Quantum Property Testing in Sparse Directed Graphs

Simon Apers
Université Paris Cité, CNRS, IRIF, Paris, France

Frédéric Magniez
Université Paris Cité, CNRS, IRIF, Paris, France

Sayantan Sen
Centre for Quantum Technologies, National University of Singapore, Singapore

Dániel Szabó
Université Paris Cité, CNRS, IRIF, Paris, France

Abstract

We initiate the study of quantum property testing in sparse directed graphs, and more particularly in the unidirectional model, where the algorithm is allowed to query only the outgoing edges of a vertex. In the classical unidirectional model, the problem of testing k-star-freeness, and more generally k-source-subgraph-freeness, is almost maximally hard for large k. We prove that this problem has almost quadratic advantage in the quantum setting. Moreover, we show that this advantage is nearly tight, by showing a quantum lower bound using the method of dual polynomials on an intermediate problem for a new, property testing version of the k-collision problem that was not studied before.

To illustrate that not all problems in graph property testing admit such a quantum speedup, we consider the problem of 3-colorability in the related undirected bounded-degree model, when graphs are now undirected. This problem is maximally hard to test classically, and we show that also quantumly it requires a linear number of queries.

2012 ACM Subject Classification Theory of computation → Streaming, sublinear and near linear time algorithms; Theory of computation → Quantum query complexity

Keywords and phrases property testing, quantum computing, bounded-degree directed graphs, dual polynomial method, collision finding

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2025.32

Category RANDOM

Related Version *Full Version*: `https://arxiv.org/abs/2410.05001` [9]

Funding Research supported in part by the ERC Advanced Grant PARQ (grant agreement No 885394), the European QuantERA project QOPT (ERA-NET Cofund 2022-25), the French PEPR integrated projects EPiQ (ANR-22-PETQ-0007) and HQI (ANR-22-PNCQ-0002), the French ANR project QUOPS (ANR-22-CE47-0003-01), the National Research Foundation, Singapore through the National Quantum Office, hosted in A*STAR, under its Centre for Quantum Technologies Funding Initiative (S24Q2d0009), and the CQT Young Researcher Career Development Grant.

Acknowledgements The authors would like to thank the anonymous reviewers for their comments which improved the presentation of the paper.

1 Introduction

1.1 Context

In this modern big data era, the size of inputs has grown so much that even just reading the full input has become extremely expensive computationally. To tackle this challenge, the framework of *property testing* was initiated. It focuses on designing ultrafast algorithms (also

© Simon Apers, Frédéric Magniez, Sayantan Sen, and Dániel Szabó;
licensed under Creative Commons License CC-BY 4.0
Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2025).
Editors: Alina Ene and Eshan Chattopadhyay; Article No. 32; pp. 32:1–32:24
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

known as "testers") that read only a small part of the input, and distinguish inputs that satisfy some property from inputs that are "far" from satisfying it. As a possible use-case, when the exact computation is expensive, one can use property testing algorithms as a precursor to running the final algorithm. If the input does not pass the property testing test, we can safely reject it, without running the expensive final computation.

At the same time, the field of quantum computing has significantly influenced many computer science paradigms, including cryptography, algorithms, and large-scale data processing. This new perspective on computer science based on quantum physics has sparked many fresh research directions. This includes the topic of this work, which combines quantum computing and property testing. More specifically, we consider quantum algorithms for *graph* property testing.

Graphs are of paramount importance for instance when it comes to understanding large datasets, since they provide a natural way to represent and analyze complex relationships inside datasets. Goldreich, Goldwasser, and Ron [28] were the first to consider graphs in the context of property testing. Formally, given some form of query access to an unknown graph G on N vertices, and a property $\mathcal{P}$ of interest, the goal is to distinguish with high probability if G satisfies the property $\mathcal{P}$, or whether it is "*far*" from all graphs that satisfy $\mathcal{P}$, with a suitable notion of farness. In [28], the "dense" model was considered, where a graph is accessed through *adjacency queries*: for a pair of vertices (u, v), the query reveals whether (u, v) is an edge in the graph. In this model, a graph G is ε-far from satisfying $\mathcal{P}$ if one needs to add or remove at least εN^2 edges of G to obtain a graph that satisfies $\mathcal{P}$.

In a later work, Goldreich and Ron [29] introduced the "bounded-degree" model for testing sparse graphs, focusing on the properties of bipartiteness and expansion. In this model, a d-bounded degree graph G with N vertices is accessed by performing *neighbor queries*: for a vertex v and an integer $i \in [d]$, the query (v, i) returns either the i-th neighbor of v, or some special symbol if v has less than i neighbors. The graph G is said to be ε-far from some property $\mathcal{P}$, if one needs to add or delete at least εdN edges of G to obtain a graph that satisfies $\mathcal{P}$. Over the last two decades, there has been a significant number of works in this model, and we refer the interested reader to the books by Goldreich [26] and Bhattacharyya and Yoshida [14] and several surveys [24, 40, 23, 41].

Some researchers have considered efficient quantum algorithms for testing both classical and quantum objects, see for instance [18, 6, 31, 12, 10] and the survey [36]. Notably, the authors in [7] initiated the study of bounded degree graph property testing in the quantum model. One important result in this context is the result of [12], who proved that there can be exponential quantum advantage in the bounded degree graph model of property testing. However, as mentioned in their paper, the graph property admitting the exponential quantum advantage is not a natural one.

1.2 Property testing of directed bounded degree graphs

While all of the aforementioned works consider undirected graphs, many real-world instances (such as the world wide web) actually correspond to *directed* graphs. Consequently, Bender and Ron [13] introduced a model of property testing for directed graphs, focusing on the properties of acyclicity and connectivity. Following that work, we open a new research line by studying quantum algorithms for testing directed graphs. As we will see, by doing so, we address new fundamental questions in the field of quantum complexity. Answering them requires using recent techniques and partially answering some new or open questions.

As described in [13], for bounded-degree directed graphs there are two natural query models: (i) the *unidirectional* model, where the algorithm is allowed to query the outgoing edges of a vertex, but not the incoming edges, and (ii) the *bidirectional* model, where the

algorithm can query both the incoming and outgoing edges of a vertex. Interestingly, [13] showed that strong connectivity is testable in the bidirectional model (i.e., it can be tested with a number of queries that depends on ε but not on N), but it requires $\Omega(\sqrt{N})$ queries in the unidirectional model. Later, the testability of other graph properties like Eulerianity, vertex and edge connectivity [37, 46, 25, 21] was also shown in the bidirectional model. While there is a clear distinction between the two models, Czumaj, Peng and Sohler [22] showed that if a property is testable in the bidirectional model, then it has a tester with *sublinear* (i.e., $o(N)$) query complexity in the unidirectional model.

In this work, we consider a particularly important problem in the unidirectional model: the problem of testing *subgraph-freeness*. More precisely, we examine the problem of testing "*k-source-subgraph-freeness*", where the goal is to test H-freeness for some constant-sized subgraph H with k "source components", where a source component is a strongly connected subgraph that has no incoming edges. This problem was first studied by Hellweg and Sohler [32], and they provided a testing algorithm that performs $O(N^{1-1/k})$ queries. They also proved a tight lower bound of $\Omega(N^{2/3})$ for the $k = 3$ case (see [32, Theorem 1 and Theorem 3]). Very recently, Peng and Wang [38] proved a matching lower bound for any constant k. In particular, they showed that $\Omega(N^{1-\frac{1}{k}})$ queries are necessary for testing k-star-freeness (which is a special case of testing k-source-subgraph-freeness) in the unidirectional model, for arbitrary k (see [38, Theorem 1.2]). Notice that asymptotically the complexity of testing k-star-freeness becomes $\Omega(N)$. This also proves that the aforementioned reduction of [22] can not be made much stronger: for the property of k-star-freeness, the separation between the query complexities in the bi- and unidirectional models is maximal, as this property can be tested using constant many queries in the bidirectional model.

1.3 Related works on collision finding

A closely related problem to finding k-stars in graphs is finding k-collisions in integer sequences. The two mentioned classical papers on subgraph-freeness testing [32, 38] actually consider a collision-type intermediate problem for proving their lower bounds. As we are also going to do so, let us look at some related, known results.

The problem of *collision finding* is a ubiquitous problem in the field of algorithm theory with wide applications in cryptography. Here, given a sequence s of N integers, the goal is to find a duplicate in s. If one has the guarantee that $\Theta(N)$ elements of the sequence are duplicated, which is the case for instance when the sequence consists of uniformly random integers from $[N]$, it is well-known that classically $\Theta(\sqrt{N})$ queries are necessary and sufficient due to the birthday paradox. In the quantum model, this can be solved with query complexity $\Theta(N^{1/3})$ by the algorithm of Brassard, Høyer and Tapp [17]. The matching lower bound was first stated for a specific set of hard instances known as 2-to-1 (i.e., each integer appears exactly twice or not at all) by Aaronson and Shi [3]. For some constant integer $k \geq 3$, those results can be further extended to finding k-collisions in a random input with suitable alphabet size, so that it contains $\Theta(N)$ k-duplicates with high probability. The classical query complexity for this problem is $\Theta(N^{1-1/k})$ [32, 38], and quantumly it is $\Theta\left(N^{\frac{1}{2}\left(1-\frac{1}{2^k-1}\right)}\right)$ [34]. The situation is more complex for non-random inputs.

Remarkably, the complexity of *testing* k-collision-freeness (i.e., the absence of k-collisions) is harder to settle on the lower bound side than the *finding* version. In this work, we are going to focus on the hardness of distinguishing inputs that have linearly many collisions from those that do not have any. For $k = 2$, the two problems have the same complexity,

since intuitively the only way to distinguish is to find a collision. This can be formalized easily in the classical case. Quantumly, this is more challenging, but the lower bound in [3] proved the hardness of distinguishing between 2-to-1 instances and ones with no duplicate.

However, for larger k, distinguishing such inputs might be easier than finding a collision. The classical upper bound of $O(N^{1-1/k})$ queries is straightforward for the finding variant. In the lower bounds of [32, 38], the authors consider the distinguishing version, so classically the question is settled. But in the quantum setting, the upper and lower bounds of [34] are tight only for finding k-collisions in random inputs, and for the distinguishing variant, we are not currently aware of anything better than the $\Omega(N^{1/3})$ lower bound corresponding to the $k = 2$ case. To our knowledge, this problem has not yet been studied in the quantum setting.

1.4 Our results

In this work, we present two lines of results for quantum property testing of graph properties.

