Robust and Space-Efficient Dual Adversary Quantum Query Algorithms

Michael Czekanski ✉
Department of Statistics and Data Science, Cornell University, Ithaca, NY, USA

Shelby Kimmel ✉
Department of Computer Science, Middlebury College, VT, USA

R. Teal Witter ✉
Department of Computer Science and Engineering, New York University, Brooklyn, NY, USA

Abstract

The general adversary dual is a powerful tool in quantum computing because it gives a query-optimal bounded-error quantum algorithm for deciding any Boolean function. Unfortunately, the algorithm uses linear qubits in the worst case, and only works if the constraints of the general adversary dual are exactly satisfied. The challenge of improving the algorithm is that it is brittle to arbitrarily small errors since it relies on a reflection over a span of vectors. We overcome this challenge and build a robust dual adversary algorithm that can handle approximately satisfied constraints. As one application of our robust algorithm, we prove that for any Boolean function with polynomially many 1-valued inputs (or in fact a slightly weaker condition) there is a query-optimal algorithm that uses logarithmic qubits. As another application, we prove that numerically derived, approximate solutions to the general adversary dual give a bounded-error quantum algorithm under certain conditions. Further, we show that these conditions empirically hold with reasonable iterations for Boolean functions with small domains. We also develop several tools that may be of independent interest, including a robust approximate spectral gap lemma, a method to compress a general adversary dual solution using the Johnson-Lindenstrauss lemma, and open-source code to find solutions to the general adversary dual.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Quantum query complexity

Keywords and phrases Quantum Computing, Robust Quantum Algorithms, Johnson-Lindenstrauss Lemma, Span Programs, Query Complexity, Space Complexity

Digital Object Identifier 10.4230/LIPIcs.ESA.2023.36

Related Version *Full Version*: https://arxiv.org/abs/2306.15040 [17]

Supplementary Material
Software (Source Code): https://github.com/rtealwitter/QuantumQueryOptimizer
archived at swh:1:dir:60941e1654f097fdcaf2c413c8f7955482f84eb1

Funding *Shelby Kimmel*: sponsored by the U.S. Army Research Office and this work was accomplished under Grant Number W911NF-20-1-0327. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.
R. Teal Witter: supported by NSF No. 1922658 and 1916505.

Acknowledgements We thank Stacey Jeffery for elucidating discussions.

31st Annual European Symposium on Algorithms (ESA 2023).
Editors: Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman; Article No. 36; pp. 36:1–36:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The query model of computation has proven a powerful model in which to prove quantum-classical separations [26, 19, 15] and to understand the limits of quantum algorithms [2, 6, 28]. The power and usefulness of this computational model in the quantum setting in part derives from the fact that bounded-error quantum query complexity for Boolean function evaluation is tightly and beautifully described by a semidefinite program (SDP) – the general adversary bound [46, 44]. The dual of this semidefinite program has played an important role in the development and understanding of quantum algorithms. In particular, the dual has been used to show the optimality of span program algorithms, which are a critical element for several algorithm design paradigms [45, 8, 30, 32] and are useful in a wide range of applications [11, 20, 12, 9, 7]. While we have methods to create a bounded-error quantum algorithm for function evaluation based on a set of vectors that exactly satisfy the constraints of the general adversary dual [44], it is natural to ask how robust this algorithm is to transformations and perturbations. For example, can we take a vector set that satisfies the dual adversary constraints and modify it to obtain an algorithm with better space complexity? Or can we create an algorithm from a vector set that satisfies relaxed dual adversary constraints? In this paper, we find criteria under which these modified or approximately satisfying vector sets yield viable algorithms, and we consider two problems where such robustness is useful.

First, we study when we can reduce the space complexity of dual adversary-derived algorithms. Almost all Boolean functions on n bits have unitary space complexity $\Omega(n)$ [31], but we hope to discover conditions under which less space might be required. We do this by determining when we can compress the dimension of a set of vectors that exactly satisfies the dual adversary constraints. While the dual adversary is most commonly studied in the context of query complexity, is it also closely related to unitary space complexity [31], and the dimension of the satisfying vectors to the dual adversary problem determines the space used by the algorithm. As building large quantum computers will likely continue to be a technical challenge in the near to medium term [29], finding ways to minimize the space used by quantum computers is important.

We use two approaches to compress the dimension of a satisfying vector set, and hence reduce the space used by the resulting algorithm. First, we consider using a unitary transformation to rotate the vectors to a smaller space. Next, we analyze applying the Johnson-Lindenstrauss (JL) lemma, which is a powerful tool for compressing the dimension of a vector set while approximately preserving the structure of the vectors. With our analysis, we find the simpler, unitary transformation always results in a better compression than the JL approach. But, with either compression method, we can show for any function with polynomially many 1-valued inputs, or polynomially many 0-valued inputs, there is a query-optimal algorithm that uses logarithmic space. While it is not surprising that there is an algorithm that uses logarithmic space for such problems (one could iterate through possible 1/0-valued inputs and run Grover's search to test each one), it is not obvious that there is a *query-optimal* algorithm that uses logarithmic space.

The second problem we consider is how to create an algorithm using the output of a numerical SDP solver applied to the general adversary dual. One can plug the general adversary dual into a classical SDP solver and find a set of vectors that is close to a query-optimal, exactly satisfying vector set. Due to finite precision, we expect that the numerical solution will almost never be *exactly* satisfying, but we would like to know if we can still produce a query-optimal quantum algorithm. With our tools for creating robust dual adversary algorithms, we bound the error that can be tolerated, and we describe how to take a vector set that does not quite satisfy the dual adversary constraints, but is within the tolerable error, and use it to create a bounded-error algorithm. In the worst case, our analysis requires an error that scales inversely with the number of 1-valued or 0-valued inputs. While this can be exponential in number of bits in the function, prior to our work, it was not clear how to create *any* algorithm from an approximately satisfying solution. Moreover, the general adversary bound is an SDP with dimension $|X|$ where X is the function domain, so in general, solvers will already take time polynomial in $|X|$ to solve classically [48], so we expect that attaining this level of precision will only contribute polynomially to the overall runtime. Additionally, we show numerically that, at least for small functions, the error bound we require is easily attainable.

One may naturally wonder why a vector set that approximately satisfies the dual adversary constraints does not immediately yield an appropriate algorithm. The challenge is that the standard algorithm is based on a reflection about the span of a subset of those vectors. For example, consider a vector set that should ideally be $\{|0\rangle, |1\rangle, |0\rangle + |1\rangle\}$, but is instead $\{|0\rangle + 10^{-10}|2\rangle, |1\rangle, |0\rangle + |1\rangle\}$. The span of the first set is 2-dimensional, but the span of the second set is 3-dimensional, even though the two vector sets are very close by almost any metric. If the algorithm reflects over a space that is much larger than it should, it might not correctly evaluate the function on all inputs. Our techniques allow us to find appropriate reflections (or in fact unitaries that are close to reflection) so that the algorithms can proceed, even with errors like the example above. Along the way towards proving our main results, we also develop several tools that may be of independent interest, including a robust approximate spectral gap lemma and open-source code to find solutions to the general adversary dual.

1.1 Related Work

Space Complexity and Compression

Reichardt observes that that the space used by the dual adversary algorithm is the log of the rank of Z, where Z is the positive semidefinite matrix that optimizes the primal general adversary bound, and he notes that this provides a worst case $\log(n|X|)$ space complexity for n-bit functions with domain X [44, 43]. Our exact compression result (Theorem 9) is of a similar flavor, except that we are using the "rank" of the dual. Barnum, Saks, and Szegedy use a different family of SDPs to characterize query complexity [5] (these SDPs can give improved performance in the case of small or zero error algorithms), and their algorithm again depends on the rank of the satisfying positive semidefinite matrix, but in the worst case uses $\log|X| + 1$ qubits.

The key tool we use in our compression application is the Johnson-Lindenstrauss lemma. The lemma guarantees that high-dimensional vectors randomly compressed into a lower-dimensional space approximately preserves the inner products of the vectors [34]. The JL lemma is used in a variety of classical applications including compressed sensing, dimensionality reduction, and machine learning [23, 49, 13], and it works even with sparse compression matrices [35]. In fact, our work, in which we compress the dual solution to an SDP, has similarities to work by So et al. [47], which uses JL compression to reduce the rank of the matrix that is the primal solution to a semidefinite programming problem, at the cost of only approximately satisfying the constraints. It is also known that the compressed dimension given by the Johnson-Lindenstrauss lemma is optimal up to constant factors [39].

The idea of relaxing SDP constraints in order to improve the space used by an algorithm has also been considered in the classical regime. Ding et al. create a storage-optimal SDP solver by relaxing constraints [22].

