A New Quantum Lower Bound Method, with Applications to Direct Product Theorems and Time-Space Tradeoffs

Andris Ambainis*
University of Waterloo
ambainis@math.uwaterloo.ca

Robert Špalek†
UC Berkeley
spalek@eecs.berkeley.edu

Ronald de Wolf‡
CWI Amsterdam
rdewolf@cwi.nl

Abstract

We give a new version of the adversary method for proving lower bounds on quantum query algorithms. The new method is based on analyzing the eigenspace structure of the problem at hand. We use it to prove a new and optimal strong direct product theorem for 2-sided error quantum algorithms computing k independent instances of a symmetric Boolean function: if the algorithm uses significantly less than k times the number of queries needed for one instance of the function, then its success probability is exponentially small in k. We also use the polynomial method to prove a direct product theorem for 1-sided error algorithms for k threshold functions with a stronger bound on the success probability. Finally, we present a quantum algorithm for evaluating solutions to systems of linear inequalities, and use our direct product theorems to show that the time-space tradeoff of this algorithm is close to optimal.

1 Introduction

1.1 A new adversary method

Most known quantum algorithms work in the black-box model of computation. Here one accesses the n-bit input via *queries*, and our measure of complexity is the number of queries made by the algorithm. In between the queries, the algorithm can make unitary transformations for free. This model includes for instance the algorithms of Deutsch and Jozsa [DJ92], Simon [Sim97], Grover [Gro96], quantum counting [BHMT02], and the recent quantum walk-based algorithms [Amb04, MSS05, MN05, BŠ06, FGG07, ACR$^+$07]. It also includes hidden-subgroup algorithms such as Shor's period-finding algorithm [Sho94] (which is the quantum core of his factoring algorithm), though there one needs the additional property that the intermediate unitaries like the quantum Fourier transform are efficiently computable.

Much work has focused on proving *lower bounds* in the black-box model. The two main methods known are the polynomial method and the adversary method. The polynomial method [NS94, FR99, BBC$^+$01] works by lower bounding the degree of a polynomial that in some way represents the desired success probability.

The adversary method was originally introduced by Ambainis [Amb02]. Many different versions have since been given [HNS02, BSS03, Amb06, LM04, Zha05], but most of them are equivalent [ŠS06]. Roughly speaking, the adversary method works as follows. Suppose we have a T-query quantum algorithm that computes some function f with high success probability. Let $|\psi_x^t\rangle$ denote the algorithm's state on input x after making the t^{th} query. Suppose x and y are two inputs with distinct function values. At the start of

*Institute for Quantum Computing and Department of Combinatorics and Optimization, University of Waterloo. Supported by NSERC, ARO, CIAR and IQC University Professorship.

†Supported by NSF Grant CCF-0524837 and ARO Grant DAAD 19-03-1-0082. Work conducted while at CWI and the University of Amsterdam, supported by the European Commission under projects RESQ (IST-2001-37559) and QAP (IST-015848).

‡Supported by a Veni grant from the Netherlands Organization for Scientific Research (NWO) and partially supported by the EU projects RESQ and QAP.

the algorithm ($t = 0$), the states $|\psi_x^0\rangle$ and $|\psi_y^0\rangle$ are the same (the input has not been queried yet), so their inner product is $\langle\psi_x^0|\psi_y^0\rangle = 1$. But at the end of the algorithm ($t = T$), the inner product $\langle\psi_x^T|\psi_y^T\rangle$ must be less than some small constant depending on the error probability, otherwise the algorithm cannot give the correct answer for both x and y. The adversary method takes a (nonnegative weighted) sum of such inner products (for x, y pairs with $f(x) \neq f(y)$) and analyzes how quickly this sum can go down after each new query. If it cannot decrease quickly in one step, then it follows that we need many steps and we obtain a lower bound on T. Recently, a new version of the adversary method has been published [HLŠ07] that goes beyond this principle of distinguishability. By taking into account that the algorithm has to perform one fixed measurement at the end of the computation to determine the answer, they have been able to extend the domain of the adversary matrices and allow arbitrary (possibly negative) weights in the sum, resulting in larger lower bounds.

The polynomial and adversary lower bound methods are incomparable. On the one hand, the adversary method proves stronger bounds than the polynomial method can give for certain iterated functions [Amb06]. It also gives tight lower bounds for constant-depth AND-OR trees [Amb02, HMW03], where we do not know how to analyze the polynomial degree. On the other hand, the polynomial method works well for analyzing zero-error or low-error quantum algorithms [BBC+01, BCWZ99] and gives optimal lower bounds for the collision problem and element distinctness [AS04]. The nonnegative adversary method fails for the latter problem, because the best bound provable with it is $O(\sqrt{C^0(f)C^1(f)})$ [ŠS06, Zha05] (it is open how the negative version of the adversary method performs for this problem). Here $C^0(f)$ and $C^1(f)$ are the certificate complexities of f on 0-inputs and 1-inputs, respectively. In the case of element distinctness one of these complexities is constant. Hence the nonnegative adversary method in its present form(s) can prove at most an $\Omega(\sqrt{N})$ bound, while the true bound is $\Theta(N^{2/3})$ [Amb04, AS04]. Similarly, the best known algorithm for detecting whether an undirected n-vertex graph contains a triangle costs $O(N^{13/20})$ queries to its $N = \binom{n}{2}$ edges [MSS05], while the best lower bound provable with the nonnegative adversary method is about $\sqrt{N}$, since 1-inputs have constant certificate complexity (you can just give the triangle). We do not know what the true bound is for triangle-finding—but if it is more than $\sqrt{N}$, then the nonnegative adversary method will not be able to prove this.

A second limitation of the adversary method is that it cannot deal well with the case where there are many different possible outputs, and a success probability much smaller than $1/2$ would still be considered good. A typical example is if there are k instead of one output bits: any success probability significantly larger than 2^{-k} could be considered nontrivial here.

In this paper we describe a new version of the adversary method that does not suffer from the second limitation, and possibly also not from the first—though we have not found an example yet where the new method breaks through the $\sqrt{C^0(f)C^1(f)}$ barrier. *Very* roughly speaking, the new method works as follows. We view the algorithm as acting on a 2-register state space $\mathcal{H}_A \otimes \mathcal{H}_I$. Here the actual algorithm's operations take place in the first register, while the second contains (a superposition of) the inputs. In particular, the query operation on $\mathcal{H}_A$ is now conditioned on the basis states in $\mathcal{H}_I$. We start the analysis with a fixed starting state (the all-0 string) in the first register and a superposition of 0-inputs and 1-inputs in the input register, and then track how this input register evolves as the computation moves along. Let ρ_t be the state of this register (tracing out the $\mathcal{H}_A$-register) after making the t^{th} query. By employing symmetries in the problem's structure, such as invariances of the function under certain permutations of its input, we can decompose the input space into orthogonal subspaces $S_0, \ldots, S_m$. We can decompose the state accordingly:

$$\rho_t = \sum_{i=0}^{m} p_{t,i}\sigma_i,$$

where σ_i is a density matrix in subspace S_i. Thus the t^{th} state can be fully described by a probability distribution $p_{t,0}, \ldots, p_{t,m}$ that describes how the input register is distributed over the various subspaces. Crucially, only some of the subspaces are "good", meaning that the algorithm will only work if most of the weight is concentrated in the good subspaces at the end of the computation. At the start of the computation, hardly any weight will be in the good subspaces. If we can show that in each query, not too much weight can move from the bad subspaces to the good subspaces, then we again get a lower bound on T.

This idea was first introduced by Ambainis in [Amb05] and used there to reprove the "strong direct product theorem" for the OR-function of [KŠW07] (we will explain in a minute what this means). In this paper we extend it and use it to prove direct product theorems for all *symmetric* functions. Very recently, this method was generalized further by one of us [Špa07] to something called the *multiplicative* adversary method.

1.2 Direct product theorems for symmetric functions

Consider an algorithm that simultaneously needs to compute k independent instances of a function f (denoted $f^{(k)}$). Direct product theorems deal with the optimal tradeoff between the resources and success probability of such algorithms. These resources could for example be time, space, ink, queries, communication, etc. Suppose we need t units of some resource to compute a single instance $f(x)$ with bounded error probability. A typical (strong) direct product theorem (DPT) has the following form:[1]

> Every algorithm with $T \leq \alpha k t$ resources for computing $f^{(k)}$ has success probability $\sigma \leq 2^{-\Omega(k)}$ (where $\alpha > 0$ is some small constant).

This expresses our intuition that essentially the best way to compute $f^{(k)}$ on k independent instances is to run separate t-resource algorithms for each of the instances. Since each of those will have success probability less than 1, we expect that the probability of simultaneously getting all k instances right goes down exponentially with k. DPTs can be stated for classical algorithms or quantum algorithms, and σ could measure worst-case success probability or average-case success probability under some input distribution. DPTs are generally hard to prove, and Shaltiel [Sha01] even gives general examples where they are just not true (with σ average success probability), the above intuition notwithstanding. Klauck, Špalek, and de Wolf [KŠW07] recently examined the case where the resource is query complexity and $f = \mathrm{OR}$, and proved an optimal DPT both for classical algorithms and for quantum algorithms (with σ worst-case success probability). This strengthened a slightly earlier result of Aaronson [Aar05], who proved that the success probability goes down exponentially with k if the number of queries is bounded by $\alpha\sqrt{kn}$ rather than the $\alpha k\sqrt{n}$ of [KŠW07].

Here we generalize their results to the case where f can be any symmetric function, i.e. a function depending only on the Hamming weight $|x|$ of its input x. In the case of classical algorithms the situation is quite simple. Every n-bit symmetric function f has classical bounded-error query complexity $R_2(f) = \Theta(n)$ and block sensitivity $bs(f) = \Theta(n)$, hence an optimal classical DPT follows immediately from [KŠW07, Theorem 3]. Classically, all symmetric functions essentially "cost the same" in terms of query complexity. This is different in the quantum world. For instance, the OR function has bounded-error quantum query complexity $Q_2(\mathrm{OR}) = \Theta(\sqrt{n})$ [Gro96, BHMT02], while Parity needs $n/2$ quantum queries [BBC+01, FGGS98]. If f is a t-threshold function ($f(x) = 1$ iff $|x| \geq t$, with $t \leq n/2$), then $Q_2(f) = \Theta(\sqrt{tn})$ [BBC+01].

Our main result is an essentially optimal quantum DPT for all symmetric functions:

> There is a constant $\alpha > 0$ such that for every symmetric f and every positive integer k: Every 2-sided error quantum algorithm with $T \leq \alpha k Q_2(f)$ queries for computing $f^{(k)}$ has success probability $\sigma \leq 2^{-\Omega(k)}$.

Our new direct product theorem generalizes the polynomial-based results of [KŠW07] (which strengthened the polynomial-based [Aar05]), but our current proof uses the above-mentioned version of the adversary method instead of polynomials.

We have not been able to prove this result using the polynomial method. We can, however, use the polynomial method to prove an incomparable DPT. This result is worse than our main result in applying only to *1-sided error* quantum algorithms[2] for *threshold* functions; but it's better in giving a much stronger upper bound on the success probability:

[1] A *strong* direct product theorem has resource bound $T \approx kt$, while a *weak* direct product theorem has resource bound $T \approx t$. Since this paper only deals with the strong variety, we will omit the word "strong" and just speak of a direct product theorem (DPT) when we mean a strong one.

[2] The error is 1-sided if 1-bits in the k-bit output vector are always correct.

There is a constant $\alpha > 0$ such that for every t-threshold function f and every positive integer k: Every 1-sided error quantum algorithm with $T \leq \alpha k Q_2(f)$ queries for computing $f^{(k)}$ has success probability $\sigma \leq 2^{-\Omega(kt)}$.

A similar theorem can be proven for the k-fold t-search problem, where in each of k inputs of n bits, we want to find at least t ones. The different error bounds $2^{-\Omega(kt)}$ and $2^{-\Omega(k)}$ for 1-sided and 2-sided error algorithms intuitively say that imposing the 1-sided error constraint makes deciding each of the k threshold problems as hard as actually *finding* t ones in each of the k inputs.

1.3 Time-Space tradeoffs for evaluating solutions to systems of linear inequalities

As an application we obtain near-optimal time-space tradeoffs for evaluating solutions to systems of linear equalities. Such tradeoffs between the two main computational resources are well known classically for problems like sorting, element distinctness, hashing, etc. In the quantum world, essentially optimal time-space tradeoffs were recently obtained for sorting and for Boolean matrix multiplication [KŠW07], but little else is known.

Let A be a fixed $N \times N$ matrix of nonnegative integers. Our inputs are column vectors $x = (x_1, \ldots, x_N)$ and $b = (b_1, \ldots, b_N)$ of nonnegative integers. We are interested in the system

$$Ax \geq b$$

of N linear inequalities, and want to find out which of these inequalities hold[3] (we could also mix $\geq$, $=$, and $\leq$, but omit that for ease of notation). Note that the output is an N-bit vector. We want to analyze the tradeoff between the time T and space S needed to solve this problem. Lower bounds on T will be in terms of query complexity. For simplicity we omit polylogarithmic factors in the following discussion.

In the classical world, the optimal tradeoff is $TS = N^2$, independent of the values in b. This follows from [KŠW07, Section 7]. The upper bounds are for deterministic algorithms and the lower bounds are for 2-sided error algorithms. In the quantum world the situation is more complex. Let us put an upper bound $\max\{b_i\} \leq t$. We show here that we have two different regimes for 2-sided error quantum algorithms:

- Quantum regime. If $S \leq N/t$ then the optimal tradeoff is $T^2 S = tN^3$ (better than classical).
- Classical regime. If $S > N/t$ then the optimal tradeoff is $TS = N^2$ (same as classical).

Our lower bounds hold even for the constrained situation where b is fixed to the all-t vector, A and x are Boolean, and A is sparse in having only $O(N/S)$ non-zero entries in each row.

Since our DPT for 1-sided error algorithms is stronger by an extra factor of t in the exponent, we obtain a stronger lower bound for 1-sided error algorithms:

- If $t \leq S \leq N/t^2$ then the optimal tradeoff for 1-sided error algorithms is $T^2 S \geq t^2 N^3$.
- If $S > N/t^2$ then the optimal tradeoff for 1-sided error algorithms is $TS = N^2$.

We do not know whether the lower bound in the first case is optimal (probably it is not), but note that it is stronger than the optimal bounds that we have for 2-sided error algorithms. This is the first separation of 2-sided and 1-sided error algorithms in the context of quantum time-space tradeoffs.[4]

Remarks:

1. Klauck et al. [KŠW07] gave direct product theorems not only for quantum query complexity, but also for 2-party quantum communication complexity, and derived some communication-space tradeoffs in analogy to the time-space tradeoffs. This was made possible by a translation of communication protocols

[3] Note that if A and x are Boolean and $b = (t, \ldots, t)$, this gives N overlapping t-threshold functions.

[4] Strictly speaking, there's a quadratic gap for OR [Gro96], but space $\log n$ suffices for the fastest 1-sided and 2-sided error algorithms for OR, so there's no real tradeoff in that case.

to polynomials due to Razborov [Raz03], and the fact that the DPTs of [KŠW07] were polynomial-based. Some of the results in this paper can similarly be ported to a communication setting, though only the ones that use the polynomial method.

2. The time-space tradeoffs for 2-sided error algorithms for $Ax \geq b$ similarly hold for a system of N equalities, $Ax = b$. The upper bound clearly carries over, while the lower holds for equalities as well, because our DPT holds even under the promise that the input has weight t or $t-1$. In contrast, the stronger 1-sided error time-space tradeoff does not automatically carry over to systems of equalities, because we do not know how to prove the DPT with bound $2^{-\Omega(kt)}$ under this promise.