In the first line, we consider the problem of testing k-source-subgraph-freeness in the unidirectional model. This problem is almost maximally hard for large k in the classical regime, and we show that it admits an almost quadratic advantage in the quantum setting.

▶ **Theorem 1** (Restated in Theorem 13). *The quantum query complexity of testing k-source-subgraph-freeness in the unidirectional model is $O\left(N^{\frac{1}{2}\left(1-\frac{1}{2^k-1}\right)}\right)$.*

In order to prove the above result, we connect it to the problem of finding k-collisions. In [34], an algorithm is given for finding k-collisions in sequences of random integers. We generalize this to the context of graph property testing in two ways: finding a subgraph (instead of a collision); and considering graphs that are far from being H-free (instead of random).

Moreover, we prove that this quantum advantage is nearly tight, by showing a quantum lower bound using the method of dual polynomials.

▶ **Theorem 2** (Corollary of Theorem 3). *The quantum query complexity of testing k-source-subgraph-freeness in the unidirectional model is $\tilde{\Omega}\left(N^{\frac{1}{2}\left(1-\frac{1}{k}\right)}\right)$.*

For proving graph property testing lower bounds, both the classical works of [32] and [38] prove collision testing lower bounds using the proportional moments technique of [39]. At the heart of this technique is a construction of two positive integer random variables, X_1 and X_2, with different expectations but with the following conditions on the first $k-1$ moments: $\mathbb{E}[X_1]/\mathbb{E}[X_2] = \mathbb{E}[X_1^2]/\mathbb{E}[X_2^2] = \ldots = \mathbb{E}[X_1^{k-1}]/\mathbb{E}[X_2^{k-1}]$. Such a construction leads to a randomized query complexity lower bound of $\Omega(N^{1-\frac{1}{k}})$ for various property testing problems such as k-collision-freeness [38]. Having a quantum version of this technique has been identified as an important open problem [6], since this could be used to pave the way to stronger quantum lower bounds in related settings. We modestly made progress to this quest for the special case of testing k-collision-freeness.

In [34], in addition to the algorithm we mentioned, they also prove a matching lower bound showing that their algorithm for *finding* k-collisions in random inputs is optimal. However, this time we cannot reuse those techniques for our purpose for two main reasons. First, the property testing variant of this problem could be easier. Moreover, their lower bound technique requires random inputs and hence it does not apply to our case. This is why we use yet another method, that of dual polynomials, to prove our lower bound.

▶ **Theorem 3** (Restated in Theorem 14). *The quantum query complexity of testing k-collision-freeness is $\tilde{\Omega}\left(N^{\frac{1}{2}\left(1-\frac{1}{k}\right)}\right)$.*

In the second line of results, we show that not all problems in graph property testing admit such a quantum speedup. This fact even remains valid for the case of undirected graphs. For this, we consider the property testing variant of the famous problem of 3-colorability: namely, distinguishing whether an unknown undirected graph G can be properly colored with 3 colors, or one needs to modify a large fraction of its edges to make it 3-colorable. In the classical bounded degree setting, this problem has been studied by [15], who proved a lower bound of $\Omega(N)$ queries. In this work, we present a simple argument that proves that there exists no sublinear quantum tester either for this problem. Our result is stated as follows:

▶ **Theorem 4** (Restated in Theorem 31)**.** *The quantum query complexity of testing of 3-colorability of undirected bounded-degree graphs is* $\Omega(N)$*.*

1.5 Technical overview

1.5.1 Subgraph-finding algorithm

We start by describing how to prove the upper bound result of Theorem 1 for testing k-source-subgraph-freeness. We view the problem as a generalization of the problem of finding k-collisions and adapt an existing quantum algorithm for the latter problem. In [34], an algorithm is given for finding k-collisions in length-N sequences of integers that contain $\Omega(N)$ k-collisions (e.g. k-to-1, or random sequence with appropriate parameters). Their algorithm generalizes the well-known collision finding algorithm of [17]. On a high level, the [34] algorithm first finds several 2-collisions using Grover search like in [17], extends some of them to 3-collisions in a similar way, and so on until a k-collision is found.

On the one hand, instead of random inputs, we consider the problem in the property testing context; and on the other hand, we generalize collision-finding to subgraph-finding. As a first step let us look at what happens when we consider the property testing version of the k-collision problem. In order to be able to use the algorithm of [34], we have to prove that if a length-N sequence is far from k-collision-freeness then it contains many k-collisions. Notice that the collisions are not necessarily distinct: if the input only contains the same integer N times, it only contains one huge collision, but it is still ε-far from k-collision-freeness for any $\varepsilon < 1 - k/N$. Thus, what we need to show is that there are $\Omega(N)$ many disjoint size-k sets of indices such that for each set, the sequence contains the same character at the positions of the set. This statement is true because otherwise, by modifying all the characters that are in positions contained in a set ($o(N)$ characters in total), we could get a k-collision-free sequence which contradicts being far from k-collision-freeness.

When we make the second step of turning to testing of subgraph-freeness, we need to prove a variant of this statement: if an N-vertex graph G is far from H-freeness (for some constant-sized subgraph H) then it contains $\Omega(N)$ many "source-disjoint" H-subgraphs. This means that there are $\Omega(N)$ many such H-subgraphs in G that the set of vertices in the source components of each H-subgraph are disjoint. We prove this fact in Proposition 12 and this allows us to further generalize the approach of [34]: first find several partial solutions where only a few source components of an H-subgraph are explored, and gradually extend these (using Grover search and constant-depth BFS) until a complete H-subgraph is found.

Notice that this way our algorithm *finds* an H-subgraph in G promised that G is far from H-freeness. This task is at least as difficult as property testing, where the algorithm only has to *distinguish* whether G is H-free or far from any H-free graph. So our algorithm provides an upper bound on the property testing variant of H-freeness.

1.5.2 Collision-freeness lower bound

Now we will discuss our approach to proving the lower bounds of collision-freeness (Theorem 2) and k-source-subgraph-freeness (Theorem 3). We first give a simple reduction from k-collision-freeness to k-star-freeness, which is a special case of k-source-subgraph-freeness. This way, it is enough to prove a lower bound on testing k-collision-freeness, and it implies the same result on testing k-source-subgraph-freeness. Since our lower bound approach crucially depends on the (dual) polynomial method, let us start by briefly discussing it.

1.5.2.1 The (dual) polynomial method

The polynomial method is a common way to prove quantum query complexity lower bounds. It relies on the fact that the acceptance probability of a T-query bounded-error quantum algorithm is a polynomial of degree at most $2T$ [11]. This way, for proving a quantum query complexity lower bound on calculating a function f, it suffices to argue that any approximating polynomial of f has large degree. One of the key properties that such lower bounds exploit is the symmetry that the function f may exhibit, such as invariance under some permutation of the input. For example, the first tight lower bound of $\Omega(n^{1/3})$ for the collision problem was proved in this way [3].

The polynomial method can be written in the form of a linear program, of which one can take the dual. By weak LP-duality, when using this dual characterization for proving a lower bound on function f, one needs to provide a "witness" of the approximating polynomial's high degree, say Δ. This witness is called the *dual polynomial* ψ and, in the easiest case of total Boolean[1] functions $f : \{-1, 1\}^n \to \{-1, 1\}$, it needs to have three properties:

(i) High correlation with f: $\sum_x f(x)\psi(x) > \delta$;
(ii) Normalization: $\sum_x |\psi(x)| = 1$;
(iii) Pure high degree Δ: $\sum_x p(x)\psi(x) = 0$, for every polynomial p with degree $< \Delta$,

where the summations are all over $x \in \{-1, 1\}^n$.

When the function f is partial, i.e., only defined on a subset $D \subset \{-1, 1\}^n$, there is some subtlety that could be handled in two ways (or even in a mixture or both): zero-out the dual polynomial outside D (corresponding to "unbounded degree"); or rewrite condition (i) accordingly (corresponding to "bounded degree"):

(i') High correlation with f: $\sum_{x\in D} f(x)\psi(x) - \sum_{x\notin D} |\psi(x)| > \delta$.

1.5.2.2 Collision function

The paper of [19] also used the dual polynomial method for proving quantum lower bounds for many problems, most of them being open before that work. Similarly to that paper, we need to take several steps to be able to use the dual polynomial method for the problem of property testing k-collision-freeness. This problem was not addressed in [19].

One of the main conceptual ideas in [19] is to re-formulate the problem we study as a composition of two simple Boolean functions. In that paper, powerful techniques are also developed in order to design dual polynomials for simple functions that can be composed. A common way of composing dual polynomials (called dual block composition) dates back to [44, 33, 43], but [19] provides new tools for handling it efficiently. We are going to reuse some of them, and also extend one in a way.

[1] We use $\{-1, 1\}$ where -1 corresponds to the "true value".

The first step is to find the right problem that can fit in the framework. We introduce a partial symmetric function F defined on input strings $s = (s_1, \ldots, s_N) \in [R]^N$. The domain of F corresponds to the following promise: either F has no k-collision, or it has many k-collisions occurring for distinct values. More formally,

$$F(s) = \begin{cases} -1 & \text{if no integer occurs at least } k \text{ times in } s, \\ 1 & \text{if more than } \gamma R \text{ distinct integers occur at least } k \text{ times in } s, \\ \text{undefined} & \text{otherwise.} \end{cases}$$

This partial function is not a property testing problem, however it corresponds to a special case of testing k-collision-freeness, which is therefore enough to prove lower bounds.

1.5.2.3 Binary encoding

Now we encode the input string $s = (s_1, \ldots, s_N) \in [R]^N$ into binary variables $x_{i,j}$ storing whether $s_i = j$, as in [1]. Doing so, starting from the function F above, we end up with a function f defined over binary variables satisfying several symmetries, under the permutation of either i or j in $x_{i,j}$.

Moreover, the symmetries of f allow the extension of the initial function f from the very restricted set of binary inputs correspond to valid strings, to the more general set of binary inputs of Hamming weight N [6]. With further technicalities one can also extend f to all binary inputs of Hamming weight *at most* N [20]. This is fundamental because instead of being forced to zero out the dual polynomial outside the domain of f, we only need to do so on inputs of Hamming weight higher than N. Using the symmetry of f, it can be shown that a lower bound on this modified, Boolean version implies a lower bound on the original k-collision problem.