In the quantum arena, a natural application of the JL lemma would be to compress the size of quantum states, but Harrow et al. found that there is no such mapping that significantly reduces the dimension of quantum states while preserving the Schatten 2-norm distance with high probability [27]. However, the JL lemma was used to compress the space used by a quantum finger printing protocol [25].

Span programs (which are equivalent to the dual adversary [44, 46]) were in fact originally formulated in order to understand classical space complexity [36], and Jeffery shows lower bounds on the space complexity of function evaluation that depend on minimum span program and approximate span program sizes [31].

Numerical Solutions to the Dual Adversary

The idea of using classical computers to design quantum algorithms is not new. The variational quantum eigensolver iteratively uses a classical computer to make a ground state ansatz, which is then tested by a quantum computer [42]. A classical machine learning algorithm can be used to guide quantum algorithm design [4]. However, the dual adversary semidefinite programming problem is different in that it automatically produces a query-optimal algorithm, rather than an iterative process guided by classical, heuristic optimization methods.

1.2 Open Questions

Our techniques for space compression preserve the quantum query complexity of the original algorithm, while attempting to reduce space complexity. It would be very interesting if they could be modified to reduce space at the cost of increased query complexity; this might provide insight into one of Aaronson's 2021 open query complexity problems [1]: better understanding space and query trade offs, specifically for the problems of collision and element distinctness.

While we analyze dual adversary algorithms, these are closely related to span program algorithms. It should be possible to translate the bounds and conditions we find on the robustness of the general adversary dual into analogous bounds and conditions on span program algorithms. We are especially curious if these relaxed constraints could be related to Approximate Span Programs [30], which are another way of relaxing the constraints of standard span programs.

While we show conditions under which it is possible to create dual adversary algorithms, we do not prove lower bounds. It would be interesting to study whether, with more detailed analysis, or under additional natural conditions, the JL approach to space compression could be improved, or whether the analysis we present is optimal.

Generalizations of the general adversary bound characterize the problems of quantum state conversion [40] and quantum subspace conversion [10]. Perhaps our techniques could be extended to these additional regimes.

2 Preliminaries

A few notational conventions: we use $\||\psi\rangle\|$ do denote the ℓ_2 norm of $|\psi\rangle$, $[n]$ to denote $\{1, 2, \ldots, n\}$, and $\delta_{i,j}$ to denote the Kronecker delta function. If $|\lambda\rangle$ is an eigenvector of U with eigenvalue $e^{i\beta}$, we say the phase of $|\lambda\rangle$ is β. For any unitary U, let $P_\Theta(U)$ be the orthogonal projector onto the eigenvectors of U with phase at most Θ. That is, $P_\Theta(U)$ is the orthogonal projector onto $\text{span}\{|\lambda\rangle : U|\lambda\rangle = e^{i\beta}|\lambda\rangle \text{ with } |\beta| \leq \Theta\}$.

We consider the quantum query complexity and space complexity of evaluating a function $f : X \to \{0,1\}$ where $X \subseteq \{0,1\}^n$. For such a function f, we define $f^{-1}(b) = \{x \in X : f(x) = b\}$. For some $x \in X \subseteq \{0,1\}^n$, we are given access to an oracle O_x that acts on $\mathbb{C}^n \otimes \mathbb{C}^2$ as $O_x|i\rangle|b\rangle = |i\rangle|b \oplus x_i\rangle$, where $|i\rangle$ for $i \in [n]$ and $|b\rangle$ for $b \in \{0,1\}$ are standard basis states, and x_i is the i^{th} bit of x. Given O_x, we would like to determine $f(x)$. We do this by implementing a bounded-error quantum query algorithm, which without loss of generality takes the form

$$U_T O_x U_{T-1} O_x \cdots U_1 O_x U_0 |\hat{0}\rangle, \tag{1}$$

followed by a two-outcome measurement that determines the output of the algorithm, where $U_0, \ldots U_T$ are unitary operations acting on a Hilbert space of size S, such that for every input $x \in X$, the probability of outputting $f(x)$ is at least $2/3$. The algorithm uses T applications of the oracle and $\log S$ qubits of space. The bounded-error query complexity of f is the minimum query complexity of any bounded-error query algorithm for f.

The general adversary dual is used in designing query-optimal quantum algorithms for function evaluation:

▶ **Definition 1** (General Adversary Dual). *Let $f : X \to \{0,1\}$ for $X \subseteq \{0,1\}^n$. The following semidefinite optimization problem is called the dual of the general adversary bound, or what we call the general adversary dual:*

$$\min_{\substack{m \in \mathbb{N} \\ |v_{x,j}\rangle \in \mathbb{C}^m}} \left\{ \max_{x \in X} \sum_j \||v_{x,j}\rangle\|^2 \right\} \tag{2}$$

$$\text{s.t. } \forall x, y \in X : f(x) \neq f(y), \qquad 1 = \sum_{j : x_j \neq y_j} \langle v_{x,j} | v_{y,j} \rangle. \tag{3}$$

While Definition 1 seeks to minimize the dimension m of the vectors $\{|v_{x,j}\rangle\}_{x \in X, j \in [n]}$ (we will drop the set-building subscript and use $\{|v_{x,j}\rangle\}$ when clear from context) that also minimizes $\max_{x \in X} \sum_j \||v_{x,j}\rangle\|^2$, we note that to design an algorithm, we only need a vector set $\{|v_{x,j}\rangle\}$ that satisfies the constraints in Equation (3). This motivates the following definition, similar to converting vector sets in [3].

▶ **Definition 2** (Deciding Vector Set and Related Terms). *Let $f : X \to \{0,1\}$ for $X \subseteq \{0,1\}^n$, and let $m \in \mathbb{N}$. Then $\{|v_{x,j}\rangle \in \mathbb{C}^m\}_{x \in X, j \in [n]}$ is an f-deciding vector set if*

$$\forall x, y \in X : f(x) \neq f(y), \qquad 1 = \sum_{j : x_j \neq y_j} \langle v_{x,j} | v_{y,j} \rangle. \tag{4}$$

We say the size *of $\{|v_{x,j}\rangle\}$ is $\max_{x \in X} \sum_j \||v_{x,j}\rangle\|^2$, the* dimension *is m, and the* maximum rank *is $\max_{j \in [n]} \quad \operatorname{rank}\{|v_{x,j}\rangle : f(x) = 1\}$.*

Given an f-deciding vector set, one can design a query algorithm to decide f:

▶ **Theorem 3** ([44, 40]). *For $f : X \to \{0,1\}$ with $X \subseteq \{0,1\}^n$ let $\{|v_{x,j}\rangle\}_{x \in X, j \in [n]}$ be an f-deciding vector set with size A and dimension m. Then there is a bounded-error quantum query algorithm that decides f with query complexity $O(A)$ and space complexity $O(\log(nm))$.*

Because any n-bit function can be decided in n queries, we assume $A = O(n)$. Additionally, applying Cauchy-Schwarz to Equation (4), we have $A \geq 1$.

We sketch the proof of Theorem 3 to make it easier to compare with our algorithms. (For a more detailed description using similar notation, see Ref. [14, Chapter 23.6]). The subroutine used in both Theorem 3 and in our algorithms is (parallelized) phase estimation.

▶ **Lemma 4** (Phase Estimation [37, 16, 41]). *Let U be a unitary that acts on n qubits, and let $\delta, \Theta > 0$. Then there is a phase estimation style circuit that has precision Θ, error δ, acts on $n+b$ qubits for $b = O\left(\log \frac{1}{\Theta} \log \frac{1}{\delta}\right)$, and uses $O\left(\frac{1}{\Theta} \log \frac{1}{\delta}\right)$ calls to control-U applied to a single instance of the state $|\psi\rangle$, such that p_0, the probability of outcome 0, satisfies*

$$\|P_{\Theta/2}(U)|\psi\rangle\|^2(1-\delta) - \delta \leq p_0 \leq \|P_\Theta(U)|\psi\rangle\|^2 + \delta. \tag{5}$$

We prove Lemma 4 in the full version of the paper [17], which relies heavily on prior analyses of phase estimation circuits.

