Negative weights make adversaries stronger

Peter Høyer*
hoyer@cpsc.ucalgary.ca

Troy Lee†
lee@lri.fr

Robert Špalek‡
spalek@eecs.berkeley.edu

Abstract

The quantum adversary method is one of the most successful techniques for proving lower bounds on quantum query complexity. It gives optimal lower bounds for many problems, has application to classical complexity in formula size lower bounds, and is versatile with equivalent formulations in terms of weight schemes, eigenvalues, and Kolmogorov complexity. All these formulations are information-theoretic and rely on the principle that if an algorithm successfully computes a function then, in particular, it is able to distinguish between inputs which map to different values.

We present a stronger version of the adversary method which goes beyond this principle to make explicit use of the existence of a measurement in a successful algorithm which gives the correct answer, with high probability. We show that this new method, which we call $\mathrm{ADV}^{\pm}$, has all the advantages of the old: it is a lower bound on bounded-error quantum query complexity, its square is a lower bound on formula size, and it behaves well with respect to function composition. Moreover $\mathrm{ADV}^{\pm}$ is always at least as large as the adversary method ADV, and we show an example of a monotone function for which $\mathrm{ADV}^{\pm}(f) = \Omega(\mathrm{ADV}(f)^{1.098})$. We also give examples showing that $\mathrm{ADV}^{\pm}$ does not face limitations of ADV such as the certificate complexity barrier and the property testing barrier. Breaking these barriers opens the possibility that $\mathrm{ADV}^{\pm}$ can prove better lower bounds where ADV cannot, notably for problems like element distinctness and triangle finding.

*Department of Computer Science, University of Calgary. Supported by Canada's Natural Sciences and Engineering Research Council (NSERC), the Canadian Institute for Advanced Research (CIAR), and The Mathematics of Information Technology and Complex Systems (MITACS).

†LRI, Université Paris-Sud. Supported by a Rubicon grant from the Netherlands Organisation for Scientific Research (NWO) and by the European Commission under the Integrated Project Qubit Applications (QAP) funded by the IST directorate as Contract Number 015848. Part of this work conducted while at CWI, Amsterdam.

‡University of California, Berkeley. Supported by NSF Grant CCF-0524837 and ARO Grant DAAD 19-03-1-0082. Work conducted in part while at CWI and the University of Amsterdam, supported by the European Commission under project QAP, IST-015848, and while visiting the University of Calgary.

1 Introduction

Quantum query complexity is a popular model for study as it seems to capture much of the power of quantum computing—in particular, the search algorithm of Grover [Gro96] and the period finding routine of Shor's factoring algorithm [Sho97] can be formulated in this model—yet is still simple enough that we can often hope to prove tight lower bounds. In this model, complexity is measured by the number of queries made to the input, and other operations are for free. For most known quantum algorithms, the time complexity is bigger than the query complexity by only a polylogarithmic factor.

The two most successful techniques for proving lower bounds on quantum query complexity are the polynomial method [BBC$^+$01] and the quantum adversary method [Amb02]. The adversary method gives tight lower bounds for many problems and is quite versatile with formulations in terms of weight schemes [Amb03, Zha05], eigenvalues [BSS03], and Kolmogorov complexity [LM04]. Špalek and Szegedy [ŠS06] show that in fact all these formulations are equivalent. All these versions of the adversary method rest on the principle that, if an algorithm is able to *compute* a function f, then in particular it is able to *distinguish* inputs which map to different values. The method actually bounds the difficulty of this distinguishing task.

We present a stronger version of the adversary method which goes beyond this principle to essentially make use of the stronger condition that the algorithm actually computes the function—namely, we make use of the existence of a measurement which gives the correct answer with high probability from the final state of the algorithm. This new method, which we call $\mathrm{ADV}^{\pm}$, is always at least as large as the adversary bound ADV, and we show an example of a monotone function f for which $\mathrm{ADV}^{\pm}(f) = \Omega(\mathrm{ADV}(f)^{1.098})$. Moreover, $\mathrm{ADV}^{\pm}$ possesses all the nice properties of the old adversary method: it is a lower bound on bounded-error quantum query complexity, its square is a lower bound on formula size, and it behaves well with respect to function composition. Using this last property, and the fact that our bound is larger than the adversary bound for the base function of Ambainis, we improve the best known separation between quantum query complexity and polynomial degree giving an f such that $Q_\epsilon(f) = \Omega(\deg(f)^{1.329})$.

The limitations of the adversary method are fairly well understood. One limitation is the "certificate complexity barrier." This says that $\mathrm{ADV}(f) \leq \sqrt{C_0(f)C_1(f)}$ for a total function f [Zha05, ŠS06], where $C_0(f)$ is the certificate complexity of the inputs x which evaluate to zero on f, and $C_1(f)$ is the certificate complexity of inputs which evaluate to one. This means that for problems like determining if a graph contains a triangle, or element distinctness, where one of the certificate complexities is constant, the best bound which can be proven by the adversary method is $\Omega(\sqrt{N})$. For triangle finding, the best known upper bound is $N^{13/20}$ [MSS05], and for element distinctness the polynomial method is able to prove a tight lower bound of $N^{2/3}$. We show that our new method can break the certificate complexity barrier—we give an example where $\mathrm{ADV}^{\pm}(f) = \Omega((C_0(f)C_1(f))^{0.549})$.

Another limitation of the adversary method is the "property testing barrier." For a partial Boolean function f where all zero-inputs have relative Hamming distance at least ϵ from all one-inputs, it holds that $\mathrm{ADV}(f) \leq 1/\epsilon$. A prime example where this limitation applies is the collision problem of determining if a function is 2-to-1 or 1-to-1. Here all zero-inputs have relative Hamming distance at least $1/2$ from all one inputs and so the best bound provable by the adversary method is 2, while the polynomial method is able to prove a tight lower bound of $n^{1/3}$ [AS04]. We show the property testing barrier does not apply in this strict sense to $\mathrm{ADV}^{\pm}$, although we do not know of an asymptotic separation for constant ϵ.

Breaking these barriers opens the possibility that $\mathrm{ADV}^{\pm}$ can prove tight lower bounds for problems like element distinctness and the collision problem, and improve the best known $\Omega(\sqrt{N})$ lower bound for triangle finding.

1.1 Comparison with previous methods

We now take a closer look at our new method and how it compares with previous adversary methods. We will use the setting of the spectral formulation of the adversary method [BSS03].

Let $f : S \to \{0,1\}$ be a Boolean function, with $S \subseteq \{0,1\}^n$. Let Γ be a Hermitian matrix with rows and columns labeled by elements of S. We say that Γ is an *adversary matrix for* f if $\Gamma[x,y] = 0$ whenever $f(x) = f(y)$. We let $\|M\|$ denote the spectral norm of the matrix M, and for a real matrix M use $M \geq 0$ to say the entries of M are nonnegative. We now give the spectral formulation of the adversary method:

Definition 1

$$\mathrm{ADV}(f) = \max_{\substack{\Gamma \geq 0 \\ \Gamma \neq 0}} \frac{\|\Gamma\|}{\max_i \|\Gamma \circ D_i\|}.$$

Here the maximum is taken over nonnegative symmetric adversary matrices Γ*, and* D_i *is a zero-one matrix where* $D_i[x,y] = 1$ *if* $x_i \neq y_i$ *and* $D_i[x,y] = 0$ *otherwise.* $\Gamma \circ D_i$ *denotes the entry-wise (Hadamard) product of* Γ *and* D_i*.*

Let $Q_\epsilon(f)$ be the two-sided ϵ-bounded error quantum query complexity of f. Barnum, Saks, and Szegedy show that the spectral version of the adversary method is a lower bound on $Q_\epsilon(f)$:

Theorem 1 ([BSS03]) *For any function* f*,* $Q_\epsilon(f) \geq \frac{1-2\sqrt{\epsilon(1-\epsilon)}}{2}\mathrm{ADV}(f)$.

Note that the definition of $\mathrm{ADV}(f)$ restricts the maximization to adversary matrices whose entries are all nonnegative and real. Our new bound removes these restrictions:

Definition 2

$$\mathrm{ADV}^{\pm}(f) = \max_{\Gamma \neq 0} \frac{\|\Gamma\|}{\max_i \|\Gamma \circ D_i\|}.$$

It is clear that $\mathrm{ADV}^{\pm}(f) \geq \mathrm{ADV}(f)$ for any function f as the maximization is taken over a larger set. Our main theorem, presented in Section 3, states that $\mathrm{ADV}^{\pm}(f)$ is a lower bound on $Q_\epsilon(f)$.

Theorem 2 $Q_\epsilon(f) \geq \frac{1-2\sqrt{\epsilon(1-\epsilon)}}{2}\mathrm{ADV}^{\pm}(f)$.