This way we end with two promises on the binary encoding of the input. The first one comes from the function F itself: we have the promise that the input contains either no k-collision or it has many ones with different values. The second promise is the consequence of the encoding: we want the binary encoding to have Hamming weight at most N. Let D denote the set of binary strings satisfying both promises, and let $H_{\leq N}$ denote the set of binary strings with Hamming weight at most N. For this case we use the "double-promise" version of the dual polynomial method, where, in order to prove that every δ-approximating polynomial of f has degree at least Δ, the dual polynomial has to satisfy four conditions, where the fourth one corresponds to zeroing out ψ on large Hamming weight inputs [19]:

(i') High correlation with f: $\sum_{x \in D} f(x)\psi(x) - \sum_{x \in H_{\leq N} \setminus D} |\psi(x)| > \delta$;
(ii) Normalization: $\sum_x |\psi(x)| = 1$;
(iii) Pure high degree Δ: $\sum_x p(x)\psi(x) = 0$, for every polynomial p with degree $< \Delta$;
(iv) No support on inputs with large Hamming weight: $\psi(x) = 0$, for every $x \notin H_{\leq N}$.

1.5.2.4 Composition

Coming back now to the definition of our Boolean function f, one can rewrite it as a composition of simpler functions: $\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k$, where by composition we mean $(g \circ h)(x) = g(h(x_1), \ldots, h(x_n))$ (where $x = (x_1, \ldots, x_n)$ and each x_i is a binary vector of appropriate dimension) and the domain is restricted to bit strings of Hamming weight at most N. Note that here, $x_j = (x_{1,j}, x_{2,j}, \ldots, x_{N,j})$. Above, THR_N^k is the threshold function: it is -1 if the input bitstring contains at least k many -1 (true) values, and is 1 otherwise; and GapOR_R^γ is the gap version of OR, which is 1 if the input is all-1, -1 if the input contains at least γR many -1 values, and is undefined otherwise.

In order to give a dual polynomial for this composed function, we start from a dual polynomial for each part of the composition (ϕ and ψ), which were already given in [19] (in different contexts). Then we use a known way [44, 33, 43] of composing dual polynomials called the *dual block composition*, which provides a nearly good dual polynomial $\phi \star \psi$ for $\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k$. Indeed, by construction, the normalization (ii) and the pure high degree (iii) are guaranteed. However, the issue is that it is not 0 on bitstrings of large Hamming weight thus (iv) is not satisfied, and and the high correlation (i') still has to be proved.

To fix (iv), we use another result of [19] which provides another dual polynomial ζ, that is close to $\phi \star \psi$ and that is 0 on inputs having Hamming weight larger than N. Also, it only changes the pure high degree by a polylogarithmic factor. Now the only remaining task is to prove a large enough correlation (i') of $\phi \star \psi$ and $\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k$ so that ζ still has high enough correlation. This high correlation proof (Lemma 30) is the most technical part of our paper.

1.5.2.5 High correlation: proof of Lemma 30

The statement we prove is the following high correlation bound:

$$\sum_{x \in D} (\phi \star \psi)(x) \cdot (\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k)(x) - \sum_{x \notin D} |(\phi \star \psi)(x)| \geq 9/10.$$

In the proof of this lemma, that is deferred to the full version of the paper [9], we use a proposition that is a more general statement of some techniques used in several proofs of [19]. But then we need to diverge from their proof because it crucially relies on a certain one-sided error property (in the sense of [19, Lemma 6.11]) of the inner function of the composition which is OR function in their case. Our inner function is the threshold function which does not satisfy this property, so we have to use some other properties of the dual polynomials in our proof. This different proof technique could be a step towards obtaining a more general lower bound technique.

1.5.3 3-colorability lower bound

Let us now discuss our approach to proving the linear lower bound on quantum query complexity of testing 3-colorability (Theorem 4). Before proceeding to present our approach, let us briefly discuss the classical lower bound of testing 3-colorability. To prove the classical lower bound of 3-colorability, the authors in [15] first studied another problem called $\mathrm{E}(3, c)$LIN-2, a problem related to deciding the satisfiability of a system of linear equations. More formally, $\mathrm{E}(3, c)$LIN-2 considers a system of linear equations modulo 2, where each equation has 3 variables and every variable appears in at most c equations. Given such a system of linear equations, the goal is to distinguish if it is satisfiable, or at least some suitable fraction of the equations need to be modified to satisfy it. [15] proved that $\Omega(N)$ classical queries to the system of linear equations are necessary for testing $\mathrm{E}(3, c)$LIN-2.

After this, they designed a reduction from $\mathrm{E}(3, c)$LIN-2 to 3-colorability such that satisfying instances of $\mathrm{E}(3, c)$LIN-2 are reduced to 3-colorable graphs, and far from satisfiable instances of $\mathrm{E}(3, c)$LIN-2 are mapped to far from 3-colorable graphs. Combining these two arguments, the authors in [15] proved that $\Omega(N)$ classical queries are necessary for testing 3-colorability for bounded degree graphs.

The authors in [15] used Yao's minimax method to prove the linear lower bound in testing $\mathrm{E}(3, c)$LIN-2. In particular, they designed two distributions D_{yes} and D_{no} such that the systems of linear equations in D_{yes} are satisfiable, whereas the systems of linear equations

in D_{no} are far from being satisfiable. A crucial ingredient of their lower bound proof is a construction of a system of linear equations (represented as a matrix) that are far from being satisfiable, but any δN rows of the matrix are linearly independent. Hence, any subset of δN entries of the matrix will look uniformly random, and therefore hard to distinguish from a satisfiable instance.

It is a known fact that distinguishing between a uniformly random string and a ℓ-wise independent string, for an appropriate integer ℓ, is hard for quantum algorithms (see e.g. [8]). Using this result, we can construct suitable hard instances for $\mathrm{E}(3,c)$LIN-2 such that testing $\mathrm{E}(3,c)$LIN-2 remains maximally hard (requires $\Omega(N)$ queries) for any quantum algorithm. Combining this hardness result with the reduction from $\mathrm{E}(3,c)$LIN-2 to 3-colorability, we finally prove that $\Omega(N)$ quantum queries are necessary for testing 3-colorability.

In [45], the authors used various reductions from 3-colorability to argue that some other important problems including testing Hamiltonian Path/Cycle, approximating Independent Set/Vertex Cover size etc, are maximally hard to test in the classical model. Our quantum lower bound implies that these problems have maximal quantum query complexity too.

1.6 Open problems

Our work raises several important open questions. First, there is still a gap between our lower and upper bounds on the quantum query complexity of testing k-collision-freeness. In [35], the authors keep using the dual polynomial method to improve the lower bound of [19] for the k-distinctness problem. They achieve this by using a slightly different dual polynomial for THR_N^k, where they allow more weight on the false positive inputs. This makes it impossible to prove the high correlation of the dual and the primal functions, so they use a modified block composition. Our technique might be combined with this other approach to improve our lower bound to $\tilde{\Omega}(N^{1/2-1/(4k)})$.

The authors in [6] stated it as an open question if one could use a variant of the proportional moments result of [39] to prove lower bounds on quantum algorithms. We leave this question open, and conjecture that a similar result holds in the quantum setting with a lower bound of $\Omega\left(N^{\frac{1}{2}\left(1-\frac{1}{k}\right)}\right)$. This work may be considered as a proof of this conjecture for the special case of k-collision-freeness, and we hope that it will serve as a step towards proving it in general.

In [22], it was proved that if a graph property can be tested with $O(1)$ queries in the bidirectional model, then it can be tested using $O(N^{1-\Omega(1)})$ queries in the unidirectional model. It would be very interesting to investigate if it also implies a quantum tester with query complexity, say $O(N^{1/2-\Omega(1)})$.

2 Preliminaries

2.1 Notations and basic definitions

Let us denote $[n] = \{1,\dots,n\}$ and $[n]_0 = \{0,\dots,n\}$. When dealing with Boolean variables, we will usually use $b \in \{-1,1\}$ instead of $b \in \{0,1\}$. We can get to one from the other easily with the mapping $b \mapsto 1-2b$, or its inverse, which means that -1 is going to be treated as the "true" or "accepting" value. The reason for using $\{-1,1\}$ is that when dealing with dual polynomials, it is easier to use this notation.

We denote by 1^n the length-n binary vector made only of 1s, and respectively -1^n. The Hamming weight $|x|$ of $x \in \{-1,1\}^n$ is then defined as the number of -1s in x, that is $|x| = \#\{i \in [n] : x_i = -1\}$. Let $H^n_{\leq w} = \{x \in \{-1,1\}^n : |x| \leq w\}$ denote the set of length-n binary vectors with Hamming weight at most w. For any $x \in \mathbb{R}$, $\mathrm{sgn}(x) = 1$ when $x \geq 0$, and -1 otherwise.

For a polynomial p, let $\deg(p)$ denote its degree. The composition $f \circ g : \{-1,1\}^{nm} \to \{-1,1\}$ of two Boolean functions $f : \{-1,1\}^n \to \{-1,1\}$ and $g : \{-1,1\}^m \to \{-1,1\}$ is defined as $(f \circ g)(x) = f(g(x_1), \ldots, g(x_n))$ where $x = (x_1, \ldots, x_n)$ with each $x_i \in \{-1,1\}^m$.

A directed graph or digraph $G = (V, E)$ is a pair of a vertex set V and an edge set E. The latter consists of directed edges that are ordered pairs of vertices: we say that $(u, v) \in E$ is directed from u to v where $u, v \in V$. We say that there is a directed path from $s = v_0$ to $t = v_{l+1}$ (with $s, t \in V$) if there exists an integer ℓ and vertices $v_1, \ldots, v_\ell \in V$ such that $\forall i \in [\ell]_0 : \ (v_i, v_{i+1}) \in E$. A digraph $G = (V, E)$ is called *strongly connected* if for every $u \in V$ and $v \in V \setminus \{u\}$, there exists a directed path from u to v. A *subgraph* of a graph $G = (V, E)$ is any graph $G' = (V', E')$ satisfying $V' \subseteq V$, $E' \subseteq E$ and $E' \subseteq V' \times V'$.

Finally, throughout this work, notations $O(\cdot)$ and $\Omega(\cdot)$ will be hiding the dependencies on parameters ε, k and d that we consider to be constants. Additionally, we will use $\widetilde{O}(\cdot)$ and $\widetilde{\Omega}(\cdot)$, where we hide poly-logarithmic dependencies on the parameters.