The algorithm of Theorem 3 applies phase estimation with precision $O(1/A)$ on a unitary U with an initial state $|\hat{0}\rangle$. U acts on the space $\mathcal{H} = \mathbb{C} \oplus \mathbb{C}^n \otimes \mathbb{C}^m \otimes \mathbb{C}^2$, and is a product of two reflections: $U = (2\Pi_x - I)(2\Delta - I)$, where $\Pi_x = |\hat{0}\rangle\langle\hat{0}| + \sum_{i\in[n]} |i\rangle\langle i| \otimes I \otimes |x_i\rangle\langle x_i|$ (here I acts on $\mathbb{C}^m$, and $|\hat{0}\rangle$ is orthogonal to $\sum_{i\in[n]} |i\rangle\langle i| \otimes I \otimes (|0\rangle\langle 0| + |1\rangle\langle 1|)$) and Δ is the orthogonal projector onto the following set of normalized vectors:

$$|\psi_y\rangle = \frac{1}{\sqrt{\nu_y}} \left(|\hat{0}\rangle + \frac{1}{\sqrt{cA}} \sum_{i\in[n]} |i\rangle|v_{y,i}\rangle|y_i\rangle \right) \qquad \forall y : f(y) = 1, \tag{6}$$

$$\text{s.t. } \nu_y = 1 + \frac{1}{cA} \sum_{i\in[n]} \||v_{y,i}\rangle\|^2 \leq 1 + 1/c, \tag{7}$$

where $\nu_y \geq 1$ is chosen to normalize $|\psi_y\rangle$, and c is a constant chosen depending on the desired success probability. We note $(2\Pi_x - I)$ requires 2 uses of O_x to implement, and $(2\Delta - I)$ depends on the choice of the deciding vector set but is independent of the input x.

Then when $f(x) = 1$ and $c = 2$, $|\hat{0}\rangle$ has high overlap with $|\psi_x\rangle$, which is easily verified to be a 0-phase eigenvector of U, so by Lemma 4, the probability of measuring a phase of 0 when we perform phase estimation is large when δ (the error of phase estimation) is a small constant.

On the other hand, when $f(x) = 0$, we consider the following normalized vector:

$$|\phi_x\rangle = \frac{1}{\sqrt{\mu_y}} \left(|\hat{0}\rangle - \sqrt{cA} \sum_{i\in[n]} |i\rangle|v_{x,i}\rangle|\bar{x}_i\rangle \right) \tag{8}$$

$$\text{s.t. } \mu_x = 1 + cA \sum_{i\in[n]} \||v_{x,i}\rangle\|^2 \leq 1 + cA^2, \tag{9}$$

where $\mu_y \geq 1$ is chosen to normalize the vector. Because $\{|v_{y,i}\rangle\}$ is a deciding vector set, we have $\forall y : f(y) = 1, \langle\psi_y|\phi_x\rangle = 0$. Thus $\{|\phi_x\rangle\}$ is orthogonal to Δ. Next we use the effective spectral gap lemma:

▶ **Lemma 5** (Effective Spectral Gap Lemma [40]). *Let Π, Δ be orthogonal projectors, let $U = (2\Pi - I)(2\Delta - I)$, and $\Delta|w\rangle = 0$. Then*

$$\|P_\Theta(U)\Pi|w\rangle\| \leq \Theta/2\||w\rangle\|. \tag{10}$$

Then the probability of measuring a phase of 0 when we perform phase estimation when $f(x) = 0$, by Lemma 4 is upper bounded by a term that depends on $\|P_\Theta(U)|\hat{0}\rangle\|^2 = \mu_y\|P_\Theta(U)\Pi_x|\phi_x\rangle\|^2$. Applying Lemma 5, and using the fact that $\mu_y = O(A^2)$, we see this is small when Θ, the precision of phase estimation, is chosen to be $O(1/A)$. Since δ (the error of phase estimation) is chosen to be a small constant, by Lemma 4 this leads to a bounded-error algorithm with query complexity $O(A)$, and space complexity $\log(1 + 2nm) + O(\log A) = O(\log(nm))$, as claimed in Theorem 3.

3 Robust Dual Adversary Algorithm

In this section, we show that the dual adversary algorithm has robustness, in that it tolerates errors and flexibility in how it is defined. As described in Section 2, we want to create a bounded-error quantum query algorithm for a Boolean function $f : X \to \{0,1\}$, for $X \subseteq \{0,1\}^n$. Similar to the standard algorithm, our robust algorithm will involve applying phase estimation to a unitary U that acts on the space $\mathcal{H} = \mathbb{C} \oplus \mathbb{C}^n \otimes \mathbb{C}^m \otimes \mathbb{C}^2$. We perform phase estimation on U with a state $|\hat{0}\rangle \in \mathcal{H}$.

We now describe U. As in Section 2, we define the orthogonal projector $\Pi_x = |\hat{0}\rangle\langle\hat{0}| + \sum_{i\in[n]} |i\rangle\langle i| \otimes I \otimes |x_i\rangle\langle x_i|$ on $\mathcal{H}$ where I acts on $\mathbb{C}^m$, and $|\hat{0}\rangle$ is orthogonal to $\sum_{i\in[n]} |i\rangle\langle i| \otimes I \otimes (|0\rangle\langle 0| + |1\rangle\langle 1|)$. Notice that Π_x can be implemented with two applications of the oracle O_x. Let R be another unitary that acts on the same space as Π_x, but R need not be a reflection. Let $U = (2\Pi_x - I)R$.

▶ **Theorem 6.** *Let $\delta, \nu_x, \mu_x > 0$, $\varepsilon_\psi, \varepsilon_\phi \geq 0$, and let $U = (2\Pi_x - I)R$ as defined above, so U acts on $O(\log nm)$ qubits. Consider $0 < \theta \leq 1$. Suppose there are sets of (not necessarily normalized) vectors $\{|\psi_x\rangle = \frac{1}{\sqrt{\nu_x}}(|\hat{0}\rangle + |\eta_x\rangle)\}_{x:f(x)=1}$ and $\{|\phi_x\rangle = \frac{1}{\sqrt{\mu_x}}(|\hat{0}\rangle + |\eta_x\rangle)\}_{x:f(x)=0}$ where $\forall x \in X, \langle\eta_x|\hat{0}\rangle = 0$, and furthermore, that*

1. $\forall x : f(x) = 1, \quad \|(I - U)|\psi_x\rangle\| \leq \varepsilon_\psi$ *and*
2. $\forall x : f(x) = 0, \quad \Pi_x|\eta_x\rangle = 0$ *and* $\|(I + R)|\phi_x\rangle\| \leq \varepsilon_\phi$.

Then the probability of measuring a phase of 0 if we do phase estimation on U with initial state $|\hat{0}\rangle$ with precision θ and error δ when $f(x) = 1$ is at least

$$\left(\sqrt{\nu_x(1 - \frac{5\varepsilon_\psi^2}{\theta^2})} - \sqrt{\nu_x\||\psi_x\rangle\|^2 - 1}\right)^2 (1 - \delta) - \delta, \tag{11}$$

and when $f(x) = 0$ is at most

$$\mu_x \left(\varepsilon_\phi/2 + \theta/2\||\phi_x\rangle\|\right)^2 + \delta. \tag{12}$$

This algorithm uses $O\left(\frac{1}{\theta}\log\frac{1}{\delta}\right)$ queries and $O\left(\log(nm) + \log\frac{1}{\theta}\log\frac{1}{\delta}\right)$ qubits.

Theorem 6 extends the robustness of the algorithm used to prove Theorem 3 (as described in Section 2) in several ways. In the standard analysis, ε_ψ and ε_ϕ are both 0, whereas we now allow them to be non-zero. In addition, Theorem 6 allows for imperfect alignment between the vector sets $\{|\psi_x\rangle\}$ and $\{|\phi_x\rangle\}$ and the unitary U. This will be the key for our applications in the following sections. Additionally, in the standard algorithm, R is chosen to be a reflection, but in Theorem 6, R can be any unitary that satisfies the criterion of Theorem 6. While none of the applications we describe in this paper use this flexibility in the design of R, it might be helpful in future use cases.

To prove Theorem 6, we need the following two lemmas, proved in the full version [17]:

▶ **Lemma 7.** *Let U be a unitary and $0 < \Theta \leq 1$. If $\|(I - U)|\psi_x\rangle\|^2 \leq \varepsilon$, then $\|P_\Theta(U)|\psi_x\rangle\|^2 \geq 1 - \frac{1.1\varepsilon}{\Theta^2}$.*

Lemma 7 tells us that if a unitary U approximately preserves a state $|\psi_x\rangle$, then $|\psi_x\rangle$ has high overlap with the low phase eigenspace of U. This gives us flexibility when $f(x) = 1$, in that our initial state need not have high overlap with the 0-phase space of U, but instead we only require high overlap with the low-phase-eigenspace of U. The proof of Lemma 7 proceeds by decomposing $|\psi_x\rangle$ into its eigenbasis with respect to U, and showing that ϵ serves to bound the amount of amplitude $|\psi_x\rangle$ can have in states with eigenvalues larger than Θ.

