

A Linear Lower Bound on Testing Graph Isomorphism in the Bounded-Degree Model

Amit Dror*

Abstract

We prove a linear query complexity lower bound for two-sided testing of graph isomorphism in the bounded-degree model. Specifically, there are constants d and ε_0 such that, for every sufficiently large n , there exists an n -vertex graph H_n of maximum degree at most d such that testing isomorphism to H_n requires $\Omega(n)$ queries.

The proof proceeds by a reduction from testing the solvability of a fixed sparse system of linear equations over $\mathbb{F}_2$. Given a fixed matrix A and an input vector b , we associate b with a graph G_b such that G_b is isomorphic to G_0 whenever $Ax = b$ has a solution. We show that each query to the graph can be emulated with at most one query to b , and that Hamming distance from $\text{Im}(A)$ translates to the distance of G_b from G_0 . The main result then follows from a linear query lower bound for testing the solvability of the underlying (sparse) linear system.

1 Introduction

Property testing asks whether a large object exactly satisfies a desired property or is far from every object satisfying it, while inspecting only a small fraction of the object. In this paper, following [7, 9], we study the query complexity of graph isomorphism testing in the bounded-degree model, introduced by Goldreich and Ron [8] (see [5, Chap. 9] for further background).

We consider two testing tasks. In the *fixed-graph* task, an n -vertex target graph H is specified in advance, and the algorithm is given oracle access to an input graph G ; the goal is to test membership in the property $\Pi_H = \{G : G \cong H\}$, where $\cong$ denotes graph isomorphism. In the two-input task, the tester receives oracle access to two unknown graphs and must decide whether they are isomorphic.

The bounded-degree graph model Fixing a degree bound d , an n -vertex graph $G = ([n], E)$ of maximum degree at most d is accessed through an incidence oracle

$$g : [n] \times [d] \longrightarrow [n] \cup \{0\}.$$

For a vertex v , the value $g(v, i)$ is its i th neighbor, or 0 when v has fewer than i neighbors. We measure proximity by edge modifications: two graphs $G = ([n], E)$ and $G' = ([n], E')$ are said to be ε -far (for $\varepsilon > 0$) if $|E \Delta E'| > \varepsilon nd/2$. Otherwise, they are said to be ε -close.

A graph property Π is a set of graphs closed under isomorphism. An n -vertex graph G is said to be ε -far from Π if it is ε -far from every n -vertex graph in Π of maximum degree at most d .

An ε -tester for a property Π is a randomized algorithm with oracle access to the input graph. It accepts every graph in Π with probability at least $2/3$ and rejects every graph that is ε -far from

*Weizmann Institute of Science, Rehovot, Israel.

Π with probability at least $2/3$. Its query complexity is the maximum number of incidence queries it makes. We say that Π can be ε -tested using $q(n, \varepsilon)$ queries if such a tester exists with query complexity at most $q(n, \varepsilon)$.

Focusing on the property of isomorphism to a fixed target graph H , it is convenient to measure the unnormalized (*absolute*) distance of G from being isomorphic to H . For a bijection $\pi : V(G) \rightarrow V(H)$, we write $\pi^{-1}(E(H)) = \{\{u, v\} : \{\pi(u), \pi(v)\} \in E(H)\}$ for the edge set of H pulled back to $V(G)$. We define

$$\text{dist}_{\text{iso}}(G, H) := \min_{\pi: V(G) \rightarrow V(H) \text{ bijective}} |E(G) \triangle \pi^{-1}(E(H))|.$$

Thus, an n -vertex graph G is ε -far from Π_H (the property of being isomorphic to H) if and only if $\text{dist}_{\text{iso}}(G, H) > \varepsilon nd/2$.

For the two-input problem, the distance measure is defined analogously: a pair (G_1, G_2) is ε -far from being isomorphic if G_1 is ε -far from every graph isomorphic to G_2 , which holds if and only if $\text{dist}_{\text{iso}}(G_1, G_2) > \varepsilon nd/2$.

1.1 A lower bound for testing isomorphism in the bounded-degree model

Our main result establishes an asymptotically optimal linear lower bound for the fixed-graph problem.

Theorem 1.1 (Lower bound for the fixed-graph version). *There are constants d and ε_0 and a family of graphs $\{H_n\}_{n \in \mathbb{N}}$ such that for every sufficiently large n :*

1. H_n has n vertices and maximum degree at most d .
2. Every ε_0 -tester for isomorphism to the fixed graph H_n makes $\Omega(n)$ queries.

We note that this lower bound is asymptotically optimal in the bounded-degree model, because the whole incidence description contains $nd = O(n)$ entries.

Corollary 1.2 (Lower bound for the two-input version). *There are constants d and ε_0 such that, for every sufficiently large n , every ε_0 -tester for isomorphism between two n -vertex graphs of maximum degree at most d makes $\Omega(n)$ queries.*

Proof. The result follows by a straightforward reduction from the fixed-graph case to the two-input case. A tester for the two-input problem can be used when the second graph is fixed: we run it on (G, H_n) , forwarding its queries to G , and answering its queries to H_n by ourselves based on the known graph. $\square$

Remark 1.3. Our construction has maximum degree $d = 13$. On the other hand, a degree bound of at least three is fundamentally necessary to achieve any lower bound that grows with n . Indeed, Goldreich and Tauber [10] showed that when the maximum degree is at most two, every graph property can be ε -tested using $\text{poly}(1/\varepsilon)$ queries, independent of n . This leaves an intriguing open problem: determining the exact minimum degree threshold (between 3 and 13) at which testing graph isomorphism requires $\Omega(n)$ queries.

1.2 Related work

The problem of testing graph isomorphism has been considered before. In particular, determining the query complexity of isomorphism testing in the bounded-degree model was explicitly highlighted as an open problem by Goldreich [6].

In the *dense graph model*, Fischer and Matsliah [4] give an almost complete picture of the query complexity of both versions of the problem. In the bounded-degree model, Goldreich [7] proves, up to polylogarithmic factors, a $\tilde{\Theta}(\sqrt{n})$ bound for the fixed-graph problem and a $\tilde{\Theta}(n^{2/3})$ bound for the two-input problem when all connected components have size at most $\text{poly}(\log n)$. Goldreich and Tauber [9] prove a $\tilde{\Theta}(\sqrt{n})$ bound for testing isomorphism to almost every fixed regular graph.

The latter result makes the contrast with our result particularly striking. Whereas, with probability $1 - o(1)$, a uniformly random d -regular graph can be tested using $\tilde{O}(\sqrt{n})$ queries, our result exhibits carefully structured bounded-degree graphs for which every tester requires $\Omega(n)$ queries.