While it is clear that $\mathrm{ADV}^{\pm}$ is always least as large as ADV, it might at first seem surprising that $\mathrm{ADV}^{\pm}$ can achieve bounds super-linear in ADV. An intuition for why negative weights help is that it is good to give negative weight to entries with large Hamming distance, entries which are easier to distinguish by queries. Consider an entry (x,y) where x and y have large Hamming distance. This entry appears in several $\Gamma \circ D_i$ matrices but only appears in the Γ matrix once. Thus by giving this entry negative weight we can simultaneously decrease $\|\Gamma \circ D_i\|$ for several i's, while doing relatively little damage to the large Γ matrix.

While in form the $\mathrm{ADV}^{\pm}$ bound is very similar to the ADV bound, our proof of Theorem 2 departs from the standard adversary principle. The standard adversary principle is based on the fact that an algorithm A which is able to *compute* a function f is, in particular, able to *distinguish* inputs x, y such that $f(x) \neq f(y)$. Distinguishing quantum states is closely related to the inner product of the states as given by the following quantitative principle:

Theorem 3 *Suppose we are given one of two known states* $|\psi_x\rangle, |\psi_y\rangle$*. Let* $0 \leq \epsilon \leq 1/2$*. There is a measurement which correctly identifies which of the two states we are given with error probability* ϵ *if and only if* $\langle\psi_x|\psi_y\rangle \leq 2\sqrt{\epsilon(1-\epsilon)}$*.*

Let $|\psi_x^t\rangle$ be the state of an algorithm on input x after t queries. The adversary method works by defining a "progress function" based on the inner product $\langle\psi_x^t|\psi_y^t\rangle$. Initially, before the algorithm has made any queries, all inputs look the same and thus $\langle\psi_x^0|\psi_y^0\rangle = 1$ for all x, y, and thus the progress function is large. On the other hand, if a T-query algorithm computes a function f within error ϵ, then by Theorem 3 for x, y with $f(x) \neq f(y)$ we must have $\langle\psi_x^T|\psi_y^T\rangle \leq 2\sqrt{\epsilon(1-\epsilon)}$, and thus the final progress function is small. In [BSS03] this is termed the Ambainis output condition. The adversary method then works by showing an upper bound on how much the progress function can change by a single query.

Our proof follows the same basic reasoning, but the Ambainis output condition no longer seems to suffice to show that the final progress function is small. We use in an essential way the stronger output condition that if a T-query algorithm A computes a Boolean function f, then there exists orthogonal projectors $\{\Pi_0, \Pi_1\}$ which sum to the identity such that $\|\Pi_b|\psi_x^T\rangle\|^2 \geq 1-\epsilon$ when $f(x) = b$, for $b \in \{0, 1\}$.

2 Preliminaries

We assume standard background from quantum computing and Boolean function complexity, see [NC00] and [BW02] for nice references. In this section, we restrict ourselves to more specific background.

2.1 Linear algebra

The background we need about matrices can be found in, for example, [Bha97]. We use standard notations such as $|\cdot|$ for absolute value, $\overline{A}$ for the entrywise complex conjugate of a matrix A, A^* for the conjugate transpose of A, and $\|x\| = \sqrt{x^*x}$ for the ℓ_2-norm of a vector x. For two matrices A, B of the same size, the Hadamard product or entrywise product is the matrix $(A \circ B)[x, y] = A[x, y]B[x, y]$.

For an indexed set of vectors $\{|\psi_x\rangle : x \in S\}$, we associate an $|S|$-by-$|S|$ Gram matrix $M = \mathrm{Gram}(|\psi_x\rangle : s \in S)$ where

$$M[x, y] = \langle\psi_x|\psi_y\rangle.$$

It is easy to see that M is Hermitian and positive semidefinite.

We will make use of several matrix norms. For a matrix A let $\|A\|$ be the spectral norm of A

$$\|A\| = \max_{x,y} \frac{|x^*Ay|}{\|x\|\|y\|}.$$

For two matrices A, B let $\langle A, B\rangle$ be the Hilbert-Schmidt inner product. This is the inner product of A, B viewed as long vectors,

$$\langle A, B\rangle = \mathrm{Tr}(A^*B) = \sum_{i,j} \overline{A[i, j]}B[i, j].$$

The Frobenius norm, denoted $\|A\|_F$, is the norm associated with this inner product,

$$\|A\|_F = \sqrt{\langle A, A\rangle} = \sqrt{\sum_{i,j} |A[i, j]|^2}.$$

Finally, we will use the trace norm, denoted $\|A\|_{tr}$, where

$$\|A\|_{tr} = \max_B \frac{|\langle A, B\rangle|}{\|B\|},$$

and B runs over all complex matrices of the same size as A.

In our proof that $\mathrm{ADV}^{\pm}$ is a lower bound on quantum query complexity we will use two tools for bounding norms. The first of these follows easily from the definition of the trace norm.

Theorem 4 *Let A, B be n-by-n matrices. Then $|\langle A, B\rangle| \leq \|A\|\|B\|_{tr}$.*

Theorem 5 (Hölder's Inequality, [Bha97] Corollary IV.2.6) *Let A, B be matrices such that AB^* is defined. Then*

$$\|AB^*\|_{tr} \leq \|A\|_F \|B\|_F.$$

2.2 Quantum query complexity

As with the classical model of decision trees, in the quantum query model we wish to compute some function f and we access the input through queries. The complexity of f is the number of queries needed to compute f on a worst-case input x. Unlike the classical case, however, we can now make queries in superposition.

The memory of a quantum query algorithm is described by three registers: the input register, H_I, which holds the input $x \in \{0,1\}^n$, the query register, H_Q, which holds an integer $0 \leq i \leq n$, and the working memory, H_W, which holds an arbitrary value. The query register and working memory together form the accessible memory, denoted H_A.

The accessible memory of a quantum query algorithm A is initialized to a fixed state. For convenience, on input x we assume the state of the algorithm is $|x, 0, 0\rangle$ where all qubits in the accessible memory are initialized to 0. The state of the algorithm then evolves through queries, which depend on the input register, and accessible memory operators which do not. We now describe these operations.

We will model a query by a unitary operator where the oracle answer is given in the phase. This operator O is defined by its action on the basis state $|x\rangle|i\rangle|w\rangle$ as

$$O|x\rangle|i\rangle|w\rangle = (-1)^{x_i}|x\rangle|i\rangle|w\rangle.$$

For every x, we define $x_0 = 0$, thus querying $i = 0$ is the identity operation or "null query" which is needed for an important technical reason.

An accesible memory operator is an arbitrary unitary operation U on the accessible memory H_A. This operation is extended to act on the whole space by interpreting it as $I_{input} \otimes U$, where I_{input} is the identity operation on the input space H_I. Thus the state of the algorithm on input x after t queries can be written

$$|\phi_x^t\rangle = U_t O U_{t-1} \cdots U_1 O U_0 |x, 0, 0\rangle.$$

As the input register is left unchanged by the algorithm, we can decompose $|\phi_x^t\rangle$ as $|\phi_x^t\rangle = |x\rangle|\psi_x^t\rangle$, where $|\psi_x^t\rangle$ is the state of the accessible memory after t queries.

The output of a T-query algorithm A on input x is chosen according to a probability distribution which depends on the final state of the accessible memory $|\psi_x^T\rangle$. Namely, the probability that the algorithm outputs the bit $b \in \{0,1\}$ on input x is $\|\Pi_b|\psi_x^T\rangle\|^2$, for a fixed set of projectors $\{\Pi_b\}$ which are orthogonal and complete, that is, sum to the identity. The ϵ-error quantum query complexity of a function f, denoted $Q_\epsilon(f)$, is the minimum number of queries made by an algorithm which outputs $f(x)$ with probability at least $1 - \epsilon$ for every x.

3 Bounded-error quantum query complexity

We now show that $\mathrm{ADV}^{\pm}(f)$ is a lower bound on the bounded-error quantum query complexity of f.

Proof of Theorem 2. Let $f : S \to \{0,1\}$ where $S \subseteq \{0,1\}^n$ be a Boolean function and let Γ be a $|S|$-by-$|S|$ Hermitian matrix such that $\Gamma[x,y] = 0$ if $f(x) = f(y)$. Notice that this property means that Γ corresponds to a weighted bipartite graph, and so the spectrum of Γ is symmetric about the origin. Thus the spectral norm $\|\Gamma\|$ is in fact an eigenvalue. Let δ be an eigenvector of Γ corresponding to the eigenvalue $\|\Gamma\|$.