2.2 Query complexity

In query complexity, we consider inputs $x \in \Sigma^I$ over a finite alphabet Σ and indexed by a set I. They are not given explicitly to the algorithm. Instead, the algorithm has query access to an input oracle $\mathcal{O}_x : I \to \Sigma$ encoding x by $\mathcal{O}_x(i) = x_i$. Quantumly, the query access is described by the unitary operator $O_x |i\rangle |z\rangle = |i\rangle |z \oplus x_i\rangle$, for $z \in \Sigma$ and $i \in I$, where $\oplus$ is usually the bit-wise exclusive-OR operation up to some binary encoding of the elements of Σ. But our lower bound technique applies to any reversible operation $\oplus$.

Query complexity measures the minimum number of queries that an algorithm has to make in order to decide whether a property $\mathcal{P} : D_{\mathcal{P}} \subseteq \Sigma^I \to \{-1,1\}$ is satisfied, for an arbitrary input $x \in D_{\mathcal{P}}$. Since the work of [11], it has been known that in the Boolean case (i.e., when $\Sigma = \{-1,1\}$), the acceptance probability of a T-query bounded-error quantum algorithm is a multivariate polynomial (in all the x_i's) of degree at most $2T$.

In this work, two kinds of inputs are going to play a crucial role. When the input is a sequence $s = (s_1, \ldots, s_N)$ of positive integers $\leq R$, then $I = [N]$ and $\Sigma = [R]$.

In the undirected bounded-degree graph model, we have query access to the adjacency list of an undirected graph $G = (V, E)$ with maximum degree d, represented as an oracle $\mathcal{O}_G : V \times [d] \to V \cup \{\perp\}$. Then we can set $I = V \times [d]$ and $\Sigma = V \cup \{\perp\}$, such that for any $v \in V$ and $i \in [d]$, we have the following:

$$\mathcal{O}_G(v, i) = \begin{cases} w, & \text{if } w \in V \text{ is the } i\text{-th neighbor of } v; \\ \perp, & \text{if } \deg(v) < i. \end{cases}$$

For bounded-degree directed graphs, there exist two query models. In the *bidirectional model*, we have access to both the outgoing and incoming edges of each vertex. Correspondingly, it is imposed that both the in- and out-degrees of a vertex are bounded by d. In the *unidirectional model*, we can only make queries to the adjacency list of the outgoing edges, and we impose only that the out-degrees of a vertex are bounded by d. Since in this work the primary focus will be on the latter model, let us formally define it below.

In the unidirectional bounded-degree graph model, we have query access to the adjacency list of a digraph $G = (V, E)$ where the out-degree of every vertex is at most d_{out}: for all $v \in V$: $\deg_{\text{out}}(v) \leq d_{\text{out}}$. This access is represented as an oracle $\mathcal{O}_G^{\text{out}} : V \times [d_{\text{out}}] \to V \cup \{\perp\}$. Then we can set $I = V \times [d_{\text{out}}]$ and $\Sigma = V \cup \{\perp\}$, such that for any $v \in V$ and $i \in [d_{\text{out}}]$, we have the following:

$$\mathcal{O}_G^{\text{out}}(v,i) = \begin{cases} w, & \text{if } w \in V \text{ is the } i\text{-th out-neighbor of } v; \\ \perp, & \deg_{\text{out}}(v) < i. \end{cases}$$

For completeness, we note that in some of the previous works on the unidirectional model, the authors do impose the degree bound on both the out- and in-degree (see, e.g., [22]). This is mostly because this makes for an easier comparison between the uni- and bidirectional models, as this way they allow the same set of graphs. In this work, we assume that only the out-degrees of the vertices are bounded by d.

2.3 Property testing

In total decision problems, the algorithm has to decide if the input satisfies a property or not. In the case of property testing, the question is relaxed: the algorithm has to distinguish inputs that satisfy the property from those that are "far" (according to some distance measure) from any input that satisfies it.

The choice of distance measure usually depends on the query model considered. As discussed before, the general query access can be viewed as black box access to the input $x \in \Sigma^I$ where querying an index $i \in I$ reveals $x_i \in \Sigma$. This way, the distance of two objects is described as the proportion of positions where they differ:

$$x \text{ is } \varepsilon\text{-far from } y \iff |\{i \in I : x_i \neq y_i\}| \geq \varepsilon |I|.$$

Applying this to the case of bounded-degree graphs with degree bound d and query access to the adjacency list, the distance of two graphs is the number of edges where they differ divided by $|V|d$. The distance of an object x from a property $\mathcal{P}$ is the minimum distance between x and any object that satisfies $\mathcal{P}$.

▶ **Definition 5** (Property Testing). *Let $0 < \varepsilon < 1$ be a constant. An algorithm $\mathcal{A}$ is an* ε-tester *for the property $\mathcal{P}$ if*

1. *For all $x \in \mathcal{P}$:* $\Pr[\mathcal{A}(x) = \text{accept}] \geq 2/3$*;*
2. *For all x that are ε-far from $\mathcal{P}$:* $\Pr[\mathcal{A}(x) = \text{accept}] \leq 1/3$.

Notice that no restriction is given on the acceptance probability of the algorithm for inputs that do not satisfy $\mathcal{P}$ but are ε-close to it.

2.4 Grover search

We are not going into details about quantum computing, because for this paper, it suffices to state very few results in the field and use them in a black box way. One of the most important results in quantum computing is Grover's algorithm for unordered search [30]: finding a marked element in an unordered database of size N takes $\Theta(N)$ queries classically, but the quantum query complexity of the same task is $\Theta(\sqrt{N})$. We are going to use a particular variant of this result that has been used many times in the literature (see e.g., [4, Item 3 in Section 2.2], which was implicitly proved in [16]).

▶ **Theorem 6.** *Let $1 \leq t_0 \leq N$. There exists a quantum algorithm that, given t_0 and query access to any function $f : S \to \{0,1\}$, makes $O(\sqrt{N/t_0})$ queries to f and outputs either "not found" or an element uniformly at random in $f^{-1}(1)$. Moreover, when $|f^{-1}(1)| \geq t_0$ the later occurs with high constant probability.*

▶ Remark 7. In practice, we will use this theorem when querying $f(z)$, for $z \in S$, requires making c queries to an input graph G with vertex set S. In that case the total query complexity to G is $O(c\sqrt{N/t_0})$.

2.5 Problem definitions

We now formally define the problems we study and argue about certain relations between them. While the problems are phrased as decision problems, ultimately we will care about the quantum query complexity for testing the corresponding properties. The complexity is going to be parameterized by a parameter k. Moreover, the parameters k, ε and the degree bound d are all considered to be constants throughout this paper.

Let us start with some definitions that will be useful to define our problems precisely.

▶ **Definition 8** (Source component). *Let $H = (V, E)$ be a digraph. A set $S \subseteq V$ is called a* source component *if it induces a strongly connected subgraph in H, and there is no edge from $V \setminus S$ to S in H.*

▶ **Definition 9** (k-star). *A k-star is a digraph on $k+1$ vertices and k edges with one center vertex, and k source vertices connected to the center vertex.*

Notice that a k-star has k source components each consisting of a single vertex.

We will now state the decision variants of several problems. The "property" corresponding to a decision problem is the set of inputs that should be accepted in the decision problem.

k-Source-Subgraph-Freeness

Parameter: Graph H of constant size with at most k source components
Query access: d-bounded out-degree directed graph G on N vertices (unidirectional model)
Task: Accept iff G is H-free, that is, no subgraph of G is isomorphic to H

In [32, 38], the authors examine the classical query complexity of testing k-source-subgraph-freeness. They consider the bounded-degree unidirectional model, albeit with a bound on both the in- and out-degrees.

For proving a lower bound, we will look at a special case of the main problem: k-star-freeness. Notice that since a k-star has k source components, a lower bound for this problem implies the same lower bound for the more general k-source-subgraph-freeness problem.

k-Star-Freeness

Parameter: Integer $k \geq 2$
Query access: d-bounded out-degree directed graph G on N vertices (unidirectional model)
Task: Accept iff G is k-star-free, that is, no subgraph of G is isomorphic to the k-star

For the lower bound on k-star-freeness testing, we are going to use as a "helper problem" the testing variant of the k-collision problem.

k-Collision-Freeness

Parameter: Integer $k \geq 2$
Query access: Sequence of integers $s = (s_1, \ldots, s_N) \in [R]^N$
Task: Accept iff s is k-collision-free, i.e., there is no $i_1, \ldots, i_k \in [N]$ with $s_{i_1} = \cdots = s_{i_k}$

As discussed in the introduction (Section 1.3), very little was known about the property testing version of this problem prior to this work. We only know the that the complexity is $\Theta(N^{1/3})$ when $k = 2$, and it is between $\Omega(N^{1/3})$ and $O\left(N^{\frac{1}{2}\left(1-\frac{1}{2^k-1}\right)}\right)$ for larger k.

Reduction from k-collision-freeness to k-star-freeness

Now we are going to prove that testing k-collision-freeness can be reduced to testing k-star-freeness (or more generally to testing k-source-subgraph-freeness). Thus, a lower bound on testing k-collision-freeness yields a lower bound on testing k-source-subgraph-freeness. Also, an algorithm for testing k-source-subgraph-freeness yields an upper bound on testing k-collision-freeness.

While the proof goes similarly to [32, Theorem 3], our reduction is not identical because we have a slightly different "helper problem". Since they consider that the in-degree of vertices to be bounded as well, for the collision problem, they assume that the sequence does not contain any collision of size larger than k (defined as *k-occurrence-freeness*).

▶ **Proposition 10.** *The problem of ε-testing k-collision-freeness of a sequence from $[R]^N$ can be reduced to $\frac{\varepsilon N}{d(N+R)}$-testing k-star-freeness of an $(N+R)$-vertex sparse directed graph with out-degree bound $d \geq 1$.*

The proof is rather standard and it can be found in the full version of the paper [9].

3 Quantum algorithm for testing subgraph-freeness

In this section, we prove that there is a quantum speedup for testing H-freeness in directed graphs with d-bounded out-degree, for any graph H that has k source components. For large but constant k, the speedup is near-quadratic. This problem was studied in [29] in the classical setting. Our algorithm can be seen as a generalization of the one in [34] to graphs. Let us start with the definition of source-disjointness which will be used in the analysis of our algorithm.

▶ **Definition 11** (Source-disjointness)**.** *Let G be a directed graph such that it contains two subgraphs H_1 and H_2. We say that H_1 and H_2 are* source-disjoint *if the union of the source components of H_1 is disjoint from the union of the source components of H_2.*

Moreover, we need the following simple proposition, that is proved in the full version of the paper [9]. It shows that if G is far from being H-free, then it contains many source-disjoint copies of H, that is, copies of H that are source-disjoint subgraphs of G.