Our construction uses a family of graph gadgets that encode parity equations; our gadgets are similar to those used in [3].

1.3 Proof idea

The proof proceeds via a reduction from testing the solvability of a fixed sparse linear system (our “base problem”) to testing isomorphism to a fixed graph. We first establish a linear query lower bound for the base problem, and then demonstrate that any isomorphism tester can be used to solve the base problem with the same query complexity.

The hard base problem. Our lower bound is based on the following hard base problem: testing the solvability of a fixed *sparse* system of linear equations over $\mathbb{F}_2$. Specifically, we fix a binary matrix $A \in \mathbb{F}_2^{n \times k}$ and define the linear subspace

$$\mathcal{C} := \text{Im}(A) = \{Ax : x \in \mathbb{F}_2^k\}.$$

The input to the tester is a vector $b \in \mathbb{F}_2^n$, accessed via coordinate queries (i.e., each query reveals a single bit of b). The problem is to decide whether $Ax = b$ has a solution (i.e., $b \in \mathcal{C}$) or whether b is far (in Hamming distance) from every vector in $\mathcal{C}$.

To ensure this problem requires $\Omega(n)$ queries, we choose $k < n$ and construct A such that every set of at most δn rows (for some constant $\delta > 0$) is linearly independent. This guarantees that, for a uniformly random x , any subset of at most δn coordinates of Ax behaves like a set of independent uniform bits. Consequently, a sublinear-query algorithm cannot distinguish a random vector in $\mathcal{C}$ from a random vector in $\mathbb{F}_2^n$. Since $\mathcal{C}$ occupies only a tiny fraction of the vector space, a random vector is far from $\mathcal{C}$ with overwhelming probability.

Encoding a vector as a graph. With the lower bound for the base problem established, the proof proceeds via a (property testing) reduction to our main graph isomorphism problem. We associate each vector b with a graph G_b and define our fixed target graph as G_0 (where 0 denotes the zero vector). The reduction is designed so that any graph isomorphism tester for G_0 can be emulated to solve the base problem with the same asymptotic query complexity.

Each graph of our construction has one **variable-gadget** for each variable and one **constraint-gadget** for each equation. A variable-gadget contains two **value-vertices**, representing the values zero and one, while a constraint-gadget for equation j (row j of A) contains **assignment-vertices** corresponding to all assignments to its variables whose parity equals b_j . Edges connect each

assignment-vertex to the variable values that it specifies. We stress that the coordinate b_j affects only the edges between the j th constraint-gadget and its variable-gadgets.

If $b = Ax$, then swapping the two values of variable i according to x_i and relabeling the assignment-vertices in each constraint-gadget gives an isomorphism from G_b to G_0 . Moreover, the answer to any incidence query is either independent of b or depends on a single coordinate of b . We can therefore emulate a graph tester on G_b with at most one coordinate query to b per each query to the graph G_b .

Distance preservation. For property testing, it is not enough that a vector outside $\mathcal{C}$ produces a graph nonisomorphic to G_0 . Unlike standard Karp reductions, property testing reductions must also *preserve distance*: a vector b that is far from having a solution (i.e., far from $\mathcal{C}$) must produce a graph G_b that is far from being isomorphic to G_0 .

To see how the construction preserves distance, we first consider bijections that map each variable-gadget set-wise to another variable-gadget. From these bijections, we can extract an assignment x such that every coordinate where the vectors Ax and b differ reflects an edge mismatch. Thus, for these “variable-preserving” bijections, the distance of b from $\mathcal{C}$ lower bounds the number of edge mismatches.

The problem is that an arbitrary bijection need not respect the variable-gadgets. To deal with such bijections, we utilize the graph structure. We show that any bijection with some number of edge mismatches can be “repaired” into a “variable-preserving” bijection while increasing the number of edge mismatches only by a constant factor.

1.4 Organization

Section 2 establishes the lower bound for our base problem while constructing a sparse matrix with the required features. Section 3 presents the graph construction, proves distance preservation, and completes the proof of the main theorem.

2 Testing solvability of a linear system

We fix a matrix $A \in \mathbb{F}_2^{n \times k}$. Given access to an input vector $b \in \mathbb{F}_2^n$, we ask whether the system $Ax = b$ has a solution or whether b is far from any vector for which the system has a solution. Testing whether this system has a solution is the same as testing membership in the linear subspace $\mathcal{C} = \text{Im}(A)$. This testing task requires distinguishing the case where the input b is in $\mathcal{C}$ from the case where b is far from every vector in $\mathcal{C}$. The distance of b from having a solution is defined by

$$\text{dist}(b, \text{Im}(A)) := \min_{x \in \mathbb{F}_2^k} |\{j \in [n] : (Ax)_j \neq b_j\}|.$$

While the existence of linear subspaces for which membership testing requires $\Omega(n)$ queries is well-known (e.g., Ben-Sasson et al. [1]), we provide a direct proof to guarantee a specific sparse representation of $\mathcal{C} = \text{Im}(A)$ such that A has bounded row and column weight.

The lower bound uses two features of A . First, we require $k \leq (1 - \rho) \cdot n$ for a constant $\rho > 0$. Since $|\text{Im}(A)| \leq 2^k$, the image occupies at most a $2^{-\rho n}$ fraction of $\mathbb{F}_2^n$, which implies that a random vector is far from it, with overwhelming probability. Second, every set of $\Omega(n)$ rows of A should be linearly independent. This makes any few coordinates of Ax , for a random x , look like independent random bits. Hence, distinguishing between a random vector and one that has a solution is impossible without inspecting more coordinates.

Claim 2.1 (Random looking). *Let $A \in \mathbb{F}_2^{n \times k}$, and suppose every set of at most q rows of A is linearly independent. Then, every adaptive algorithm making at most q queries has the same transcript distribution in the following two cases:*

1. x is chosen uniformly at random from $\mathbb{F}_2^k$ and the queried vector is Ax ;
2. the queried vector is chosen uniformly at random from $\mathbb{F}_2^n$.