2 Preliminaries

2.1 Quantum query complexity

We assume familiarity with quantum computing [NC00] and sketch the model of quantum query complexity. We refer to [BW02] for more details, also on the close relation between query complexity and degrees of multivariate polynomials. 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, $\mathcal{H}_I$, which holds the input $x \in \{0,1\}^n$.
- The query register, $\mathcal{H}_Q$, which holds an integer $0 \leq i \leq n$.
- The working memory, $\mathcal{H}_W$, which holds an arbitrary value w.

The query register and working memory together form the accessible memory, denoted $\mathcal{H}_A$. Thus the state of the algorithm is described by a vector

$$|\psi\rangle = \sum_{x,i,w} \alpha_{x,i,w} |x, i, w\rangle$$

where $\sum_{x,i,w} |\alpha_{x,i,w}|^2 = 1$.

The accessible memory of a quantum query algorithm $\mathcal{A}$ 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.

There are two common ways to generalize the notion of a query to the quantum setting, where it must be a unitary operation. We will use the model where the oracle answer is given in the phase. This model is a unitary operator O that is defined by its action on basis states $|x\rangle|i\rangle|w\rangle$ as

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

For every $x = x_1 \ldots x_n$, we additionally define $x_0 = 0$. Hence querying $i = 0$ is the identity operation or "null query". This is needed to make the above "phase"-query equivalent to the alternative model of a query, which is as a map

$$O'|x\rangle|i\rangle|b\rangle|w\rangle = |x\rangle|i\rangle|b \oplus x_i\rangle|w\rangle,$$

where $b \in \{0,1\}$.

An accessible memory operator is an arbitrary unitary operation U on the accessible memory $\mathcal{H}_A$. 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 $\mathcal{H}_I$. Thus the state of the algorithm on input x after t queries can be written as

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

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 $\mathcal{A}$ 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. More general POVM measurement schemes can be considered, but these are essentially equivalent in power—see the discussion in [BSS03]. 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. $Q_2(f)$ is the query complexity for the standard value $\epsilon = \frac{1}{3}$. The subscript '2' here refers to the 2-sided nature of the errors, which can occur on 1-inputs as well as on 0-inputs.

2.2 Some quantum algorithms

We mention some well known quantum algorithms that we will use as subroutines.

- **Quantum search.** Grover's search algorithm [Gro96, BBHT98] can find an index of a 1-bit in an n-bit input in expected number of $O(\sqrt{n/(|x|+1)})$ queries, where $|x|$ is the Hamming weight (number of ones) in the input. If $|x|$ is known, the algorithm can be made to find the index in exactly $O(\sqrt{n/(|x|+1)})$ queries, instead of the expected number [BHMT02]. By repeated Grover search, we can find t ones in an n-bit input with $|x| \geq t$, using $\sum_{i=|x|-t+1}^{|x|} O(\sqrt{n/(i+1)}) = O(\sqrt{tn})$ queries.

- **Quantum counting** [BHMT02, Theorem 13]. For every integer $M \geq 1$, there is a quantum algorithm that uses M queries to n-bit x to compute an estimate w of $|x|$ such that with probability at least $8/\pi^2$

$$|w - |x|| \leq 2\pi \frac{\sqrt{|x|(n-|x|)}}{M} + \pi^2 \frac{n}{M^2}.$$

For investigating time-space tradeoffs we use a variant of the circuit model. We fix a universal set of elementary gates (for instance CNOT and all 1-qubit unitaries and measurements), and consider larger operations to be built up from these elementary gates. A circuit accesses its input via an oracle like a query algorithm. The oracle is also considered an elementary gate. *Time* corresponds to the number of elementary gates in the circuit. We often, however, only consider the number of queries to the input, which is obviously a lower bound on time. A circuit uses *space* S if it works with S bits/qubits only. We require that the outputs are made at predefined gates in the circuit, by writing their value to some extra bits/qubits that may not be used later on and that are not part of the workspace.

3 Direct Product Theorem for Symmetric Functions (2-sided)

The main result of this paper is the following theorem. In this section we first give an outline of the proof. Most of the proofs of technical claims are deferred to later subsections.

Theorem 1 *There is a constant $\alpha > 0$ such that for every symmetric f and every positive integer k: Every 2-sided error quantum algorithm with $T \leq \alpha k Q_2(f)$ queries for computing $f^{(k)}$ has success probability $\sigma \leq 2^{-\Omega(k)}$.*

Implicit threshold. Let us first say something about $Q_2(f)$ for a symmetric function $f : \{0,1\}^n \to \{0,1\}$. Let t denote the smallest non-negative integer such that f is constant on the interval $|x| \in [t, n-t]$. We call this value t the *implicit threshold* of f. For instance, functions like OR and AND have $t = 1$, while parity and majority have $t = \frac{n}{2}$. If f is the t-threshold function with $t \leq \frac{n}{2}$, then the implicit threshold is just the threshold t. The implicit threshold is related to the parameter $\Gamma(f)$ introduced by Paturi [Pat92] via $t = \frac{n}{2} - \frac{\Gamma(f)}{2} \pm 1$. It characterizes the bounded-error quantum query complexity of f:

$Q_2(f) = \Theta(\sqrt{tn})$ [BBC+01]. Hence our resource bound in the above theorem will be $\alpha k\sqrt{tn}$ for some small constant $\alpha > 0$.

We actually prove a stronger statement, applying to any Boolean function f (total or partial) for which $f(x) = 0$ if $|x| = t-1$ and $f(x) = 1$ if $|x| = t$.

Input register. Let $\mathcal{A}$ be an algorithm that computes k instances of this weight-$(t-1)$ versus weight-t problem. Let $\mathcal{H}_A$ be the accessible memory of $\mathcal{A}$. Let $\mathcal{H}_I$ be an $(\binom{n}{t-1} + \binom{n}{t})^k$-dimensional Hilbert space whose basis states correspond to inputs $(x^1, \ldots, x^k)$ with Hamming weights $|x^1| \in \{t-1, t\}, \ldots, |x^k| \in \{t-1, t\}$. The algorithm $\mathcal{A}$ is thus a sequence of transformations on a Hilbert space $\mathcal{H} = \mathcal{H}_I \otimes \mathcal{H}_A$, as described in Section 2.1. The starting state of the algorithm is

$$|\varphi_0\rangle = |\psi_{\text{start}}\rangle_A \otimes |\psi_0\rangle_I$$

where $|\psi_{\text{start}}\rangle$ is the fixed starting state of $\mathcal{A}$ as an algorithm acting on $\mathcal{H}_A$ (the all-0 state). The state $|\psi_0\rangle = |\psi_{\text{one}}\rangle^{\otimes k}$ in the input register is a tensor product of k copies of the state $|\psi_{\text{one}}\rangle$ in which half of the weight is on $|x\rangle$ with $|x| = t$, the other half is on $|x\rangle$ with $|x| = t-1$, and any two states $|x\rangle$ with the same $|x|$ have equal amplitudes:

$$|\psi_{\text{one}}\rangle = \frac{1}{\sqrt{2\binom{n}{t}}} \sum_{x:|x|=t} |x\rangle + \frac{1}{\sqrt{2\binom{n}{t-1}}} \sum_{x:|x|=t-1} |x\rangle \ .$$

Let $|\varphi_d\rangle \in \mathcal{H}$ be the state of the algorithm $\mathcal{A}$ after the d^{th} query. Let ρ_d be the mixed state in $\mathcal{H}_I$ obtained from $|\varphi_d\rangle$ by tracing out the $\mathcal{H}_A$ register.

Subspaces of the input register. We define two decompositions of $\mathcal{H}_I$ into a direct sum of subspaces. We have $\mathcal{H}_I = (\mathcal{H}_{\text{one}})^{\otimes k}$ where $\mathcal{H}_{\text{one}}$ is the input Hilbert space for one instance, with basis states $|x\rangle$, $x \in \{0,1\}^n, |x| \in \{t-1, t\}$. Let

$$|\psi^0_{i_1,\ldots,i_j}\rangle = \frac{1}{\sqrt{\binom{n-j}{t-1-j}}} \sum_{\substack{x_1,\ldots,x_n: \\ x_1+\cdots+x_n=t-1, \\ x_{i_1}=\cdots=x_{i_j}=1}} |x_1 \ldots x_n\rangle$$

and let $|\psi^1_{i_1,\ldots,i_j}\rangle$ be a similar state with $x_1 + \cdots + x_n = t$ instead of $x_1 + \cdots + x_n = t-1$. Let $T_{j,0}$ (resp. $T_{j,1}$) be the space spanned by all states $|\psi^0_{i_1,\ldots,i_j}\rangle$ (resp. $|\psi^1_{i_1,\ldots,i_j}\rangle$) and let $S_{j,a} = T_{j,a} \cap T^{\perp}_{j-1,a}$. For a subspace S, we use Π_S to denote the projector onto S. Let $|\tilde{\psi}^a_{i_1,\ldots,i_j}\rangle = \Pi_{T^{\perp}_{j-1,a}} |\psi^a_{i_1,\ldots,i_j}\rangle$. For $j < t$, let $S_{j,+}$ be the subspace spanned by the states

$$\frac{|\tilde{\psi}^0_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^0_{i_1,\ldots,i_j}\|} + \frac{|\tilde{\psi}^1_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^1_{i_1,\ldots,i_j}\|}$$

and $S_{j,-}$ be the subspace spanned by the states

$$\frac{|\tilde{\psi}^0_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^0_{i_1,\ldots,i_j}\|} - \frac{|\tilde{\psi}^1_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^1_{i_1,\ldots,i_j}\|} \ .$$

For $j = t$, we define $S_{t,-} = S_{t,1}$ and there is no subspace $S_{t,+}$. Thus $\mathcal{H}_{\text{one}} = \bigoplus_{j=0}^{t-1} (S_{j,+} \oplus S_{j,-}) \oplus S_{t,-}$. Let us try to give some intuition. In the spaces $S_{j,+}$ and $S_{j,-}$, we may be said to "know" j positions of ones. In the *good* subspaces $S_{j,-}$ we have distinguished the zero-inputs from one-inputs by the relative phase, while in the *bad* subspaces $S_{j,+}$ we have not distinguished them. Accordingly, the algorithm is doing well on this one instance if most of the state sits in the good subspaces $S_{j,-}$.

First decomposition. For the space $\mathcal{H}_I$ (representing k independent inputs) and $r_1, \ldots, r_k \in \{+, -\}$, we define

$$S_{j_1,\ldots,j_k,r_1,\ldots,r_k} = S_{j_1,r_1} \otimes S_{j_2,r_2} \otimes \cdots \otimes S_{j_k,r_k} \ .$$

Let $\mathcal{S}_{m-}$ be the direct sum of all $S_{j_1,\ldots,j_k,r_1,\ldots,r_k}$ such that exactly m of the signs $r_1, \ldots, r_k$ are equal to $-$. Then $\mathcal{H}_I = \bigoplus_m \mathcal{S}_{m-}$. This is the first decomposition.

The above intuition for one instance carries over to k instances: the more minuses the better for the algorithm. Conversely, if most of the input register sits in S_{m-} for low m, then its success probability will be small. More precisely, in Section 3.1 we prove the following lemma.

Lemma 2 (Measurement in bad subspaces) *Let ρ be the reduced density matrix of $\mathcal{H}_I$. If the support of ρ is contained in $\mathcal{S}_{0-} \oplus \mathcal{S}_{1-} \oplus \cdots \oplus \mathcal{S}_{m-}$, then the probability that measuring $\mathcal{H}_A$ gives the correct answer is at most $\dfrac{\sum_{m'=0}^{m} \binom{k}{m'}}{2^k}$.*

Note that this probability is exponentially small in k for, say, $m = \frac{k}{3}$. The following consequence of this lemma is proven in Section 3.2.

Corollary 3 (Total success probability) *Let ρ be the reduced density matrix of $\mathcal{H}_I$. The probability that measuring $\mathcal{H}_A$ gives the correct answer is at most*

$$\frac{\sum_{m'=0}^{m} \binom{k}{m'}}{2^k} + 4\sqrt{\operatorname{Tr} \Pi_{(\mathcal{S}_{0-} \oplus \mathcal{S}_{1-} \oplus \cdots \oplus \mathcal{S}_{m-})^\perp} \rho} \ .$$

Second decomposition. The first decomposition, described above, bounds the probability of success for the measurement on the final state of the algorithm. Our second decomposition will bound the probability of success of an algorithm, if the algorithm can still perform more queries before the final measurement.

To define the second decomposition into subspaces, we express $\mathcal{H}_{\text{one}} = \bigoplus_{j=0}^{t/2} R_j$ with $R_j = S_{j,+}$ for $j < t/2$ and

$$R_{t/2} = \bigoplus_{j \geq t/2} S_{j,+} \oplus \bigoplus_{j \geq 0} S_{j,-} \ .$$

Intuitively, all subspaces except for $R_{t/2}$ are bad for the algorithm, since they equal the bad subspaces $S_{j,+}$. (Unlike in the first decomposition, we have included $S_{j,+}$ for $j \geq t/2$ in the good subspace $R_{t/2}$. The reason for this distinction is that, when j is sufficiently large, it is possible to move from $S_{j,+}$ to the good subspaces $S_{j,-}$ with relatively few queries.)

Let $\mathcal{R}_\ell$ be the direct sum of all $R_{j_1} \otimes \cdots \otimes R_{j_k}$ satisfying $j_1 + \cdots + j_k = \ell$. Then $\mathcal{H}_I = \bigoplus_{\ell=0}^{tk/2} \mathcal{R}_\ell$. This is the second decomposition.

Intuitively, the algorithm can only have good success probability if for most of the k instances, most of the input register sits in $R_{t/2}$. Aggregated over all k instances, this means that the algorithm will only work well if most of the k-input register sits in $\mathcal{R}_\ell$ for ℓ large, meaning fairly close to $kt/2$. Our goal below is to show that this cannot happen if the number of queries is small.

Let $\mathcal{R}'_j = \bigoplus_{\ell=j}^{tk/2} \mathcal{R}_\ell$. Note that $\mathcal{S}_{m-} \subseteq \mathcal{R}'_{tm/2}$ for every m: $\mathcal{S}_{m-}$ is the direct sum of subspaces $S = S_{j_1,r_1} \otimes \cdots \otimes S_{j_k,r_k}$ having m minuses among $r_1, \ldots, r_k$; each such minus-subspace sits in the corresponding $R_{t/2}$ and hence $S \subseteq \mathcal{R}'_{tm/2}$. This implies

$$(\mathcal{S}_{0-} \oplus \mathcal{S}_{1-} \oplus \cdots \oplus \mathcal{S}_{(m-1)-})^\perp \subseteq \mathcal{R}'_{tm/2} \ .$$

Accordingly, if we prove an upper bound on $\operatorname{Tr} \Pi_{\mathcal{R}'_{tm/2}} \rho_T$, where T is the total number of queries, this bound together with Corollary 3 implies an upper bound on the success probability of $\mathcal{A}$.

