, H \sim G(n, 1/2)$.

Clearly D_P satisfies the property with probability 1. By large deviation inequalities, it is also clear that in an input chosen according to D_N, the graphs G and H are $\frac{1}{8}$-far with probability $1 - 2^{\Omega(n^2)}$. The next step is to replace D_N with D'_N, in which the graphs are $\frac{1}{8}$-far from being isomorphic with probability 1. We just set D'_N to be the distribution that results from conditioning D_N on the event that G is indeed $\frac{1}{8}$-far from H.

We now consider any fixed set $Q = \{p_1, \ldots, p_{\frac{n}{4}}\}$ of vertex pairs, some from the first graph, and others from the second graph. For an input chosen according to the distribution D_N, the values of these pairs (the answers for corresponding queries) are $\frac{n}{4}$ uniformly and independently chosen random bits. We now analyze the distribution D_P. Let $e_1, \ldots, e_k$ and $f_1, \ldots, f_l$ be all vertex pairs of the first and the second graph respectively, that appear in Q. Clearly $k, l \leq |Q| = \frac{n}{4}$. Let $\sigma : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$ be the permutation according to which the second graph is chosen in D_P. Let E denote the event that $\sigma(e_i) \neq f_j$ for every $1 \leq i \leq k$ and $1 \leq j \leq l$, where for $e = \{u, v\}$ we denote by $\sigma(e)$ the pair $\{\sigma(u), \sigma(v)\}$. Clearly, if E occurs then $\{p_1, \ldots, p_{\frac{n}{4}}\}$ will be a set of $\frac{n}{4}$ uniformly and independently chosen random bits.

Claim 4.19. *The event E as defined above occurs with probability at least* $3/4$.

Proof. For a single pair e_i and a random permutation σ, the probability that $e_i = \sigma(f_j)$ for some $1 \leq j \leq l$ is bounded by $\frac{n}{2\binom{n}{2}}$. Hence by the union bound, $\Pr[E] \geq 1 - \frac{kn}{2\binom{n}{2}} > 3/4$. ∎

Since E occurs with probability at least $3/4$, and since the event upon which we conditioned D_N to get D'_N occurs with probability $1 - 2^{-\Omega(n^2)} = 1 - o(2^{-|\mathcal{Q}|})$, we get that for any $g : \mathcal{Q} \rightarrow \{0, 1\}$, we have $\Pr_{D'_N|\mathcal{Q}}[g] < \frac{3}{2}\Pr_{D_P|\mathcal{Q}}[g]$ and therefore the distributions D_P and D'_N satisfy the conditions of of Lemma 2.3.

∎

5 Concluding Remarks

While our two-sided error algorithms run in time quasi-polynomial in n (like the general approximation algorithm of [6]), the one-sided algorithms presented here require an exponential running time. It would be interesting to reduce the running time of the one-sided algorithms to be quasi-polynomial while still keeping them one-sided.

Another issue goes back to [1]. There, the graph isomorphism question was used to prove that certain first order graph properties are impossible to test with a constant number of queries. However, in view of the situation with graph isomorphism, the question now is whether every first order graph property is testable with $O(n^{2-\alpha})$ many queries for some $\alpha > 0$ that depends on the property to be tested.

Finally, it would be interesting to close the remaining gap between $\Omega(n)$ and $\widetilde{O}(n^{5/4})$ in the setting of two graphs that need to be queried, and a two-sided error algorithm. It appears (with the aid of martingale analysis on the same distributions D_P, D_N as above) that at least for non-adaptive algorithms the lower bound can be increased a little to a bound of the form $\Omega(n \log^\alpha n)$, but we are currently unable to give tighter bounds on the power of n.

Acknowledgements

We would like to thank the editor and two anonymous referees for their useful comments.

References

[1] N. Alon, E. Fischer, M. Krivelevich and M. Szegedy, Efficient testing of large graphs, *Combinatorica* 20 (2000), 451–476.

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

[3] N. Alon and A. Shapira, A Characterization of the (natural) Graph Properties Testable with One-Sided Error, *Proceedings of the* 46^{th} *IEEE FOCS* (2005), 429–438, Also *SIAM Journal on Computing*, to appear.

[4] N. Alon and A. Shapira, Every monotone graph property is testable, *Proceedings of the* 37^{th} *ACM STOC* (2005), 128–137, Also *SIAM Journal on Computing*, to appear.

[5] N. Alon and J. H. Spencer, *The probabilistic method*. Wiley-Interscience (John Wiley & Sons), New York, 1992 (1^{st} edition) and 2000 (2^{nd} edition).

[6] S. Arora, A. Frieze, and H. Kaplan, A new rounding procedure for the assignment problem with applications to dense graph arangement problems, *Mathematical programming* 92 (2002), 1–36.

[7] T. Batu, E. Fischer, L. Fortnow, R. Kumar, R. Rubinfeld and P. White, Testing random variables for independence and identity, *Proceedings of the* 42^{nd} *IEEE FOCS* (2001), 442–451.

[8] M. Blum, M. Luby and R. Rubinfeld, Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences* 47 (1993), 549–595 (a preliminary version appeared in Proc. 22^{nd} STOC, 1990).

[9] E. Fischer, The art of uninformed decisions: A primer to property testing, *Current Trends in Theoretical Computer Science: The Challenge of the New Century, G. Paun, G. Rozenberg and A. Salomaa (editors), World Scientific Publishing* (2004), Vol. I 229-264.

[10] E. Fischer, The difficulty of testing for isomorphism against a graph that is given in advance, *SIAM Journal on Computing* 34 (2005), 1147-1158.

[11] E. Fischer and I. Newman, Testing versus estimation of graph properties, *Proceedings of the* 37^{th} *ACM STOC* (2005), Also *SIAM Journal on Computing*, to appear.

[12] E. Fischer, I. Newman and J. Sgall, Functions that have read-twice constant width branching programs are not necessarily testable, *Random Structures and Algorithms* 24 (2004), 175–193.

[13] O. Goldreich, S. Goldwasser and D. Ron, Property testing and its connection to learning and approximation, *Journal of the ACM* 45 (1998), 653–750 (a preliminary version appeared in Proc. 37^{th} FOCS, 1996).

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

[15] P. Hajnal and M. Szegedy, On packing bipartite graphs. *Combinatorica* 12 (1992), 295–301.

[16] D. Ron, Property testing (a tutorial), In: *Handbook of Randomized Computing* (S. Rajasekaran, P. M. Pardalos, J. H. Reif and J. D. P. Rolim eds), Kluwer Press (2001), Vol. II Chapter 15.

[17] R. Rubinfeld and M. Sudan, Robust characterization of polynomials with applications to program testing, *SIAM Journal on Computing* 25 (1996), 252–271 (first appeared as a technical report, Cornell University, 1993).

[18] A. C. Yao, Probabilistic computation, towards a unified measure of complexity. *Proceedings of the* 18^{th} *IEEE FOCS* (1977), 222–227.