Proof. We fix the algorithm's internal randomness and consider the resulting decision tree. At a node reached after answers on a set S of distinct coordinates, suppose the next query is a new coordinate j . Since $|S \cup \{j\}| \leq q$, the rows indexed by $S \cup \{j\}$ are independent. Therefore, when x ranges uniformly over $\mathbb{F}_2^k$, the vector $(Ax)_{S \cup \{j\}}$ takes every value equally often. Conditioned on the previous answers, the next answer is therefore an unbiased bit, just as it is for a random vector in $\mathbb{F}_2^n$. $\square$

Theorem 2.2 (Hardness of image membership). *Fix constants $\rho, \delta \in (0, 1)$. There exists a constant $\beta > 0$ such that the following holds for every sufficiently large n . Let $A \in \mathbb{F}_2^{n \times k}$ satisfy $k \leq (1 - \rho) \cdot n$, and suppose every set of at most δn rows of A is linearly independent. Then, every tester that distinguishes the case $b \in \text{Im}(A)$ from the case where $\text{dist}(b, \text{Im}(A)) > \beta n$ requires more than δn queries.*

Proof. Choose $\beta > 0$ sufficiently small such that the volume of a Hamming ball of radius βn in $\mathbb{F}_2^n$ is strictly less than $2^{\rho n - 2}$. Since $|\text{Im}(A)| \leq 2^{(1 - \rho)n}$, a union bound implies that a uniformly random vector in $\mathbb{F}_2^n$ is β -far from $\text{Im}(A)$ with probability at least $3/4$. Thus, a tester accepts a uniformly random vector with probability at most $\frac{3}{4} \cdot \frac{1}{3} + \frac{1}{4} \cdot 1 = \frac{1}{2}$.

Assume towards contradiction that a tester makes at most δn queries. By Claim 2.1, the tester cannot distinguish a random vector in $\text{Im}(A)$ from a random vector in $\mathbb{F}_2^n$. Because the tester must accept vectors in $\text{Im}(A)$ with probability at least $2/3$, it must also accept a random vector in $\mathbb{F}_2^n$ with probability at least $2/3 > 1/2$, which is a contradiction. $\square$

Theorem 2.2 is only useful if such matrices exist. We next show this is indeed the case and that we can even require additional features.

2.1 A hard sparse system

The preceding theorem does not require A to be sparse, but our reduction requires bounded row and column weights. We show such matrices exist for every sufficiently large (even) n .

Lemma 2.3. *There are constants $\delta > 0$ and n_0 such that, for every even $n \geq n_0$, there is a binary matrix $A \in \mathbb{F}_2^{n \times (n/2)}$ with row weight three and column weight six such that every set of at most δn rows of A is linearly independent.*

Proof. The proof proceeds in two main parts: first, we construct A as the adjacency matrix of a biregular bipartite graph that exhibits a “unique neighbor expansion” feature, and second, we show how this feature is used to deduce the required row independence.

We begin with the graph construction. Let $(L \sqcup R, E)$ be a bipartite graph with $|L| = n$, $|R| = n/2$, degree three on L , and degree six on R (see Figure 1(a)). We require that for a (small) constant $\delta > 0$, the graph is sufficiently expanding:

$$|N(S)| > \frac{3}{2}|S| \quad \text{for any non-empty } S \subseteq L \text{ such that } |S| \leq \delta n. \quad (1)$$

This specific level of expansion guarantees that the set S will have “unique neighbors”. A **unique neighbor** of S is a vertex in R that is adjacent to exactly one vertex in S . Indeed, since every vertex in S has degree three, there are exactly $3|S|$ edges leaving S . If no unique neighbor existed, then every vertex in the neighborhood $N(S)$ would be incident to at least two edges from S , which would imply $|N(S)| \leq \frac{3}{2}|S|$, contradicting Equation (1). Figure 1(b) illustrates this argument.

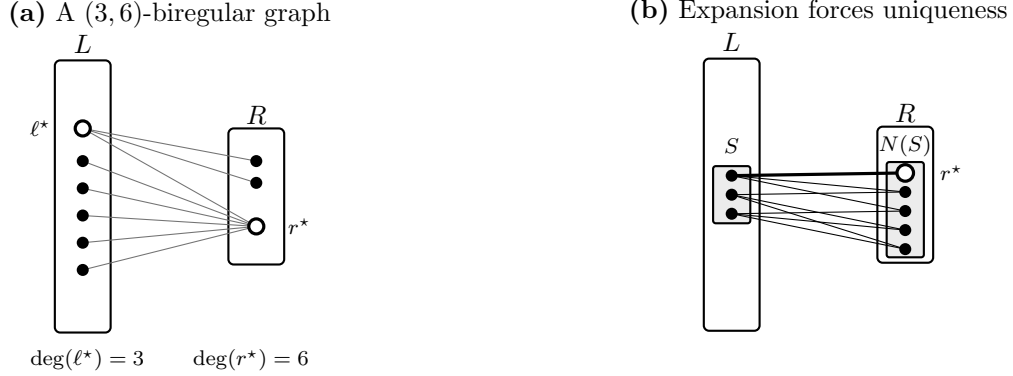


Figure 1: In (a), vertices in L and R have degrees three and six, respectively. Part (b) shows how expansion forces a unique neighbor. Since the set S is expanding (see Equation (1)), its neighborhood must contain at least five vertices. On the other hand, the degree of each vertex in S is three. Therefore, there are only nine edges leaving S . Hence, it must be the case that at least one vertex of $N(S)$ is a unique neighbor.

We establish the expansion feature of Equation (1) using the probabilistic method. Specifically, we use the configuration model [2]. We pair uniformly at random the $3n$ half-edges on L with the $3n$ half-edges on R . We fix $S \subseteq L$ with $|S| = s \geq 2$. If Equation (1) fails for S , then all edges leaving S enter some fixed set $T \subseteq R$ of size at most $t = 3s/2$. For fixed subsets S and T , the probability of this event is at most

$$\frac{\binom{6t}{3s}}{\binom{3n}{3s}} = \frac{\binom{9s}{3s}}{\binom{3n}{3s}},$$

because the partners of the $3s$ half-edges incident to S form a uniformly distributed $3s$ -subset of the $3n$ half-edges on R . A union bound over all choices of S , T , and $2 \leq s \leq \delta n$ bounds the probability of any violation by

$$\sum_{s=2}^{\delta n} \binom{n}{s} \cdot \binom{n/2}{3s/2} \cdot \frac{\binom{9s}{3s}}{\binom{3n}{3s}}.$$

We bound each summand as follows:

$$\begin{aligned} \binom{n}{s} \cdot \binom{n/2}{3s/2} \cdot \frac{\binom{9s}{3s}}{\binom{3n}{3s}} &= O\left(\frac{1}{s}\right) \cdot \exp\left(H\left(\frac{s}{n}\right) \cdot n + H\left(\frac{3s/2}{n/2}\right) \cdot \frac{n}{2} + H\left(\frac{1}{3}\right) \cdot 9s - H\left(\frac{s}{n}\right) \cdot 3n\right) \\ &\leq \exp\left(-H\left(\frac{s}{n}\right) \cdot 2n + H\left(\frac{3s}{n}\right) \cdot \frac{n}{2} + H\left(\frac{1}{3}\right) \cdot 9s\right) \\ &\leq \exp\left(-H\left(\frac{s}{n}\right) \cdot \frac{n}{2} + O(s)\right) \\ &\leq \exp\left(-s \cdot \left(\frac{1}{2} \log\left(\frac{n}{s}\right) - O(1)\right)\right) \end{aligned}$$

For a sufficiently small constant $\delta > 0$, since $s \leq \delta n$, the $\log(n/s)$ term dominates the $O(1)$ constant. The union bound sum over $s \geq 2$ is therefore bounded by a geometric series and hence dominated by its initial term, which is $O(1/n)$. Thus, with probability $1 - o(1)$, there is no bad set $S \subseteq L$ of size at most δn .