$\vert\psi^a_{i_1,\dots,i_j}\rangle$	uniform superposition of states with $\vert x\vert = t-1+a$ and with j fixed bits set to 1
$T_{j,a}$	spanned by $\vert\psi^a_{i_1,\dots,i_j}\rangle$ for all j-tuples i
$S_{j,a} = T_{j,a} \cap T^{\perp}_{j-1,a}$	we remove the subspace $T_{j-1,a}$ from $T_{j,a}$
$\vert\tilde{\psi}^a_{i_1,\dots,i_j}\rangle$	projection of $\vert\psi^a_{i_1,\dots,i_j}\rangle$ onto $S_{j,a}$
$S_{j,\pm}$	spanned by $\frac{\vert\tilde{\psi}^0\rangle}{\Vert\tilde{\psi}^0\Vert} \pm \frac{\vert\tilde{\psi}^1\rangle}{\Vert\tilde{\psi}^1\Vert}$
$R_j = S_{j,+}$	for $j < \frac{t}{2}$ … bad subspaces
$R_{t/2}$	direct sum of $S_{j,+}$ for $j \geq t/2$, and all $S_{j,-}$ … good subspaces
$\mathcal{S}_{m-} = \bigoplus_{\substack{\vert r\vert=m \\ j_1,\dots,j_k}} \bigotimes_{i=1}^{k} S_{j_i,r_i}$	where $\vert r\vert$ is the number of minuses in $r = r_1,\dots,r_k$
$\mathcal{R}_\ell = \bigoplus_{\vert j\vert_1=\ell} \bigotimes_{i=1}^{k} R_{j_i}$	where $\vert j\vert_1$ is the sum of all entries in $j = j_1,\dots,j_k$
$\mathcal{R}'_j = \bigoplus_{\ell\geq j} \mathcal{R}_\ell$	
$\vert\psi^{a,b}_{i_1,\dots,i_j}\rangle$	uniform superposition of states with $\vert x\vert = t-1+a$, with j fixed bits set to 1, and $x_1 = b$
$T_{j,a,b}$	spanned by $\vert\psi^{a,b}_{i_1,\dots,i_j}\rangle$ for all j-tuples i
$S_{j,a,b} = T_{j,a,b} \cap T^{\perp}_{j-1,a,b}$	we remove the subspace $T_{j-1,a,b}$ from $T_{j,a,b}$
$\vert\tilde{\psi}^{a,b}_{i_1,\dots,i_j}\rangle$	projection of $\vert\psi^{a,b}_{i_1,\dots,i_j}\rangle$ into $S_{j,a,b}$
$S^{\alpha,\beta}_{j,a}$	spanned by $\alpha\frac{\vert\tilde{\psi}^{a,0}\rangle}{\Vert\tilde{\psi}^{a,0}\Vert} + \beta\frac{\vert\tilde{\psi}^{a,1}\rangle}{\Vert\tilde{\psi}^{a,1}\Vert}$

Table 1: States and subspaces used in the proof

Potential function. To bound $\operatorname{Tr}\Pi_{\mathcal{R}'_{tm/2}}\rho_T$, we consider the following *potential function*

$$P(\rho) = \sum_{\ell=0}^{tk/2} q^\ell \operatorname{Tr}\Pi_{\mathcal{R}_\ell}\rho \ ,$$

where $q = 1 + \frac{1}{t}$. Then for every d we have

$$\operatorname{Tr}\Pi_{\mathcal{R}'_{tm/2}}\rho_d = \sum_{\ell=tm/2}^{tk/2} \operatorname{Tr}\Pi_{\mathcal{R}_\ell}\rho_d \leq q^{-tm/2} \sum_{\ell=tm/2}^{tk/2} q^\ell \operatorname{Tr}\Pi_{\mathcal{R}_\ell}\rho_d \leq P(\rho_d) q^{-tm/2} = P(\rho_d) e^{-(1+o(1))m/2} \ . \quad (1)$$

We have $P(\rho_0) = 1$, because the initial state $|\psi_0\rangle$ is a tensor product of the states $|\psi_{\text{one}}\rangle$ on each copy of $\mathcal{H}_{\text{one}}$ and $|\psi_{\text{one}}\rangle$ belongs to $S_{0,+}$, hence $|\psi_0\rangle$ belongs to $\mathcal{R}_0$. In Section 3.5 we prove

Lemma 4 (Bounding the growth of the potential) *There is a constant C such that*

$$P(\rho_{j+1}) \leq \left(1 + \frac{C}{\sqrt{tn}}(q^{t/2}-1) + \frac{C\sqrt{t}}{\sqrt{n}}(q-1)\right) P(\rho_j) \ .$$

Since $q = 1 + \frac{1}{t}$, Lemma 4 means that $P(\rho_{j+1}) \leq (1 + \frac{C\sqrt{e}}{\sqrt{tn}})P(\rho_j)$ and $P(\rho_j) \leq (1 + \frac{C\sqrt{e}}{\sqrt{tn}})^j \leq e^{\frac{2Cj}{\sqrt{tn}}}$. By equation (1), for the final state after T queries we have

$$\operatorname{Tr}\Pi_{\mathcal{R}'_{tm/2}}\rho_T \leq e^{\frac{2CT}{\sqrt{tn}} - (1+o(1))\frac{m}{2}} \ .$$

We take $m = \frac{k}{3}$. Then if $T \leq \frac{1}{8C} m\sqrt{tn}$, this expression is exponentially small in k. Together with Corollary 3, this implies Theorem 1 for $\alpha = \frac{1}{24C}$.

3.1 Measurement in bad subspaces

In this section we prove Lemma 2.

The measurement of $\mathcal{H}_A$ decomposes the state in the $\mathcal{H}_I$ register as follows:

$$\rho = \sum_{a_1,\ldots,a_k \in \{0,1\}} p_{a_1,\ldots,a_k} \sigma_{a_1,\ldots,a_k} \ ,$$

with $p_{a_1,\ldots,a_k}$ being the probability of the measurement giving $(a_1, \ldots, a_k)$ (where $a_j = 1$ means the algorithm outputs—not necessarily correctly—that $|x^j| = t$, and $a_j = 0$ means $|x^j| = t-1$) and $\sigma_{a_1,\ldots,a_k}$ being the density matrix of $\mathcal{H}_I$, conditional on this outcome of the measurement. Since the support of ρ is contained in $\mathcal{S}_{0-} \oplus \cdots \oplus \mathcal{S}_{m-}$, the support of the states $\sigma_{a_1,\ldots,a_k}$ is also contained in $\mathcal{S}_{0-} \oplus \cdots \oplus \mathcal{S}_{m-}$. The probability that the answer $(a_1, \ldots, a_k)$ is correct is equal to

$$\operatorname{Tr} \Pi_{\otimes_{j=1}^k \oplus_{l=0}^{t-1+a_j} S_{l,a_j}} \sigma_{a_1,\ldots,a_k} \ . \tag{2}$$

We show that, for any $\sigma_{a_1,\ldots,a_k}$ with support contained in $\mathcal{S}_{0-} \oplus \cdots \oplus \mathcal{S}_{m-}$, (2) is at most $\frac{\sum_{m'=0}^{m} \binom{k}{m'}}{2^k}$.

For brevity, we now write σ instead of $\sigma_{a_1,\ldots,a_k}$. A measurement with respect to $\otimes_{j=1}^k \oplus_l S_{l,a_j}$ and its orthogonal complement commutes with a measurement with respect to the collection of subspaces

$$\otimes_{j=1}^k (S_{l_j,0} \oplus S_{l_j,1}) \ ,$$

where $l_1, \ldots, l_k$ range over $\{0, \ldots, t\}$. Therefore

$$\operatorname{Tr} \Pi_{\otimes_{j=1}^k \oplus_l S_{l,a_j}} \sigma = \sum_{l_1,\ldots,l_k} \operatorname{Tr} \Pi_{\otimes_{j=1}^k \oplus_l S_{l,a_j}} \Pi_{\otimes_{j=1}^k (S_{l_j,0} \oplus S_{l_j,1})} \sigma \ .$$

Hence to bound (2) it suffices to prove the same bound with

$$\sigma' = \Pi_{\otimes_{j=1}^k (S_{l_j,0} \oplus S_{l_j,1})} \sigma$$

instead of σ. Since

$$\left(\otimes_{j=1}^k (S_{l_j,0} \oplus S_{l_j,1})\right) \cap \left(\otimes_{j=1}^k (\oplus_l S_{l,a_j})\right) = \otimes_{j=1}^k S_{l_j,a_j} \ ,$$

we have

$$\operatorname{Tr} \Pi_{\otimes_{j=1}^k (\oplus_l S_{l,a_j})} \sigma' = \operatorname{Tr} \Pi_{\otimes_{j=1}^k S_{l_j,a_j}} \sigma' \ . \tag{3}$$

We prove this bound for the case when σ' is a pure state: $\sigma' = |\psi\rangle\langle\psi|$. Then equation (3) is equal to

$$\|\Pi_{\otimes_{j=1}^k S_{l_j,a_j}} \psi\|^2 \ . \tag{4}$$

The bound for mixed states σ' follows by decomposing σ' as a mixture of pure states $|\psi\rangle$, bounding (4) for each of those states and then summing up the bounds.

We have

$$(\mathcal{S}_{0-} \oplus \cdots \oplus \mathcal{S}_{m-}) \cap \bigotimes_{j=1}^k (S_{l_j,0} \oplus S_{l_j,1}) = \bigoplus_{\substack{r_1,\ldots,r_k \in \{+,-\}, \\ |\{i : r_i = -\}| \le m}} \bigotimes_{j=1}^k S_{l_j,r_j} \ .$$

We express

$$|\psi\rangle = \sum_{\substack{r_1,\ldots,r_k \in \{+,-\}, \\ |\{i : r_i = -\}| \le m}} \alpha_{r_1,\ldots,r_k} |\psi_{r_1,\ldots,r_k}\rangle \ ,$$

with $|\psi_{r_1,\ldots,r_k}\rangle \in \otimes_{j=1}^k S_{l_j,r_j}$. Therefore

$$\|\Pi_{\otimes_{j=1}^k S_{l_j,a_j}}\psi\|^2 \le \left(\sum_{r_1,\ldots,r_k} |\alpha_{r_1,\ldots,r_k}| \cdot \|\Pi_{\otimes_{j=1}^k S_{l_j,a_j}}\psi_{r_1,\ldots,r_k}\|\right)^2 \le \sum_{r_1,\ldots,r_k} \|\Pi_{\otimes_{j=1}^k S_{l_j,a_j}}\psi_{r_1,\ldots,r_k}\|^2 \ , \tag{5}$$

where the second inequality uses Cauchy-Schwarz and the fact that

$$\|\psi\|^2 = \sum_{r_1,\ldots,r_k} |\alpha_{r_1,\ldots,r_k}|^2 = 1 \ .$$

Claim 5 $\|\Pi_{\otimes_{j=1}^k S_{l_j,a_j}}\psi_{r_1,\ldots,r_k}\|^2 \le \frac{1}{2^k}$.

Proof. Let $|\varphi_i^{j,0}\rangle$, $i \in [\dim S_{l_j,0}]$ form an orthonormal basis for the subspace $S_{l_j,0}$. Define a map $\mathsf{U}_j : S_{l_j,0} \to S_{l_j,1}$ by $\mathsf{U}_j|\tilde{\psi}^0_{i_1,\ldots,i_{l_j}}\rangle = |\tilde{\psi}^1_{i_1,\ldots,i_{l_j}}\rangle$. Then U_j is a multiple of a unitary transformation: $\mathsf{U}_j = c_j\mathsf{U}'_j$ for some unitary U'_j and a constant c_j. (This follows from Claim 8 below.)

Let $|\varphi_i^{j,1}\rangle = \mathsf{U}'_j|\varphi_i^{j,0}\rangle$. Since U'_j is a unitary transformation, the states $|\varphi_i^{j,1}\rangle$ form a basis for $S_{l_j,1}$. Therefore the set of states

$$\bigotimes_{j=1}^k |\varphi_{i_j}^{j,a_j}\rangle \tag{6}$$

is a basis for $\otimes_{j=1}^k S_{l_j,a_j}$. Moreover, the states

$$|\varphi_i^{j,+}\rangle = \frac{1}{\sqrt{2}}|\varphi_i^{j,0}\rangle + \frac{1}{\sqrt{2}}|\varphi_i^{j,1}\rangle, \quad |\varphi_i^{j,-}\rangle = \frac{1}{\sqrt{2}}|\varphi_i^{j,0}\rangle - \frac{1}{\sqrt{2}}|\varphi_i^{j,1}\rangle$$

are a basis for $S_{l_j,+}$ and $S_{l_j,-}$, respectively. Therefore there exist $\alpha_{i_1,\ldots,i_k}$ such that

$$|\psi_{r_1,\ldots,r_k}\rangle = \sum_{i_1,\ldots,i_k} \alpha_{i_1,\ldots,i_k} \bigotimes_{j=1}^k |\varphi_{i_j}^{j,r_j}\rangle \ . \tag{7}$$

The inner product between $\otimes_{i=1}^k|\varphi_{i'_j}^{j,a_j}\rangle$ and $\otimes_{j=1}^k|\varphi_{i_j}^{j,r_j}\rangle$ is

$$\prod_{j=1}^k \langle\varphi_{i_j}^{j,r_j}|\varphi_{i'_j}^{j,a_j}\rangle \ .$$

Note that $r_j \in \{+,-\}$ and $a_j \in \{0,1\}$. The terms in this product are $\pm\frac{1}{\sqrt{2}}$ if $i'_j = i_j$ and 0 otherwise. This means that $\otimes_{j=1}^k|\varphi_{i_j}^{j,r_j}\rangle$ has inner product $\pm\frac{1}{2^{k/2}}$ with $\otimes_{i=1}^k|\varphi_{i_j}^{j,a_j}\rangle$ and inner product 0 with all other basis states of the form (6). Therefore,

$$\Pi_{\otimes_{j=1}^k S_{l_j,a_j}} \otimes_{j=1}^k |\varphi_{i_j}^{j,r_j}\rangle = \pm\frac{1}{2^{k/2}} \otimes_{i=1}^k |\varphi_{i_j}^{j,a_j}\rangle \ .$$

Together with equation (7), this means that

$$\|\Pi_{\otimes_{j=1}^k S_{l_j,a_j}}\psi_{r_1,\ldots,r_k}\| \le \frac{1}{2^{k/2}}\|\psi_{r_1,\ldots,r_k}\| = \frac{1}{2^{k/2}} \ .$$

Squaring both sides completes the proof of the claim. □

Since there are $\binom{k}{m'}$ tuples $(r_1,\ldots,r_k)$ with $r_1,\ldots,r_k \in \{+,-\}$ and $|\{i : r_i = -\}| = m'$, Claim 5 together with equation (5) implies

$$\|\Pi_{\otimes_{j=1}^k S_{l_j,a_j}}\psi\|^2 \le \frac{\sum_{m'=0}^m \binom{k}{m'}}{2^k} \ .$$

3.2 Total success probability

In this section we prove Corollary 3.

Let $|\psi\rangle$ be a purification of ρ in $\mathcal{H}_I \otimes \mathcal{H}_A$. Let

$$|\psi\rangle = \sqrt{1-\delta}|\psi'\rangle + \sqrt{\delta}|\psi''\rangle \ ,$$

where $|\psi'\rangle$ is in the subspace $\mathcal{H}_A \otimes (\mathcal{S}_{0-} \oplus \mathcal{S}_{1-} \oplus \cdots \oplus \mathcal{S}_{m-})$ and $|\psi''\rangle$ is in the subspace $\mathcal{H}_A \otimes (\mathcal{S}_{0-} \oplus \mathcal{S}_{1-} \oplus \cdots \oplus \mathcal{S}_{m-})^{\perp}$. Then $\delta = \operatorname{Tr} \Pi_{(\mathcal{S}_{0-}\oplus\cdots\oplus\mathcal{S}_{m-})^{\perp}} \rho$.

The success probability of $\mathcal{A}$ is the probability that, if we measure both the register $\mathcal{H}_A$ containing the result of the computation and $\mathcal{H}_I$, then we get $a_1, \ldots, a_k$ and $x^1, \ldots, x^k$ such that x^j contains $t-1+a_j$ ones for every $j \in \{1, \ldots, k\}$.

Consider the probability of getting $a_1, \ldots, a_k \in \{0,1\}$ and $x^1, \ldots, x^k \in \{0,1\}^n$ with this property, when measuring $|\psi'\rangle$ (instead of $|\psi\rangle$). By Lemma 2, this probability is at most $\frac{\sum_{m'=0}^{m} \binom{k}{m'}}{2^k}$. We have

$$\|\psi - \psi'\| \leq (1-\sqrt{1-\delta})\|\psi'\| + \sqrt{\delta}\|\psi''\| = (1-\sqrt{1-\delta}) + \sqrt{\delta} \leq 2\sqrt{\delta} \ .$$

We now apply the following lemma by Bernstein and Vazirani.