▶ **Lemma 8** (Robust Approximate Spectral Gap Lemma). *Let $\varepsilon \geq 0$, Π be an orthogonal projector, R be a unitary, and $U = (2\Pi - I)R$. For $\Theta > 0$, if $\|(I+R)|w\rangle\| \leq \varepsilon$, then*

$$\|P_\Theta(U)\Pi|w\rangle\| \leq \frac{\varepsilon}{2} + \frac{\Theta}{2}\||w\rangle\|. \tag{13}$$

Lemma 8 generalizes the standard approximate spectral gap lemma (Lemma 5), in which R is a reflection and $\varepsilon = 0$. In particular, Lemma 8 shows that when R is not a reflection and when $|w\rangle$ is not exactly given a phase of -1 by R, a variant of the approximate spectral gap lemma still holds.[1] The proof of Lemma 8 closely follows the proof approach of [40, Lemma 4.2], which does not use Jordan's Lemma, but instead directly uses a series of triangle inequalities and observations of preserved subspaces to obtain the result.

Proof of Theorem 6. We first consider the case of x such that $f(x) = 1$. By Lemma 4, the probability that we get an outcome of 0 when we perform phase estimation on the unitary U with initial state $|\hat{0}\rangle$ with precision Θ and accuracy δ is at least $\|P_{\Theta/2}(U)|\hat{0}\rangle\|^2(1-\delta) - \delta$. Now

$$\begin{aligned}\|P_{\Theta/2}(U)|\hat{0}\rangle\| &= \|P_{\Theta/2}(U)\sqrt{\nu_x}|\psi_x\rangle - P_{\Theta/2}(U)|\eta_x\rangle\| \\ &\geq \sqrt{\nu_x}\|P_{\Theta/2}(U)|\psi_x\rangle\| - \|P_{\Theta/2}(U)|\eta_x\rangle\| && \text{(reverse triangle inequality)} \\ &\geq \sqrt{\nu_x\left(1 - \frac{5\varepsilon_\psi^2}{\Theta^2}\right)} - \||\eta_x\rangle\|, && (14)\end{aligned}$$

where the first term in the final line combines Lemma 7 and the assumption that $\|(I - U)|\psi_x\rangle\| \leq \varepsilon_\psi$, so $\|(I-U)|\psi_x\rangle\|^2 \leq \varepsilon_\psi^2$, and the second term uses the fact that projectors can only decrease the ℓ_2 norm of a vector. Thus, using that $\||\eta_x\rangle\| = \sqrt{\nu_x\||\psi_x\rangle\|^2 - 1}$, we have

$$\|P_{\Theta/2}(U)|\hat{0}\rangle\|^2(1-\delta) - \delta \geq \left(\sqrt{\nu_x(1 - \frac{5\varepsilon_\psi^2}{\Theta^2})} - \sqrt{\nu_x\||\psi_x\rangle\|^2 - 1}\right)^2 (1-\delta) - \delta. \tag{15}$$

When $f(x) = 0$, by Lemma 4, the probability that we get an outcome of 0 when we perform phase estimation on U with initial state $|\hat{0}\rangle$ with precision Θ and accuracy δ is at most

$$\|P_\Theta(U)|\hat{0}\rangle\|^2 + \delta = \mu_x\|P_\Theta(U)\Pi_x|\phi_x\rangle\|^2 + \delta, \tag{16}$$

since by assumption, $\Pi_x|\phi_x\rangle = 1/\sqrt{\mu_x}|\hat{0}\rangle$. Then from Lemma 8 and our assumption that $\|(I+R)|\phi_x\rangle\| \leq \varepsilon_\phi$, we have

$$\|P_\Theta(U)\Pi_x|\phi_x\rangle\| \leq \varepsilon_\phi/2 + \Theta/2\||\phi_x\rangle\|. \tag{17}$$

Combining Equation (16) and Equation (17) gives us a probability of outcome 0 of at most

$$\mu_x\left(\varepsilon_\phi/2 + \Theta/2\||\phi_x\rangle\|\right)^2 + \delta. \tag{18}$$

Finally the query complexity and space complexity come from the requirements of phase estimation Lemma 4, and that $2\Pi_x - I$ can be implemented with two uses of the oracle. ◀

[1] We note Lemma 8 is similar to Lemma 3.4 in [33], except we allow R to not be a reflection.

4 Compressing the Dual Adversary Algorithm

In this section, we consider how and when it is possible to reduce the space complexity of the quantum algorithm built from the general adversary dual. In particular, our goal is to take an f-deciding vector set, reduce its dimension and hence create an algorithm which requires fewer qubits to implement.

Our first result, Theorem 9, is a simple compression scheme that shows that an f-deciding vector set on an n-bit function with maximum rank κ' and size A can be compressed to an f-deciding vector set with dimension κ' and size at most A (see Definition 2 for terminology). The number of qubits required by the resulting algorithm is then $O(\log(n\kappa'))$ and the query complexity is $O(A)$, by Theorem 3 and using that $A = O(n)$. Notice that κ' is at most the number of 1-valued inputs, but if many of the f-deciding vectors are linearly dependent, the maximum rank could be much smaller.

We note that an f-deciding vector set is also an $\neg f$-deciding vector set by Definition 2, where $\neg f$ is the negation of f. Also, a $\neg f$-deciding vector set can be used to design a bounded error quantum algorithm for deciding f by negating the output of the algorithm. Thus for a given deciding vector set, we can minimize the space used by the algorithm by considering either $\neg f$ or f.

▶ **Theorem 9.** *Given an f-deciding vector set with maximum rank κ' and size A, we can construct an f-deciding vector set with dimension κ' and size at most A, resulting in an algorithm that decides f with query complexity $O(A)$ and space complexity $O(\log(n\kappa'))$.*

To prove Theorem 9, we apply a series of unitaries to rotate the vectors of the deciding vector set into a smaller dimensional space. This is possible because if we apply the same unitary to two vectors, their inner product is preserved. Our full proof appears in Ref. [17].

The next natural question is whether we can *approximate* the solution to the general adversary dual in a lower dimension. We answer this question by considering a compression of the vectors in a deciding vector set with guarantees from the Johnson-Lindenstrauss lemma, which approximately preserve the inner products of the vectors.

It turns out that this straightforward idea is not trivial to implement. The first challenge is that we must preserve the tensor product structure of our vectors in order to ensure that the query algorithm can apply queries, so we must be careful about the part of the vectors that we compress. The second challenge is that the compression only approximately preserves the inner products of the compressed vectors so we need the robust dual adversary algorithm described in Theorem 6.

Formally stated in Theorem 14, given an f-deciding vector set with maximum rank κ' and size A, we show how to build a quantum algorithm that succeeds with probability $2/3$ and operates in a Hilbert space of dimension $O((\kappa'^2 + A^4\kappa')n)$ with quantum query complexity $O(A)$. Since $A = O(n)$, as discussed below Theorem 3, the number of qubits needed to run the algorithm is no more than $O(\log(\kappa' n))$.

Thus, this approximate compression using the JL lemma achieves the same space complexity as the exact compression of Theorem 9, to within a constant multiplicative factor. This may seem surprising that we are not able to do better, since we are no longer requiring the constraints are exactly satisfied. However, the compression dimension in the JL lemma has a polynomial dependence on the allowed error, and since the amount of error we can tolerate roughly scales with maximum rank, we do not get as much compression as one might hope for.

We now describe at a high level how we prove Theorem 14. While an optimal deciding vector set might use complex vectors, the following lemma, which we prove in Ref. [17], shows that at a small cost in increased vector dimension, we can restrict to real vectors:

▶ **Lemma 10.** *If there is an f-deciding vector set with complex numbers of dimension m and size A, there exists an f-deciding vector set with only real numbers of dimension $2m$ and size A.*

The proof of Lemma 10 proceeds by creating a new vector set from the original where each new real vector consists of the real part of the original complex vector stacked on top of the imaginary part of the original complex vector.

While the standard Johnson-Lindenstrauss lemma guarantees that there is a compression matrix that approximately preserves the l_2-norm difference of any two vectors in a set, we use the following corollary, which shows that the compression approximately preserves inner products in addition to distances, which we prove in Ref. [17]. Our proof of Corollary 11 is similar to a similar result in [38], except that we do not assume the vectors have norm 1.

▶ **Corollary 11.** *Given $\varepsilon > 0$, a set of finite vectors $V \subset \mathbb{R}^d$, and a number $N > 8\ln(|V|)/\varepsilon^2$, there is a compression matrix $S \in \mathbb{R}^{N\times d}$ such that for $|v\rangle, |u\rangle \in V$,*

$$(S|u\rangle)^\dagger(S|v\rangle) = \langle u|v\rangle \pm 2\varepsilon(\||u\rangle\|^2 + \||v\rangle\|^2). \tag{19}$$

The Johnson-Lindenstrauss lemma guarantees the existence of such a compression matrix S, and it can be found probabilistically by sampling random projections. It requires $O(|X|n)$ time to find such a satisfying projection via random sampling [18]. For our purposes, this contributes to classical preprocessing time and space resources, and not towards the quantum query complexity or quantum space use of the quantum algorithm itself. In particular, this sampling requires no queries, so does not contribute to the query complexity.