We are not done yet, as the configuration model allows parallel edges, whereas we require our adjacency matrix to have boolean entries. The enumeration formula of McKay and Wang shows that the probability that the configuration is simple (i.e., contains no parallel edges or self-loops) tends to $e^{-(3-1)(6-1)/2} = e^{-5}$ [12]. Therefore, the probability that the configuration is both simple and free of bad sets is at least $e^{-5} - o(1) > 0$. Hence, for every sufficiently large even n , there is a simple graph with the required degrees and expansion.

Now, let A be the bipartite adjacency matrix of the foregoing simple graph, with rows indexed by L and columns by R . In terms of this matrix, the unique neighbor expansion feature means that for any non-empty set of at most δn rows, there is at least one column that contains exactly a single 1 in these rows.

We now deduce the required row independence over $\mathbb{F}_2$. If a non-empty set S of at most δn rows summed to the zero vector over $\mathbb{F}_2$, then this requires that for every column, the sum of the entries in the rows corresponding to S is even. However, the unique neighbor feature guarantees the existence of a column with exactly a single 1 among the rows of S . $\square$

Remark 2.4. The exact row and column weights (three and six, respectively) are convenient but not essential. Bounded row and column weights would have sufficed. We use the biregular $(3, 6)$ choice for simplicity.

3 The graph construction

Let $A \in \mathbb{F}_2^{n \times k}$ be the matrix from Lemma 2.3, where $k = n/2$. For any vector $b \in \mathbb{F}_2^n$, we construct a graph G_b . We require this family of graphs to satisfy three features:

1. if $b \in \text{Im}(A)$ (i.e., if the system $Ax = b$ has a solution), then $G_b \cong G_0$;
2. each incidence query to G_b can be answered by reading at most one bit of b ;
3. if b is far from $\text{Im}(A)$, then G_b is far from every copy of G_0 .

The first feature establishes the completeness of the reduction, the second preserves its query complexity, and the third guarantees soundness. We next describe the construction and then prove these three features.

3.1 The construction

Each row of A corresponds to an equation over three variables and is represented by a *constraint-gadget*. Each column is a variable and is represented by a *variable-gadget*. The edges between constraint- and variable-gadgets encode the right-hand side value (i.e., b) of the system $Ax = b$. Specifically, for every $j \in [n]$, fix an ordering of the variables in the j th constraint and denote by $C_j = \{c_j(1), c_j(2), c_j(3)\} \subseteq [k]$ the three variables that participate in the j th constraint. The j th equation (row) of $Ax = b$ is therefore

$$\sum_{\ell=1}^3 x_{c_j(\ell)} = b_j.$$

Let $P_0 = \{000, 011, 101, 110\}$ and $P_1 = \{111, 100, 010, 001\}$ be the four assignments of even and odd parity, respectively. For $i \in [k]$, variable-gadget i consists of a *variable-anchor* $v_{(b,i)}^\natural$ and two *value-vertices* $v_{(b,i)}^0$ and $v_{(b,i)}^1$, representing the possible values for x_i . For $j \in [n]$, constraint-gadget j consists of a *constraint-anchor* $a_{(b,j)}^\natural$ and four *assignment-vertices* $\{a_{(b,j)}^\tau : \tau \in P_{b_j}\}$. The label of an assignment-vertex in constraint-gadget j represents an assignment to the constraint's three variables, and its parity is b_j . We include b in the notation for each vertex in order to identify the graph to which the vertex belongs. When that graph is clear from context, we suppress the graph identifier and use the shorthand

$$v_i^\natural = v_{(b,i)}^\natural, \quad v_i^\sigma = v_{(b,i)}^\sigma, \quad a_j^\natural = a_{(b,j)}^\natural, \quad a_j^\tau = a_{(b,j)}^\tau.$$

We use the fully qualified notation whenever vertices of two graphs are discussed together.

Edges connect anchors to the vertices of their gadgets; we call these *spoke-edges*:

$$\begin{aligned} \{v_i^\natural, v_i^\sigma\} & \text{ for every } i \in [k] \text{ and } \sigma \in \mathbb{F}_2, \\ \{a_j^\natural, a_j^\tau\} & \text{ for every } j \in [n] \text{ and } \tau \in P_{b_j}. \end{aligned}$$

Finally, for every $j \in [n]$, $\tau \in P_{b_j}$, and $\ell \in [3]$, we add the edge

$$\{a_j^\tau, v_{c_j(\ell)}^{\tau_\ell}\}.$$

In words, each assignment-vertex is connected to the specific value-vertices that match the boolean values it represents. For an assignment-vertex labeled τ in constraint-gadget j , an edge is added between that assignment-vertex and the corresponding value-vertex $v_{c_j(\ell)}^{\tau_\ell}$ (where $c_j(\ell)$ is the ℓ th variable in constraint j). This wiring links each constraint to the exact variable values that satisfy it. Figure 2 shows one constraint-gadget and the three variable-gadgets the constraint involves.