▶ **Proposition 12.** *Let H be an h-vertex graph with k source components. Assume that a d-bounded out-degree directed graph G on N vertices is ε-far from H-free. Then G contains at least $\varepsilon N/h = \Omega(N)$ source-disjoint copies of H.*

We will use Breadth-first search (BFS) in order to explore G layer by layer: starting from a given vertex, first it explores its direct (out-)neighbors, then their unexplored (out-)neighbors etc. We will run BFS up to some limited depth ℓ, so that the depth-ℓ BFS algorithm has query complexity at most d^ℓ where d is the maximum (out-)degree of G. In our application, d and ℓ are constants, and thus the query complexity is $d^\ell = O(1)$.

3.1 The algorithm for $k = 2$

To illustrate our algorithm, we first consider the $k = 2$ case to build some intuition. Here, our algorithm generalizes the BHT algorithm for collision finding [17] in the context of graphs. The high level idea is that if we manage to sample a vertex from each of the two source components of an H-subgraph (a collision), then by querying their "surroundings" we will discover the H-instance. In the following, we set $h = |V(H)|$, the number of vertices in H.

1. Sample a uniformly random vertex subset $\mathcal{S}$ of size $t = \Theta(N^{1/3})$ in G. Perform a depth-h BFS from every vertex in $\mathcal{S}$.
2. Perform Grover search over the remaining vertices $V \backslash \mathcal{S}$ in the following way. A vertex v is marked if there exists another vertex $u \in \mathcal{S}$ such that u and v are from the 2 different source components of an H subgraph of G.
3. If any occurrence of H in G is found, output **Reject**. Otherwise, output **Accept**.

Note that if G is H-free, then the above algorithm will always accept. Now we need to argue that if G is ε-far from being H-free, then with constant probability, the above algorithm will find a copy of H and thus will output reject.

By Proposition 12, with high probability, a constant fraction of the t vertices in $\mathcal{S}$ are part of a source component in source-disjoint H-subgraphs of G. For such vertices, the BFS in step 1 will discover the entire source component, as well as all other vertices reachable from that source component in H. Then, in step 2 we search for a vertex that is in the remaining source component of such an instance of H that we already partly discovered. This can be verified by doing a depth-h BFS from it and checking if this completes an H-instance with one of the previously sampled vertices' neighborhoods. As we mentioned, by Proposition 12 with high probability there are $\Omega(t)$ many marked vertices. This proves the correctness of the algorithm.

Finally, we bound the algorithm's query complexity. Step 1 makes $O(t) = O(N^{1/3})$ many (classical) queries. In step 2 we use Theorem 6 and Remark 7: checking whether a vertex is marked requires running a depth-h BFS from it, which costs $c = O(1)$ queries. We argued that there are $\Omega(t)$ many marked vertices, so Grover search makes $O(\sqrt{N/t}) = O(N^{1/3})$ quantum queries.

3.2 The algorithm for general k

We are now ready to state our general upper bound result.

▶ **Theorem 13** (Restatement of Theorem 1). *Let H be a digraph of constant size with k source components. The quantum query complexity of testing H-freeness of an N-vertex graph with bounded out-degree in the unidirectional model is $O\left(N^{\frac{1}{2}\left(1-\frac{1}{2^k-1}\right)}\right)$.*

The algorithm and proof follow the same lines as the $k = 2$ case. In order to extend the $k = 2$ case described above to larger k, we first try to find many partial H-instances with $k - 1$ source components found, and then extend one of them to a complete H-instance. We present a brief description of our algorithm, where h is the number of vertices of H:

1. Sample a uniformly random vertex subset $\mathcal{S}_1$ in G of size t_1. Perform a depth-h BFS from every vertex in $\mathcal{S}_1$. Let $\mathcal{S}'_1 = \mathcal{S}_1$.
2. For iterations $i = 2$ to $k - 1$, do the following:
   a. Perform a Grover search t_i times on the vertices $V \setminus \mathcal{S}'_{i-1}$ in the following way. A vertex v is marked if there exist $i - 1$ other vertices $u_j \in \mathcal{S}_j$ for each $j \in [i - 1]$ such that $u_1, \ldots, u_{i-1}$ and v are from i different source components of an H subgraph of G. If we do not find t_i vertices like this, output **Reject**, otherwise let $\mathcal{S}_i$ denote the set of the vertices v that we found.
   b. Set $\mathcal{S}'_i = \mathcal{S}'_{i-1} \cup \mathcal{S}_i$.

3. Perform Grover search on $V \setminus \mathcal{S}'_{k-1}$ to find a complete H-instance. I.e., a vertex v is marked if there exist $k-1$ other vertices $u_j \in \mathcal{S}_j$ for each $j \in [k-1]$ such that $u_1, \ldots, u_{k-1}$ and v are from the k different source components of an H subgraph of G.
4. If any occurrence of H in G is found, output **Reject** and terminate the algorithm. Otherwise, output **Accept**.

The correctness proof can be found in the full version of the paper [9].

4 Collision-freeness lower bound

As discussed in Section 2, we are going to prove a lower bound on the problem of testing k-collision-freeness.

▶ **Theorem 14** (Restatement of Theorem 3). *Let $k \geq 3$ and $0 < \varepsilon < 1/(4^{k-1}\lceil 20(2k)^{k/2}\rceil)$ be constants and N be a large enough positive integer. Then the quantum query complexity of the ε-testing of k-collision-freeness of a sequence of integers $S = (s_1, \ldots, s_N) \in [N]^N$ with parameter ε is $\Omega(N^{1/2-1/(2k)}/\ln^2 N)$.*

The proof of the theorem is at the end of Section 4.3. Observe that Theorem 2 is implied by Theorem 14 and the reduction in Proposition 10. Our proof mostly follows the structure of [19, Section 6.1], and in particular it uses the notion of dual polynomial for non-Boolean partial symmetric functions. Our main technical contribution in this section is the proof of Lemma 30, because the corresponding proof in [19] crucially relies on a fact that does not hold for our problem. We will discuss it in detail below.

In the following, we first state some general results related to the polynomial method for non-Boolean functions, then we use these results for our problem to state the exact statement that we prove in the technical part.

4.1 The (dual) polynomial method

For Boolean functions

We consider a property on Boolean vectors as a function $f : D \subseteq \{-1,1\}^n \to \{-1,1\}$. Since the work of [11], it has been known that the acceptance probability $\mathsf{acc}(x)$ of a T-query bounded-error quantum algorithm on input $x \in D$ is a polynomial of degree at most $2T$. Thus, the polynomial $p(x) = 1 - 2 \cdot \mathsf{acc}(x)$ must be a good approximation of f. Observe that $p(x)$ remains bounded outside D since $\mathsf{acc}(x)$ remains a probability defined by the algorithm, with no constraint.

In order to formalize this, we first define the notion of approximate degree of a Boolean function, and then relate it to its query complexity.

▶ **Definition 15** (Approximate bounded degree). *Let $f : D \subseteq \{-1,1\}^n \to \{-1,1\}$ and $\delta > 0$. A polynomial $p : \{-1,1\}^n \to \mathbb{R}$ δ-approximates f on D if*

$$\forall x \in D : \ |f(x) - p(x)| < \delta \quad \text{and} \quad \forall x \in \{-1,1\}^n \setminus D : \ |p(x)| < 1 + \delta.$$

Moreover, the δ-approximate bounded degree $\mathrm{bdeg}_\delta(f)$ of f on D is the smallest degree of such a polynomial.

The following lemma connects quantum query complexity to approximate bounded degree.

▶ **Lemma 16** ([11, 2]). *Let $f : D \subseteq \{-1,1\}^n \to \{-1,1\}$ and $\delta > 0$. If a quantum algorithm computes f on D with error δ using T queries, then there is a polynomial p of degree at most $2T$ that 2δ-approximates f on D.*

This implies that the quantum query complexity for computing f with error δ is $\mathrm{bdeg}_{2\delta}(f)/2$, and so we will focus on proving lower bounds on the approximate bounded degree.

We now turn to a dual characterization of this polynomial approximation. This method of dual polynomials dates back to [42, 44] for initially studying communication complexity. Below we refer to some results stated in [19] for studying query complexity.

▶ **Definition 17** (Pure high degree). *A function $\psi : \{-1,1\}^n \to \mathbb{R}$ has pure high degree at least Δ if for every polynomial $p : \{-1,1\}^n \to \mathbb{R}$ with $\deg(p) < \Delta$ it satisfies $\sum_{x\in\{-1,1\}^n} p(x)\psi(x) = 0$. We denote this as $\mathrm{phd}(\psi) \geq \Delta$.*

One can observe that $\mathrm{phd}(\psi) \geq \Delta$ is equivalent to the fact that all the monomials of ψ are of degree at least Δ. Then by weak LP duality we get the following result.

▶ **Theorem 18** ([19, Proposition 2.3]). *Let $f : D \subseteq \{-1,1\}^n \to \{-1,1\}$ and $\delta > 0$. Then $\mathrm{bdeg}_\delta(f) \geq \Delta$ iff there exists a function $\psi : \{-1,1\}^n \to \mathbb{R}$ such that*

$$\sum_{x\in D} \psi(x)f(x) - \sum_{x\in\{-1,1\}^n\setminus D} |\psi(x)| > \delta; \tag{1}$$

$$\|\psi\|_1 = \sum_{x\in\{-1,1\}^n} |\psi(x)| = 1; \tag{2}$$

$$\mathrm{phd}(\psi) \geq \Delta. \tag{3}$$

Now we are going to discuss how to extend these results to non-boolean functions, which is the interesting case for us.

For non-Boolean partial symmetric functions

We now consider a property of a sequence of integers as a function $F : D \subseteq [R]_0^N \to \{-1,1\}$. The symbol 0 will play a special role that will be exhibited later on. Unfortunately one cannot just take the polynomial of those integers. The standard approach (see [1]) is to encode $s = (s_1, \ldots, s_N) \in [R]_0^N$ into binary variables $x = (x_{i,j})_{i\in[N],j\in[R]_0} \in \{-1,1\}^{N(R+1)}$ encoding whether $s_i = j$ as follows: $x_{i,j} = -1$ if $s_i = j$, and $x_{i,j} = 1$ otherwise. Let $H_b^{N(R+1)} \subseteq \{-1,1\}^{N(R+1)}$ be the set of all possible encodings of vectors s, that is, for every $i \in [N]$ there is exactly one $j \in [R]_0$ such that $x_{i,j} = -1$.