Lemma 6 ([BV97]) *For any states $|\psi\rangle$ and $|\psi'\rangle$ and any measurement M, the variational distance between the probability distributions obtained by applying M to $|\psi\rangle$ and $|\psi'\rangle$ is at most $2\|\psi - \psi'\|$.*

Hence the success probability of $\mathcal{A}$ is at most

$$\frac{\sum_{m'=0}^{m} \binom{k}{m'}}{2^k} + 4\sqrt{\delta} = \frac{\sum_{m'=0}^{m} \binom{k}{m'}}{2^k} + 4\sqrt{\operatorname{Tr} \Pi_{(\mathcal{S}_{0-}\oplus\cdots\oplus\mathcal{S}_{m-})^{\perp}} \rho} \ .$$

3.3 Subspaces when asking one query

Let $|\psi_d\rangle$ be the state of $\mathcal{H}_A \otimes \mathcal{H}_I$ after d queries. Write

$$|\psi_d\rangle = \sum_{i=0}^{kn} a_i |\psi_{d,i}\rangle \ ,$$

with $|\psi_{d,i}\rangle$ being the part in which the query register contains $|i\rangle$. Let $\rho_{d,i} = \operatorname{Tr}_{\mathcal{H}_A} |\psi_{d,i}\rangle\langle\psi_{d,i}|$. Then

$$\rho_d = \sum_{i=0}^{kn} a_i^2 \rho_{d,i} \ . \tag{8}$$

Because of

$$\operatorname{Tr} \Pi_{\mathcal{R}_m} \rho_d = \sum_{i=0}^{kn} a_i^2 \operatorname{Tr} \Pi_{\mathcal{R}_m} \rho_{d,i} \ ,$$

we have $P(\rho_d) = \sum_{i=0}^{kn} a_i^2 P(\rho_{d,i})$. Let ρ'_d be the state after the $(d+1)^{\text{st}}$ query and let $\rho'_d = \sum_{i=0}^{kn} a_i^2 \rho'_{d,i}$ be a decomposition similar to equation (8). Lemma 4 follows by showing

$$P(\rho'_{d,i}) \leq \left(1 + \frac{C}{\sqrt{tn}}(q^{t/2}-1) + \frac{C\sqrt{t}}{\sqrt{n}}(q-1)\right) P(\rho_{d,i}) \tag{9}$$

for each i. For $i=0$, the query does not change the state if the query register contains $|i\rangle$. Therefore, $\rho'_{d,0} = \rho_{d,0}$ and $P(\rho'_{d,0}) = P(\rho_{d,0})$. This means that equation (9) is true for $i=0$. To prove the $i \in \{1, \ldots, kn\}$ case, it suffices to prove the $i=1$ case (because of symmetry).

Subspaces of $\rho_{d,1}$. We now define analogues of the spaces $T_{j,a}$, $S_{j,a}$, etc., but treating x_1 separately.

Let $|\psi^{a,b}_{i_1,\ldots,i_j}\rangle$ (with $a, b \in \{0,1\}$ and $i_1, \ldots, i_j \in \{2, \ldots, n\}$) be the uniform superposition over basis states $|b, x_2, \ldots, x_n\rangle$ (of $\mathcal{H}_{\text{one}}$) with $b + x_2 + \cdots + x_n = t - 1 + a$ and $x_{i_1} = \cdots = x_{i_j} = 1$. Let $T_{j,a,b}$ be the space spanned by all states $|\psi^{a,b}_{i_1,\ldots,i_j}\rangle$ and let $S_{j,a,b} = T_{j,a,b} \cap T^{\perp}_{j-1,a,b}$. Let $|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle = \Pi_{T^{\perp}_{j-1,a,b}} |\psi^{a,b}_{i_1,\ldots,i_j}\rangle$.

Let $S^{\alpha,\beta}_{j,a}$ be the subspace spanned by all states

$$\alpha \frac{|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle}{\||\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|} + \beta \frac{|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle}{\||\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} \quad . \tag{10}$$

Claim 7 *Let* $\alpha_a = \sqrt{\frac{n-(t-1+a)}{n-j}} \|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|$ *and* $\beta_a = \sqrt{\frac{(t-1+a)-j}{n-j}} \|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|$*. Then (i)* $S^{\alpha_a,\beta_a}_{j,a} \subseteq S_{j,a}$ *and (ii)* $S^{\beta_a,-\alpha_a}_{j,a} \subseteq S_{j+1,a}$.

Proof. For part (i), consider the states $|\psi^{a}_{i_1,\ldots,i_j}\rangle$ in $T_{j,a}$, for $1 \notin \{i_1, \ldots, i_j\}$. We have

$$|\psi^{a}_{i_1,\ldots,i_j}\rangle = \sqrt{\frac{n-(t-1+a)}{n-j}} |\psi^{a,0}_{i_1,\ldots,i_j}\rangle + \sqrt{\frac{(t-1+a)-j}{n-j}} |\psi^{a,1}_{i_1,\ldots,i_j}\rangle, \tag{11}$$

because among the states $|x_1 \ldots x_n\rangle$ with $|x| = t-1+a$ and $x_{i_1} = \cdots = x_{i_j} = 1$, a $\frac{n-(t-1+a)}{n-j}$ fraction have $x_1 = 0$ and the rest have $x_1 = 1$. The projections of these states to $T^{\perp}_{j-1,a,0} \cap T^{\perp}_{j-1,a,1}$ are

$$\sqrt{\frac{n-(t-1+a)}{n-j}} |\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle + \sqrt{\frac{(t-1+a)-j}{n-j}} |\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle.$$

By equation (10), these are exactly the states spanning $S^{\alpha_a,\beta_a}_{j,a}$. Furthermore, we claim that

$$T_{j-1,a} \subseteq T_{j-1,a,0} \oplus T_{j-1,a,1} \subseteq T_{j,a} \quad . \tag{12}$$

The first containment is true because $T_{j-1,a}$ is spanned by the states $|\psi^{a}_{i_1,\ldots,i_{j-1}}\rangle$. These states either belong to $T_{j-2,a,1} \subseteq T_{j-1,a,1}$ (if $1 \in \{i_1, \ldots, i_{j-1}\}$), or they are a linear combination of states $|\psi^{a,0}_{i_1,\ldots,i_{j-1}}\rangle$ and $|\psi^{a,1}_{i_1,\ldots,i_{j-1}}\rangle$ (by equation (11)), which belong to $T_{j-1,a,0}$ and $T_{j-1,a,1}$. The second containment follows because the states $|\psi^{a,1}_{i_1,\ldots,i_{j-1}}\rangle$ spanning $T_{j-1,a,1}$ are the same as the states $|\psi^{a}_{1,i_1,\ldots,i_{j-1}}\rangle$ which belong to $T_{j,a}$, and the states $|\psi^{a,0}_{i_1,\ldots,i_{j-1}}\rangle$ spanning $T_{j-1,a,0}$ can be expressed as linear combinations of $|\psi^{a}_{i_1,\ldots,i_{j-1}}\rangle$ and $|\psi^{a}_{1,i_1,\ldots,i_{j-1}}\rangle$ which both belong to $T_{j,a}$.

The first part of (12) now implies

$$S^{\alpha_a,\beta_a}_{j,a} \subseteq T^{\perp}_{j-1,a,0} \cap T^{\perp}_{j-1,a,1} \subseteq T^{\perp}_{j-1,a} \quad .$$

Also, $S^{\alpha_a,\beta_a}_{j,a} \subseteq T_{j,a}$, because $S^{\alpha_a,\beta_a}_{j,a}$ is spanned by the states

$$\Pi_{T^{\perp}_{j-1,a,0} \cap T^{\perp}_{j-1,a,1}} |\psi^{a}_{i_1,\ldots,i_j}\rangle = |\psi^{a}_{i_1,\ldots,i_j}\rangle - \Pi_{T_{j-1,a,0} \oplus T_{j-1,a,1}} |\psi^{a}_{i_1,\ldots,i_j}\rangle$$

and $|\psi^{a}_{i_1,\ldots,i_j}\rangle$ belongs to $T_{j,a}$ by the definition of $T_{j,a}$, and $\Pi_{T_{j-1,a,0} \oplus T_{j-1,a,1}} |\psi^{a}_{i_1,\ldots,i_j}\rangle$ belongs to $T_{j,a}$ because of the second part of (12). Therefore, $S^{\alpha_a,\beta_a}_{j,a} \subseteq T_{j,a} \cap T^{\perp}_{j-1,a} = S_{j,a}$.

For part (ii), we have

$$S^{\beta_a,-\alpha_a}_{j,a} \subseteq S_{j,a,0} \oplus S_{j,a,1} \subseteq T_{j,a,0} \oplus T_{j,a,1} \subseteq T_{j+1,a} \quad .$$

Here the first containment is true because $S^{\beta_a,-\alpha_a}_{j,a}$ is spanned by linear combinations of vectors $|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle$ (which belong to $S_{j,a,0}$) and vectors $|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle$ (which belong to $S_{j,a,1}$). The last containment is true because of the second part of equation (12). Now let

$$|\psi\rangle = \beta_a \frac{|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|} - \alpha_a \frac{|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} \tag{13}$$

be one of the vectors spanning $S_{j,a}^{\beta_a,-\alpha_a}$. To prove that $|\psi\rangle$ is in $S_{j+1,a} = T_{j+1,a} \cap T_{j,a}^{\perp}$, it remains to prove that $|\psi\rangle$ is orthogonal to $T_{j,a}$. This is equivalent to proving that $|\psi\rangle$ is orthogonal to each of the vectors $|\psi^a_{i'_1,\ldots,i'_j}\rangle$ spanning $T_{j,a}$. We distinguish two cases (note that $1 \notin \{i_1,\ldots,i_j\}$):

1. $1 \in \{i'_1,\ldots,i'_j\}$.

   For simplicity, assume $1 = i'_j$. Then $|\psi^a_{i'_1,\ldots,i'_j}\rangle$ is the same as $|\psi^{a,1}_{i'_1,\ldots,i'_{j-1}}\rangle$, which belongs to $T_{j-1,a,1}$. By definition, the vector $|\psi\rangle$ belongs to $T^{\perp}_{j-1,a,0} \cap T^{\perp}_{j-1,a,1}$ and is therefore orthogonal to $|\psi^{a,1}_{i'_1,\ldots,i'_{j-1}}\rangle$.

2. $1 \notin \{i'_1,\ldots,i'_j\}$.

   We will prove this case by induction on $\ell = |\{i'_1,\ldots,i'_j\} - \{i_1,\ldots,i_j\}|$.

   In the base step ($\ell = 0$), we have $\{i'_1,\ldots,i'_j\} = \{i_1,\ldots,i_j\}$. Since $|\psi\rangle$ belongs to $T^{\perp}_{j-1,a,0} \cap T^{\perp}_{j-1,a,1}$, it suffices to prove $|\psi\rangle$ is orthogonal to the projection of $|\psi^a_{i_1,\ldots,i_j}\rangle$ to $T^{\perp}_{j-1,a,0} \cap T^{\perp}_{j-1,a,1}$ which, by the discussion after equation (11), equals

   $$\alpha_a \frac{|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|} + \beta_a \frac{|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} \ . \tag{14}$$

   From equations (13) and (14), we see that the inner product of the two states is $\alpha_a\beta_a - \beta_a\alpha_a = 0$.

   For the inductive step ($\ell \geq 1$), assume $i'_j \notin \{i_1,\ldots,i_j\}$. Up to a constant multiplicative factor, we have

   $$|\psi^a_{i'_1,\ldots,i'_{j-1}}\rangle = \sum_{i' \notin \{i'_1,\ldots,i'_{j-1}\}} |\psi^a_{i'_1,\ldots,i'_{j-1},i'}\rangle \ .$$

   Because $|\psi^a_{i'_1,\ldots,i'_{j-1}}\rangle$ is in $T_{j-1,a,0} \oplus T_{j-1,a,1}$, we have

   $$\sum_{i' \notin \{i'_1,\ldots,i'_{j-1}\}} \langle\psi^a_{i'_1,\ldots,i'_{j-1},i'}|\psi\rangle = \langle\psi^a_{i'_1,\ldots,i'_{j-1}}|\psi\rangle = 0 \ . \tag{15}$$

   As proven in the previous case, $\langle\psi^a_{i'_1,\ldots,i'_{j-1},1}|\psi\rangle = 0$. Moreover, by the induction hypothesis we have $\langle\psi^a_{i'_1,\ldots,i'_{j-1},i'}|\psi\rangle = 0$ whenever $i' \in \{i_1,\ldots,i_j\}$. Therefore equation (15) reduces to

   $$\sum_{i' \notin \{i'_1,\ldots,i'_{j-1},i_1,\ldots,i_j,1\}} \langle\psi^a_{i'_1,\ldots,i'_{j-1},i'}|\psi\rangle = 0 \ . \tag{16}$$

   By symmetry, the inner products in this sum are the same for every i'. Hence they are all 0, in particular for $i' = i'_j$.

We conclude that $|\psi\rangle$ is orthogonal to the subspace $T_{j,a}$ and therefore $|\psi\rangle$ is in $S_{j+1,a} = T_{j+1,a} \cap T^{\perp}_{j,a}$. □

Claim 8 *The maps* $\mathsf{U}_{01} : S_{j,0,0} \to S_{j,0,1}$*,* $\mathsf{U}_{10} : S_{j,0,0} \to S_{j,1,0}$ *and* $\mathsf{U}_{11} : S_{j,0,0} \to S_{j,1,1}$ *defined by* $\mathsf{U}_{ab}|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle = |\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle$ *are multiples of unitary transformations:* $\mathsf{U}_{ab} = c_{ab}\mathsf{U}'_{ab}$ *for some unitary* U'_{ab} *and some constant* c_{ab}*.*

Proof. We define $\mathsf{M} : T_{j,0,0} \to T_{j,0,1}$ by

$$\mathsf{M}|0x_2\ldots x_n\rangle = \sum_{\ell: x_\ell = 1} |1x_2\ldots x_{\ell-1}0x_{\ell+1}\ldots x_n\rangle \ .$$

Note that M does not depend on j. We claim

$$\begin{aligned}\mathsf{M}|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle &= c|\tilde{\psi}^{0,1}_{i_1,\ldots,i_j}\rangle \\ \mathsf{M}^\dagger|\tilde{\psi}^{0,1}_{i_1,\ldots,i_j}\rangle &= c'|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle\end{aligned} \tag{17}$$

for some constants c and c' that may depend on n, t and j but not on $i_1, \ldots, i_j$.