We now describe how we use Johnson-Lindenstrauss to compress our deciding vector set. Let $\{|v_{x,i}\rangle\}$ be a real f-deciding vector set with size A, dimension m, and maximum rank κ'. Define $\{|\psi_x\rangle\}_{x:f(x)=1}$ and $\{|\phi_x\rangle\}_{x:f(x)=0}$ as in Equations (6) and (8) in Section 2. Let κ be the rank of $\{|\psi_x\rangle\}_{x:f(x)=1}$. (We show in Ref. [17] that $\kappa' \le \kappa \le 2n\kappa'$.)

To compress our vectors, it suffices to compress their orthonormal basis. Let $\{|\zeta_j\rangle\}_{j\in[\kappa]}$ be an orthonormal basis for the space spanned by $\{|\psi_x\rangle\}_{x:f(x)=1}$. Then there are (non-unique) real numbers $\{\alpha_{j,x} \in \mathbb{R}\}_{j\in[n],x\in f^{-1}(1)}$ such that

$$|\zeta_j\rangle = \sum_{x:f(x)=1} \alpha_{j,x}|\psi_x\rangle. \tag{20}$$

We will approximately preserve the structure of the vectors $|\zeta_j\rangle$ in the compression. Thus we define their components

$$|v_{j,i,b}\rangle = \sum_{\substack{x:f(x)=1\\ x_i=b}} \frac{\alpha_{j,x}}{\sqrt{\nu}_x}|v_{x,i}\rangle, \quad \forall j \in [\kappa], i \in [n], b \in \{0,1\}. \tag{21}$$

We will use the random compression matrix S from Corollary 11 to compress the following set of vectors to error ε, as in Corollary 11, (and the compression dimension N will be chosen later to achieve the desired value of ε):

$$\left\{|v_{j,i,b}\rangle\right\}_{j\in[\kappa],i\in[n],b\in\{0,1\}} \bigcup \left\{|v_{y,i}\rangle\right\}_{y:f(y)=0,i\in[n]}. \tag{22}$$

We use these compressed vectors to define

$$\begin{aligned}
\forall x : f(x) = 1, \quad & |\psi'_x\rangle = \left[|\hat{0}\rangle\langle\hat{0}| + (I\otimes S\otimes I)\right]|\psi_x\rangle,\\
\forall x : f(x) = 0, \quad & |\phi'_x\rangle = \left[|\hat{0}\rangle\langle\hat{0}| + (I\otimes S\otimes I)\right]|\phi_x\rangle,\\
\forall j \in [\kappa], \quad & |\zeta'_j\rangle = \left[|\hat{0}\rangle\langle\hat{0}| + (I\otimes S\otimes I)\right]|\zeta_j\rangle.
\end{aligned} \tag{23}$$

As one would expect, the primed, compressed versions of these vectors have approximately the properties of the uncompressed version, as follows:

▶ **Lemma 12.** *For* $\{|\zeta'_j\rangle\}_{j\in[\kappa]}$, $\{|\psi'_x\rangle\}_{x:f(x)=1}$ *and* $\{|\phi'_x\rangle\}_{x:f(x)=0}$ *as described in Equation* (23)*, these vectors have the following properties*

1. $\forall j,l \in [\kappa], \langle\zeta'_j|\zeta'_l\rangle \in \delta_{j,l} \pm 4\varepsilon$
2. $\forall j \in [\kappa], x \in f^{-1}(0), |\langle\zeta'_j|\phi'_x\rangle| \le 2\varepsilon(c+1)A.$
3. $\forall x : f(x) = 1, \left|\||\psi'_x\rangle\|^2 - 1\right| \le 4\varepsilon\kappa$ *and* $\forall x : f(x) = 0, |\||\phi'_x\rangle\| - 1| \le 3\varepsilon.$

The proof of Lemma 12 uses Corollary 11 in fairly straightforward ways.

Let Δ' be the orthogonal projector onto the space spanned by $\{|\psi'_x\rangle\}_{x:f(x)=1}$ and the reflection R be $2\Delta' - I$. By definition, observe that $R|\psi'_x\rangle = |\psi'_x\rangle$, so Theorem 6 Item 1 is satisfied with $\varepsilon_\psi = 0$. Additionally, because $|\phi'_x\rangle$ has the structure

$$|\phi'_x\rangle \propto |\hat{0}\rangle + \sum_{i\in[n]} |i\rangle|v'_{x,i}\rangle|\bar{x}_i\rangle \tag{24}$$

we have $\Pi_x|\phi'_x\rangle \propto |\hat{0}\rangle$, as required by Theorem 6 Item 2. All that is left is to show that $\|(I+R)|\phi'_x\rangle\| \le \varepsilon_\phi$.

▶ **Lemma 13.** *Consider* ε *so that* $\varepsilon\kappa < 1/12$*. For* R *as defined below Lemma 12 and* $|\phi'_x\rangle$ *defined using Equations* (8) *and* (23)*, we have* $\|(I+R)|\phi'_x\rangle\| \le 8\varepsilon(c+1)A\sqrt{\kappa}$*.*

We prove Lemma 13 in Ref. [17]. The main idea is to write Δ' in terms of the vectors $\{|\zeta'\rangle\}$, and then use Lemma 12 Item 2. Along the way, we show how to use Gram-Schmidt to build an orthonormal basis from $\{|\zeta'_j\rangle\}$, which is already almost orthonormal by Lemma 12 Item 1 (see Ref. [17] for details).

With Lemma 13 in hand, the conditions of Theorem 6 are satisfied and we apply it to our compressed vectors.

▶ **Theorem 14.** *Consider a Boolean function* $f : X \to \{0,1\}$ *where* $X \subseteq \{0,1\}^n$ *and an* f*-deciding vector set with maximum rank* κ *and size* A*. Using Johnson-Lindenstrauss compression, we can compress the* f*-deciding vector set to produce a quantum algorithm that correctly evaluates* $f(x)$ *with probability* $2/3$ *for every input* $x \in X$ *with* $O(A)$ *quantum query complexity. The algorithm uses* $O(\log \kappa n)$ *qubits.*

Proof of Theorem 14. We set

$$\Theta = \frac{1}{4\sqrt{c}A}, \quad \varepsilon = \min\left\{\frac{1}{4c\kappa}, \frac{1}{32\sqrt{c}(c+1)A^2\sqrt{\kappa}}\right\}, \quad \delta = \frac{1}{25}, \quad \text{and} \quad c = 100. \tag{25}$$

With these choices, we bound the failure probability below $1/3$ for all inputs x. (Note c appears in Equations (6) and (8).)

Case $\mathbf{f(x) = 1}$. By Theorem 6, for x such that $f(x) = 1$, we have that when $\varepsilon_\psi = 0$, the probability of measuring a phase of 0 is at least

$$\left(\sqrt{\nu_x} - \sqrt{\nu_x\||\psi'_x\rangle\|^2 - 1}\right)^2 (1-\delta) - \delta. \tag{26}$$

We know that $1 \le \nu_x \le 1 + 1/c = 1.01$ from Equation (7) and $|\||\psi'_x\rangle\|^2 - 1| \le 4\varepsilon\kappa \le 1/100$ from Lemma 12 Item 3 and Equation (25) when $c = 100$, so Equation (26) is at least

$$\left(\sqrt{1} - \sqrt{1.01 \cdot 1.01 - 1}\right)^2 (1-\delta) - \delta \ge 2/3 \tag{27}$$

where the final inequality is satisfied for $\delta = 1/25$.

Case $\mathbf{f(x) = 0}$. By Theorem 6, for x such that $f(x) = 0$, we have that the probability of measuring a phase of 0 is at most

$$\mu_x \left(\varepsilon_\phi + \frac{\Theta}{2} \| |\phi'_x\rangle \| \right)^2 + \delta. \tag{28}$$

We know that $\mu_x \leq 1 + cA^2$ from Equation (9), $\varepsilon_\phi \leq 8\varepsilon(c+1)A\sqrt{\kappa}$ from Lemma 13, and $\| |\phi'_x\rangle \| \leq 1 + 3\varepsilon$ from Lemma 12 Item 3. So the probability of measuring a 0 phase is at most

$$(1 + cA^2) \left(8\varepsilon(c+1)A\sqrt{\kappa} + \frac{\Theta}{2}(1 + 3\varepsilon) \right)^2 + \delta. \tag{29}$$

With Θ, ε, δ, and c as set in Equation (25), continuing from Equation (29), we have that probability of failure when $f(x) = 0$ is at most

$$(1 + cA^2) \left(\frac{1}{4\sqrt{c}A} + \frac{1}{4\sqrt{c}A} \right)^2 + \delta \leq \frac{1}{4cA^2} + \frac{1}{4} + \delta \leq \frac{1}{3} \tag{30}$$

since $A \geq 1$.