We imagine that we initially prepare the state $|\Psi^0\rangle = \sum_x \delta_x |x\rangle|0\rangle|0\rangle$ and run the algorithm on this superposition. Thus after t queries we have the state

$$|\psi^t\rangle = U_t O U_{t-1} \dots U_1 O U_0 \sum_x \delta_x |x\rangle|0\rangle|0\rangle = \sum_x \delta_x |x\rangle|\psi_x^t\rangle,$$

where ψ_x^t is the state of the accesible memory of the algorithm on input x after t queries. We define $\rho^{(t)}$ to be the reduced density matrix of the state $|\Psi^t\rangle$ on the input register, that is we trace out the accessible memory. In other words, $\rho^{(t)} = \mathrm{Gram}(\delta_x|\psi_x^t\rangle : x \in S)$.

We define a progress function W^t based on $\rho^{(t)}$ as $W^t = \langle \Gamma, \rho^{(t)} \rangle$. Although phrased differently, this is in fact the same progress function used by Høyer and Špalek [HŠ05] in their proof that the regular adversary method is a lower bound on bounded-error quantum query complexity. Our proof rests on three claims:

1. At the beginning of the algorithm $W^0 = \|\Gamma\|$.
2. At the end of the algorithm $|W^T| \leq 2\sqrt{\epsilon(1-\epsilon)}\|\Gamma\|$.
3. With any one query, the progress measure changes by at most $|W^t - W^{t+1}| \leq 2 \max_i \|\Gamma \circ D_i\|$.

The theorem clearly follows from these three claims. The main novelty of the proof lies in the second step. This is where we depart from the standard adversary principle in using a stronger output condition implied by a successful algorithm.

Item 1: As the state of the accessible memory $|\psi_u^0\rangle$ is independent of the oracle, $\langle \psi_u^0 | \psi_v^0 \rangle = 1$ for every u, v, and so $\rho^{(0)} = \delta\delta^*$. Thus $W^0 = \langle \Gamma, \delta\delta^* \rangle = \mathrm{Tr}(\delta^* \Gamma^* \delta) = \|\Gamma\|$.

Item 2: Now consider the algorithm at the final time T. We want to upper bound $|\langle \Gamma, \rho^{(T)} \rangle|$. The first thing to notice is that as $\Gamma[x,y] = 0$ when $f(x) = f(y)$, we have $\Gamma = \Gamma \circ F$, where F is a zero-one matrix such that $F[x,y] = 1$ if $f(x) \neq f(y)$ and $F[x,y] = 0$ otherwise.

As F is a real symmetric matrix, it is clear from the definition of the Hilbert-Schmidt inner product that $\langle \Gamma \circ F, \rho^{(T)} \rangle = \langle \Gamma, F \circ \rho^{(T)} \rangle$. Now applying Theorem 4 we have $\langle \Gamma, F \circ \rho^{(T)} \rangle \leq \|\Gamma\| \|\rho^{(T)} \circ F\|_{tr}$. It remains to upper bound $\|\rho^{(T)} \circ F\|_{tr}$, which we do using Theorem 5. By this theorem, it suffices to bound $\|X\|_F \|Y\|_F$ for some X, Y such that $XY^* = \rho^{(T)} \circ F$.

Let Π_0, Π_1 be a complete set of orthogonal projectors which determine the output probabilities, that is, the probability that the algorithm outputs b on input x is $\|\Pi_b |\psi_x^T\rangle\|^2$. The correctness of the algorithm tells us that $\|\Pi_{f(x)} |\psi_x^T\rangle\|^2 \geq 1 - \epsilon$ and $\|\Pi_{1-f(x)} |\psi_x^T\rangle\|^2 \leq \epsilon$. We choose X to be the matrix with rows $\Pi_{f(x)} \delta_x |\psi_x^T\rangle$, and Y to be the matrix with rows $\Pi_{1-f(x)} \delta_x |\psi_x^T\rangle$. That is, X is the matrix where we project

onto the correct answers, and Y is the matrix where we project onto the incorrect answers. Using the fact that $\Pi_0\Pi_1 = 0$, a little computation shows that

$$(XY^* + YX^*)[x,y] = \begin{cases} \delta_x\delta_y(\langle\psi_x^T|\Pi_0|\psi_y^T\rangle + \langle\psi_x^T|\Pi_1|\psi_y^T\rangle) & \text{if } f(x) \neq f(y) \\ 0 & \text{if } f(x) = f(y). \end{cases}$$

Since $\Pi_0 + \Pi_1 = I$, we get $XY^* + YX^* = \rho^{(T)} \circ F$. Thus, using the triangle inequality and Theorem 5,

$$\|\rho^{(T)} \circ F\|_{tr} \leq \|XY^*\|_{tr} + \|YX^*\|_{tr} \leq 2\|X\|_F\|Y\|_F.$$

Notice that

$$\|X\|_F^2 + \|Y\|_F^2 = \sum_{x\in S} |\delta_x|^2(\|\Pi_0|\psi_x^T\rangle\|^2 + \|\Pi_1|\psi_x^T\rangle\|^2) = 1$$

and

$$\|Y\|_F^2 = \sum_{x\in S} |\delta_x|^2\|\Pi_{1-f(x)}|\psi_x^T\rangle\|^2 \leq \epsilon\sum_{x\in S}|\delta_x|^2 = \epsilon.$$

The maximum of $\|X\|_F^2\|Y\|_F^2$ under these constraints is $\epsilon(1-\epsilon)$, thus $W^T \leq 2\|\Gamma\|\sqrt{\epsilon(1-\epsilon)}$.

Item 3: We now bound how much the progress function can drop with any single query. To do this, we first look at how a single query affects the inner product between two states $|\psi_x^t\rangle$ and $|\psi_y^t\rangle$. Let O_x denote the oracle operator when the input register has value x, that is $O_x|i\rangle|w\rangle = (-1)^{x_i}|i\rangle|w\rangle$. For each $0 \leq i \leq n$ let $P_i = \sum_{z\geq 0}|i;z\rangle\langle i;z|$ denote the projection onto the subpace querying the ith oracle bit. The $t+1^{\text{st}}$ query changes the inner product by at most the overlap between the projections onto the subspace that corresponds to indices i where x_i and y_i differ.

$$\begin{aligned}\langle\psi_x^t|\psi_y^t\rangle - \langle\psi_x^{t+1}|\psi_y^{t+1}\rangle &= \langle\psi_x^t|\psi_y^t\rangle - \langle\psi_x^t|O_xO_y|\psi_y^t\rangle \\ &= \langle\psi_x^t|(I - O_xO_y)|\psi_y^t\rangle = \sum_{i:x_i\neq y_i} 2\langle\psi_x^t|P_i|\psi_y^t\rangle.\end{aligned}$$

As before, let $\rho^{(t)} = \mathrm{Gram}(\delta_x|\psi_x^t\rangle)$, and let $\rho_i^{(t)} = \mathrm{Gram}(\delta_x P_i|\psi_x^t\rangle)$. Consider $\rho^{(t)} - \rho^{(t+1)}$. Using the above expression we see that $\rho^{(t)} - \rho^{(t+1)} = 2\sum_i \rho_i^{(t)} \circ D_i$. Thus

$$\begin{aligned}W^t - W^{t+1} &= \langle\Gamma, \rho^{(t)} - \rho^{(t+1)}\rangle = 2\langle\Gamma, \sum_i \rho_i^{(t)} \circ D_i\rangle \\ &= 2\sum_i\langle\Gamma, \rho_i^{(t)} \circ D_i\rangle = 2\sum_i\langle\Gamma \circ D_i, \rho_i^{(t)}\rangle,\end{aligned}$$

where the last step follows as D_i is a real symmetric matrix and so can be shifted in the Hilbert-Schmidt inner product. Now applying Theorem 4 gives

$$\begin{aligned}|W^t - W^{t+1}| &\leq 2\sum_i \|\Gamma \circ D_i\|\|\rho_i^{(t)}\|_{tr} \\ &\leq 2\max_i\|\Gamma \circ D_i\|\sum_i\|\rho_i^{(t)}\|_{tr} = 2\max_i\|\Gamma \circ D_i\|.\end{aligned}$$

To see the last equality notice that as each $\rho_i^{(t)}$ is positive semidefinite, $\|\rho_i^{(t)}\|_{tr} = \mathrm{Tr}(\rho_i^{(t)})$. As $\{P_i\}_{0\leq i\leq n}$ form a complete orthonormal set of projectors, $\sum_{i=1}^n \rho_i^{(t)} = \rho^{(t)}$. Finally, $\mathrm{Tr}(\rho^{(t)}) = 1$ as $\rho^{(t)}$ is a density matrix. □

4 Formula size

Laplante, Lee, and Szegedy [LLS06] show that the adversary method can also be used to prove classical lower bounds—they show that $\mathrm{ADV}(f)^2$ is a lower bound on the formula size of f. A formula is circuit with AND, OR, and NOT gates with the restriction that every gate has out-degree exactly one. The size of a formula is the number of leaves and the size of a smallest formua computing f is denoted $L(f)$. We show that $\mathrm{ADV}^{\pm}(f)^2$ remains a lower bound on the formula size of f, denoted $L(f)$.