To prove that, we need to prove two things. First, we claim that

$$\mathsf{M}|\psi^{0,0}_{i_1,\ldots,i_j}\rangle = c|\psi^{0,1}_{i_1,\ldots,i_j}\rangle + |\psi'\rangle \ , \tag{18}$$

where $|\psi'\rangle \in T_{j-1,0,1}$ (note that $1 \notin \{i_1, \ldots, i_j\}$). Equation (18) follows by

$$\begin{aligned}
\mathsf{M}|\psi^{0,0}_{i_1,\ldots,i_j}\rangle &= \frac{1}{\sqrt{\binom{n-j-1}{t-1-j}}} \sum_{\substack{x:|x|=t-1,x_1=0\\ x_{i_1}=\cdots=x_{i_j}=1}} \mathsf{M}|x\rangle \\
&= \frac{1}{\sqrt{\binom{n-j-1}{t-1-j}}} \sum_{\substack{x:|x|=t-1,x_1=0\\ x_{i_1}=\cdots=x_{i_j}=1}} \sum_{\ell:x_\ell=1} |1x_2\ldots x_{\ell-1}0x_{\ell+1}\ldots x_n\rangle \\
&= \frac{n-t+1}{\sqrt{\binom{n-j-1}{t-1-j}}} \sum_{\substack{y:|y|=t-1,y_1=1\\ y_{i_1}=\cdots=y_{i_j}=1}} |y\rangle + \frac{1}{\sqrt{\binom{n-j-1}{t-1-j}}} \sum_{\ell=1}^{j} \sum_{\substack{y:|y|=t-1,y_1=1,y_{i_\ell}=0\\ y_{i_1}=\cdots=y_{i_j}=1}} |y\rangle \\
&= \frac{n-t-j+1}{\sqrt{\binom{n-j-1}{t-1-j}}} \sum_{\substack{y:|y|=t-1,y_1=1\\ y_{i_1}=\cdots=y_{i_j}=1}} |y\rangle + \frac{1}{\sqrt{\binom{n-j-1}{t-1-j}}} \sum_{\ell=1}^{j} \sum_{\substack{y:|y|=t-1,y_1=1\\ y_{i_1}=\cdots=y_{i_{\ell-1}}=1\\ y_{i_{\ell+1}}=\cdots=y_{i_j}=1}} |y\rangle \\
&= (n-t-j+1)\sqrt{\tfrac{t-1-j}{n-t+1}}|\psi^{0,1}_{i_1,\ldots,i_j}\rangle + \sqrt{\tfrac{n-j}{n-t+1}} \sum_{\ell=1}^{j} |\psi^{0,1}_{i_1,\ldots,i_{\ell-1},i_{\ell+1},\ldots,i_j}\rangle \ .
\end{aligned}$$

This proves (18), with $|\psi'\rangle$ equal to the second term.

Second, for every j, $\mathsf{M}(T_{j,0,0}) \subseteq T_{j,0,1}$ and $\mathsf{M}(T^\perp_{j,0,0}) \subseteq T^\perp_{j,0,1}$. The first statement follows from equation (18), because the subspaces $T_{j,0,0}$, $T_{j,0,1}$ are spanned by the states $|\psi^{0,0}_{i_1,\ldots,i_j}\rangle$ and $|\psi^{0,1}_{i_1,\ldots,i_j}\rangle$, respectively, and $T_{j-1,0,1} \subseteq T_{j,0,1}$. To prove the second statement, let $|\psi\rangle \in T^\perp_{j,0,0}$, $|\psi\rangle = \sum_x a_x|x\rangle$. We would like to prove $\mathsf{M}|\psi\rangle \in T^\perp_{j,0,1}$. This is equivalent to $\langle\psi^{0,1}_{i_1,\ldots,i_j}|\mathsf{M}|\psi\rangle = 0$ for all $i_1, \ldots, i_j$. We have

$$\begin{aligned}
\langle\psi^{0,1}_{i_1,\ldots,i_j}|\mathsf{M}|\psi\rangle &= \frac{1}{\sqrt{\binom{n-j-1}{t-j-2}}} \sum_{\substack{y:|y|=t-1,y_1=1\\ y_{i_1}=\cdots=y_{i_j}=1}} \langle y|\mathsf{M}|\psi\rangle \\
&= \frac{1}{\sqrt{\binom{n-j-1}{t-j-2}}} \sum_{\substack{x:|x|=t-1,x_1=0\\ x_{i_1}=\cdots=x_{i_j}=1}} \sum_{\substack{\ell:x_\ell=1\\ \ell\notin\{i_1,\ldots,i_j\}}} a_x \\
&= \frac{t-1-j}{\sqrt{\binom{n-j-1}{t-j-2}}} \sum_{\substack{x:|x|=t-1,x_1=0\\ x_{i_1}=\cdots=x_{i_j}=1}} a_x = 0 \ .
\end{aligned}$$

The first equality follows by writing out $\langle\psi^{0,1}_{i_1,\ldots,i_j}|$, the second equality follows by writing out M. The third equality follows because, for every x with $|x| = t-1$ and $x_{i_1} = \cdots = x_{i_j} = 1$, there are $t-1-j$ more $\ell \in [n]$ satisfying $x_\ell = 1$. The fourth equality follows because $\sum_{\substack{x:|x|=t-1,x_1=0\\ x_{i_1}=\cdots=x_{i_j}=1}} a_x$ is a constant times $\langle\psi^{0,0}_{i_1,\ldots,i_j}|\psi\rangle$, and $\langle\psi^{0,0}_{i_1,\ldots,i_j}|\psi\rangle = 0$ because $|\psi\rangle \in T^\perp_{j,0,0}$.

To deduce equation (17), we write

$$|\psi^{0,0}_{i_1,\ldots,i_j}\rangle = |\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle + \Pi_{T_{j-1,0,0}}|\psi^{0,0}_{i_1,\ldots,i_j}\rangle \ .$$

Since $\mathsf{M}(T_{j-1,0,0}) \subseteq T_{j-1,0,1}$ and $\mathsf{M}(T^{\perp}_{j-1,0,0}) \subseteq T^{\perp}_{j-1,0,1}$,

$$\mathsf{M}|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle = \Pi_{T^{\perp}_{j-1,0,1}}\mathsf{M}|\psi^{0,0}_{i_1,\ldots,i_j}\rangle = c\Pi_{T^{\perp}_{j-1,0,1}}|\psi^{0,1}_{i_1,\ldots,i_j}\rangle = c|\tilde{\psi}^{0,1}_{i_1,\ldots,i_j}\rangle \ ,$$

with the second equality following from (18) and $|\psi'\rangle \in T_{j-1,0,1}$. This proves the first half of (17). The second half follows similarly. Therefore

$$\langle\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}|\mathsf{M}^{\dagger}\mathsf{M}|\tilde{\psi}^{0,0}_{i'_1,\ldots,i'_j}\rangle = c\cdot c'\langle\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}|\tilde{\psi}^{0,0}_{i'_1,\ldots,i'_j}\rangle \ .$$

Hence M is a multiple of a unitary transformation. By equation (17), $\mathsf{U}_{01} = \mathsf{M}/c$ and, therefore, U_{01} is also a multiple of a unitary transformation.

Next, we define M by $\mathsf{M}|0x_2\ldots x_n\rangle = |1x_2\ldots x_n\rangle$. Then M is a unitary transformation from the space spanned by $|0x_2\ldots x_n\rangle$, $x_2+\cdots+x_2 = t-1$, to the space spanned by $|1x_2\ldots x_n\rangle$, $1+x_2+\cdots+x_n = t$. We claim that $\mathsf{U}_{11} = \mathsf{M}$. To prove that, we first observe that

$$\begin{aligned}\mathsf{M}|\psi^{0,0}_{i_1,\ldots,i_j}\rangle &= \frac{1}{\sqrt{\binom{n-j-1}{t-j-1}}}\sum_{\substack{x_2,\ldots,x_n:\\ x_{i_1}=\cdots=x_{i_j}=1\\ x_2+\cdots x_n=t-1}}\mathsf{M}|0x_2\ldots x_n\rangle \\ &= \frac{1}{\sqrt{\binom{n-j-1}{t-j-1}}}\sum_{\substack{x_2,\ldots,x_n:\\ x_{i_1}=\cdots=x_{i_j}=1\\ x_2+\cdots x_n=t-1}}|1x_2\ldots x_n\rangle = |\psi^{1,1}_{i_1,\ldots,i_j}\rangle \ .\end{aligned}$$

Since $T_{j,a,b}$ is defined as the subspace spanned by all $|\psi^{a,b}_{i_1,\ldots,i_j}\rangle$, this means that $\mathsf{M}(T_{j,0,0}) = T_{j,1,1}$ and similarly $\mathsf{M}(T_{j-1,0,0}) = T_{j-1,1,1}$. Since M is unitary, this implies $\mathsf{M}(T^{\perp}_{j-1,0,0}) = T^{\perp}_{j-1,1,1}$ and

$$\mathsf{M}|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle = \mathsf{M}\Pi_{T^{\perp}_{j-1,0,0}}|\psi^{0,0}_{i_1,\ldots,i_j}\rangle = \Pi_{T^{\perp}_{j-1,1,1}}|\psi^{1,1}_{i_1,\ldots,i_j}\rangle = |\tilde{\psi}^{1,1}_{i_1,\ldots,i_j}\rangle \ .$$

Finally, we have $\mathsf{U}_{10} = \mathsf{U}''_{10}\mathsf{U}_{11}$, where U''_{10} is defined by $\mathsf{U}''_{10}|\tilde{\psi}^{1,1}_{i_1,\ldots,i_j}\rangle = |\tilde{\psi}^{1,0}_{i_1,\ldots,i_j}\rangle$. Since U_{11} is unitary, it suffices to prove that U''_{10} is a multiple of a unitary transformation and this follows similarly to U_{01} being a multiple of a unitary transformation. □

Claim 9 *Let $|\psi_{00}\rangle$ be an arbitrary state in $S_{j,0,0}$ for some $j \in \{0,\ldots,t-1\}$. Define $|\psi_{ab}\rangle = \mathsf{U}'_{ab}|\psi_{00}\rangle$ for $ab \in \{01,10,11\}$. Let $\alpha'_a = \sqrt{\frac{n-(t-1+a)}{n-j}}\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|$, $\beta'_a = \sqrt{\frac{(t-1+a)-j}{n-j}}\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|$, $\alpha_a = \frac{\alpha'_a}{\sqrt{(\alpha'_a)^2+(\beta'_a)^2}}$, $\beta_a = \frac{\beta'_a}{\sqrt{(\alpha'_a)^2+(\beta'_a)^2}}$. Then*

1. *$|\phi_1\rangle = \alpha_0|\psi_{00}\rangle + \beta_0|\psi_{01}\rangle + \alpha_1|\psi_{10}\rangle + \beta_1|\psi_{11}\rangle$ belongs to $S_{j,+}$;*
2. *$|\phi_2\rangle = \beta_0|\psi_{00}\rangle - \alpha_0|\psi_{01}\rangle + \beta_1|\psi_{10}\rangle - \alpha_1|\psi_{11}\rangle$ belongs to $S_{j+1,+}$;*
3. *Any linear combination of $|\psi_{00}\rangle$, $|\psi_{01}\rangle$, $|\psi_{10}\rangle$ and $|\psi_{11}\rangle$ which is orthogonal to $|\phi_1\rangle$ and $|\phi_2\rangle$ belongs to $S_- = \bigoplus_{j=0}^{t} S_{j,-}$.*

Proof. Let $i_1, \ldots, i_j$ be j distinct elements of $\{2, \ldots, n\}$. As shown in the beginning of the proof of Claim 7,

$$\begin{aligned}
|\tilde{\psi}^a_{i_1,\ldots,i_j}\rangle &= \sqrt{\tfrac{n-(t-1+a)}{n-j}}|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle + \sqrt{\tfrac{(t-1+a)-j}{n-j}}|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle \\
&= \alpha'_a \frac{|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|} + \beta'_a \frac{|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} \ .
\end{aligned}$$

This means that $\|\tilde{\psi}^a_{i_1,\ldots,i_j}\| = \sqrt{(\alpha'_a)^2 + (\beta'_a)^2}$ and

$$\frac{|\tilde{\psi}^a_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^a_{i_1,\ldots,i_j}\|} = \alpha_a \frac{|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|} + \beta_a \frac{|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} \ .$$

Since the states $|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle$ span $S_{j,0,0}$, $|\psi_{00}\rangle$ is a linear combination of states $\frac{|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\|}$. By Claim 8, the states $|\psi_{ab}\rangle$ are linear combinations of $\frac{|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\|}$ with the same coefficients. Therefore, $|\phi_1\rangle$ is a linear combination of

$$\alpha_0 \frac{|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\|} + \beta_0 \frac{|\tilde{\psi}^{0,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{0,1}_{i_1,\ldots,i_j}\|} + \alpha_1 \frac{|\tilde{\psi}^{1,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{1,0}_{i_1,\ldots,i_j}\|} + \beta_1 \frac{|\tilde{\psi}^{1,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{1,1}_{i_1,\ldots,i_j}\|} = \frac{|\tilde{\psi}^{0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{0}_{i_1,\ldots,i_j}\|} + \frac{|\tilde{\psi}^{1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{1}_{i_1,\ldots,i_j}\|} \ ,$$

each of which, by definition, belongs to $S_{j,+}$. This establishes the first part of the claim.

In order to prove the second part, let $i_1, \ldots, i_j$ be distinct elements of $\{2, \ldots, n\}$. We claim

$$\frac{|\tilde{\psi}^a_{1,i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^a_{1,i_1,\ldots,i_j}\|} = \beta_a \frac{|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|} - \alpha_a \frac{|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} \ . \tag{19}$$

By Claim 7, the right hand side of (19) belongs to $S_{j+1,a}$. We need to show it equals $|\tilde{\psi}^a_{1,i_1,\ldots,i_j}\rangle$. We have

$$\begin{aligned}
|\tilde{\psi}^a_{1,i_1,\ldots,i_j}\rangle &= \Pi_{T^\perp_{j,a}} |\psi^a_{1,i_1,\ldots,i_j}\rangle = \Pi_{T^\perp_{j,a}} |\psi^{a,1}_{i_1,\ldots,i_j}\rangle \\
&= \Pi_{T^\perp_{j,a}} \Pi_{T^\perp_{j-1,a,1}} |\psi^{a,1}_{i_1,\ldots,i_j}\rangle = \Pi_{T^\perp_{j,a}} |\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle \ ,
\end{aligned}$$

where the third equality follows from $T_{j-1,a,1} \subseteq T_{j,a}$. This is because the states $|\psi^{a,1}_{i_1,\ldots,i_{j-1}}\rangle$ spanning $T_{j-1,a,1}$ are the same as the states $|\psi^a_{1,i_1,\ldots,i_{j-1}}\rangle$ in $T_{j,a}$. Write

$$|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle = c_1|\delta_1\rangle + c_2|\delta_2\rangle \ ,$$

where

$$\begin{aligned}
|\delta_1\rangle &= \alpha_a \frac{|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|} + \beta_a \frac{|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} \ , \\
|\delta_2\rangle &= \beta_a \frac{|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|} - \alpha_a \frac{|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} \ .
\end{aligned}$$

By Claim 7, we have $|\delta_1\rangle \in S_{j,a} \subseteq T_{j,a}$, $|\delta_2\rangle \in S_{j+1,a} \subseteq T^\perp_{j,a}$. Therefore, $\Pi_{T^\perp_{j,a}} |\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle = c_2|\delta_2\rangle$ and

$$\frac{|\tilde{\psi}^a_{1,i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^a_{1,i_1,\ldots,i_j}\|} = |\delta_2\rangle = \beta_a \frac{|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\rangle}{\|\psi^{a,\tilde{0}}_{i_1,\ldots,i_j}\|} - \alpha_a \frac{|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} \ ,$$

proving (19).

Similarly to the argument for $|\phi_1\rangle$, equation (19) implies that $|\phi_2\rangle$ is a linear combination of

$$\beta_0 \frac{|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\|} - \alpha_0 \frac{|\tilde{\psi}^{0,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{0,1}_{i_1,\ldots,i_j}\|} + \beta_1 \frac{|\tilde{\psi}^{1,0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{1,0}_{i_1,\ldots,i_j}\|} - \alpha_1 \frac{|\tilde{\psi}^{1,1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{1,1}_{i_1,\ldots,i_j}\|} = \frac{|\tilde{\psi}^{0}_{1,i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{0}_{1,i_1,\ldots,i_j}\|} + \frac{|\tilde{\psi}^{1}_{1,i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{1}_{1,i_1,\ldots,i_j}\|}$$

and each of those states belongs to $S_{j+1,+}$.