This way we can represent F as a function $F_b : D_b \to \{-1,1\}$ where $D_b \subseteq H_b^{N(R+1)}$ is the set of valid encodings of D. More precisely, each $x \in D_b$ satisfies two constraints: (1) $x \in H_b^{N(R+1)}$; and (2) x encodes some $s \in D$. Since only inputs $x \in H_b^{N(R+1)}$ correspond to possible input sequences of an algorithm, the polynomials derived from a quantum query algorithm might not be bounded outside of that set. This implies a slight modification on the definition of approximate degree, in order to relate it to query complexity as in [1].

But before doing this, we are going to relax the constraints on the domain D_b in the case of symmetric functions, while we decrease its dimension. When F is *symmetric* (i.e., $F(s) = F(s \circ \pi_N)$ for any permutation π_N of $[N]$), one can instead define a function $F_{\leq N}$ with weaker constraints by removing the variables corresponding to the symbol 0. Define $H_{\leq N}^{NR}$ as the set of length-(NR) binary vectors with Hamming weight at most N. Given any $x \in H_{\leq N}^{NR}$, we define its frequency vector $z(x) = (z_0, z_1, \ldots, z_R)$ with $z_j = \#\{i : x_{ij} = -1\}$, for $1 \leq j \leq R$, and $z_0 = N - z_1 - \ldots - z_R$. From the vector $z(x)$, one can define a valid sequence of integers $s(x) \in [R]_0^N$: it can be any sequence from $[R]_0^N$ that has frequency vector $z(x)$. Now we can define $F_{\leq N}$ on domain $D_{\leq N}$ as

$$D_{\leq N} = \{x \in H_{\leq N}^{NR} : s(x) \in D\} \quad \text{and} \quad F_{\leq N}(x) = F(s(x)).$$

In fact, for the special case of total symmetric functions F, we can transform F_b on $H_b^{N(R+1)}$ to $F_{\leq N}$ on $H_{\leq N}^{NR}$ due to the symmetry of F.

In [5] it was proved implicitly that for symmetric F, both F_b and $F_{\leq N}$ variants are equally hard to approximate by polynomials. We now define the appropriate notion of approximate degree for $F_{\leq N}$ and relate it to the query complexity of F as in [19, Theorem 6.5].

▶ **Definition 19** (Double-promise approximate degree). *Let $F : D \subseteq [R]_0^N \to \{-1,1\}$ be symmetric and $\delta > 0$. Define $H_{\leq N}^{NR} \subseteq \{-1,1\}^{NR}$ and $F_{\leq N} : D_{\leq N} \subseteq H_{\leq N}^{NR} \to \{-1,1\}$ as above. A polynomial $p : \{-1,1\}^{NR} \to \mathbb{R}$* double-promise δ-approximates *F on D if*

$$\forall x \in D_{\leq N} : \ |F_{\leq N}(x) - p(x)| < \delta \quad \text{and} \quad \forall x \in H_{\leq N}^{NR} \setminus D_{\leq N} : \ |p(x)| < 1 + \delta.$$

Moreover, the double-promise δ-approximate degree $\mathrm{dpdeg}_\delta(F_{\leq N})$ of $F_{\leq N}$ on $D_{\leq N}$ *is the smallest degree of such a polynomial.*

The following lemma connects the quantum query complexity and double-promise approximate degree.

▶ **Lemma 20** ([1, 5],[20, Theorem 3.9]). *Let $F : D \subseteq [R]_0^N \to \{-1,1\}$ be symmetric and $\delta > 0$. Define $H_{\leq N}^{NR} \subseteq \{-1,1\}^{NR}$ and $F_{\leq N} : D_{\leq N} \subseteq H_{\leq N}^{NR} \to \{-1,1\}$ as above. If a quantum algorithm computes F on D with error δ using T queries, then there is a polynomial p of degree at most $2T$ that double-promise 2δ-approximates $F_{\leq N}$ on $D_{\leq N}$.*

As for the Boolean case, this implies that a quantum algorithm computing F with error δ must make at least $\mathrm{dpdeg}_{2\delta}(F_{\leq N})/2$ queries. We can now also take the dual of this characterization.

▶ **Theorem 21** ([19, Proposition 6.6]). *Let $F : D \subseteq [R]_0^N \to \{-1,1\}$ be symmetric. Define $F_{\leq N} : D_{\leq N} \to \{-1,1\}$ as above. Then $\mathrm{dpdeg}_\delta(F_{\leq N}) \geq \Delta$ iff there exists a function $\psi : \{-1,1\}^{NR} \to \mathbb{R}$ such that*

$$\forall x \in \{-1,1\}^{NR} \setminus H_{\leq N}^{NR}, \quad \psi(x) = 0; \tag{4}$$

$$\sum_{x \in D_{\leq N}} \psi(x) F^{\leq N}(x) - \sum_{x \in H_{\leq N}^{NR} \setminus D_{\leq N}} |\psi(x)| > \delta; \tag{5}$$

$$\|\psi\|_1 = 1 \quad \text{and} \quad \mathrm{phd}(\psi) \geq \Delta. \tag{6}$$

4.2 Preparation

Technically, the problem we use in the proof of Theorem 14 is slightly more restricted than k-collision-freeness: we want to distinguish no k-collision from many distinct collisions of size at least k.

▶ **Definition 22** (Collision function). *Let $\gamma \in (0,1)$. The symmetric function $\mathrm{Collision}_{N,R}^{k,\gamma} : D_{\mathrm{Collision}_{N,R}^{k,\gamma}} \subset [R]^N \to \{-1,1\}$ is defined by $\mathrm{Collision}_{N,R}^{k,\gamma}(s) = -1$ if no integer occurs at least k times in s, $\mathrm{Collision}_{N,R}^{k,\gamma}(s) = 1$ if there are more than γR distinct integers that occur at least k times in s, and it is undefined otherwise.*

Notice that this problem is not a property testing problem, as the outcome is not determined based on the distance between inputs. Nevertheless, it is a valid promise problem and a special case of testing k-collision-freeness, which we use to prove lower bounds on the other problems of interest.

To prove a bound on the Collision function, we will actually relate it to the composition of two more elementary functions g, h, where by composition we mean $(g \circ h)(x) = g(h(x_1), \ldots, h(x_n))$ (where $x = (x_1, \ldots, x_n)$ and each x_i is a binary vector of appropriate dimension). Let us define (i) the threshold function $\mathrm{THR}_N^k : \{-1,1\}^N \to \{-1,1\}$ which is -1 if the input bitstring contains at least k many -1s, and it is 1 otherwise; and (2) the gap version of OR, that is $\mathrm{GapOR}_R^\gamma : D_{\mathrm{GapOR}_R^\gamma} \subset \{-1,1\}^R \to \{-1,1\}$ which takes value 1 if the input is 1^R, -1 if the input contains at least γR many -1s, and is undefined otherwise. We show that the double-promise approximate degree of $\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k$ lower bounds the quantum query complexity of the collision problem.

▶ **Lemma 23.** *Let $k \geq 3$, $0 < \gamma < 1$, $\delta > 0$ and $c > 2$ be constants such that $N/c \leq R \leq N/2$. If the double-promise δ-approximate degree of $\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k$ on domain further restricted to $H_{\leq N}^{NR}$ is at least Δ, then every quantum algorithm computing $\mathrm{Collision}_{N,N}^{k,\gamma/c}$ with error $\delta/2$ must require at least $\Delta/2$ queries.*

Before proving this lemma, we prove some helper propositions. In order to apply the dual polynomial method for partial symmetric functions, we start by proving that $\mathrm{Collision}_{N,R'}^{k,\gamma'}$ is at least as hard as a very similar problem. We introduce a "dummy-augmented" version $\mathrm{dCollision}_{N,R}^{k,\gamma} : D_{\mathrm{dCollision}_{N,R}^{k,\gamma}} \subseteq [R]_0^N \to \{-1,1\}$ of the problem $\mathrm{Collision}_{N,R}^{k,\gamma}$ for the purpose of proving Lemma 23, where now the input sequence can have integer 0, but those 0s are just ignored when they occur. We show that it is enough to prove a lower bound for this second version.

▶ **Proposition 24.** *Let $k \geq 3$, $0 < \gamma < 1$ and $c > 2$ be constants such that $N/c \leq R \leq N/2$. Then $\mathrm{dCollision}_{N,R}^{k,\gamma}$ can be reduced to $\mathrm{Collision}_{N,N}^{k,\gamma/c}$.*

The proof is deferred to the full version of the paper [9]. The following proposition relates dCollision to GapOR $\circ$ THR.

▶ **Proposition 25.** *The domain of $\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k$ is*

$$D_{\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k} = \{x \in \{-1,1\}^{NR} : (\mathrm{THR}_N^k(x_1), \ldots, \mathrm{THR}_N^k(x_R)) \in H_{\geq \gamma R}^R \cup \{1^R\}\}.$$

where $x = (x_1, \ldots, x_R)$ with each $x_i \in \{-1,1\}^N$.

The domain of $(\mathrm{dCollision}_{N,R}^{k,\gamma})^{\leq N}$ is

$$D_{(\mathrm{dCollision}_{N,R}^{k,\gamma})^{\leq N}} = H_{\leq N}^{NR} \cap D_{\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k}.$$

Moreover, restricted to the latter domain they are the same function:

$$(\mathrm{dCollision}_{N,R}^{k,\gamma})^{\leq N} = \mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k.$$

We are now ready to give the proof of Lemma 23.

Proof of Lemma 23. By Proposition 24, instead of $\mathrm{Collision}_{N,N}^{k,\gamma/c}$ we can consider $\mathrm{dCollision}_{N,R}^{k,\gamma}$ (with the appropriate parameters) to show a lower bound. By Proposition 25, we can use Lemma 20 to relate the query complexity of $\mathrm{dCollision}_{N,R}^{k,\gamma}$ to the double-promise degree of $\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k$ with domain further restricted to $H_{\leq N}^{NR}$. ◀

4.3 Main lower bound

Let us fix $f = (\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k)$ with domain $D = D_{(\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k)}$ (see Proposition 25). For technical reasons, in the rest of the section we fix $k \geq 3$ and $N = \lceil 20(2k)^{k/2} \rceil R$. (These parameters are used in the proof of Lemma 29, that omitted from this version of the paper.)

We first define a construction used in order to compose dual polynomials, which was introduced in earlier line of work [44, 33, 43].