Theorem 6 $L(f) \geq \mathrm{ADV}^{\pm}(f)^2$.

The proof is given in Appendix A. Note that this theorem does imply a limitation of $\mathrm{ADV}^{\pm}(f)$—it is upper bounded by the square root of the formula size of f. Thus for the binary AND-OR tree—or read-once formulae in general— the largest lower bounds provable by $\mathrm{ADV}^{\pm}$ are $\sqrt{n}$. On the other hand, this theorem can also be seen as giving further evidence to the conjecture of Laplante, Lee, and Szegedy that this is not a limitation at all—they conjecture that bounded-error quantum query complexity squared is in general a lower bound on formula size.

5 Composition theorem

One nice property of the adversary method is that it behaves very well with respect to function composition. For a function $f : \{0,1\}^n \to \{0,1\}$ we define the dth iteration of f, $f^d : \{0,1\}^{n^d} \to \{0,1\}$ recursively as $f^1 = f$ and $f^d = f \circ (f^{d-1}, \ldots, f^{d-1})$ for $d > 1$. Ambainis [Amb03] shows that $\mathrm{ADV}(f^d) \geq \mathrm{ADV}(f)^d$. Thus by proving a good adversary bound on the base function f, one can easily obtain good lower bounds on the iterates of f. In this way, Ambainis shows a super-linear gap between the bound given by the polynomial degree of a function and the adversary method, thus separating polynomial degree and quantum query complexity.

Laplante, Lee, and Szegedy [LLS06] show a matching upper bound for iterated functions, namely that $\mathrm{ADV}(f^d) \leq \mathrm{ADV}(f)^d$. Thus we conclude that the adversary method possesses the following composition property.

Theorem 7 ([Amb03, LLS06]) *For any function $f : S \to \{0,1\}$, with $S \subseteq \{0,1\}^n$ and natural number $d > 0$,*

$$\mathrm{ADV}(f^d) = \mathrm{ADV}(f)^d.$$

We show that one direction of this theorem, the lower bound, also holds for the $\mathrm{ADV}^{\pm}$ bound. This is the direction which is useful for proving separations.

Theorem 8 $\mathrm{ADV}^{\pm}(f^d) \geq \mathrm{ADV}^{\pm}(f)^d$.

The proof is given in Appendix B. We actually prove a more general version of this theorem which gives a lower bound on the adversary bound of any two-level decision tree $h = f \circ (g_1, \ldots, g_k)$ in terms of the adversary bounds of the component functions f, g_i. As with the proof that $\mathrm{ADV}^{\pm}$ is a lower bound on quantum query complexity, the presence of negative entries again causes new difficulties here and our proof is substantially different from previous composition theorems. Also, the dual of the $\mathrm{ADV}^{\pm}$ bound is more complicated than that of the ADV bound, and we have not yet been able to show the upper bound in this theorem.

6 Examples

In this section, we look at some examples to see how negative weights can help to achieve larger lower bounds. We consider two examples in detail: a 4-bit function giving the largest known separation between the polynomial degree and the quantum query complexity, and a 6-bit function breaking the certificate complexity and property testing barriers.

To help find good adversary matrices, we implemented both adversary bounds as semidefinite programs and used the convex optimization package SeDuMi for Matlab. Using these programs, we tested both ADV and $\mathrm{ADV}^{\pm}$ bounds for all 222 functions on 4 or fewer variables which are not equivalent under negation of output and input variables and permutation of input variables (see sequence number A000370 in [Slo]). The $\mathrm{ADV}^{\pm}$ bound is strictly larger than the ADV bound for 128 of these functions. The source code of our semidefinite programs and more examples can be downloaded from [HLŠ06].

6.1 Ambainis function

In order to separate quantum query complexity and polynomial degree, Ambainis defines a Boolean function $f : \{0,1\}^4 \to \{0,1\}$ which is one if and only if the four input bits are sorted[1], that is they are either in a non-increasing or non-decreasing order. This function has polynomial degree 2, and an adversary bound of 2.5. Thus by the composition theorem for the standard adversary method, Ambainis obtains a separation between quantum query complexity and polynomial degree of $Q_\epsilon(f^d) = \Omega(\deg(f^d)^{1.321})$. We have verified that this function indeed gives the largest separation between adversary bounds and polynomial degree over all functions on 4 or fewer variables.

In the next theorem, we construct an adversary matrix with negative weights which shows that $\mathrm{ADV}^{\pm}(f) \geq 2.5135$. Using the composition theorem Theorem 8 we obtain $\mathrm{ADV}^{\pm}(f) \geq \mathrm{ADV}(f)^{1.005}$ and improve the separation between quantum query complexity and polynomial degree to $Q_\epsilon(f^d) = \Omega(\deg(f^d)^{1.3296})$.

Theorem 9 *Let $f : \{0,1\}^4 \to \{0,1\}$ be Ambainis' function. Then $\mathrm{ADV}^{\pm}(f) \geq 2.5135$.*

Proof. We first look at some basic properties of Ambainis' function. It is a balanced function, with 8 inputs which map to zero and 8 inputs which map to one. Every input $x \in \{0,1\}^4$ has 2 sensitive bits and 2 insensitive bits, where flipping both insensitive bits also changes the function value. The function is invariant under complementation of inputs, thus the Hamming distance between any zero-input and one-input is either 1, 2, or 3. We define an adversary matrix Γ where $\Gamma[x,y] = 0$ if $f(x) = f(y)$ and otherwise: $\Gamma[x,y] = a$ if the Hamming distance between (x,y) is 1, $\Gamma[x,y] = b$ if the Hamming distance is 2 and the different bits are both sensitive or both not, $\Gamma[x,y] = c$ if the Hamming distance is 2 and one different bit is sensitive and the other is not, and $\Gamma[x,y] = d$ if the Hamming distance is 3, for some constants a, b, c, d.

It can be shown that for every $i = 1, \ldots, 4$, the matrix $\Gamma \circ D_i$ consists of four 4-by-4 disjoint blocks, and each of these blocks is some permutation of rows and columns of the following matrix B:

$$B = \begin{pmatrix} c & b & d & d \\ b & c & d & a \\ d & d & c & b \\ d & a & b & c \end{pmatrix}. \tag{1}$$

[1]The function was first described in this way by Laplante, Lee, and Szegedy [LLS06]. The function defined by Ambainis [Amb03] can be obtained from this function by exchanging the first and third input bits and negating the output.

The particular block B above is one of the four blocks of $\Gamma \circ D_1$ with columns indexed by zero-inputs 0010, 0100, 0101, 0110, and rows indexed by one-inputs 1000, 1110, 1111, 1100. We choose this particular order of rows instead of the lexicographical order, so that B is a symmetric matrix; its eigenvalues correspond to singular values of a matrix with any different ordering. We maximize the spectral norm of Γ while keeping $\|\Gamma \circ D_i\| \leq 1$. The optimal setting of the four variables can be found numerically by semidefinite programming and is the following:

	ADV	$\mathrm{ADV}^{\pm}$
a	$3/4$	0.5788
b	$1/2$	0.7065
c	0	0.1834
d	0	-0.2120
λ	$5/2$	2.5135

(2)

The eigenvalues of $\Gamma \circ D_i$ are $\{1, 1, \frac{1}{4}, \frac{1}{4}\}$, and the eigenvalues of $\Gamma^{\pm} \circ D_i$ are $\{1, 1, -1, -0.2664\}$. The spectral norm of Γ is at least $2a + 2b + 2c + 2d$, witnessed by the all one vector. Both spectral bounds are tight due to the existence of matching dual solutions; we, however, omit them here. □

6.2 Breaking the certificate complexity barrier

We now consider a function on six bits. We will consider this function in two guises. We first define a partial function f to show that $\mathrm{ADV}^{\pm}$ can break the property testing barrier. We then extend this partial function to a total monotone function g which gives a larger separation between the ADV and $\mathrm{ADV}^{\pm}$ bounds, and also shows that $\mathrm{ADV}^{\pm}$ can break the certificate complexity barrier.

We define the partial function f on six bits as follows:

- The zero inputs of f are: 111000, 011100, 001110, 100110, 110010, 101001, 100101, 010101, 010011, 001011.
- The one inputs of f are: 110100, 110001, 101100, 101010, 100011, 011010, 011001, 010110, 001101, 000111.

Notice that f is defined on all inputs with Hamming weight three, and only on these inputs. This function is inspired by a function defined by Kushilevitz which appears in [NW95] and is also discussed by Ambainis [Amb03]. Kushilevitz's function has the same behavior as the above on inputs of Hamming weight three; it is additionally defined to be 0 on inputs with Hamming weight 0, 4, or 5, and to be 1 on inputs with Hamming weight 1, 2, or 6.