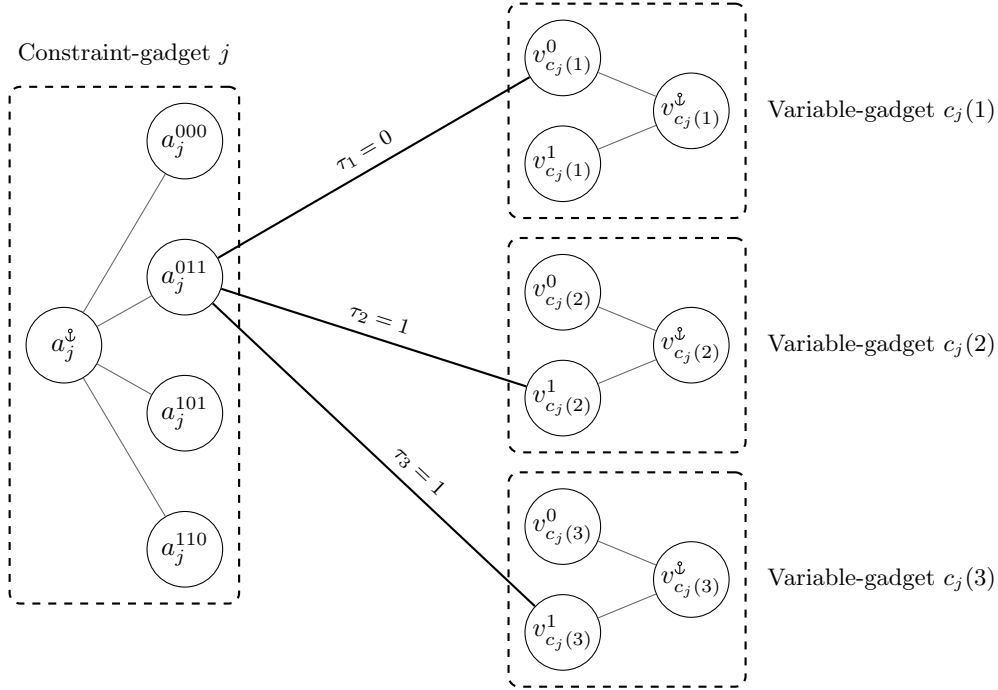


Figure 2: The local gadget structure involving constraint j in G_b , assuming $b_j = 0$. The constraint-gadget consists of its anchor and four assignment-vertices, and each variable-gadget consists of its anchor and two value-vertices. For readability, constraint-to-variable edges are shown only for a_j^{011} .

Since $k = n/2$, the graph has $N = (4 + 1) \cdot n + (2 + 1) \cdot k = 13n/2$ vertices. We also note that it has bounded degree. Each value-vertex has degree 13: each of the six constraints containing the corresponding variable contributes the two assignments that select that value, and there is one additional edge to its variable-anchor. Each variable-anchor has degree two. Each assignment-vertex has degree four, since it is connected to the three value-vertices it specifies and to its constraint-anchor; each constraint-anchor also has degree four. Thus, the maximum degree is $d = 13$.

3.2 Completeness

Recall that the first feature we required is that if $b \in \text{Im}(A)$, then $G_b \cong G_0$. If the system $Ax = b$ has a solution x , then this solution specifies how to map G_0 to G_b . For every variable i such that $x_i = 1$, we swap the 0 and 1 value-vertices of that variable's gadget. These local variable flips shift the required parity of every constraint-gadget j from 0 to b_j . We then map the assignment-vertices as expected; that is, each assignment-vertex is mapped to match its new (possibly after bit-flips) variable's values. Details are provided in the following proof.

Lemma 3.1. *If $b \in \text{Im}(A)$, then $G_b \cong G_0$.*

Proof. Let $x \in \mathbb{F}_2^k$ be such that $Ax = b$. For each constraint j , define the local assignment to the j th constraint by $\mathbf{x}_j := (x_{c_j(1)}, x_{c_j(2)}, x_{c_j(3)})$. The parity of $\mathbf{x}_j$ is $(Ax)_j = b_j$. Therefore, translation by $\mathbf{x}_j$ (i.e., the mapping $y \mapsto y + \mathbf{x}_j$) constitutes a bijection from P_0 onto P_{b_j} .

In order to show G_0 is isomorphic to G_b , we show there exists an isomorphism π between them. Define $\pi : V(G_0) \rightarrow V(G_b)$ by setting

$$\pi(v_{(0,i)}^\flat) = v_{(b,i)}^\flat, \quad \pi(v_{(0,i)}^\sigma) = v_{(b,i)}^{\sigma+x_i} \quad (2)$$

and

$$\pi(a_{(0,j)}^\flat) = a_{(b,j)}^\flat, \quad \pi(a_{(0,j)}^\tau) = a_{(b,j)}^{\tau+\mathbf{x}_j}. \quad (3)$$

In words, for the variable-gadgets (Equation (2)), π shifts the binary label of each value-vertex according to the corresponding solution bit x_i . For the constraint-gadgets (Equation (3)), π shifts the assignment-vertices of each constraint by the local assignment $\mathbf{x}_j$.

As discussed, translation by $\mathbf{x}_j$ is a bijection from P_0 onto P_{b_j} . Therefore, $\pi(a_{(0,j)}^\tau) = a_{(b,j)}^{\tau+\mathbf{x}_j}$ is a valid vertex in G_b (i.e., the parity of $\tau + \mathbf{x}_j$ is exactly b_j). Moreover, π is a bijection between the vertex sets; in fact, it is a bijection on the assignment-vertices, value-vertices, and each of the two anchor classes. Because the two graphs have the same number of edges, it suffices to show that π maps every edge of G_0 to an edge of G_b .

Clearly, all spoke-edges are preserved by π . Consider an arbitrary constraint-to-variable edge $\{a_{(0,j)}^\tau, v_{(0,c_j(\ell))}^{\tau_\ell}\}$ in G_0 . Under π , this edge is mapped to $\{a_{(b,j)}^{\tau+\mathbf{x}_j}, v_{(b,c_j(\ell))}^{\tau_\ell+x_{c_j(\ell)}}\}$. Let $\tau' = \tau + \mathbf{x}_j$. Since the ℓ th coordinate of $\mathbf{x}_j$ is $x_{c_j(\ell)}$, we have $\tau'_\ell = \tau_\ell + x_{c_j(\ell)}$. Thus, the mapped edge is exactly $\{a_{(b,j)}^{\tau'}, v_{(b,c_j(\ell))}^{\tau'_\ell}\}$, which is by definition an edge in G_b . Hence, π is an isomorphism. $\square$

3.3 Locality: emulating a graph query

To establish the query complexity in the bounded-degree model, we define the incidence oracle g_b . Recall that the labels for assignment-vertices depend on b (the labels consist of $\tau \in P_{b_j}$). For the incidence oracle, however, it will be convenient to use a fixed vertex set independent of b . We achieve

this by denoting the assignment-vertices for each constraint $j \in [n]$ strictly over the even-parity assignments, denoted $\{a_j^\alpha : \alpha \in P_0\}$. Now, the assignment represented by an assignment-vertex a_j^α in the graph G_b is actually $(\alpha_1 + b_j, \alpha_2 + b_j, \alpha_3 + b_j)$.

With this fixed set of vertices, we define the incidence oracle $g_b(v, p)$, which returns the neighbor of vertex v along port p as follows:

- **Assignment-vertex ports:** For an assignment-vertex a_j^α , we connect ports $\ell \in \{1, 2, 3\}$ to its three variables:

$$g_b(a_j^\alpha, \ell) = v_{c_j(\ell)}^{\alpha_\ell + b_j}. \quad (4)$$

Port 4 is connected to the constraint-anchor: $g_b(a_j^\alpha, 4) = a_j^\alpha$.