To prove the third part of Claim 9, we observe that any vector orthogonal to $|\phi_1\rangle$ and $|\phi_2\rangle$ is a linear combination of vectors of the form

$$|\phi_3\rangle = \alpha_0|\psi_{00}\rangle + \beta_0|\psi_{01}\rangle - \alpha_1|\psi_{10}\rangle - \beta_1|\psi_{11}\rangle \ ,$$

and

$$|\phi_4\rangle = \beta_0|\psi_{00}\rangle - \alpha_0|\psi_{01}\rangle - \beta_1|\psi_{10}\rangle + \alpha_1|\psi_{11}\rangle \ .$$

A vector of the form $|\phi_3\rangle$ is a linear combination of vectors

$$\frac{|\tilde{\psi}^{0}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{0}_{i_1,\ldots,i_j}\|} - \frac{|\tilde{\psi}^{1}_{i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{1}_{i_1,\ldots,i_j}\|} \ .$$

A vector of the form $|\phi_4\rangle$ is a linear combination of vectors

$$\frac{|\tilde{\psi}^{0}_{1,i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{0}_{1,i_1,\ldots,i_j}\|} - \frac{|\tilde{\psi}^{1}_{1,i_1,\ldots,i_j}\rangle}{\|\tilde{\psi}^{1}_{1,i_1,\ldots,i_j}\|} \ .$$

This means that we have $|\phi_3\rangle \in S_{j,-}$ and $|\phi_4\rangle \in S_{j+1,-}$, and the third part of the claim follows. □

3.4 Norms of projected basis states

We use the notation $x^{\underline{j}} = x(x-1)\cdots(x-j+1)$.

Claim 10 *The norms of the various projected basis states are as follows*

1. $\|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\| = \sqrt{\frac{(n-t-a+b)^{\underline{j}}}{(n-j)^{\underline{j}}}}$.
2. $\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\| = \sqrt{\frac{n-t_a-j}{n-t_a}}\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|$.
3. $\frac{\|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\| \cdot \|\tilde{\psi}^{1,1}_{i_1,\ldots,i_j}\|}{\|\tilde{\psi}^{0,1}_{i_1,\ldots,i_j}\| \cdot \|\tilde{\psi}^{1,0}_{i_1,\ldots,i_j}\|} = \sqrt{\frac{(n-t)(n-t-j+1)}{(n-t+1)(n-t-j)}}$.

Proof. Define $t_a = t - 1 + a$. We calculate the vector

$$|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle = \Pi_{T^{\perp}_{j-1,a,b}}|\psi^{a,b}_{i_1,\ldots,i_j}\rangle \ .$$

Both vector $|\psi^{a,b}_{i_1,\ldots,i_j}\rangle$ and subspace $T_{j-1,a,b}$ are fixed by

$$\mathsf{U}_\pi|x\rangle = |x_{\pi(1)}\ldots x_{\pi(n)}\rangle$$

for any permutation π that fixes 1 and maps $\{i_1,\ldots,i_j\}$ to itself. Thus $|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle$ is fixed by any such U_π as well. Therefore, the amplitude of $|x\rangle$ with $|x|=t_a$, $x_1=b$ in $|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle$ only depends on $|\{i_1,\ldots,i_j\}\cap\{i: x_i=1\}|$. Hence there exist numbers $\kappa_0,\ldots,\kappa_j$ such that $|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle$ is of the form

$$|\upsilon_{a,b}\rangle=\sum_{m=0}^{j}\kappa_m\sum_{\substack{x:|x|=t_a,x_1=b\\ |\{i_1,\ldots,i_j\}\cap\{i:x_i=1\}|=m}}|x\rangle\ .$$

To simplify the following calculations, we multiply $\kappa_0,\ldots,\kappa_j$ by the same constant so that $\kappa_j=1/\sqrt{\binom{n-j-1}{t_a-j-b}}$. Then $|\upsilon_{a,b}\rangle$ remains a multiple of $|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle$ but may no longer be equal to $|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle$.

The numbers $\kappa_0,\ldots,\kappa_{j-1}$ should be such that the state $|\upsilon_{a,b}\rangle$ is orthogonal to $T_{j-1,a,b}$ and, in particular, orthogonal to the states $|\psi^{a,b}_{i_1,\ldots,i_\ell}\rangle$ for all $\ell\in\{0,\ldots,j-1\}$. By writing out $\langle\upsilon_{a,b}|\psi^{a,b}_{i_1,\ldots,i_\ell}\rangle=0$, we get

$$\sum_{m=\ell}^{j}\kappa_m\binom{n-j-1}{t_a-m-b}\binom{j-\ell}{m-\ell}=0\ . \tag{20}$$

To show that, we first note that $|\psi^{a,b}_{i_1,\ldots,i_\ell}\rangle$ is a uniform superposition of all $|x\rangle$ with $|x|=t_a$, $x_1=b$, $x_{i_1}=\cdots=x_{i_\ell}=1$. If we want to choose x subject to those constraints and also satisfying $|\{i_1,\ldots,i_j\}\cap\{i: x_i=1\}|=m$, then we have to set $x_i=1$ for $m-\ell$ different $i\in\{i_{\ell+1},\ldots,i_j\}$ and for t_a-m-b different $i\notin\{1,i_1,\ldots,i_j\}$. This can be done in $\binom{j-\ell}{m-\ell}$ and $\binom{n-j-1}{t_a-m-b}$ different ways, respectively.

By solving the system of equations (20), starting from $\ell=j-1$ and going down to $\ell=0$, we get that the only solution is

$$\kappa_m=(-1)^{j-m}\frac{\binom{n-j-1}{t_a-j-b}}{\binom{n-j-1}{t_a-m-b}}\kappa_j\ . \tag{21}$$

Let $|\upsilon'_{a,b}\rangle=\frac{|\upsilon_{a,b}\rangle}{\|\upsilon_{a,b}\|}$ be the normalized version of $|\upsilon_{a,b}\rangle$. Then

$$\begin{aligned}|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\rangle&=\langle\upsilon'_{a,b}|\psi^{a,b}_{i_1,\ldots,i_j}\rangle|\upsilon'_{a,b}\rangle\ ,\\ \|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\|&=\langle\upsilon'_{a,b}|\psi^{a,b}_{i_1,\ldots,i_j}\rangle=\frac{\langle\upsilon_{a,b}|\psi^{a,b}_{i_1,\ldots,i_j}\rangle}{\|\upsilon_{a,b}\|}\ .\end{aligned} \tag{22}$$

We have

$$\langle\upsilon_{a,b}|\psi^{a,b}_{i_1,\ldots,i_j}\rangle=1\ , \tag{23}$$

because $|\psi^{a,b}_{i_1,\ldots,i_j}\rangle$ consists of $\binom{n-j-1}{t_a-j-b}$ basis states $|x\rangle$, $x_1=b$, $x_{i_1}=\cdots=x_{i_j}=1$, each having amplitude $1/\sqrt{\binom{n-j-1}{t_a-j-b}}$ in both $|\upsilon_{a,b}\rangle$ and $|\psi^{a,b}_{i_1,\ldots,i_j}\rangle$. Furthermore,

$$\begin{aligned}\|\upsilon_{a,b}\|^2&=\sum_{m=0}^{j}\binom{j}{m}\binom{n-j-1}{t_a-m-b}\kappa_m^2\\ &=\sum_{m=0}^{j}\binom{j}{m}\frac{\binom{n-j-1}{t_a-j-b}^2}{\binom{n-j-1}{t_a-m-b}}\kappa_j^2\\ &=\sum_{m=0}^{j}\binom{j}{m}\frac{\binom{n-j-1}{t_a-j-b}}{\binom{n-j-1}{t_a-m-b}}\\ &=\sum_{m=0}^{j}\binom{j}{m}\frac{(t_a-m-b)!(n-t_a+m-j-1+b)!}{(t_a-j-b)!(n-t_a-1+b)!}\end{aligned}$$

$$= \sum_{m=0}^{j} \binom{j}{m} \frac{(t_a - m - b)^{\underline{j-m}}}{(n - t_a - 1 + b)^{\underline{j-m}}} \ . \tag{24}$$

Here the first equality follows because there are $\binom{j}{m}\binom{n-j-1}{t_a-m-b}$ vectors x such that $|x| = t_a$, $x_1 = b$, $x_i = 1$ for m different $i \in \{i_1, \ldots, i_j\}$ and $t_a - m$ different $i \notin \{1, i_1, \ldots, i_j\}$, the second equality follows from equation (21) and the third equality follows from our choice $\kappa_j = 1/\sqrt{\binom{n-j-1}{t_a-j-b}}$.

From equations (22), (23), and (24), we have

$$\|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\| = \frac{1}{\sqrt{A_{a,b}}},$$

where

$$A_{a,b} = \sum_{m=0}^{\infty} C_{a,b}(m) \quad \text{and} \quad C_{a,b}(m) = \binom{j}{m} \frac{(t_a - m - b)^{\underline{j-m}}}{(n - t_a - 1 + b)^{\underline{j-m}}} \ .$$

The terms with $m > j$ are zero because $\binom{j}{m} = 0$ for $m > j$.

We compute the combinatorial sum $A_{a,b}$ using hyper-geometric series [GKP98, Section 5.5]. Since

$$\frac{C_{a,b}(m+1)}{C_{a,b}(m)} = \frac{(m-j)(m+n-t_a-j+b)}{(m+1)(m-t_a+b)}$$

is a rational function of m, $A_{a,b}$ is a hyper-geometric series and its value is

$$A_{a,b} = \sum_{m=0}^{\infty} C_{a,b}(m) = C_{a,b}(0) \cdot F\left({-j, \ n - t_a - j + b \atop -t_a + b} \Big| 1 \right) \ .$$

We apply Vandermonde's convolution $F({-j, \ x \atop y}|1) = (x-y)^{\underline{j}}/(-y)^{\underline{j}}$ [GKP98, Equation 5.93 on page 212], which holds for every integer $j \geq 0$, and obtain

$$A_{a,b} = \frac{(t_a - b)^{\underline{j}}}{(n - t_a - 1 + b)^{\underline{j}}} \cdot \frac{(n-j)^{\underline{j}}}{(t_a - b)^{\underline{j}}} = \frac{(n-j)^{\underline{j}}}{(n - t_a - 1 + b)^{\underline{j}}} \ .$$

This proves the first part of the claim, that $\|\tilde{\psi}^{a,b}_{i_1,\ldots,i_j}\| = \sqrt{\frac{(n-t_a-1+b)^{\underline{j}}}{(n-j)^{\underline{j}}}}$.

The second part of the claim follows because

$$\frac{\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\|}{\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|} = \sqrt{\frac{(n-t_a-1)^{\underline{j}}}{(n-t_a)^{\underline{j}}}} = \sqrt{\frac{n-t_a-j}{n-t_a}} \ .$$

The expression in the third part of the claim is the square root of the following value:

$$\frac{A_{1,0}A_{0,1}}{A_{0,0}A_{1,1}} = \frac{((n-t)^{\underline{j}})^2}{(n-t+1)^{\underline{j}}(n-t-1)^{\underline{j}}} = \frac{(n-t)(n-t-j+1)}{(n-t+1)(n-t-j)} \ .$$

□

Claim 11 $\alpha_a = \sqrt{\dfrac{n-t_a-j}{n-2j}}$ *and* $\beta_a = \sqrt{\dfrac{t_a-j}{n-2j}}$.

Proof. Define $t_a = t - 1 + a$. By Claim 10, $\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\| = \sqrt{\frac{n-t_a-j}{n-t_a}}\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\|$. That implies

$$\alpha'_a = \frac{\sqrt{n-t_a}}{\sqrt{n-j}}\|\tilde{\psi}^{a,0}_{i_1,\ldots,i_j}\| = \sqrt{\frac{n-t_a}{n-j}}\sqrt{\frac{n-t_a-j}{n-t_a}}\|\tilde{\psi}^{a,1}_{i_1,\ldots,i_j}\| = \sqrt{\frac{n-t_a-j}{t_a-j}}\beta'_a$$

and hence

$$\sqrt{(\alpha_a')^2 + (\beta_a')^2} = \beta_a' \sqrt{\frac{n - t_a - j}{t_a - j} + 1} = \beta_a' \sqrt{\frac{n - 2j}{t_a - j}} \ .$$

Then,

$$\beta_a = \frac{\beta_a'}{\sqrt{(\alpha_a')^2 + (\beta_a')^2}} = \sqrt{\frac{t_a - j}{n - 2j}}$$

and $\alpha_a = \sqrt{1 - \beta_a^2}$. □

Claim 12 *If $t \le \frac{n}{2}$ and $j < t - 1$, then $\alpha_0\beta_1 - \alpha_1\beta_0 \le \frac{1}{2\sqrt{(t-1-j)(n-t-j)}}$. This is $O(\frac{1}{\sqrt{tn}})$ if $j \le \frac{t}{2}$.*

Proof. By the definition from Claim 9 and by Claim 10, we have

$$\begin{aligned}
\frac{\alpha_0\beta_1}{\alpha_1\beta_0} = \frac{\alpha_0'\beta_1'}{\alpha_1'\beta_0'} &= \frac{\sqrt{(n-t+1)(t-j)}}{\sqrt{(n-t)(t-1-j)}} \cdot \frac{\|\tilde{\psi}^{0,0}_{i_1,\ldots,i_j}\|\|\tilde{\psi}^{1,1}_{i_1,\ldots,i_j}\|}{\|\tilde{\psi}^{1,0}_{i_1,\ldots,i_j}\|\|\tilde{\psi}^{0,1}_{i_1,\ldots,i_j}\|} \\
&= \frac{\sqrt{(n-t+1)(t-j)}}{\sqrt{(n-t)(t-1-j)}} \cdot \sqrt{\frac{(n-t)(n-t-j+1)}{(n-t+1)(n-t-j)}} \\
&= \sqrt{\frac{(t-j)(n-t-j+1)}{(t-1-j)(n-t-j)}} \\
&= \sqrt{1 + \frac{n-2j}{(t-1-j)(n-t-j)}} \\
&\le 1 + \frac{n-2j}{2(t-1-j)(n-t-j)} \ .
\end{aligned}$$

We thus have

$$\begin{aligned}
\alpha_0\beta_1 - \beta_0\alpha_1 &= \left(\frac{\alpha_0\beta_1}{\beta_0\alpha_1} - 1\right)\beta_0\alpha_1 \\
&\le \frac{n-2j}{2(t-1-j)(n-t-j)} \sqrt{\frac{t-1-j}{n-2j}} \sqrt{\frac{n-t-j}{n-2j}} \\
&= \frac{1}{2\sqrt{(t-1-j)(n-t-j)}} \ ,
\end{aligned}$$

thanks to Claim 11. □

3.5 Bounding the growth of the potential

Proof of Lemma 4. We first analyze the case when $\rho_{d,1}$ belongs to the subspace $\mathcal{H}_4$ spanned by $|\psi_{ab}\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_k\rangle$, where $|\psi_2\rangle, \ldots, |\psi_k\rangle$ are some vectors from subspaces $R_{j_2}, \ldots, R_{j_k}$ for some $j_2, \ldots, j_k$, $|\psi_{00}\rangle$ is an arbitrary state in $S_{j,0,0}$ for some $j \in \{0, \ldots, t-1\}$, $|\psi_{ab}\rangle = \mathsf{U}'_{ab}|\psi_{00}\rangle$ for $ab \in \{01, 10, 11\}$, and U'_{ab} are the unitaries from Claim 8.