▶ **Definition 26** (Dual block composition). *The dual block composition of two functions $\phi : \{-1,1\}^n \to \mathbb{R}$ and $\psi : \{-1,1\}^m \to \mathbb{R}$ is a function $\phi \star \psi : \{-1,1\}^{nm} \to \mathbb{R}$ defined as*

$$(\phi \star \psi)(x) = 2^n \, \phi(\mathrm{sgn}(\psi(x_1)), \ldots, \mathrm{sgn}(\psi(x_n))) \prod_{i \in [n]} |\psi(x_i)|$$

where $x = (x_1, \ldots, x_n)$ and $x_i \in \{-1,1\}^m$, for $i \in [n]$.

This subsection is dedicated to the proof of the following lemma which, together with Lemma 23, implies Theorem 14. Observe that we have to zero out the support of the dual polynomial outside of $H_{\leq N}^{NR}$, since our target domain is not D but $D \cap H_{\leq N}^{NR}$ in Lemma 23.

▶ **Lemma 27.** *Let $N = \lceil 20(2k)^{k/2} \rceil R$ and $0 < \gamma < 1/4^{k-1}$. Then there exists a function $\zeta : \{-1,1\}^{NR} \to \mathbb{R}$ such that*

$$\forall x \in \{-1,1\}^{NR} \setminus H_{\leq N}^{NR}, \quad \zeta(x) = 0; \tag{7}$$

$$\sum_{x \in H_{\leq N}^{NR} \cap D} \zeta(x) f(x) - \sum_{x \in H_{\leq N}^{NR} \setminus D} |\zeta(x)| > 2/3; \tag{8}$$

$$\|\zeta\|_1 = 1 \quad \text{and} \quad \mathrm{phd}(\zeta) \in \Omega\left(\sqrt{N^{1-1/k}} / \ln^2 N\right). \tag{9}$$

Proof. The construction of ζ starts by *block composing* (Definition 26) two dual polynomials ϕ, ψ, one for GapOR_R^γ and one for THR_N^k. The dual polynomial ϕ for GapOR_R^γ is given by Proposition 28. The dual polynomial ψ for THR_N^k is given by the first part of Lemma 29.

The block composition $\phi \star \psi$ is a good candidate for the dual polynomial of f. Indeed, Lemma 30 shows that it satisfies Equation (8), showing correlation at least $9/10 > 2/3$. One could also check that it satisfies Equation (9). Nonetheless it does not satisfy Equation (7).

We can now use the second part of Lemma 29 to argue that there exists another dual polynomial ζ that satisfies Equation (7) and Equation (9). Moreover, this ζ is close to $\phi \star \psi$ so that it also satisfies Equation (8), with the weaker but sufficient correlation $9/10 - 2/9 > 2/3$. This concludes the proof. ◀

As we have seen, the previous proof relies on the following results. The first one is direct and we omit its proof.

▶ **Proposition 28.** *Let $\phi : \{-1,1\}^R \to \mathbb{R}$ be such that $\phi(-1^R) = -1/2$, $\phi(1^R) = 1/2$, and $\phi(z) = 0$ for all $z \in \{-1,1\}^R \setminus \{-1^R, 1^R\}$. Then $\|\phi\|_1 = 1$, $\mathrm{phd}(\phi) \geq 1$, and*

$$\sum_{x \in \{-1,1\}^R} \phi(x) \mathrm{OR}(x) = 1.$$

The second lemma is the rephrasing of several scattered results in [19] that we unify into one statement for more clarity. For the sake of completeness, we explain how to prove it in the full version of the paper [9].

▶ **Lemma 29.** *Let $N = \lceil 20(2k)^{k/2} \rceil R$ and $\phi : \{-1,1\}^R \to \mathbb{R}$ from Proposition 28. Then there exist $\psi : \{-1,1\}^N \to \mathbb{R}$ and $\zeta : \{-1,1\}^{NR} \to \mathbb{R}$ such that*

1. $\|\psi\|_1 = 1$, $\mathrm{phd}(\psi) \geq c_1 \sqrt{k^{-1} N^{1-1/k}}$,

$$\sum_{x \in D_+} |\psi(x)| \leq \frac{1}{48N}, \quad \text{and} \quad \sum_{x \in D_-} |\psi(x)| \leq \frac{1}{2} - \frac{2}{4^k},$$

   where $D_+ = \{x \in \{-1,1\}^N : \psi(x) > 0, \mathrm{THR}_N^k(x) = -1\}$ and $D_- = \{x \in \{-1,1\}^N : \psi(x) < 0, \mathrm{THR}_N^k(x) = 1\}$.
2. $\|\zeta\|_1 = 1$, $\mathrm{phd}(\zeta) = \Omega(\sqrt{N^{1-1/k}}/\ln^2 N)$, $\|\zeta - \phi \star \psi\|_1 \leq 2/9$, *and* $\zeta(x) = 0$ *for all* $x \in \{-1,1\}^{NR} \setminus H_{\leq N}^{NR}$.

Finally we are ready to state the last missing statement, the proof of which is our main technical contribution to this part. The proof of [19, Lemma 6.9] does not apply directly to this problem: they use the fact that the dual polynomial ψ of their inner function (OR) has one sided error, which is not the case here.

As now we focus on the composed function f (and the dual composition $\phi \star \psi$), the domain is not restricted to small Hamming weight inputs anymore. For space saving purposes, the proof can be found in the paper's full version [9].

▶ **Lemma 30.** *Let $N = \lceil 20(2k)^{k/2} \rceil R$ and $0 < \gamma < 1/4^{k-1}$. Functions ϕ from Proposition 28 and ψ from Lemma 29 satisfy*

$$\sum_{x \in D} (\phi \star \psi)(x) \cdot f(x) - \sum_{x \in \{-1,1\}^{NR} \setminus D} |(\phi \star \psi)(x)| \geq 9/10.$$

Finally, we can conclude the proof of Theorem 14.

Proof of Theorem 14. By Lemma 27, there is a dual polynomial for $\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k$ of pure high degree $\Omega(\sqrt{N^{1-1/k}}/\ln^2 N)$ that is only supported on $H_{\leq N}^{NR}$. By Theorem 21 this means that the double-promise δ-approximate degree of $\mathrm{GapOR}_R^\gamma \circ \mathrm{THR}_N^k$ with domain restricted to $H_{\leq N}^{NR}$ is $\Omega(\sqrt{N^{1-1/k}}/\ln^2 N)$. Using Lemma 23 with $c = \lceil 20(2k)^{k/2} \rceil$, we obtain that the bounded-error quantum query complexity of $\mathrm{Collision}_{N,N}^{k,\gamma'}$ is $\Omega(\sqrt{N^{1-1/k}}/\ln^2 N)$ if $\gamma' = \gamma/c < 1/(4^{k-1} \lceil 20(2k)^{k/2} \rceil)$. This implies the same lower bound on testing k-collision-freeness with $\varepsilon = \gamma'$ as Collision is just a more restricted version of the same problem. ◀

5 Testing 3-colorability

In this section, we will prove that the problem of testing 3-colorability in bounded degree graphs remains maximally hard-to-test in the quantum setting. Our lower bound proof will roughly follow the same approach as that of [15], see [14, Section 5.6] also for a reference. Our result is stated below.

▶ **Theorem 31** (Restatement of Theorem 4)**.** *Let G be an unknown undirected N-vertex graph with maximum degree d, and $\varepsilon \in (0,1)$ be a parameter. Given quantum query access to G in the undirected bounded-degree graph model, in order to distinguish if G is 3-colorable, or if it is ε-far from being 3-colorable, $\Omega(N)$ quantum queries are necessary.*

To prove the above theorem, we adapt the classical lower bound for testing 3-colorability. Let us start with the notion of k-wise independent string which will be used in the quantum lower bound proof.

▶ **Definition 32** (k-wise independent string). *A string $s = (s_1, \ldots, s_N) \in \{0,1\}^N$ is said to be k-*wise independent *if for any set of k-indices $i_1, i_2 \ldots, i_k$, the probability of any particular assignment $(b_{i_1}, b_{i_2}, \ldots, b_{i_k}) \in \{0,1\}^k$ to the indices $i_1, i_2 \ldots, i_k$ is equal to $1/2^k$.*

As we discussed in the overview, to prove the lower bound of 3-colorability, the authors in [15] studied another problem called $\mathrm{E}(3,c)$LIN-2, a problem related to deciding the satisfiability of a system of linear equations. Then the authors designed a reduction to 3-colorability from $\mathrm{E}(3,c)$LIN-2, which finally proves the linear query complexity lower bound for testing 3-colorability. We will also follow a similar approach here. Let us first formally define the problem of $\mathrm{E}(3,c)$LIN-2, where below $\mathbb{F}_2$ denotes the field with two elements.

▶ **Definition 33** ($\mathsf{E}(3,c)$LIN-2). *Let $\mathcal{E}$ be a system of linear equations with N variables from $\mathbb{F}_2$, where there are 3 variables in each equation, and each variable occurs in at most c equations. This system $\mathcal{E}$ is represented as a matrix and we have query access to its entries. Given a parameter $\alpha \in (0,1)$, the goal is to distinguish if $\mathcal{E}$ is satisfiable, or at least an α-fraction of the equations need to be modified to make $\mathcal{E}$ satisfiable.*

We will first prove the quantum lower bound for testing $\mathrm{E}(3,c)$LIN-2. Our result is stated as follows.

▶ **Lemma 34.** *For every $\alpha > 0$ there are constants c and $\delta > 0$ such that every algorithm that distinguishes satisfiable instances of $E(3,c)$LIN-2 with N variables from instances that are $(1/2 - \alpha)$-far from satisfiable must have quantum query complexity at least $\delta N/2$.*

In order to prove the above lemma, we will use the following observation from [8], which states that distinguishing between a uniformly random string and a ℓ-wise independent string, for an appropriate integer ℓ, is hard for quantum algorithms.

▶ **Proposition 35** (Fact 1 from [8]). *The output distribution of a quantum algorithm making q queries to a uniformly random string is identical to the same algorithm making q queries to a $2q$-wise independent string.*

Finally, we have the reduction that maps satisfying instances of testing $\mathrm{E}(3,c)$LIN-2 to satisfying instances of testing 3-colorability and vice-versa.