- **Value-vertex ports:** For a value-vertex v_i^σ , we recall that variable i appears in six constraints. We fix an ordering of the six occurrences of variable i , and let the m th occurrence be in constraint j_m at position ℓ_m . Let $\alpha^{(m, \sigma, 1)}, \alpha^{(m, \sigma, 2)} \in P_0$ be the two assignments satisfying $\alpha_{\ell_m} + b_{j_m} = \sigma$, ordered lexicographically. We connect ports $2m - 1$ and $2m$ to these edges:

$$g_b(v_i^\sigma, 2m - 1) = a_{j_m}^{\alpha^{(m, \sigma, 1)}} \quad \text{and} \quad g_b(v_i^\sigma, 2m) = a_{j_m}^{\alpha^{(m, \sigma, 2)}}. \quad (5)$$

Port 13 is connected to the variable-anchor: $g_b(v_i^\sigma, 13) = v_i^\sigma$.

- **Anchor ports:** We set $g_b(v_i^\sigma, 1) = v_i^0$ and $g_b(v_i^\sigma, 2) = v_i^1$. At a_j^α , ports 1 through 4 are connected to the four assignment-vertices a_j^α , $\alpha \in P_0$, in lexicographic order.

With the oracle explicitly defined, we can now show that its evaluation is extremely local.

Claim 3.2 (One-bit emulation). *Each incidence query to G_b can be answered by inspecting at most one bit of b .*

Proof. Consider the evaluation of the incidence oracle g_b . For a query to an assignment-vertex a_j^α along port $\ell \in [3]$, Equation (4) shows that the returned value-vertex depends on the variable index $c_j(\ell)$ and the binary value $\alpha_\ell + b_j$. The latter binary value requires obtaining only the single bit b_j .

For a query to a value-vertex v_i^σ along port $p \in [12]$, with the corresponding occurrence index m , the constraint index j_m , and the position ℓ_m entirely determined by i and p . As shown in Equation (5), finding the correct target vertices requires identifying $\alpha \in P_0$ such that $\alpha_{\ell_m} = \sigma + b_{j_m}$ and this requires obtaining only the single bit b_{j_m} .

Queries along spoke-edges require no access to b , since their answers are fixed independently of b . $\square$

3.4 Distance preservation

The remaining question is whether Hamming distance from $\text{Im}(A)$ translates to distance from G_0 . Recall that graph isomorphism distance is obtained by aligning the vertex sets of two graphs with a bijection so to minimize the resulting number of edge mismatches. Accordingly, for a bijection $\pi : V(G_0) \rightarrow V(G_b)$, we denote its *number of edge mismatches* by

$$\Delta_b(\pi) := |\pi(E(G_0)) \triangle E(G_b)|.$$

Thus,

$$\text{dist}_{\text{iso}}(G_b, G_0) = \min_{\pi \text{ bijection}} \Delta_b(\pi).$$

Recall that the graph G_b is said to be ε -far from being isomorphic to the graph G_0 exactly when $\text{dist}_{\text{iso}}(G_b, G_0) > \varepsilon \cdot (N \cdot d/2)$. Hence, for soundness, our goal is to lower bound $\Delta_b(\pi)$ (for every bijection π) by the distance of the vector b from $\text{Im}(A)$. Actually, we first focus on a specific family of bijections.

We call a bijection $\nu : V(G_0) \rightarrow V(G_b)$ **variable-preserving** if there is a permutation ρ of $[k]$ such that

$$\nu(v_{(0,i)}^\circ) = v_{(b,\rho(i))}^\circ \text{ and } \nu(\{v_{(0,i)}^0, v_{(0,i)}^1\}) = \{v_{(b,\rho(i))}^0, v_{(b,\rho(i))}^1\} \quad (\forall i \in [k]). \quad (6)$$

That is, ν maps variable-gadgets (set-wise) to other variable-gadgets. Moreover, it maps variable-anchors only to other variable-anchors. We stress that ν need not be the identity: entire variable-gadgets may move, and the two value-vertices within each gadget may be swapped; moreover, ν may act arbitrarily on constraint-gadget vertices. Indeed, such variable-preserving bijections preserve the structure of each variable-gadget (hence their name). We show that for variable-preserving bijections, the distance of the vector b from $\text{Im}(A)$ directly lower bounds the number of edge mismatches.

Lemma 3.3 (Distance preservation under a variable-preserving bijection). *For every variable-preserving bijection $\nu : V(G_0) \rightarrow V(G_b)$, $\text{dist}(b, \text{Im}(A)) \leq \Delta_b(\nu)$.*

Proof. The idea is to use ν to extract a candidate assignment x such that Ax is close to b whenever ν has few edge mismatches. Let ρ be the permutation induced by ν on the variable-gadgets. For each variable i , variable preservation (Equation (6)) gives a unique bit $x_{\rho(i)}$ such that

$$\nu(v_{(0,i)}^\sigma) = v_{(b,\rho(i))}^{\sigma+x_{\rho(i)}} \quad (\forall \sigma \in \mathbb{F}_2). \quad (7)$$

In words, when mapping variable-gadget i of G_0 to variable-gadget $\rho(i)$ of G_b , the bijection ν either preserves or swaps the two binary labels of the value-vertices. We set $x_{\rho(i)} = 0$ if ν preserves the labels and $x_{\rho(i)} = 1$ if it swaps them. Roughly speaking, x can be viewed as transporting the all-zero assignment of G_0 through ν . Since the all-zero assignment satisfies every constraint of G_0 , we expect x to satisfy every constraint of G_b . We will show that each constraint of G_b unsatisfied by x forces an edge mismatch.

Let $U := \{j \in [n] : (Ax)_j \neq b_j\}$ be the set of constraints unsatisfied by x . We will show that each constraint in U forces an edge mismatch. Fixing $j \in U$ and an assignment-vertex $a_{(b,j)}^\tau$, we show that $a_{(b,j)}^\tau$ is incident to an edge mismatch. We denote $s := \nu^{-1}(a_{(b,j)}^\tau)$ for the preimage of $a_{(b,j)}^\tau$ under ν . The assignment-vertex $a_{(b,j)}^\tau$ has three value-vertex neighbors, which we denote by $v_{(b,c_j(\ell))}^{\tau_\ell}$ for $\ell \in [3]$. We claim that at least one of their preimages is not adjacent to s . In other words, at least one of the three edges incident to $a_{(b,j)}^\tau$ and its value-vertices does not belong to $\nu(E(G_0))$. Figure 3 illustrates our claim.