To achieve this compression with ε as desired we look to Corollary 11 to see what compression dimension is achievable. From Equation (22), we see we are compressing at most $3|X|n$ vectors. Thus we require a compression dimension

$$N = O(\log(|X|n)/\varepsilon^2) = O((\kappa^2 + A^4\kappa)\log(|X|n)). \tag{31}$$

Then since our unitary U acts on $O(\log(nN))$ qubits, by Theorem 6, the space complexity is

$$O(\log nN + \log A) = O\left(\log\left(An(\kappa^2 + A^4\kappa)\log(|X|n)\right)\right) = O(\log(\kappa n)). \tag{32}$$

where we've used that $|X| \leq 2^n$ and $A = O(n)$. Also, from Theorem 6, the query complexity is $O(A)$. ◀

▶ **Corollary 15.** *If there are polynomially many 1-valued or 0-valued inputs to a function $f : X \to \{0,1\}$, then there is a query-optimal quantum algorithm that evaluates f, and that uses $O(\log n)$ qubits.*

Proof. Notice that $\kappa \leq n_1$ where n_1 the number of 1-valued inputs to f. When $\kappa = O(n^d)$ for $d = O(1)$, by Theorem 9 or Theorem 14, we have that for any deciding vector set for f with size A, we can create an algorithm with query complexity $O(A)$ whose space complexity is $O(\log n)$. Since there is always a deciding vector set for f with size A such that $O(A)$ is the optimal query complexity of f [44], our results imply that there exists a query-optimal algorithm that uses logarithmic space. For the case of polynomially many 0-valued inputs, we use $\neg f$. ◀

5 Algorithm from a Numerical Solution

While the dual adversary provides a method of designing optimal query algorithms, in general it might be hard to find an optimal solution. However, since the problem can be formulated as a semidefinite program, we can find a numerical solution. We show that numerical solutions that only approximately satisfy the dual adversary constraints can be used to produce bounded-error quantum algorithms within a constant factor of the objective function value of the numerical solution.

We first provide a theoretical result that shows how an algorithm can be constructed from an *approximate* deciding vector set, a vector set that only approximately satisfies Equation (3). Then we introduce a classical Python package implemented specifically to solve the general adversary dual SDP and show that its solutions often satisfy the error bounds required by our theoretical results.

5.1 Application of Robust Dual Adversary Algorithm

In previous sections we assumed access to an exact solution to the general adversary dual which we used to produce approximate solutions. In this section, we use a finite precision SDP solver to obtain an approximate solution and then build a bounded-error quantum algorithm with the robust dual adversary algorithm described in Theorem 6.

▶ **Theorem 16.** *Consider a Boolean function $f : X \to \{0,1\}$ where $X \subseteq \{0,1\}^n$ and an approximate f-deciding vector set $\{|v_{x,j}\rangle\}_{x\in X, j\in[n]}$ in the sense that*

$$\varepsilon := \max_{x,y: f(x)\neq f(y)} |\langle\psi_x|\phi_y\rangle|, \tag{33}$$

is small as defined below, where $|\psi_x\rangle$, $|\phi_x\rangle$ are as in Equations (6) and (8). Let A be the size and m be the dimension of the approximate f-deciding vector set, with size and dimension defined as in Definition 2. Consider a matrix M with rows $|\psi_x\rangle_{x:f(x)=1}$. Let the singular values of M be $s_1 \geq s_2 \geq \cdots \geq s_\kappa \geq s_{\kappa+1} := 0$ and n_1 be $|\{x : f(x) = 1\}|$. If there exists $\kappa^* \in [\kappa]$ such that

$$\varepsilon \leq \frac{1}{\sqrt{n_1}}\left(\frac{s_{\kappa^*}}{2\sqrt{c}A} - s_{\kappa^*+1}\right) \quad \textit{and} \quad s_{\kappa^*+1} \leq \frac{1}{2\sqrt{1000c}A}, \tag{34}$$

then there is a quantum algorithm that correctly evaluates f with probability at least $2/3$ with at most $O(A)$ queries and $O(\log(nm))$ qubits.

Given any numerical solution, we can simply set $\kappa^* = \kappa$ in which case we require $\varepsilon < s_\kappa/(2\sqrt{cn_1}A)$. However, if the singular values fall off sharply then we can obtain a less stringent constraint on ε, which in turn leads to less precision required by the numerical solver. So in practice, we search for any $\kappa^* \in [\kappa]$ that gives a large enough bound to accommodate the ε we observe in our numerical solution.

To prove Theorem 16, we apply Theorem 6. For this application, we set R to be equal to a reflection over the space spanned by the first κ^* right singular vectors of M, the matrix whose rows are the vectors $\{|\psi_x\rangle\}_{x:f(x)=1}$. We use the following two lemmas to show that this reflection approximately preserves the vectors $\{|\psi_x\rangle\}_{x:f(x)=1}$ and mostly destroys vectors that are almost orthogonal to all vectors in $\{|\psi_x\rangle\}_{x:f(x)=1}$. Their proofs, found in Ref. [17], use standard results from the singular value decomposition approach to approximating of matrices.

▶ **Lemma 17.** *Let M be the matrix whose rows are the vectors $\{|\psi_x\rangle\}_{x\in X_1}$ for some set X_1, and denote M's singular values by $s_1 \geq s_2 \geq \cdots \geq s_\kappa$. Let Δ' be the orthogonal projector onto the first κ^* right singular vectors of M. Then $\forall x \in X_1$, $(2\Delta' - I)|\psi_x\rangle = |\psi_x\rangle + |\eta\rangle$, where $\||\eta\rangle\| \leq 2s_{\kappa^*+1}$ and $s_{\kappa^*+1} = 0$ if $\kappa^* = \kappa$.*

▶ **Lemma 18.** *For disjoint sets X_1 and Y and sets of vectors $\{|\psi_x\rangle\}_{x\in X_1}$ and $\{|\phi_y\rangle\}_{y\in Y}$ such that*

$$|\langle\psi_x|\phi_y\rangle| \leq \varepsilon \quad \forall x \in X_1, y \in Y, \tag{35}$$

let M be the matrix whose rows are the vectors $\{|\psi_x\rangle\}_{x\in X_1}$ and denote its singular values by $s_1 \geq s_2 \geq \cdots \geq s_\kappa$. Let Δ' be the orthogonal projector onto the first κ^* right singular vectors of M. Then $\forall y \in Y$, $(2\Delta' - I)|\phi_y\rangle = |\eta\rangle$, where $\||\eta\rangle\| \leq \frac{s_{\kappa^*+1}+\varepsilon\sqrt{|X_1|}}{s_{\kappa^*}}$ and $s_{\kappa^*+1} = 0$ if $\kappa^* = \kappa$.

We now use Lemmas 17 and 18 to apply Theorem 6 and prove Theorem 16.

Proof of Theorem 16. Let Δ' be the orthogonal projector onto the first κ^* right singular vectors of M, and set $R = 2\Delta' - I$ so as in Theorem 6, $U = (2\Pi_x - I)R$. Then

$$\begin{aligned}\varepsilon_\psi := \|(I-U)|\psi_x\rangle\| &= \||\psi_x\rangle - (2\Pi_x - I)R|\psi_x\rangle\| \\ &= \||\psi_x\rangle - (2\Pi_x - I)(|\psi_x\rangle + |\eta_x\rangle)\| \\ &= \|(2\Pi_x - I)|\eta_x\rangle\| \leq 2s_{\kappa^*+1} \quad \text{(By Lemma 17 and defs of } \Pi_x, |\psi_x\rangle) \\ \varepsilon_\phi := \|(I+R)|\phi_y\rangle\| = \|2\Delta'|\phi_y\rangle\| &\leq 2\frac{s_{\kappa^*+1} + \varepsilon\sqrt{n_1}}{s_{\kappa^*}}. \qquad \text{(By Lemma 18)}\end{aligned}$$

Then by Theorem 6, if $f(x) = 1$, the probability of measuring a phase of 0 is at least

$$\left(\sqrt{\nu_x(1 - \frac{5\varepsilon_\psi^2}{\Theta^2})} - \sqrt{\nu_x\||\psi_x\rangle\|^2 - 1}\right)^2 (1-\delta) - \delta \tag{36}$$

$$\geq \left(\sqrt{(1 - \frac{20s_{\kappa^*+1}^2}{\Theta^2})} - \sqrt{\nu_x\||\psi_x\rangle\|^2 - 1}\right)^2 (1-\delta) - \delta \tag{37}$$

$$\geq \left(\sqrt{1 - \frac{20s_{\kappa^*+1}^2}{\Theta^2}} - \sqrt{1/c}\right)^2 (1-\delta) - \delta \geq 2/3 \tag{38}$$

where we set $c = 100$, $\delta = 1/25$, and $\Theta = (2\sqrt{c}A)^{-1}$. In addition, we used that ν_x is between 1 and $1 + 1/c$ and that $\||\psi_x\rangle\| = 1$, both by Equation (6). Recall that $s_{\kappa^*+1}^2 \leq \Theta^2/1000$ by assumption.