We pick an orthonormal basis for $\mathcal{H}_4$ that has $|\phi_1\rangle$ and $|\phi_2\rangle$ from Claim 9 as its first two vectors. Let $|\phi_3\rangle$ and $|\phi_4\rangle$ be the other two basis vectors. We define

$$|\chi_i\rangle = |\phi_i\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_k\rangle \ . \tag{25}$$

By Claim 9, $|\chi_1\rangle$ belongs to $S_{j,+} \otimes R_{j_2} \otimes \cdots \otimes R_{j_k}$ which is contained in $\mathcal{R}_{\min(j,t/2)+j_2+\cdots+j_k}$. Similarly, $|\chi_2\rangle$ belongs to $\mathcal{R}_{\min(j+1,t/2)+j_2+\cdots+j_k}$ and $|\chi_3\rangle$, $|\chi_4\rangle$ belong to $\mathcal{R}_{t/2+j_2+\cdots+j_k}$. If $j < t/2$, this means that

$$P(\rho_{d,1}) = q^{j_2+\cdots+j_k} \cdot \left(q^j \langle \chi_1|\rho_{d,1}|\chi_1\rangle + q^{j+1}\langle \chi_2|\rho_{d,1}|\chi_2\rangle + q^{\frac{t}{2}}\langle \chi_3|\rho_{d,1}|\chi_3\rangle + q^{\frac{t}{2}}\langle \chi_4|\rho_{d,1}|\chi_4\rangle \right) \ . \tag{26}$$

If $j \geq t/2$, then $|\chi_1\rangle$, $|\chi_2\rangle$, $|\chi_3\rangle$, $|\chi_4\rangle$ are all in $\mathcal{R}_{t/2+j_2+\cdots+j_k}$. This means that $P(\rho_{d,1}) = q^{t/2+j_2+\cdots+j_k}$ and it remains unchanged by a query.

We define $\gamma_\ell = \langle \chi_\ell|\rho_{d,1}|\chi_\ell\rangle$. Since the support of $\rho_{d,1}$ is contained in the subspace spanned by $|\chi_\ell\rangle$, we have $\gamma_1 + \gamma_2 + \gamma_3 + \gamma_4 = \operatorname{Tr}\rho_{d,1} = 1$. This means that equation (26) can be rewritten as

$$\begin{aligned} P(\rho_{d,1}) &= q^{j+j_2+\cdots+j_k}\gamma_1 + q^{j+j_2+\cdots+j_k+1}\gamma_2 + q^{t/2+j_2+\cdots+j_k}(\gamma_3+\gamma_4) \\ &= q^{t/2+j_2+\cdots+j_k} + q^{j_2+\cdots+j_k}(q^{j+1} - q^{t/2})(\gamma_1+\gamma_2) + q^{j_2+\cdots+j_k}(q^j - q^{j+1})\gamma_1 \ . \end{aligned} \tag{27}$$

$P(\rho'_{d,1})$ can be also expressed in a similar way, with $\gamma'_j = \langle \chi_j|\rho'_{d,1}|\chi_j\rangle$ instead of γ_j. By combining equations (27) for $P(\rho_{d,1})$ and $P(\rho'_{d,1})$, we get

$$P(\rho'_{d,1}) - P(\rho_{d,1}) = q^{j+j_2+\cdots+j_k}(q^{t/2-j} - q)(\gamma_1+\gamma_2-\gamma'_1-\gamma'_2) \qquad + q^{j+j_2+\cdots+j_k}(q-1)(\gamma_1-\gamma'_1) \ .$$

Therefore, it suffices to bound $|\gamma_1+\gamma_2-\gamma'_1-\gamma'_2|$ and $|\gamma_1-\gamma'_1|$. Without loss of generality we can assume that $\rho_{d,1}$ is a pure state $|\varphi\rangle\langle\varphi|$. Let

$$|\varphi\rangle = (a|\psi_{00}\rangle + b|\psi_{01}\rangle + c|\psi_{10}\rangle + d|\psi_{11}\rangle) \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_k\rangle \ .$$

Then the state after a query is

$$|\varphi'\rangle = (a|\psi_{00}\rangle - b|\psi_{01}\rangle + c|\psi_{10}\rangle - d|\psi_{11}\rangle) \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_k\rangle$$

and we have to bound

$$\gamma_\ell - \gamma'_\ell = |\langle \chi_\ell|\varphi\rangle|^2 - |\langle \chi_\ell|\varphi'\rangle|^2$$

for $\ell \in \{1,2\}$. For $\ell = 1$, we have

$$\langle \chi_1|\varphi\rangle = a\alpha_0 + b\beta_0 + c\alpha_1 + d\beta_1 \ .$$

The expression for φ' is similar, with minus signs in front of $b\beta_0$ and $d\beta_1$. Therefore,

$$\left| |\langle \chi_1|\varphi\rangle|^2 - |\langle \chi_1|\varphi'\rangle|^2 \right| \leq 4|a||b|\alpha_0\beta_0 + 4|c||d|\alpha_1\beta_1 + 4|a||d|\alpha_0\beta_1 + 4|b||c|\alpha_1\beta_0 \ . \tag{28}$$

Since $|a|$, $|b|$, $|c|$, $|d|$ are all at most $\|\varphi\| = 1$ and α_0, α_1 are less than 1, equation (28) is at most $8\beta_0 + 8\beta_1$. By Claim 11, we have

$$|\gamma_1 - \gamma'_1| \leq 8\beta_0 + 8\beta_1 \leq 16\sqrt{\frac{2t}{n}} \ .$$

We also have

$$\begin{aligned} |\gamma_1+\gamma_2-\gamma'_1-\gamma'_2| &= \left| |\langle \chi_1|\varphi\rangle|^2 + |\langle \chi_2|\varphi\rangle|^2 - |\langle \chi_1|\varphi'\rangle|^2 - |\langle \chi_2|\varphi'\rangle|^2 \right| \\ &\leq 4|a||d||\alpha_0\beta_1 - \alpha_1\beta_0| + 4|b||c||\alpha_1\beta_0 - \alpha_0\beta_1| \\ &\leq 8|\alpha_0\beta_1 - \alpha_1\beta_0| \leq \frac{8C}{\sqrt{tn}} \ , \end{aligned}$$

where C is the big-O constant from Claim 12. By taking into account that $P(\rho_{d,1}) \geq q^{j+j_2+\cdots+j_k}$,

$$\begin{aligned} P(|\varphi'\rangle\langle\varphi'|) - P(|\varphi\rangle\langle\varphi|) &\leq \left((q^{t/2-j} - q)\frac{8C}{\sqrt{tn}} + (q-1)\frac{16\sqrt{2t}}{\sqrt{n}} \right) P(|\varphi\rangle\langle\varphi|) \\ &\leq \left((q^{t/2} - 1)\frac{8C}{\sqrt{tn}} + (q-1)\frac{16\sqrt{2t}}{\sqrt{n}} \right) P(|\varphi\rangle\langle\varphi|) \ . \end{aligned} \tag{29}$$

This proves Lemma 4 for the case when the support of $\rho_{d,1}$ is contained in $\mathcal{H}_4$. (If $\rho_{d,1}$ is a mixed state, we just express it as a mixture of pure states $|\varphi\rangle$. The bound for $\rho_{d,1}$ follows by summing equations (29) for every $|\varphi\rangle$.)

For the general case, we divide the entire state space $\mathcal{H}_I$ into 4-dimensional subspaces. To do that, we first subdivide $\mathcal{H}_I$ into subspaces

$$(S_{j,0,0} \oplus S_{j,0,1} \oplus S_{j,1,0} \oplus S_{j,1,1}) \otimes R_{j_2} \otimes \cdots \otimes R_{j_k} \ . \tag{30}$$

Let states $|\psi_{1,i}^{0,0}\rangle$, $i \in [\dim S_{j,0,0}]$ form a basis for $S_{j,0,0}$ and let $|\psi_{1,i}^{a,b}\rangle = \mathsf{U}'_{ab}|\psi_{1,i}^{0,0}\rangle$ for $(a,b) \in \{(0,1),(1,0),(1,1)\}$, where the U'_{ab} are the unitaries from Claim 8. Then the $|\psi_{1,i}^{a,b}\rangle$ form a basis for $S_{j,a,b}$.

Let $|\psi_{l,i}\rangle$, $i \in [\dim R_{j_l}]$, form a basis for R_{j_l}, $l \in \{2,\ldots,k\}$. We subdivide (30) into 4-dimensional subspaces $H_{i_1,\ldots,i_k}$ spanned by

$$|\psi_{1,i_1}^{a,b}\rangle \otimes |\psi_{2,i_2}\rangle \otimes \cdots \otimes |\psi_{k,i_k}\rangle \ ,$$

where a, b range over $\{0,1\}$. Let $\mathcal{H}_{all}$ be the collection of all $H_{i_1,\ldots,i_k}$ obtained by subdividing all subspaces (30). We claim that

$$P(\rho) = \sum_{H \in \mathcal{H}_{all}} P(\Pi_H \rho) \ . \tag{31}$$

Equation (31) together with equation (29) implies Lemma 4. Since $P(\rho)$ is defined as a weighted sum of traces $\operatorname{Tr}\Pi_{\mathcal{R}_m}\rho$, we can prove equation (31) by showing

$$\operatorname{Tr}\Pi_{\mathcal{R}_m}\rho_{d,1} = \sum_{H \in \mathcal{H}_{all}} \operatorname{Tr}\Pi_{\mathcal{R}_m}\Pi_H\rho_{d,1} \ . \tag{32}$$

To prove (32), we define a basis for $\mathcal{H}_I$ by first decomposing $\mathcal{H}_I$ into subspaces $H \in \mathcal{H}_{all}$, and then for each subspace, taking the basis consisting of $|\chi_1\rangle$, $|\chi_2\rangle$, $|\chi_3\rangle$ and $|\chi_4\rangle$ defined by equation (25). By Claim 9, each of the basis states belongs to one of the subspaces $\mathcal{R}_m$. This means that each $\mathcal{R}_m$ is spanned by some subset of this basis.

The left hand side of (32) is equal to the sum of squared projections of $\rho_{d,1}$ to basis states $|\chi_j\rangle$ that belong to $\mathcal{R}_m$. Each of the terms $\operatorname{Tr}\Pi_{\mathcal{R}_m}\Pi_H\rho_{d,1}$ on the right hand side is equal to the sum of squared projections to basis states $|\chi_j\rangle$ that belong to $\mathcal{R}_m \cap H$. Summing over all H gives the sum of squared projections of $\rho_{d,1}$ to all $|\chi_j\rangle$ that belong to $\mathcal{R}_m$. Therefore, the two sides of (32) are equal. □

4 Direct Product Theorem for Threshold Functions (1-sided)

The previous section used the adversary method to prove a direct product theorem for 2-sided error algorithms computing k instances of some symmetric function. In this section we use the polynomial method to obtain stronger direct product theorems for *1-sided error* algorithms for threshold functions. An algorithm for $f^{(k)}$ is said to have 1-sided error if the 1's in its k-bit output vector are always correct.

Our use of polynomials is a relatively small extension of the argument in [KŠW07]. We use three results about polynomials, also used in [BCWZ99, KŠW07]. The first is by Coppersmith and Rivlin [CR92, p. 980] and gives a general bound for polynomials bounded by 1 at integer points:

Theorem 13 (Coppersmith & Rivlin [CR92]) *Every polynomial p that has degree $d \le n$ and absolute value*

$$|p(i)| \le 1 \text{ for all integers } i \in [0,n] \ ,$$

satisfies

$$|p(x)| < ae^{bd^2/n} \text{ for all real } x \in [0,n] \ ,$$

where $a, b > 0$ are universal constants (no explicit values for a and b are given in [CR92]).

The other two results concern the Chebyshev polynomials T_d, defined by (see for example [Riv90]):

$$T_d(x) = \frac{1}{2}\left(\left(x + \sqrt{x^2-1}\right)^d + \left(x - \sqrt{x^2-1}\right)^d\right) \ .$$

This T_d has degree d and its absolute value $|T_d(x)|$ is bounded by 1 if $x \in [-1, 1]$. On the interval $[1, \infty)$, T_d exceeds all others polynomials with those two properties ([Riv90, p.108] and [Pat92, Fact 2]).

Theorem 14 *If q is a polynomial of degree d such that $|q(x)| \leq 1$ for all $x \in [-1, 1]$, then $|q(x)| \leq |T_d(x)|$ for all $x \geq 1$.*

Paturi [Pat92, before Fact 2] proved the following:

Lemma 15 (Paturi [Pat92]) *$T_d(1 + \mu) \leq e^{2d\sqrt{2\mu+\mu^2}}$ for all $\mu \geq 0$.*

Proof. For $x = 1 + \mu$: $T_d(x) \leq (x + \sqrt{x^2-1})^d = (1 + \mu + \sqrt{2\mu+\mu^2})^d \leq (1 + 2\sqrt{2\mu+\mu^2})^d \leq e^{2d\sqrt{2\mu+\mu^2}}$ (using that $1 + z \leq e^z$ for all real z). $\square$

Key lemma. The following lemma is the key. It analyzes polynomials that are 0 on the first m integer points, and that significantly "jump" a bit later.

Lemma 16 *Suppose E, N, m are integers satisfying $10 \leq E \leq \frac{N}{2m}$, and let p be a degree-D polynomial such that*

$$\begin{aligned} p(i) &= 0 && \text{for all } i \in \{0, \ldots, m-1\} \\ p(8m) &= \sigma && \\ p(i) &\in [0, 1] && \text{for all } i \in \{0, \ldots, N\} \ . \end{aligned}$$

Then $\sigma \leq 2^{O(D^2/N + D\sqrt{Em/N} - m\log E)}$.

Proof. Divide p by $\prod_{j=0}^{m-1}(x - j)$ to obtain

$$p(x) = q(x)\prod_{j=0}^{m-1}(x - j) \ ,$$

where $d = \deg(q) = D - m$. This implies the following about the values of the polynomial q:

$$\begin{aligned} |q(8m)| &= \frac{\sigma}{\prod_{j=0}^{m-1}(8m - j)} \geq \frac{\sigma}{(8m)^m} \ , && \\ |q(i)| &\leq \frac{1}{\prod_{j=0}^{m-1}(i - j)} \leq \frac{1}{((E-1)m)^m} && \text{for } i \in \{Em, \ldots, N\} \ . \end{aligned}$$

Theorem 13 implies that there are constants $a, b > 0$ such that

$$|q(x)| \leq \frac{a}{((E-1)m)^m} e^{bd^2/(N-Em)} = B \qquad \text{for all real } x \in [Em, N] \ .$$

We now divide q by B to normalize it, and re-scale the interval $[Em, N]$ to $[1, -1]$ to get a degree-d polynomial t satisfying

$$\begin{aligned} |t(x)| &\leq 1 && \text{for all } x \in [-1, 1] \ , \\ t(1 + \mu) &= \frac{q(8m)}{B} && \text{for } \mu = \frac{2(E-8)m}{N - Em} \ . \end{aligned}$$

Since t cannot grow faster than the degree-d Chebyshev polynomial, Theorem 14 and Lemma 15 imply

$$t(1+\mu) \leq e^{2d\sqrt{2\mu+\mu^2}} \ .$$

Combining our upper and lower bounds on $t(1+\mu)$ gives

$$\frac{\sigma}{(8m)^m} \cdot \frac{((E-1)m)^m}{ae^{O(d^2/N)}} \leq e^{O(d\sqrt{Em/N})} \ ,$$

which implies the lemma. $\square$

Theorem 17 *There exists $\alpha > 0$ such that for every threshold function T_t and positive integer k: Every 1-sided error quantum algorithm with $T \leq \alpha k Q_2(T_t)$ queries for computing $T_t^{(k)}$ has success probability $\sigma \leq 2^{-\Omega(kt)}$.*

Proof. We assume without loss of generality that $t \leq \frac{n}{20}$, the other cases can easily be reduced to this. We know that $Q_2(T_t) = \Theta(\sqrt{tn})$ [BBC+01]. Consider a quantum algorithm A with $T \leq \alpha k\sqrt{tn}$ queries that computes $f^{(k)}$ with success probability σ. Roughly speaking, below we use A to solve one big threshold problem on the total input, and then invoke the polynomial lemma to upper bound the success probability.

Define a new quantum algorithm B on an input x of $N = kn$ bits as follows. Algorithm B runs A on a random permutation $\pi(x)$, and then outputs 1 if and only if the k-bit output vector has at least $\frac{k}{2}$ ones.