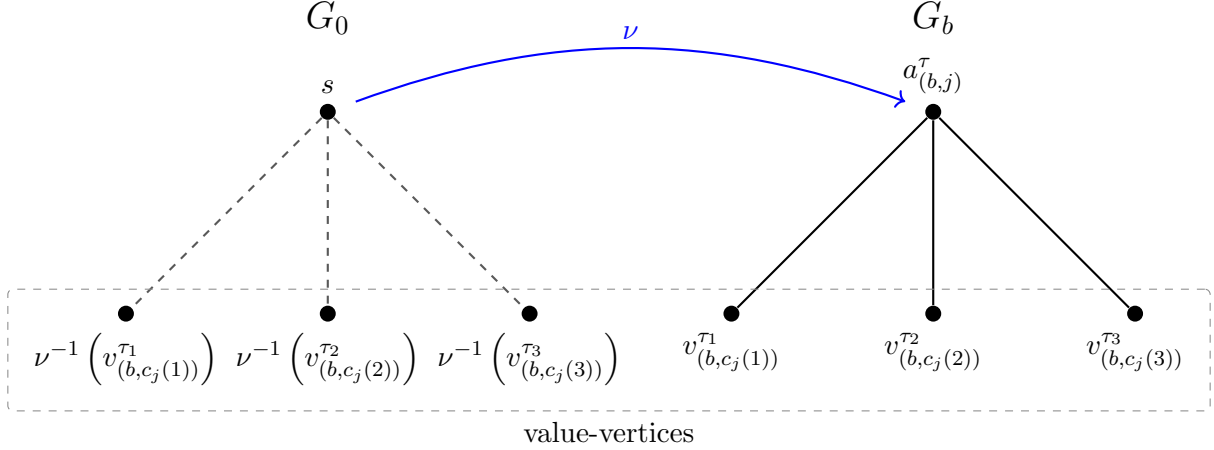


Figure 3: In G_b , vertex $a_{(b,j)}^\tau$ is connected to its three value-vertices. In G_0 , we see the preimages of $a_{(b,j)}^\tau$ and its three value-vertices. We note that since ν is variable-preserving, the preimages of the value-vertices must also be value-vertices of G_0 . For $j \in U$, our goal is to show that at least one of the dashed lines is not an edge of G_0 .

Suppose, toward a contradiction, that all three preimages (i.e., $\nu^{-1}(v_{(b,c_j(\ell))}^{\tau_\ell})$ for $\ell \in [3]$) are adjacent to s . Since ν is variable-preserving, these preimages are value-vertices of G_0 . Moreover, since $\nu(s) = a_{(b,j)}^\tau$ is an assignment-vertex, it must be the case that s belongs to a constraint-gadget. Within a constraint-gadget, constraint-anchors have no value-vertex neighbors; hence, we conclude that s is an assignment-vertex and the three preimages are its value-vertices. Since every assignment-vertex in G_0 represents an even-parity assignment, the binary labels of these three preimages sum to zero. By Equation (7), the three preimages are

$$\nu^{-1}(v_{(b,c_j(\ell))}^{\tau_\ell}) = v_{(0,\rho^{-1}(c_j(\ell)))}^{\tau_\ell + x_{c_j(\ell)}} \quad (\ell \in [3]).$$

Thus

$$0 = \sum_{\ell=1}^3 (\tau_\ell + x_{c_j(\ell)}) = \sum_{\ell=1}^3 \tau_\ell + \sum_{\ell=1}^3 x_{c_j(\ell)} = b_j + (Ax)_j$$

Thus, $(Ax)_j = b_j$, contradicting $j \in U$.

We conclude that $a_{(b,j)}^\tau$ is incident to at least one edge mismatch. Choose one such edge for one assignment-vertex of each constraint in U . These edges are distinct because they are incident to distinct vertices. Therefore,

$$\text{dist}(b, \text{Im}(A)) \leq |U| \leq \Delta_b(\nu)$$

□

Lemma 3.3 establishes the desired bound for variable-preserving bijections. However, graph isomorphism distance is defined by minimizing over *all* bijections. This introduces a problem: what if an arbitrary bijection reduces the number of edge mismatches by splitting the vertices of a variable-gadget? To establish a lower bound, we show that these non-variable-preserving bijections offer no real benefit.

The core idea is to show that any bijection is penalized (in edge mismatches) if it attempts to split variable-gadgets. Splitting a variable gadget will “break” the spoke edges that connect the gadget, hence introducing edge mismatches. This allows us to conclude that an arbitrary bijection with a small number of edge mismatches essentially behaves like a variable-preserving one.

The proof proceeds by two “repair” steps. At each step, we modify the bijection while increasing the number of edge mismatches by at most a constant factor. When we are done, we get a variable-preserving bijection with a number of edge mismatches that is only a constant factor larger than the original bijection. We conclude that arbitrary bijections offer no (asymptotic) advantage over variable-preserving ones. Hence, we can use the lower bound from Lemma 3.3.

For the first repair, we call a bijection **role-preserving** if it maps variable-anchors onto variable-anchors and value-vertices onto value-vertices. We stress that such a bijection may still split the vertices of a variable-gadget among several gadgets and has no guarantee regarding constraint-gadget vertices.

Claim 3.4 (Recovering the variable roles). *For every bijection $\pi : V(G_0) \rightarrow V(G_b)$, there is a role-preserving bijection $\mu : V(G_0) \rightarrow V(G_b)$ such that $\Delta_b(\mu) \leq (1 + 4d) \cdot \Delta_b(\pi)$.*

Proof. Partition the vertex set of each of the graphs into three classes: variable-anchors, value-vertices, and the remaining vertices. The degrees within each vertex class are 2, 13, and 4, respectively. Let S be the set of vertices in G_0 whose class differs from that of their image under π . For every $u \in S$, the degrees of u and $\pi(u)$ differ, so at least one edge mismatch is incident to $\pi(u)$. Since each edge mismatch has two endpoints, $|S| \leq 2\Delta_b(\pi)$.