All zero inputs of f have Hamming distance at least 2 from any one input, thus the relative Hamming distance between any zero and one input is $\epsilon = 1/3$. In Theorem 10 we show that $\mathrm{ADV}^{\pm}(f) \geq 2+3\sqrt{5}/5 \approx 3.341$. This implies $\mathrm{ADV}^{\pm}(f) \geq (1/\epsilon(f))^{1.098}$, and as both bounds compose we obtain $\mathrm{ADV}^{\pm}(f^d) \geq (1/\epsilon(f^d))^{1.098}$. This shows that the property testing barrier does not apply to $\mathrm{ADV}^{\pm}$ as it does to ADV. The relative Hamming distance $\epsilon(f^d)$, however, goes to zero when d increases. We don't know of an asymptotic separation for constant ϵ.

We now consider a monotone extension of f to a total function, denoted g. It is additionally defined to be 0 on inputs with Hamming weight 0, 1, or 2, and to be 1 on inputs with Hamming weight 4, 5, or 6. Recall that the maxterms of a monotone Boolean function are the maximal, under subset ordering, inputs x which evaluate to 0, and similarly the minterms are the minimal inputs which evaluate to 1. The zero inputs

of f become maxterms of g and the one inputs become minterms. Since f is defined on all inputs with Hamming weight three, g is a total function. The extended function g is at least as hard as its sub-function f, hence $\mathrm{ADV}^{\pm}(g) \geq \mathrm{ADV}^{\pm}(f)$. The 0-certificates of g are given by the location of 0's in the maxterms and the 1-certificates are given by the location of 1's in the minterms, thus $C_0(g) = C_1(g) = 3$. Both bounds compose thus $C_0(g^d) = C_1(g^d) = 3^d$.

Applying the composition theorem Theorem 8 we obtain $\mathrm{ADV}^{\pm}(g^d) \geq (C_0(g)C_1(g))^{0.549}$. As $\mathrm{ADV}(h) \leq \sqrt{C_0(h)C_1(h)}$ for a total function h, we also conclude $\mathrm{ADV}^{\pm}(g^d) \geq \mathrm{ADV}(g^d)^{1.098}$.

Theorem 10 $\mathrm{ADV}^{\pm}(f) \geq 2 + 3\sqrt{5}/5$.

Proof. In the adversary matrix for f we only give nonzero weight to pairs (x, y) where one is a maxterm and one is a minterm. Furthermore, for a maxterm-minterm pair (x, y), the corresponding entry of the adversary matrix depends only on the Hamming distance between x and y. As all minterms and maxterms have Hamming weight three, the Hamming distance between x and y is even and is either two, four, or six. We label the matrix entries a, b, c respectively for Hamming distances two, four, six. The optimal settings turn out to be $a = (1 + \sqrt{5})/5, b = (1 - \sqrt{5})/5, c = 1/5$.

The function is very regular, thus for any maxterm x, there are six minterms at Hamming distance two, three minterms at Hamming distance four, and one minterm at Hamming distance six. It follows that all rows have the same sum, namely $6a + 3b + c$. This implies that the all ones vector is an eigenvector—it turns out the principal eigenvector—corresponding to the eigenvalue $2 + 3\sqrt{5}/5$.

To complete the proof it remains to verify that $\|\Gamma \circ D_i\| \leq 1$ for every i. By inspection we see that all $\Gamma \circ D_i$ matrices are equivalent up to permutation, which does not change the spectral norm, to the following matrix B:

$$B = \begin{pmatrix} c & b & b & a & a \\ b & c & a & a & b \\ b & a & c & b & a \\ a & a & b & c & b \\ a & b & a & b & c \end{pmatrix}.$$

Calculation shows that the eigenvalues of B are $\{1, 1, 1, -1, -1\}$. □

7 Conclusion

7.1 Open questions

Breaking the certificate complexity and property testing barriers opens the possibility that $\mathrm{ADV}^{\pm}$ can prove better lower bounds where we know ADV cannot. Salient examples are element distinctness, the collision problem, and triangle finding. For element distinctness, the best bound provable by the standard adversary method is $O(\sqrt{n})$ while the polynomial method is able to prove a tight lower bound of $\Omega(n^{2/3})$ [AS04]. For the collision problem, the adversary method is only able to prove a constant lower bound while the polynomial method again proves a tight lower bound of $\Omega(n^{1/3})$ [AS04]. Finally, for the problem of determining if a graph contains a triangle, the best bound provable by the adversary method is $O(n)$ and the best known algorithm is $O(n^{1.3})$ [MSS05]. We have seen that the square of $\mathrm{ADV}^{\pm}$ is a lower bound on formula size. Is this indeed a limitation, or, as conjectured by Laplante, Lee, and Szegedy, is the square of bounded-error quantum query complexity in general a lower bound on formula size?

Acknowledgements

We would like to thank Aram Harrow and Umesh Vazirani for interesting discussions on the topics of this paper, and Ronald de Wolf for many valuable comments on an earlier draft.

References

[AS04] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–767, 2004.

[Amb02] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64:750–767, 2002.

[Amb03] A. Ambainis. Polynomial degree vs. quantum query complexity. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science*, pages 230–239. IEEE, 2003.

[AŠW06] A. Ambainis, R. Špalek, and R. de Wolf. A new quantum lower bound method, with applications to direct product theorems and time-space tradeoffs. In *Proceedings of the 38th ACM Symposium on the Theory of Computing*, pages 618–633. ACM, 2006.

[BBC+01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001.

[BSS03] H. Barnum, M. Saks, and M. Szegedy. Quantum decision trees and semidefinite programming. In *Proceedings of the 18th IEEE Conference on Computational Complexity*, pages 179–193, 2003.

[Bha97] R. Bhatia. *Matrix Analysis*. Springer-Verlag, 1997.

[BW02] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288:21-43, 2002.

[Gro96] L. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th ACM Symposium on the Theory of Computing*, pages 212–219. ACM, 1996.

[HLŠ05] P. Høyer, T. Lee, and R. Špalek. Tight adversary bounds for composite functions. quant-ph/0509067, 2005.

[HLŠ06] P. Høyer, T. Lee, and R. Špalek. Source codes of semidefinite programs for $\mathrm{ADV}^{(\pm)}$. `http://www.ucw.cz/~robert/papers/adv/`

[HŠ05] P. Høyer and R. Špalek. Lower bounds on quantum query complexity. *Bulletin of the European Association for Theoretical Computer Science*, 87:78–103, 2005.

[KN97] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[KW88] M. Karchmer and A. Wigderson. Monotone connectivity circuits require super-logarithmic depth. In *Proceedings of the 20th ACM Symposium on the Theory of Computing*, pages 539–550, 1988.

[LLS06] S. Laplante, T. Lee, and M. Szegedy. The quantum adversary method and classical formula size lower bounds. *Computational Complexity*, 15:163–196, 2006.

[LM04] S. Laplante and F. Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. In *Proceedings of the 19th IEEE Conference on Computational Complexity*, pages 294–304. IEEE, 2004.

[MSS05] F. Magniez, M. Santha, and M. Szegedy. Quantum algorithms for the triangle problem. In *Proc. of 16th ACM-SIAM SODA*, pages 1109–1117, 2005.

[NC00] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[NW95] N. Nisan and A. Wigderson. A note on rank vs. communication complexity. *Combinatorica*, 15(4):557–566, 1995.

[Sho97] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26:1484–1509, 1997.

[Slo] N. Sloane. On-line encyclopedia of integer sequences. `http://www.research.att.com/~njas/sequences/`

[ŠS06] R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2(1):1–18, 2006.

[Zha05] S. Zhang. On the power of Ambainis's lower bounds. *Theoretical Computer Science*, 339(2–3):241–256, 2005.

A Formula size

In this section we give the proof of Theorem 6. We will work in the setting of Karchmer and Wigderson, who characterize formula size in terms of a communication complexity game [KW88]. Since this seminal work, nearly all formula size lower bounds have been formulated in the language of communication complexity.

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function. Following Karchmer and Wigderson, we associate with f a relation $R_f \subseteq \{0,1\}^n \times \{0,1\}^n \times [n]$ where

$$R_f = \{(x, y, z) : f(x) = 0, f(y) = 1, x_z \neq y_z\}.$$

For a relation R, let $C^P(R)$ denote the number of leaves in a smallest communication protocol for R, and let $L(f)$ be the number of leaves in a smallest formula for f. Karchmer and Wigderson show the following:

Theorem 11 $L(f) = C^P(R)$.

We say that a set $S \subseteq X \times Y$ is *monochromatic* with respect to R if there exists $z \in Z$ such that $(x, y, z) \in R$ for all $(x, y) \in S$. It is well known, see for example [KN97], that a successful communication protocol for a relation $R \subseteq X \times Y \times Z$ partitions $X \times Y$ into disjoint combinatorial rectangles which are monochromatic with respect to R. Let $C^D(R)$ be the size of a smallest decomposition of $X \times Y$ into disjoint rectangles monochromatic with respect to R. Clearly, $C^D(R) \leq C^P(R)$. We are actualy able to show the stronger statement that the square of $\mathrm{ADV}^{\pm}(f)$ is a lower bound on the size of a smallest rectangle decomposition of R_f.