In the other case, where $f(x) = 0$, the probability of measuring a phase of 0 is at most

$$\mu_x\left(2\frac{s_{\kappa^*+1} + \varepsilon\sqrt{n_1}}{s_{\kappa^*}} + \Theta/2\||\phi_x\rangle\|\right)^2 + \delta \leq (1 + cA^2)\left(2\frac{s_{\kappa^*+1} + \varepsilon\sqrt{n_1}}{s_{\kappa^*}} + \Theta/2\right)^2 + \delta \tag{39}$$

$$= (1 + cA^2)\left(\frac{1}{4\sqrt{c}A} + \frac{1}{4\sqrt{c}A}\right)^2 + \delta \tag{40}$$

$$\leq 1/3 \qquad \text{(by the proof of Theorem 14)}$$

when

$$\frac{s_{\kappa^*+1} + \varepsilon\sqrt{n_1}}{s_{\kappa^*}} < \frac{1}{8\sqrt{c}A} \iff \varepsilon < \frac{1}{\sqrt{n_1}}\left(\frac{s_{\kappa^*}}{2\sqrt{c}A} - s_{\kappa^*+1}\right). \tag{41}$$

The algorithm uses within a multiplicative factor of $\frac{1}{\Theta}\log(\frac{1}{\delta}) = 20A\log(20)$ queries by Lemma 4. ◀

The obvious question is whether the conditions in Theorem 16 are met in practice. We show in the next subsection that the conditions are met in the vast majority of numerical solutions to the general adversary dual for the random Boolean functions we find on functions of up to 25 bits with domain size 32.

5.2 Experiments

In this section, we describe how the numerical error (defined in Equation (33)) behaves in practice.

The general adversary dual (Definition 1) is a semidefinite programming (SDP) problem. In order to solve the SDP numerically, we first reformulate it in the following standard form:

$$\min_{\mathbf{X}\in\mathbf{S}^n} \operatorname{tr}(\mathbf{CX}) \quad \text{s.t.} \quad \mathcal{A}(\mathbf{X}) = \mathbf{b} \quad \text{and} \quad \mathbf{X} \succeq 0 \tag{42}$$

where $\mathbf{S}^n$ is the set of $n \times n$ symmetric matrices, $\mathbf{C} \in \mathbf{S}^n$, $\operatorname{tr}(\cdot)$ is the trace, $\mathbf{b} \in \mathbb{R}^m$ is the constraint vector, and

$$\mathcal{A}(\mathbf{X}) := [\operatorname{tr}(\mathbf{A}^1\mathbf{X}), \ldots, \operatorname{tr}(\mathbf{A}^m\mathbf{X})]^\top \tag{43}$$

for constraint matrices $\mathbf{A}^i \in \mathbf{S}^n$. The reformulation requires converting the constraints and introducing appropriate slack variables. The constraints for our SDP are particularly sparse which makes it inefficient to use standard packages like CVXOPT and SDPA [21, 24]. Instead, we use the alternating direction method of [50] which is specifically designed for SDP problems with sparse structure and orthogonal constraints. The pseudocode appears in Algorithm 1.

Algorithm 1 Alternating direction augmented Lagrangian method [50].

```
Require: Constraint matrices $\mathbf{A}, \mathbf{C} \in \mathbf{S}^n$, constraint vector $\mathbf{b} \in \mathbb{R}^m$, iterations $T$, tolerance $t \geq 0$
Ensure: Output $\mathbf{X}$ approximately satisfies Equation (42)
  $\mathbf{X}^0 \leftarrow \mathbf{0}_{n\times n}$                                   ▷ Zero matrix
  $\mathbf{S}^0 \leftarrow \mathbf{I}_{n\times n}$
  for $k = 0, 1, \ldots, T-1$ do
    $y^{k+1} \leftarrow -(\mathcal{A}\mathcal{A}^*)^+(\mathcal{A}(\mathbf{X}^k) - \mathbf{b} + \mathcal{A}(\mathbf{S}^k - \mathbf{C}))$     ▷ $(\cdot)^+$ denotes Moore-Penrose pseudoinverse
    $\mathbf{V}^{k+1} \leftarrow \mathbf{C} - \mathcal{A}^*(y^{k+1}) - \mathbf{X}^k$
    $\mathbf{S}^{k+1} \leftarrow \mathbf{Q}_+\mathbf{\Sigma}_+\mathbf{Q}_+^\top$   ▷ $\mathbf{Q}_+, \mathbf{\Sigma}_+$ contain the non-negative eigendecomposition of $\mathbf{V}^{k+1}$
    $\mathbf{X}^{k+1} \leftarrow \mathbf{S}^{k+1} - \mathbf{V}^{k+1}$
    $\mathbf{X}^{k+1} \leftarrow \operatorname{round}(\mathbf{X}^{k+1})$        ▷ For entries within $t$ of 0 or 1, round to nearest integer
  end for
  $\mathbf{X} \leftarrow \mathbf{X}^{T-1}$
```

We slightly adapt Algorithm 1 to our problem: We store each matrix in a sparse format, round entries of the solution after every iteration, and use the Moore-Penrose pseudoinverse.

The first natural question is how well Algorithm 1 performs. We test this by generating random Boolean functions $f : X \to \{0,1\}$ where $X \subseteq \{0,1\}^n$ for different values of n. Since the dimension of the SDP grows exponentially with $|X|$, we fix $|X| = 32$ for our experiments. For each instance, we compute the maximum numerical error

$$\varepsilon := \max_{x,y: f(x)\neq f(y)} |\langle\psi_x|\phi_y\rangle|. \tag{44}$$

The results in Theorem 16 depend on the error ε being small so it's important that we can obtain solutions with small error efficiently. Figure 1 shows how ε decreases with the number of iterations T of the SDP solver.

The next question is whether the error is sufficiently small to satisfy the requirements of Theorem 16. Figure 2 shows how ε is typically much smaller than the bound required in Theorem 16 for random functions of up to 25 bits with domain size 32. Our experiments test for whether ε satisfies the bound of Theorem 16 in the case that $\kappa^* = \kappa$. It could be that even larger ε is tolerated by considering $\kappa^* < \kappa$.

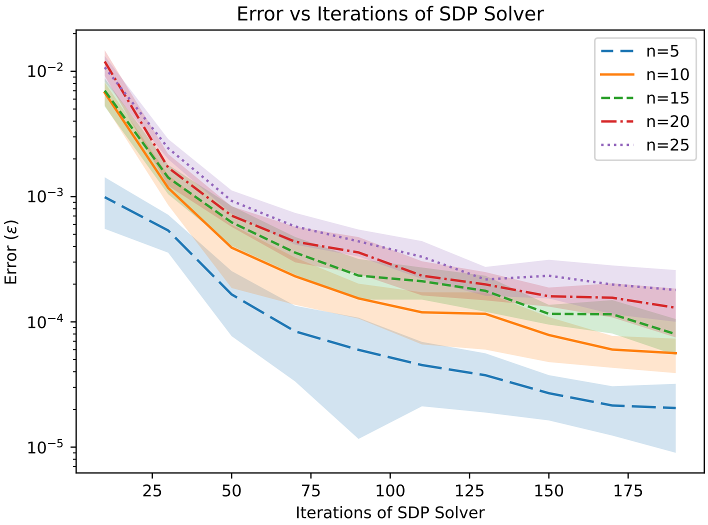

Figure 1 For random Boolean functions with a domain of fixed size, the maximum numerical error decreases with the number of iterations of the SDP solver. Note: the vertical axis is on a logarithmic scale and the shaded regions contain one standard deviation from 20 random instances.