▶ **Lemma 36** ([15, Section 4]). *There exists a reduction φ that maps instances of testing $E(3,c)$LIN-2 to instances of testing 3-colorability such that the following hold:*

1. *If an input x to $E(3,c)$LIN-2 is satisfiable, then $\varphi(x)$ is a 3-colorable graph;*
2. *If an input x to $E(3,c)$LIN-2 is far from being satisfiable, then $\varphi(x)$ is a graph that is far from being 3-colorable.*

The proofs of Lemma 34 and of Theorem 31 are simple combinations of the classical results and of Proposition 35. For the sake of completeness, they are included in the full version of the paper [9].

Using various reductions following [45, 27], we obtain the following corollaries.

▶ **Corollary 37** (Hamiltonian Path/Cycle). *Given quantum query access to an unknown undirected (directed) d-bounded degree N-vertex graph G for some integer d, and a parameter $\varepsilon \in (0,1)$, in order to distinguish if G has an undirected (directed) Hamiltonian path/cycle or ε-far from having an undirected (directed) Hamiltonian path/cycle, $\Omega(N)$ quantum queries are necessary.*

▶ **Corollary 38** (Approximating Independent Set/Vertex Cover size). *Given query access to an unknown undirected d-bounded degree N-vertex graph G for some integer d, and a parameter $\varepsilon \in (0,1)$, approximating the independent set size/vertex cover of G, $\Omega(N)$ quantum queries are necessary.*

References

1. Scott Aaronson. Quantum lower bound for the collision problem. In *Proceedings of 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 635–642. Association for Computing Machinery, 2002. doi:10.1145/509907.509999.
2. Scott Aaronson, Andris Ambainis, Janis Iraids, Martins Kokainis, and Juris Smotrovs. Polynomials, quantum query complexity, and Grothendieck's inequality. In *Proceedings of the 31st Conference on Computational Complexity (CCC)*, volume 50, pages 25:1–25:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.25.
3. Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004. doi:10.1145/1008731.1008735.
4. A. Ambainis. Quantum search algorithms. *SIGACT News*, 35(2):22–35, 2004. doi:10.1145/992287.992296.
5. Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1(3):37–46, 2005. doi:10.4086/toc.2005.v001a003.
6. Andris Ambainis, Aleksandrs Belovs, Oded Regev, and Ronald de Wolf. Efficient quantum algorithms for (gapped) group testing and junta testing. In *Proceedings of the 27th annual Symposium on Discrete Algorithms (SODA)*, pages 903–922, 2016. doi:10.1137/1.9781611974331.ch65.
7. Andris Ambainis, Andrew M Childs, and Yi-Kai Liu. Quantum property testing for bounded-degree graphs. In *Proceedings of the 14th International Workshop and 15th International Conference on Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques (APPROX-RANDOM)*, pages 365–376. Springer Berlin Heidelberg, 2011. doi:10.5555/2033252.2033285.
8. Simon Apers and Ronald De Wolf. Quantum speedup for graph sparsification, cut approximation, and Laplacian solving. *SIAM Journal on Computing*, 51(6):1703–1742, 2022. doi:10.1137/21M1391018.
9. Simon Apers, Frédéric Magniez, Sayantan Sen, and Dániel Szabó. Quantum property testing in sparse directed graphs. *arXiv:2410.05001*, 2024. doi:10.48550/arXiv.2410.05001.
10. Simon Apers and Alain Sarlette. Quantum fast-forwarding: Markov chains and graph property testing. *Quantum Information & Computation*, 19(3–4):181–213, 2019. doi:10.5555/3370245.3370246.
11. Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. doi:10.1145/502090.502097.
12. Shalev Ben-David, Andrew M Childs, András Gilyén, William Kretschmer, Supartha Podder, and Daochen Wang. Symmetries, graph properties, and quantum speedups. In *61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 649–660. IEEE, 2020. doi:10.1109/FOCS46700.2020.00066.
13. Michael A Bender and Dana Ron. Testing properties of directed graphs: acyclicity and connectivity. *Random Structures & Algorithms*, 20(2):184–205, 2002. doi:10.1002/rsa.10023.
14. Arnab Bhattacharyya and Yuichi Yoshida. *Property Testing - Problems and Techniques*. Springer Singapore, 1 edition, 2022. doi:10.1007/978-981-16-8622-1.
15. Andrej Bogdanov, Kenji Obata, and Luca Trevisan. A lower bound for testing 3-colorability in bounded-degree graphs. In *43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 93–102. IEEE Computer Society, 2002. doi:10.1109/SFCS.2002.1181886.
16. Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. *Tight Bounds on Quantum Searching*, chapter 10, pages 187–199. John Wiley & Sons, Ltd, 1999. doi:10.1002/3527603093.ch10.

17 Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In *Proceedings of the 3rd Latin American Symposium on Theoretical Informatics (LATIN)*, pages 163–169. Springer Berlin Heidelberg, 1998. doi:10.1007/BFb0054319.

18 Harry Buhrman, Lance Fortnow, Ilan Newman, and Hein Röhrig. Quantum property testing. *SIAM Journal on Computing*, 37(5):1387–1400, 2008. doi:10.1137/S0097539704442416.

19 Mark Bun, Robin Kothari, and Justin Thaler. The polynomial method strikes back: Tight quantum query bounds via dual polynomials. *Theory of Computing*, 16(10):1–71, 2020. doi:10.4086/toc.2020.v016a010.

20 Mark Bun and Justin Thaler. A nearly optimal lower bound on the approximate degree of AC^0. *SIAM Journal on Computing*, 49(4):FOCS17–59–FOCS17–96, 2020. doi:10.1137/17M1161737.

21 Hubie Chen and Yuichi Yoshida. Testability of homomorphism inadmissibility: Property testing meets database theory. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, pages 365–382. Association for Computing Machinery, 2019. doi:10.1145/3294052.3319679.

22 Artur Czumaj, Pan Peng, and Christian Sohler. Relating two property testing models for bounded degree directed graphs. In *Proceedings of the 48th annual Symposium on Theory of Computing (STOC)*, pages 1033–1045. Association for Computing Machinery, 2016. doi:10.1145/2897518.2897575.

23 Artur Czumaj and Christian Sohler. Sublinear-time algorithms. In Oded Goldreich, editor, *Property Testing: Current Research and Surveys*, pages 41–64. Springer Berlin Heidelberg, 2010. doi:10.1007/978-3-642-16367-8_5.

24 Eldar Fischer. The art of uninformed decisions. *Bulletin of the EATCS*, 75:97, 2001.

25 Sebastian Forster, Danupon Nanongkai, Liu Yang, Thatchaphol Saranurak, and Sorrachai Yingchareonthawornchai. Computing and testing small connectivity in near-linear time and queries via fast local cut algorithms. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2046–2065. Society for Industrial and Applied Mathematics, 2020. doi:10.5555/3381089.3381215.

26 Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017. doi:10.1017/9781108135252.

27 Oded Goldreich. *On Testing Hamiltonicity in the Bounded Degree Graph Model*, pages 282–292. Springer Nature Switzerland, 2025. doi:10.1007/978-3-031-88946-2_15.

28 Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, 1998. doi:10.1145/285055.285060.

29 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. doi:10.1007/s00453-001-0078-7.

30 Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 212–219. Association for Computing Machinery, 1996. doi:10.1145/237814.237866.

31 Aram W Harrow, Cedric Yen-Yu Lin, and Ashley Montanaro. Sequential measurements, disturbance and property testing. In *Proceedings of the 28th Annual Symposium on Discrete Algorithms (SODA)*, pages 1598–1611. Society for Industrial and Applied Mathematics, 2017. doi:10.5555/3039686.3039791.

32 Frank Hellweg and Christian Sohler. Property testing in sparse directed graphs: Strong connectivity and subgraph-freeness. In *20th Annual European Symposium on Algorithms (ESA)*, pages 599–610. Springer Berlin Heidelberg, 2012. doi:10.1007/978-3-642-33090-2_52.

33 Troy Lee. A note on the sign degree of formulas. *arXiv:0909.4607*, 2009. doi:10.48550/arXiv.0909.4607.

34 Qipeng Liu and Mark Zhandry. On finding quantum multi-collisions. In *Advances in Cryptology (EUROCRYPT)*, pages 189–218. Springer International Publishing, 2019. doi:10.1007/978-3-030-17659-4_7.

35 Nikhil S. Mande, Justin Thaler, and Shuchen Zhu. Improved approximate degree bounds for k-distinctness. In *Proceedings of the 15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC)*, volume 158, pages 2:1–2:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.TQC.2020.2.

36 Ashley Montanaro and Ronald de Wolf. *A Survey of Quantum Property Testing*. Number 7 in Graduate Surveys. Theory of Computing Library, 2016. doi:10.4086/toc.gs.2016.007.

37 Yaron Orenstein and Dana Ron. Testing Eulerianity and connectivity in directed sparse graphs. *Theoretical Computer Science*, 412(45):6390–6408, 2011. doi:10.1016/j.tcs.2011.06.038.

38 Pan Peng and Yuyang Wang. An optimal separation between two property testing models for bounded degree directed graphs. In *50th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 261, pages 96:1–96:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ICALP.2023.96.

39 Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM Journal on Computing*, 39(3):813–842, 2009. doi:10.1137/070701649.

40 Dana Ron. Algorithmic and analysis techniques in property testing. *Foundations and Trends in Theoretical Computer Science*, 5(2):73–205, 2010. doi:10.1561/0400000029.

41 Ronitt Rubinfeld and Asaf Shapira. Sublinear time algorithms. *SIAM Journal on Discrete Mathematics*, 25(4):1562–1588, 2011. doi:10.1137/100791075.

42 Alexander A. Sherstov. The pattern matrix method. *SIAM Journal on Computing (SICOMP)*, 40(6):1969–2000, 2011. doi:10.1137/080733644.

43 Alexander A. Sherstov. The intersection of two halfspaces has high threshold degree. *SIAM Journal on Computing*, 42(6):2329–2374, 2013. doi:10.1137/100785260.

44 Yaoyun Shi and Yufan Zhu. Quantum communication complexity of block-composed functions. *Quantum Information & Computation*, 9(5):444–460, 2009. doi:10.5555/2011791.2011798.

45 Yuichi Yoshida and Hiro Ito. Query-number preserving reductions and linear lower bounds for testing. *IEICE transactions on Information and Systems*, E93.D(2):233–240, 2010. doi:10.1587/transinf.E93.D.233.

46 Yuichi Yoshida and Hiro Ito. Testing k-edge-connectivity of digraphs. *Journal of Systems Science and Complexity*, 23(1):91–101, 2010. doi:10.1007/s11424-010-9280-5.