Theorem 12 $L(f) \geq C^D(R_f) \geq (\mathrm{ADV}^{\pm}(f))^2$.

Proof. Laplante, Lee, and Szegedy [LLS06] show that two conditions are sufficient for a measure to lower bound formula size. The first is rectangle subadditivity—they show that the spectral norm squared is subadditive over rectangles, and this result holds for an arbitrary, possibly negative, matrix.

Lemma 13 (Laplante, Lee, Szegedy) *Let A be an arbitrary $|X|$-by-$|Y|$ matrix and $\mathcal{R}$ a rectangle partition of $|X| \times |Y|$. Then $\|A\|^2 \leq \sum_{R \in \mathcal{R}} \|A_R\|^2$.*

The second property is monotonicity, and here we need to modify their argument to handle negative entries. They use the property that if A, B are nonnegative matrices, and if $A \leq B$ entrywise, then $\|A\| \leq \|B\|$. In our application, however, we actually know more: if R is a rectangle monochromatic with respect to a color i, then A_R is a *submatrix* of A_i. And, for arbitrary matrices A, B, if A is a submatrix of B then $\|A\| \leq \|B\|$.

This allows us to complete the proof: let $\mathcal{R}$ be a monochromatic partition of R_f with $|\mathcal{R}| = C^D(R_f)$. Then for any matrix A

$$\begin{aligned}\|A\|^2 &\leq \sum_{R \in \mathcal{R}} \|A_R\|^2 \leq C^D(R_f) \cdot \max_R \|A_R\|^2 \\ &\leq C^D(R_f) \cdot \max_i \|A_i\|^2.\end{aligned}$$

And so we conclude

$$L(f) \geq C^D(R_f) \geq \max_{A \neq 0} \frac{\|A\|^2}{\max_i \|A_i\|^2}.$$

□

B Composition theorem

In this section, we prove Theorem 8. We will actually show a more general result which applies to functions that can be written in the form

$$h = f \circ (g_1, \ldots, g_k). \tag{3}$$

One may think of h as a two-level decision tree with the top node being labeled by a function $f : \{0,1\}^k \to \{0,1\}$, and each of the k internal nodes at the bottom level being labeled by a function $g_i : \{0,1\}^{n_i} \to \{0,1\}$. We do not require that the inputs to the inner functions g_i have the same length. An input $x \in \{0,1\}^n$ to h is a bit string of length $n = \sum_i n_i$, which we think of as being comprised of k parts, $x = (x^1, x^2, \ldots, x^k)$, where $x^i \in \{0,1\}^{n_i}$. We may evaluate h on input x by first computing the k bits $\tilde{x}_i = g_i(x^i)$, and then evaluating f on input $\tilde{x} = (\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_k)$.

Adversary bound with costs On the way to proving our composition theorem we consider as an intermediate step a generalization of the adversary method allowing input bits to be given an arbitrary positive cost. For any function $f : \{0,1\}^n \to \{0,1\}$, and any vector $\alpha \in \mathbb{R}_+^n$ of length n of positive reals, we define a quantity $\mathrm{ADV}_\alpha(f)$ as follows:

$$\mathrm{ADV}_\alpha(f) = \max_{\substack{\Gamma \geq 0 \\ \Gamma \neq 0}} \min_i \left\{ \alpha_i \frac{\|\Gamma\|}{\|\Gamma \circ D_i\|} \right\}.$$

We define the analogous quantity $\mathrm{ADV}^{\pm}_{\alpha}(f)$ by enlarging the maximization over all nonzero adversary matrices. We will use the notation $\mathrm{ADV}^{(\pm)}$ to simultaneously refer to both ADV and $\mathrm{ADV}^{\pm}$. One may think of α_i as expressing the cost of querying the ith input bit x_i. For example, x_i could be equal to the parity of α_i new input bits, or, alternatively, each query to x_i could reveal only a fraction of $1/\alpha_i$ bits of information about x_i. When $\alpha = (a, \ldots, a)$ and all costs are equal to a, the new adversary bound $\mathrm{ADV}^{(\pm)}_{\alpha}(f)$ reduces to $a \cdot \mathrm{ADV}^{(\pm)}(f)$, the product of a and the adversary bound $\mathrm{ADV}^{(\pm)}(f)$. In particular, when all costs $a = 1$ we have $Q_\epsilon(f) = \Omega(\mathrm{ADV}^{(\pm)}_{\vec{1}}(f))$. When α is not the all-one vector, then $\mathrm{ADV}^{(\pm)}_{\alpha}(f)$ will not necessarily be a lower bound on the quantum query complexity of f, but this quantity can still be very useful in computing the adversary bound of composed functions.

We are able to give a lower bound on $\mathrm{ADV}^{(\pm)}(h)$ in terms of the adversary bounds $\mathrm{ADV}^{(\pm)}(f), \mathrm{ADV}^{(\pm)}(g_i)$.

Theorem 14 *For any function $h : S \to \{0,1\}$ of the form $h = f \circ (g_1, \ldots, g_k)$ with domain $S \subseteq \{0,1\}^n$, and any cost function $\alpha \in \mathbb{R}^n_+$,*

$$\mathrm{ADV}^{(\pm)}_{\alpha}(h) \geq \mathrm{ADV}^{(\pm)}_{\beta}(f),$$

where $\beta_i = \mathrm{ADV}^{(\pm)}_{\alpha^i}(g_i)$, $\alpha = (\alpha^1, \alpha^2, \ldots, \alpha^k)$, and $\beta = (\beta_1, \ldots, \beta_k)$.

We remark that for the ADV bound one can in fact show that this holds with equality [HLŠ05]. The usefulness of such a theorem is that it allows one to divide and conquer—it reduces the computation of the adversary bound for h into the disjoint subproblems of first computing the adversary bound for each g_i, and then, having determined $\beta_i = \mathrm{ADV}^{(\pm)}(g_i)$, computing $\mathrm{ADV}^{(\pm)}_{\beta}(f)$, the adversary bound for f with costs β.

B.1 Composition Lemma

We now turn to the proof of the composition theorem. Given an adversary matrix Γ_f realizing the adversary bound for f and adversary matrices Γ_{g_i} realizing the adversary bound for g_i where $i = 1, \ldots, k$, we build an adversary matrix Γ_h for the function $h = f \circ (g_1, \ldots, g_k)$. Lemma 15 expresses the spectral norm of this Γ_h in terms of the spectral norms of Γ_f and Γ_{g_i}.

Let Γ_f be an adversary matrix for f, i.e. a Hermitian matrix satisfying $\Gamma_f[x, y] = 0$ if $f(x) = f(y)$, and let δ_f be a prinicipal eigenvector of Γ_f with unit norm. Similarly, let Γ_{g_i} be a spectral matrix for g_i and let δ_{g_i} be a principal eigenvector of unit norm, for every $i = 1, \ldots, k$.

It is helpful to visualize an adversary matrix in the following way. Let $X_f = f^{-1}(0)$ and $Y_f = f^{-1}(1)$. We order the rows first by elements from X_f and then by elements of Y_f. In this way, the matrix has the following form:

$$\Gamma_f = \begin{bmatrix} 0 & {\Gamma_f}^{(0,1)} \\ {\Gamma_f}^{(1,0)} & 0 \end{bmatrix}$$

where ${\Gamma_f}^{(0,1)}$ is the submatrix of Γ_f with rows labeled from X_f and columns labeled from Y_f and ${\Gamma_f}^{(1,0)}$ is the conjugate transpose of ${\Gamma_f}^{(0,1)}$.

Thus one can see that an adversary matrix for a Boolean function corresponds to a (weighted) bipartite graph where the two color classes are the domains where the function takes the values 0 and 1. For $b \in \{0,1\}$ let $\delta_{g_i}^{\restriction b}[x] = \delta_{g_i}[x]$ if $g_i(x) = b$ and $\delta_{g_i}^{\restriction b}[x] = 0$ otherwise. In other words, $\delta_{g_i}^{\restriction b}$ is the vector δ_{g_i} restricted to the color class b.

Before we define our composition matrix, we need one more piece of notation. Let ${\Gamma_f}^{(0,0)} = \|\Gamma_f\| I_{|X_f|}$, where I is a $|X_f|$-by-$|X_f|$ identity matrix and similarly ${\Gamma_f}^{(1,1)} = \|\Gamma_f\| I_{|Y_f|}$.