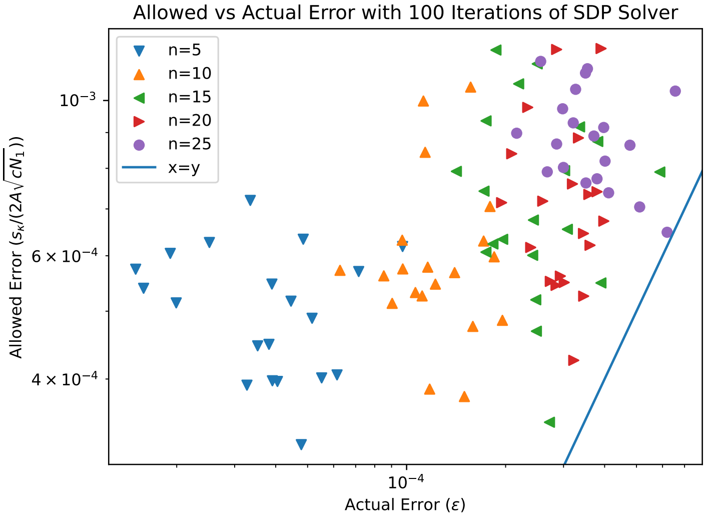

Figure 2 For random Boolean functions with a domain of fixed size, the maximum numerical error stays below the threshold required to construct a provably correct bounded-error quantum algorithm. Note: both axes are on a logarithmic scale and the green region above the diagonal line indicates the error is small enough.

References

1 Scott Aaronson. Open problems related to quantum query complexity. *ACM Transactions on Quantum Computing*, 2(4):1–9, 2021.

2 Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 636–643, 2000.

3 Noel T Anderson, Jay-U Chung, Shelby Kimmel, Da-Yeon Koh, and Xiaohan Ye. Improved quantum query complexity on easier inputs. *arXiv preprint*, 2023. arXiv:2303.00217.

4 Jeongho Bang, Junghee Ryu, Seokwon Yoo, Marcin Pawłowski, and Jinhyoung Lee. A strategy for quantum algorithm design assisted by machine learning. *New Journal of Physics*, 16(7):073017, 2014.

5 Howard Barnum, Michael Saks, and Mario Szegedy. Quantum query complexity and semi-definite programming. In *18th IEEE Annual Conference on Computational Complexity, 2003. Proceedings.*, pages 179–193. IEEE, 2003.

6 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald De Wolf. Quantum lower bounds by polynomials. *Journal of the ACM (JACM)*, 48(4):778–797, 2001.

7 Salman Beigi and Leila Taghavi. Quantum Speedup Based on Classical Decision Trees. *arXiv*, September 2019. arXiv:1905.13095.

8 Salman Beigi and Leila Taghavi. Span Program for Non-binary Functions. *arXiv*, May 2019. arXiv:1805.02714.

9 Aleksandrs Belovs. Learning-graph-based quantum algorithm for k-distinctness. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 207–216. IEEE, 2012.

10 Aleksandrs Belovs and Duyal Yolcu. One-way ticket to las vegas and the quantum adversary. *arXiv preprint*, 2023. arXiv:2301.02003.

11 Chris Cade, Ashley Montanaro, and Aleksandrs Belovs. Time and space efficient quantum algorithms for detecting cycles and testing bipartiteness. *Quantum Information & Computation*, 18(1-2):18–50, February 2018.

12 Titouan Carette, Mathieu Laurière, and Frédéric Magniez. Extended learning graphs for triangle finding. *Algorithmica*, 82(4):980–1005, 2020.

13 Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

14 Andrew Childs. Lecture notes on quantum algorithms, 2022. URL: https://www.cs.umd.edu/~amchilds/qa/qa.pdf.

15 Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68, 2003.

16 R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454(1969):339–354, January 1998. doi:10.1098/rspa.1998.0164.

17 Michael Czekanski, Shelby Kimmel, and R. Teal Witter. Robust and space-efficient dual adversary quantum query algorithms, 2023. arXiv:2306.15040.

18 Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

19 J Niel De Beaudrap, Richard Cleve, John Watrous, et al. Sharp quantum versus classical query complexity separations. *Algorithmica*, 34(4):449–461, 2002.

20 Kai DeLorenzo, Shelby Kimmel, and R. Teal Witter. Applications of the Quantum Algorithm for st-Connectivity. In Wim van Dam and Laura Mancinska, editors, *14th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2019)*, volume 135 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.TQC.2019.6.

21 Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

22 Lijun Ding, Alp Yurtsever, Volkan Cevher, Joel A Tropp, and Madeleine Udell. An optimal-storage approach to semidefinite programming using approximate complementarity. *SIAM Journal on Optimization*, 31(4):2695–2725, 2021.

23 Simon Foucart, Holger Rauhut, Simon Foucart, and Holger Rauhut. *An invitation to compressive sensing*. Springer, 2013.

24 Katsuki Fujisawa, Masakazu Kojima, Kazuhide Nakata, and Makoto Yamashita. Sdpa (semidefinite programming algorithm) user's manual—version 6.2. 0. *Department of Mathematical and Com-puting Sciences, Tokyo Institute of Technology. Research Reports on Mathematical and Computing Sciences Series B: Operations Research*, 2002.

25 Dmitry Gavinsky, Julia Kempe, and Ronald De Wolf. Strengths and weaknesses of quantum fingerprinting. In *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 8–pp. IEEE, 2006.

26 Lov K. Grover. Quantum Mechanics Helps in Searching for a Needle in a Haystack. *Physical Review Letters*, 79(2):325–328, July 1997. doi:10.1103/PhysRevLett.79.325.

27 Aram W Harrow, Ashley Montanaro, and Anthony J Short. Limitations on quantum dimensionality reduction. *International Journal of Quantum Information*, 13(04):1440001, 2015.

28 Peter Hoyer, Troy Lee, and Robert Spalek. Negative weights make adversaries stronger. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 526–535, 2007.

29 IBM. The IBM quantum development roadmap, 2022. URL: https://www.ibm.com/quantum/roadmap.

30 Tsuyoshi Ito and Stacey Jeffery. Approximate Span Programs. *Algorithmica*, 81(6):2158–2195, June 2019. doi:10.1007/s00453-018-0527-1.

31 Stacey Jeffery. Span programs and quantum space complexity. *arXiv preprint*, 2019. arXiv:1908.04232.

32 Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum algorithm for path-edge sampling. *arXiv preprint*, 2023. arXiv:2303.03319.

33 Stacey Jeffery and Sebastian Zur. Multidimensional quantum walks, with application to k-distinctness. *arXiv preprint*, 2022. arXiv:2208.13492.

34 William B Johnson and Joram Lindenstraus. Extensions of lipschitz mappings into a hilbert space. *Contemp. Math.*, 26:189–206, 1984.

35 Daniel M Kane and Jelani Nelson. Sparser johnson-lindenstrauss transforms. *Journal of the ACM (JACM)*, 61(1):1–23, 2014.

36 Mauricio Karchmer and Avi Wigderson. On span programs. In *[1993] Proceedings of the Eigth Annual Structure in Complexity Theory Conference*, pages 102–111. IEEE, 1993.

37 A. Yu Kitaev. Quantum measurements and the Abelian Stabilizer Problem. *arXiv*, November 1995. arXiv:quant-ph/9511026.

38 K. G. Larsen and J. Nelson. Optimality of the Johnson-Lindenstrauss Lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 633–638, October 2017. doi:10.1109/FOCS.2017.64.

39 Kasper Green Larsen and Jelani Nelson. Optimality of the johnson-lindenstrauss lemma. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 633–638. IEEE, 2017.

40 Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum Query Complexity of State Conversion. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 344–353, October 2011. doi:10.1109/FOCS.2011.75.

41 Daniel Nagaj, Pawel Wocjan, and Yong Zhang. Fast amplification of qma. *arXiv preprint*, 2009. arXiv:0904.1549.

42 Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.

43 Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function. *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 544–551, October 2009. doi:10.1109/FOCS.2009.55.

44 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the 2011 Annual ACM-SIAM Symposium on Discrete Algorithms*, Proceedings, pages 560–569. Society for Industrial and Applied Mathematics, January 2011. doi:10.1137/1.9781611973082.44.

45 Ben W Reichardt and Robert Spalek. Span-program-based quantum algorithm for evaluating formulas. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 103–112, 2008.

46 B.W. Reichardt. Span programs are equivalent to quantum query algorithms. *SIAM Journal on Computing*, 43(3):1206–1219, 2014. doi:10.1137/100792640.

47 Anthony Man-Cho So, Yinyu Ye, and Jiawei Zhang. A unified theorem on sdp rank reduction. *Mathematics of Operations Research*, 33(4):910–920, 2008.

48 Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

49 Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

50 Zaiwen Wen, Donald Goldfarb, and Wotao Yin. Alternating direction augmented lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2(3-4):203–230, 2010.