Changing a bijection on t vertices affects only the images of edges incident to those vertices, of which there are at most dt . Applying this with $t = |S| \leq 2\Delta_b(\pi)$, we obtain a role-preserving bijection μ that differs from π only on S such that

$$\Delta_b(\mu) \leq \Delta_b(\pi) + 2d|S| \leq (1 + 4d) \cdot \Delta_b(\pi).$$

□

Claim 3.5 (Recovering the variable gadgets). *For every role-preserving bijection $\mu : V(G_0) \rightarrow V(G_b)$, there is a variable-preserving bijection $\nu : V(G_0) \rightarrow V(G_b)$ such that $\Delta_b(\nu) \leq (1 + 4d) \cdot \Delta_b(\mu)$.*

Proof. Since μ is role-preserving, there exists a permutation ρ of the variable-anchors, defined by $\mu(v_{(0,i)}^\natural) = v_{(b,\rho(i))}^\natural$. Call variable i *good* if both of its spoke-edges are mapped to edges of G_b . Since the only neighbors of variable-anchor $v_{(b,\rho(i))}^\natural$ are the value-vertices $v_{(b,\rho(i))}^0$ and $v_{(b,\rho(i))}^1$, a good variable satisfies

$$\mu(\{v_{(0,i)}^0, v_{(0,i)}^1\}) = \{v_{(b,\rho(i))}^0, v_{(b,\rho(i))}^1\}.$$

Let B be the set of variables that are not good. Each variable in B contributes a distinct spoke-edge that is mapped to a non-edge, so $|B| \leq \Delta_b(\mu)$.

Let S contain the two value-vertices of every variable in B . We keep μ unchanged outside S , and remap the pair of values of each $i \in B$ bijectively onto the variable value pair indexed by $\rho(i)$. This defines a bijection ν that is variable-preserving and differs from μ on at most $|S| = 2|B| \leq 2\Delta_b(\mu)$ vertices. As in the previous claim, we get

$$\Delta_b(\nu) \leq \Delta_b(\mu) + 2d|S| \leq (1 + 4d) \cdot \Delta_b(\mu).$$

□

We can now combine the two repairs with Lemma 3.3 to obtain distance preservation under arbitrary bijections.

Corollary 3.6 (Distance transfer). *There exists a constant K such that, for every $b \in \mathbb{F}_2^n$ and every bijection $\pi : V(G_0) \rightarrow V(G_b)$, $\text{dist}(b, \text{Im}(A)) \leq K \cdot \Delta_b(\pi)$.*

Proof. Apply Claims 3.4 and 3.5 successively to obtain a variable-preserving bijection ν satisfying

$$\Delta_b(\nu) \leq K \cdot \Delta_b(\pi), \quad \text{for } K := (1 + 4d)^2.$$

Lemma 3.3 then gives

$$\text{dist}(b, \text{Im}(A)) \leq \Delta_b(\nu) \leq K \cdot \Delta_b(\pi).$$

□

3.5 Proof of the main theorem

Corollary 3.6 provides the soundness needed to complete the reduction. Let $\beta > 0$ be the constant from Theorem 2.2. If $\text{dist}(b, \text{Im}(A)) > \beta n$, then Corollary 3.6 implies that every bijection $\pi : V(G_0) \rightarrow V(G_b)$ satisfies $\Delta_b(\pi) > \beta n/K$. Since G_b has $N = 13n/2$ vertices, we set

$$\varepsilon_0 := \frac{4\beta}{13dK}.$$

Then, G_b is ε_0 -far from being isomorphic to G_0 whenever b is βn -far from $\text{Im}(A)$.

Proof of Theorem 1.1. Completeness follows from Lemma 3.1, soundness from Corollary 3.6, and one-bit query emulation from Claim 3.2. Thus, an ε_0 -tester for isomorphism to G_0 making q queries can be emulated to give a β -tester for membership in $\text{Im}(A)$ making at most q queries. By Theorem 2.2 and Lemma 2.3, there exists a constant δ such that $q > \delta n = \Omega(N)$. □

Remark 3.7 (Arbitrary graph sizes). We note that our construction produces graphs of sizes that are multiples of 13. In order to allow any (sufficiently large) graph size, one can augment the construction with a bounded-size gadget (for example, connecting a path of at most 13 vertices to one of the anchors). This constant-size gadget will have negligible effect on the analysis above, allowing us to conclude that the lower bound holds for any sufficiently large n .

Remark 3.8 (Connected graphs). Our result may be strengthened by allowing the tester to assume the graph is connected. To show the lower bounds also hold for connected graphs, consider adding a fixed cycle through the constraint-anchors $a_1^\$, \dots, a_n^\$$. This connects all constraint-gadgets and, since every variable occurs in a constraint, makes the whole graph connected. The variable-gadget vertices still have their unique degrees and constraint-anchors still have no value-vertex neighbors. Thus, the two repair claims and Lemma 3.3 continue to apply.

Remark 3.9. Our original construction overlaid a robustly self-ordered graph (see [11]) on all anchors to ensure that isomorphisms map each gadget to the gadget with the same index. As it turns out, this additional structure is unnecessary as the distance-preservation argument only requires variable-gadgets to be mapped set-wise to other variable-gadgets.

Acknowledgments

I am grateful to Oded Goldreich for his guidance throughout the development of this work and for his valuable feedback during the writing stages.

References

- [1] E. Ben-Sasson, P. Harsha, and S. Raskhodnikova. Some 3CNF properties are hard to test. *SIAM Journal on Computing*, 35(1):1–21, 2005.
- [2] B. Bollobás. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European Journal of Combinatorics*, 1(4):311–316, 1980.
- [3] J.-Y. Cai, M. Fürer, and N. Immerman. An optimal lower bound on the number of variables for graph identification. *Combinatorica*, 12(4):389–410, 1992.
- [4] E. Fischer and A. Matsliah. Testing graph isomorphism. *SIAM Journal on Computing*, 38(1):207–225, 2008.
- [5] O. Goldreich. *Introduction to property testing*. Cambridge University Press, 2017.
- [6] O. Goldreich. Open problems in property testing of graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 28(88), 2021.
- [7] O. Goldreich. Testing isomorphism in the bounded-degree graph model. In *Computational Complexity and Local Algorithms: On the Interplay Between Randomness and Computation*, pages 293–323. Springer, 2025.
- [8] O. Goldreich and D. Ron. Property testing in bounded degree graphs. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 406–415, 1997.
- [9] O. Goldreich and L. Tauber. On testing isomorphism to a fixed graph in the bounded-degree graph model. In *Computational Complexity and Local Algorithms: On the Interplay Between Randomness and Computation*, pages 324–337. Springer, 2025.
- [10] O. Goldreich and L. Tauber. Testing in the bounded-degree graph model with degree bound two. In *Computational Complexity and Local Algorithms: On the Interplay Between Randomness and Computation*, pages 373–382. Springer, 2025.
- [11] O. Goldreich and A. Wigderson. Robustly self-ordered graphs: Constructions and applications to property testing. *TheoretiCS*, 1, 2022.
- [12] B. D. McKay and X. Wang. Asymptotic enumeration of 0–1 matrices with equal row sums and equal column sums. *Linear Algebra and its Applications*, 373:273–287, 2003.