We are now ready to define the matrix Γ_h:

Definition 3 $\Gamma_h[x,y] = \Gamma_f[\tilde{x},\tilde{y}] \cdot \left(\bigotimes_i \Gamma_{g_i}^{(\tilde{x}_i,\tilde{y}_i)}\right)[x,y]$

Lemma 15 *Let Γ_h be as in Definition 3. Then $\|\Gamma_h\| = \|\Gamma_f\| \cdot \prod_{i=1}^k \|\Gamma_{g_i}\|$ and a principal eigenvector of Γ_h is $\delta_h[x] = \delta_f[\tilde{x}] \cdot \prod_{i=1}^k \delta_{g_i}[x^i]$.*

Proof. The more difficult direction is to show $\|\Gamma_h\| \leq \|\Gamma_f\| \cdot \prod_{i=1}^k \|\Gamma_{g_i}\|$, and we do this first. The outline of this direction is as follows:

1. We first define 2^{k+n} many vectors $\delta_{\alpha,c} \in \mathbb{C}^{2^n}$.
2. We show that each $\delta_{\alpha,c}$ is an eigenvector of Γ_h.
3. We show that $\{\delta_{\alpha,c}\}_{\alpha,c}$ span a space of dimension 2^n. This implies that every eigenvalue of Γ_h is an eigenvalue associated to at least one of the $\delta_{\alpha,c}$ as eigenvectors corresponding to different eigenvalues of a Hermitian matrix are orthogonal.
4. We upper bound the absolute value of the eigenvalues corresponding to the $\delta_{\alpha,c}$ by $\|\Gamma_f\| \cdot \prod_{i=1}^k \|\Gamma_{g_i}\|$.

Let $c = (c_1, \ldots, c_k)$ where $c_i \in [2^{n_i}]$ for $i = 1, \ldots, k$. Let δ_{c_i} be an eigenvector of unit norm corresponding to the c_i^{th} largest eigenvalue of Γ_{g_i}—that is $\Gamma_{g_i}\delta_{c_i} = \lambda_{c_i}(\Gamma_{g_i})\delta_{c_i}$.

It is helpful to look at the matrix Γ_h as composed of blocks labeled by $a, b \in \{0,1\}^k$ where the (a,b) block of the matrix consists of all x, y pairs with $\tilde{x} = a$ and $\tilde{y} = b$. Notice that the (a,b) block of Γ_h is the matrix $\Gamma_f[a,b] \cdot \otimes\Gamma_{g_i}^{(a_i,b_i)}$.

Let $\lambda_{c_i}^0(A) = \|A\|$ and $\lambda_{c_i}^1(A) = \lambda_{c_i}(A)$. We claim that $\Gamma_{g_i}^{(a_i,b_i)}\delta_{c_i}^{\restriction b_i} = \lambda_{c_i}^{a_i \oplus b_i}(\Gamma_{g_i})\delta_{c_i}^{\restriction a_i}$. This is because if $a_i \neq b_i$ then $\Gamma_{g_i}^{(a_i,b_i)}$ is one half of the bipartite matrix Γ_{g_i} and so $\Gamma_{g_i}^{(a_i,b_i)}\delta_{c_i}^{\restriction b_i} = \lambda_{c_i}(\Gamma_{g_i})\delta_{c_i}^{\restriction a_i}$. On the other hand, if $a_i = b_i$ then $\Gamma_{g_i}^{(a_i,b_i)} = \|\Gamma_{g_i}\|I$ and so $\Gamma_{g_i}^{(a_i,b_i)}\delta_{c_i}^{\restriction b_i} = \|\Gamma_{g_i}\|\delta_{c_i}^{\restriction b_i} = \|\Gamma_{g_i}\|\delta_{c_i}^{\restriction a_i}$.

Thus for the tensor product matrix $\otimes\Gamma_{g_i}^{(a_i,b_i)}$ we have that

$$\otimes\Gamma_{g_i}^{(a_i,b_i)} \otimes \delta_{c_i}^{\restriction b_i} = \prod_{i=1}^k \lambda_{c_i}^{a_i \oplus b_i}(\Gamma_{g_i}) \cdot \otimes\delta_{c_i}^{\restriction a_i}.$$

Expanding this equation gives that for every x such that $\tilde{x} = a$

$$\sum_{y:\tilde{y}=b} \otimes\Gamma_{g_i}^{(a_i,b_i)}[x,y] \cdot (\otimes\delta_{c_i})[y] = \prod_{i=1}^k \lambda_{c_i}^{a_i \oplus b_i}(\Gamma_{g_i}) \cdot (\otimes\delta_{c_i})[x]. \tag{4}$$

Now consider a 2^k-by-2^k matrix A_c where

$$A_c[a,b] = \Gamma_f[a,b] \cdot \prod_{i=1}^k \lambda_{c_i}^{a_i \oplus b_i}(\Gamma_{g_i}).$$

Let α be a unit norm eigenvector of this matrix, say with eigenvalue $\mu_{\alpha,c}$. Explicitly writing out the eigenvalue equation means that for every a,

$$\sum_b \Gamma_f[a,b] \cdot \prod_{i=1}^k \lambda_{c_i}^{a_i \oplus b_i}(\Gamma_{g_i}) \cdot \alpha[b] = \mu_{\alpha,c}\, \alpha[a]. \tag{5}$$

Item 1: We are ready to define our proposed eigenvectors of Γ_h. For any $c = (c_1, \ldots, c_k)$ and α an eigenvector of A_c let

$$\delta_{\alpha,c}[x] = \alpha[\tilde{x}] \cdot \prod_{i=1}^{k} \delta_{c_i}[x^i] = \alpha[\tilde{x}] \cdot (\otimes \delta_{c_i})[x].$$

Item 2: We claim that $\delta_{\alpha,c}$ is an eigenvector of Γ_h with eigenvalue $\mu_{\alpha,c}$. This can be verified as follows: for any x,

$$\begin{aligned}\sum_y \Gamma_h[x,y]\delta_{\alpha,c}[y] &= \sum_y \Gamma_f[\tilde{x},\tilde{y}]\alpha[\tilde{y}] \cdot (\otimes \Gamma_{g_i}^{(\tilde{x}_i,\tilde{y}_i)})[x,y] \cdot (\otimes \delta_{c_i})[y] \\ &= \sum_b \Gamma_f[\tilde{x},b]\alpha[b] \cdot \sum_{y:\tilde{y}=b} (\otimes \Gamma_{g_i}^{(\tilde{x}_i,\tilde{y}_i)})[x,y] \cdot (\otimes \delta_{c_i})[y]\end{aligned}$$

Applying Equation (4) gives

$$\begin{aligned}\sum_y \Gamma_h[x,y]\delta_{\alpha,c}[y] &= \sum_b \Gamma_f[\tilde{x},b]\alpha[b] \cdot \prod_{i=1}^{k} \lambda_{c_i}^{\tilde{x}_i \oplus b_i}(\Gamma_{g_i}) \cdot (\otimes \delta_{c_i})[x] \\ &= (\otimes \delta_{c_i})[x] \cdot \sum_b \Gamma_f[\tilde{x},b] \cdot \prod_{i=1}^{k} \lambda_{c_i}^{\tilde{x}_i \oplus b_i}(\Gamma_{g_i})\alpha[b].\end{aligned}$$

And now applying Equation (5) gives

$$\sum_y \Gamma_h[x,y]\delta_{\alpha,c}[y] = \mu_{\alpha,c}\alpha[\tilde{x}] \cdot (\otimes \delta_{c_i})[x] = \mu_{\alpha,c}\ \delta_{\alpha,c}[x].$$

Thus $\delta_{\alpha,c}$ is an eigenvector of Γ_h with eigenvalue $\mu_{\alpha,c}$. This completes the second step of the proof.

Item 3: We now claim that the vectors $\{\delta_{\alpha,c}\}_{\alpha,c}$ span $\mathbb{C}^{2^n}$. For a fixed c, the set of eigenvectors $\{\alpha_\ell\}_{\ell=1}^{2^k}$ of A_c forms an orthogonal basis for the space of vectors of dimension 2^k, hence there is a linear combination γ of α_ℓ's such that $\sum_\ell \gamma_\ell \alpha_\ell = (1, 1, \ldots, 1)$. Then $\sum_\ell \gamma_\ell \delta_{\alpha_\ell,c} = \otimes \delta_{c_i}$. Now, since $\{\delta_{c_i}\}_{c_i=1}^{2^{n_i}}$ form an orthogonal basis for every i, linear combinations of $\delta_{\alpha,c}$ span the whole space of dimension $2^{\sum_i n_i}$, which is the dimension of Γ_h. Hence every eigenvector of Γ_h can be expressed in this form. This completes step three of the proof.

Item 4: It now remains to show that $\mu_{\alpha,c} \leq \|\Gamma_f\| \cdot \prod_i \|\Gamma_{g_i}\|$ for every α, c. To do this, fix c and consider the matrix A_c.

$$\mu_{\alpha,c} = \alpha^* A_c \alpha = \sum_{a,b} \Gamma_f[a,b] \cdot \prod_{i=1}^{k} \lambda_{c_i}^{a_i \oplus b_i}(\Gamma_{g_i}) \cdot \alpha[a]\alpha[b]. \tag{6}$$

Notice that $-\|\Gamma_{g_i}\| \leq \lambda_{c_i}(\Gamma_{g_i}) \leq \|\Gamma_{g_i}\|$. Our first claim is that we can replace $\lambda_{c_i}(\Gamma_{g_i})$ by either $\|\Gamma_{g_i}\|$ or $-\|\Gamma_{g_i}\|$ in such a way that the sum in (6) does not decrease. To see this, we can first factor out $\lambda_{c_1}(\Gamma_{g_1})$ of the above sum and look at the term it multiplies. If this term is positive, then setting $\lambda_{c_1}(\Gamma_{g_1})$ to $\|\Gamma_{g_1}\|$

will not decrease the sum; on the other hand, if the term it multiplies is negative, then replacing $\lambda_{c_1}(\Gamma_{g_1})$ by $-\|\Gamma_{g_1}\|$ will not decrease the sum. We continue this process in turn with $i = 2, \ldots, k$.

Let $d_i = 1$ if in this process we replaced $\lambda_{c_i}(\Gamma_{g_i})$ by $-\|\Gamma_{g_i}\|$ and $d_i = 0$ if $\lambda_{c_i}(\Gamma_{g_i})$ was replaced by $\|\Gamma_{g_i}\|$. Note that if $a_i = b_i$, then no replacement was made and the coefficient remains $\|\Gamma_{g_i}\|$. We thus now have

$$\mu_{\alpha,c} \leq \sum_{a,b} \Gamma_f[a,b]\alpha[a]\alpha[b] \cdot \prod_{i=1}^{k} (-1)^{d_i(a_i+b_i)} \|\Gamma_{g_i}\|, \tag{7}$$

A key fact here is that the sign of $\|\Gamma_{g_i}\|$ will be the same everywhere $a_i \neq b_i$—the signs of entries cannot be flipped at will.

We now mimic the pattern of signs in Equation (7) by defining a new unit vector α'. Let $\alpha'[a] = \alpha[a] \prod_i (-1)^{d_i \cdot a_i}$. Then

$$\begin{aligned}
\mu_{\alpha,c} &\leq \sum_{a,b} \Gamma_f[a,b]\alpha[a]\alpha[b] \cdot \prod_{i=1}^{k} (-1)^{d_i(a_i+b_i)} \|\Gamma_{g_i}\| \\
&= \prod_{i=1}^{k} \|\Gamma_{g_i}\| \sum_{a,b} \Gamma_f[a,b]\alpha'[a]\alpha'[b] \\
&\leq \|\Gamma_f\| \cdot \prod \|\Gamma_{g_i}\|,
\end{aligned}$$

which we wished to show.

Other direction: We now show that $\|\Gamma_h\| \geq \|\Gamma_f\| \cdot \prod_{i=1}^{k} \|\Gamma_{g_i}\|$. Let δ_f be a principal eigenvector of Γ_f and δ_{g_i} a principal eigenvector for Γ_{g_i} for $i = 1, \ldots, k$. We have already argued that $\delta_h = \delta_f[\tilde{x}] \cdot \prod_{i=1}^{k} \delta_{g_i}[x^i]$ is an eigenvector of Γ_h whose eigenvalue is the eigenvalue of the matrix $A_{\vec{1}}$ where

$$A_{\vec{1}}[a,b] = \Gamma_f[a,b] \cdot \prod_{i=1}^{k} \|\Gamma_{g_i}\|.$$

Factoring out $\prod_{i=1}^{k} \|\Gamma_{g_i}\|$ from $A_{\vec{1}}$ we are simply left with the matrix Γ_f, thus the largest eigenvalue of $A_{\vec{1}}$ is $\|\Gamma_f\| \cdot \prod_{i=1}^{k} \|\Gamma_{g_i}\|$. □

B.2 Composition lower bound

With Lemma 15 in hand, it is a relatively easy matter to show a lower bound on the adversary value of the composed function h.

Lemma 16 $\mathrm{ADV}_{\alpha}^{(\pm)}(h) \geq \mathrm{ADV}_{\beta}^{(\pm)}(f)$, *where* $\beta_i = \mathrm{ADV}_{\alpha^i}^{(\pm)}(g_i)$,

Proof. Due to the maximization over all matrices Γ, the spectral bound of the composite function h is at least $\mathrm{ADV}_{\alpha}^{(\pm)}(h) \geq \min_{\ell=1}^{n}(\alpha_\ell \|\Gamma_h\| / \|\Gamma_h \circ D_\ell\|)$, where Γ_h is defined as in Lemma 15. We compute $\|\Gamma_h \circ D_\ell\|$ for $\ell = 1, \ldots, n$. Let the ℓth input bit be the qth bit in the pth block. Recall that

$$\Gamma_h[x,y] = \Gamma_f[\tilde{x},\tilde{y}] \cdot \prod_{i=1}^{k} \Gamma_{g_i}^{(\tilde{x}_i,\tilde{y}_i)}[x^i, y^i].$$

We prove that

$$(\Gamma_h \circ D_\ell)[x,y] = (\Gamma_f \circ D_p)[\tilde{x},\tilde{y}] \cdot (\Gamma_{g_p} \circ D_q)^{(\tilde{x}_p,\tilde{y}_p)}[x^p,y^p] \cdot \prod_{i\neq p} \Gamma_{g_i}^{(\tilde{x}_i,\tilde{y}_i)}[x^i,y^i].$$

If $x_\ell \neq y_\ell$ and $\tilde{x}_p \neq \tilde{y}_p$ then both sides are equal because all multiplications by D_p, D_q, D_ℓ are multiplications by 1. If this is not the case—that is, if $x_\ell = y_\ell$ or $\tilde{x}_p = \tilde{y}_p$—then both sides are zero. We see this by means of two cases:

1. $x_\ell = y_\ell$: In this case the left hand side is zero due to $(\Gamma_h \circ D_\ell)[x,y] = 0$. The right hand side is also zero because
   (a) if $\tilde{x}_p = \tilde{y}_p$ then the right hand side is zero as $(\Gamma_f \circ D_p)[\tilde{x},\tilde{y}] = 0$.
   (b) else if $\tilde{x}_p \neq \tilde{y}_p$ then the right hand side is zero as $(\Gamma_{g_p} \circ D_q)[x^p,y^p] = 0$.
2. $x_\ell \neq y_\ell$, $\tilde{x}_p = \tilde{y}_p$: The left side is zero because $\Gamma_{g_p}^{(\tilde{x}_p,\tilde{y}_p)}[x^p,y^p] = \|\Gamma_{g_p}\| I[x^p,y^p] = 0$ since $x^p \neq y^p$. The right side is also zero due to $(\Gamma_f \circ D_p)[\tilde{x},\tilde{y}] = 0$.

Since $\Gamma_h \circ D_\ell$ has the same structure as Γ_h, by Lemma 15, $\|\Gamma_h \circ D_\ell\| = \|\Gamma_f \circ D_p\| \cdot \|\Gamma_{g_p} \circ D_q\| \cdot \prod_{i\neq p} \|\Gamma_{g_i}\|$. By dividing the two spectral norms,

$$\frac{\|\Gamma_h\|}{\|\Gamma_h \circ D_\ell\|} = \frac{\|\Gamma_f\|}{\|\Gamma_f \circ D_p\|} \cdot \frac{\|\Gamma_{g_p}\|}{\|\Gamma_{g_p} \circ D_q\|}. \tag{8}$$

Since the spectral adversary maximizes over all adversary matrices, we conclude that

$$\begin{aligned}
\mathrm{ADV}_\alpha^{(\pm)}(h) &\geq \min_{\ell=1}^{n} \frac{\|\Gamma_h\|}{\|\Gamma_h \circ D_\ell\|} \cdot \alpha_\ell \\
&= \min_{i=1}^{k} \frac{\|\Gamma_f\|}{\|\Gamma_f \circ D_i\|} \cdot \min_{j=1}^{n_i} \frac{\|\Gamma_{g_i}\|}{\|\Gamma_{g_i} \circ D_j\|} \cdot \alpha_j^i \\
&= \min_{i=1}^{k} \frac{\|\Gamma_f\|}{\|\Gamma_f \circ D_i\|} \cdot \mathrm{ADV}_{\alpha^i}^{(\pm)}(g_i) \\
&= \min_{i=1}^{k} \frac{\|\Gamma_f\|}{\|\Gamma_f \circ D_i\|} \cdot \beta_i \\
&= \mathrm{ADV}_\beta^{(\pm)}(f),
\end{aligned}$$

which we had to prove. □