Let $m = \frac{kt}{2}$. Note that if $|x| < m$, then B always outputs 0 because the 1-sided error output vector must have fewer than $\frac{k}{2}$ ones. Now suppose $|x| = 8m = 4kt$. Call an n-bit input block *full* if $\pi(x)$ contains at least t ones in that block. Let F be the random variable counting how many of the k blocks are full. We claim that $\Pr[F \geq \frac{k}{2}] \geq \frac{1}{9}$. To prove this, observe that the number B of ones in one fixed block is a random variable distributed according to a hyper-geometric distribution ($4kt$ balls into N boxes, n of which count as success) with expectation $\mu = 4t$ and variance $V \leq 4t$. Using Chebyshev's inequality we bound the probability that this block is not full:

$$\Pr[B < t] \leq \Pr[|B-\mu| > 3t] \leq \Pr\left[|B-\mu| > \frac{3\sqrt{t}}{2}\sqrt{V}\right] < \frac{1}{(3\sqrt{t}/2)^2} \leq \frac{4}{9}.$$

Hence the probability that the block is full ($B \geq t$) is at least $\frac{5}{9}$. This is true for each of the k blocks, so using linearity of expectation we have

$$\frac{5k}{9} \leq \mathrm{Exp}[F] \leq \Pr[F \geq \tfrac{k}{2}] \cdot k + (1 - \Pr[F \geq \tfrac{k}{2}]) \cdot \frac{k}{2} \ .$$

This implies $\Pr[F \geq \frac{k}{2}] \geq \frac{1}{9}$, as claimed. But then on all inputs with $|x| = 8m$, B outputs 1 with probability at least $\frac{\sigma}{9}$.

Algorithm B uses $\alpha k\sqrt{tn}$ queries. By [BBC+01] and symmetrization, the acceptance probability of B is a single-variate polynomial p of degree $D \leq 2\alpha k\sqrt{tn}$ such that

$$\begin{aligned} p(i) &= 0 && \text{for all } i \in \{0, \ldots, m-1\} \\ p(8m) &\geq \tfrac{\sigma}{9} && \\ p(i) &\in [0,1] && \text{for all } i \in \{0, \ldots, N\} \ . \end{aligned}$$

The result now follows by applying Lemma 16 with $N = kn$, $m = \frac{kt}{2}$, $E = 10$, and α a sufficiently small positive constant. $\square$

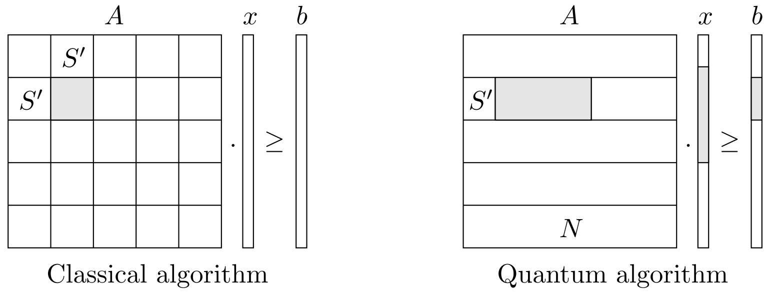

Figure 1: Parallel computation of S' rows in a matrix-vector product

5 Time-Space Tradeoff for Systems of Linear Inequalities

Let A be a fixed $N \times N$ matrix of non-negative integers and let x, b be two input vectors of N non-negative integers smaller or equal to t. A *matrix-vector product with upper bound*, denoted by $y = (Ax)_{\le b}$, is a vector y such that $y_i = \min((Ax)[i], b_i)$. An *evaluation of a system of linear inequalities* $Ax \ge b$ is the N-bit vector of the truth values of the individual inequalities. Here we present a quantum algorithm for matrix-vector product with upper bound that satisfies time-space tradeoff $T^2 S = O(tN^3(\log N)^5)$. We then use our direct product theorems to show this is close to optimal.

5.1 Upper bound

Classical algorithm. It is easy to show that matrix-vector products with upper bound t can be computed by a classical algorithm with $TS = O(N^2 \log t)$, as follows. Let $S' = \frac{S}{\log t}$ and divide the matrix A into $(\frac{N}{S'})^2$ blocks of size $S' \times S'$ each. The output vector is evaluated S' rows at a time, as follows:

1. Clear S' counters, one for each of the S' rows considered at this time, and read the upper bounds b_i corresponding to these rows.

2. For each block, read the corresponding S' input variables from x, multiply them by the corresponding sub-matrix of A, and update the counters, but do not let them grow larger than b_i.

3. Output the S' counters.

The space used is $O(S' \log t) = O(S)$ and the total query complexity is $T = O(\frac{N}{S'} \cdot \frac{N}{S'} \cdot S') = O(N^2 (\log t)/S)$.

Quantum algorithm. The quantum algorithm Bounded Matrix Product works in a similar way and it is outlined in Table 2. We compute the matrix product in groups of $S' = \frac{S}{\log N}$ rows, read input variables, and update the counters accordingly. The advantage over the classical algorithm is that we use faster quantum search and quantum counting for finding nonzero entries. The two algorithms are compared in Figure 1.

The u^{th} row is called *open* if its counter has not yet reached b_u. The subroutine Small Matrix Product maintains a set of open rows $U \subseteq \{1, \ldots, S'\}$ and counters $0 \le y_u \le b_u$ for all $u \in U$. We process the input x in blocks, each containing between $S' - O(\sqrt{S'})$ and $2S' + O(\sqrt{S'})$ nonzero numbers at the positions j where $A[u, j] \ne 0$ for some $u \in U$. See Figure 2—the dashed vertical lines correspond to these numbers. The length ℓ of such a block is first found by quantum counting and the nonzero input numbers are then found by a Grover search. For each such number, we update all counters y_u and close all rows that have exceeded their threshold b_u.

Theorem 18 Bounded Matrix Product *has bounded error, space complexity $O(S)$, and query complexity $T = O(N^{3/2}\sqrt{t} \cdot (\log N)^{5/2}/\sqrt{S})$.*

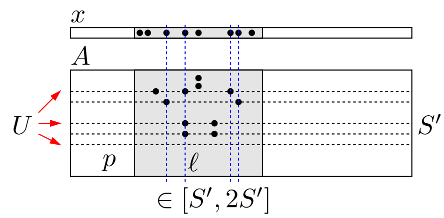

Figure 2: Nonzero numbers in the input string are only searched for at certain positions

Proof. The space complexity of Small Matrix Product is $O(S' \log N) = O(S)$, because it stores a subset $U \subseteq \{1, \dots, S'\}$, integer vectors y, b of dimension S' with numbers at most $t \le N$, the set J of size $O(S')$ with numbers at most N, and a few counters. Let us estimate its query complexity.

Consider the i^{th} block found by Small Matrix Product; let p_i be its left column, let ℓ_i be its length, and let U_i be the set of open rows at the beginning of processing of this block. The scalar product c_{p_i,ℓ_i} is estimated by quantum counting with $\sqrt{\ell_i}$ queries. Finding a proper ℓ_i requires $O(\log \ell_i)$ iterations. Let r_i be the number of rows closed during processing of this block and let s_i be the total number added to the counters for other (still open) rows in this block. The numbers ℓ_i, r_i, s_i are random variables. If we instantiate them at the end of the quantum subroutine, the following inequalities hold:

$$\sum_i \ell_i \le N, \quad \sum_i r_i \le S', \text{ and } \quad \sum_i s_i \le tS' \ .$$

The iterated Grover search finds ones for two purposes: closing rows and increasing counters. Since each b_i is at most t, the total cost in the i^{th} block is at most $\sum_{j=1}^{r_i t} O(\sqrt{\ell_i/j}) + \sum_{j=1}^{s_i} O(\sqrt{\ell_i/j}) = O(\sqrt{\ell_i r_i t} + \sqrt{\ell_i s_i})$. By the Cauchy-Schwarz inequality, the total number of queries that Small Matrix Product spends in the Grover searches is at most

$$\sum_{i=1}^{\#\text{blocks}} (\sqrt{\ell_i r_i t} + \sqrt{\ell_i s_i}) \le \sqrt{\sum_i \ell_i}\sqrt{t \sum_i r_i} + \sqrt{\sum_i \ell_i}\sqrt{\sum_i s_i} \le \sqrt{N}\sqrt{tS'} + \sqrt{N}\sqrt{tS'} = O(\sqrt{NS't}) \ .$$

The error probability of the Grover searches can be made polynomially small in N, at the cost of a factor of $O(\log N)$ in time. It remains to analyze the outcome and error probability of quantum counting. Let $c_i = c_{p_i,\ell_i} \in [S', 2S']$. One quantum counting call with $M = \sqrt{\ell_i}$ queries gives an estimate w such that

$$|w - c_i| = O\left(\sqrt{\frac{c_i(\ell_i - c_i)}{\ell_i}} + \frac{\ell_i}{\ell_i}\right) = O(\sqrt{c_i}) = O(\sqrt{S'})$$

with probability at least $\frac{8}{\pi^2} \approx 0.8$ (see page 6). We do this $O(\log N)$ times and take the median, thus obtaining an estimate $\tilde{c}$ of c_i with accuracy $O(\sqrt{S'})$ with polynomially small error probability. The result of quantum counting is compared with the given threshold, that is with S' or $2S'$. Binary search for $\ell \in [\frac{k}{2}, k]$ costs another factor of $\log k \le \log N$. By a Cauchy-Schwarz inequality, the total number of queries spent in the quantum counting is at most $(\log N)^2$ times

$$\sum_i \sqrt{\ell_i} \le \sqrt{\sum_i \ell_i}\sqrt{\sum_i 1} \le \sqrt{N}\sqrt{\#\text{blocks}} \le \sqrt{N}\sqrt{S' + t} \le \sqrt{NS't},$$

because in every block the algorithm closes a row or adds $\Theta(S')$ in total to the counters. The number of closed rows is at most S' and the number S' can be added at most t times.

The total query complexity of Small Matrix Product is thus $O(\sqrt{NS't} \cdot (\log N)^2)$ and the query complexity of Bounded Matrix Product is $\frac{N}{S'}$-times bigger. By the union bound, the overall error

Bounded Matrix Product (fixed matrix $A_{N\times N}$, threshold t, input vectors x and b of dimension N) returns output vector $y = (Ax)_{\leq b}$.

For $i = 1, 2, \ldots, \frac{N}{S'}$, where $S' = \frac{S}{\log N}$:

1. Run Small Matrix Product on the i^{th} block of S' rows of A.
2. Output the S' obtained results for those rows.

Small Matrix Product (fixed matrix $A_{S'\times N}$, input vectors $x_{N\times 1}$ and $b_{S'\times 1}$) returns $y_{S'\times 1} = (Ax)_{\leq b}$.

3. Initialize $y := (0, 0, \ldots, 0)$, $p := 1$, $U := \{1, \ldots, S'\}$, and read b. Let $a_{1\times N}$ denote an on-line computed N-bit row-vector with $a_j = 1$ if $A[u, j] = 1$ for some $u \in U$, and $a_j = 0$ otherwise.
4. While $p \leq N$ and $U \neq \emptyset$, do the following:
   (a) Let $\tilde{c}_{p,k}$ denote an estimate of $c_{p,k} = \sum_{j=p}^{p+k-1} a_j x_j$; we estimate it by computing the median of $O(\log N)$ calls to Quantum Counting $((a_p x_p) \ldots (a_{p+k-1} x_{p+k-1}), \sqrt{k})$.
      - Initialize $k = S'$.
      - While $p + k - 1 < N$ and $\tilde{c}_{p,k} < S'$, double k.
      - Find by binary search the maximal $\ell \in [\frac{k}{2}, k]$ for which $p + \ell - 1 \leq N$ and $\tilde{c}_{p,\ell} \leq 2S'$.

   (b) Use repeated Grover search to find the set J of all positions $j \in [p, p + \ell - 1]$ such that $a_j x_j > 0$.
   (c) For all $j \in J$, read x_j, and then do the following for all $u \in U$:
      - Increase y_u by $A[u, j]x_j$.
      - If $y_u \geq b_u$, then set $y_u := b_u$ and remove u from U.

   (d) Increase p by ℓ.
5. Return y.

Table 2: Algorithm Bounded Matrix Product

probability is at most the sum of the polynomially small error probabilities of the different subroutines, hence it can be kept below $\frac{1}{3}$. $\square$

5.2 Matching quantum lower bound

Here we use our direct product theorems to lower bound the quantity T^2S for T-query, S-space quantum algorithms for systems of linear inequalities. The lower bound even holds if we fix b to the all-t vector $\vec{t}$ and let A and x be Boolean.

Theorem 19 *Let $S \leq \min(O(\frac{N}{t}), o(\frac{N}{\log N}))$. There exists an $N \times N$ Boolean matrix A such that every 2-sided error quantum algorithm for evaluating a system $Ax \geq \vec{t}$ of N inequalities that uses T queries and space S satisfies $T^2S = \Omega(tN^3)$.*

Proof. The proof is a modification of Theorem 22 of [KŠW07] (quant-ph version). They use the probabilistic method to establish the following

Fact: For every $k = o(N/\log N)$, there exists an $N \times N$ Boolean matrix A, such that all rows of A have weight $N/2k$, and every set of k rows of A contains a set R of $k/2$ rows with the following property: each row in R contains at least $n = N/6k$ ones that occur in no other row of R.

Fix such a matrix A for $k = cS$, for some constant c to be chosen later. Consider a quantum circuit with T queries and space S that solves the problem with success probability at least $\frac{2}{3}$. We slice the quantum circuit into disjoint consecutive slices, each containing $Q = \alpha\sqrt{tNS}$ queries, where α is the constant from our direct product theorem (Theorem 1). The total number of slices is $L = \frac{T}{Q}$. Together, these disjoint slices contain all N output gates. Our aim below is to show that with sufficiently small constant α and sufficiently large constant c, no slice can produce more than k outputs. This will imply that the number of slices is $L \geq \frac{N}{k}$, hence

$$T = LQ \geq \frac{\alpha N^{3/2}\sqrt{t}}{c\sqrt{S}} \ .$$

Now consider any slice. It starts with an S-qubit state that is delivered by the previous slice and that depends on the input, then it makes Q queries and outputs some ℓ results that are jointly correct with probability at least $\frac{2}{3}$. Suppose, by way of contradiction, that $\ell \geq k$. Then there exists a set of k rows of A such that our slice produces the k corresponding results (t-threshold functions) with probability at least $\frac{2}{3}$. By the above Fact, some set R of $\frac{k}{2}$ of those rows has the following property: each row from R contains a set of $n = \frac{N}{6k} = \Theta(\frac{N}{S})$ ones that do not occur in any of the $\frac{k}{2} - 1$ other rows of R. By setting all other $N - \frac{kn}{2}$ bits of x to 0, we naturally get that our slice, with the appropriate S-qubit starting state, solves $\frac{k}{2}$ independent t-threshold functions T_t on n bits each. (Note that we need $t \leq \frac{n}{2} = O(\frac{N}{S})$; this follows from our assumption $S = O(\frac{N}{t})$ with appropriately small constant in the $O(\cdot)$.) Now we replace the initial S-qubit state by the completely mixed state, which has overlap 2^{-S} with every S-qubit state. This turns the slice into a stand-alone algorithm solving $T_t^{(k/2)}$ with success probability

$$\sigma \geq \frac{2}{3} \cdot 2^{-S} \ .$$

But this algorithm uses only $Q = \alpha\sqrt{tNS} = O(\alpha k\sqrt{tn})$ queries, so our direct product theorem (Theorem 1) with sufficiently small constant α implies

$$\sigma \leq 2^{-\Omega(k/2)} = 2^{-\Omega(cS/2)} \ .$$

Choosing c a sufficiently large constant (independent of this specific slice), our upper and lower bounds on σ contradict. Hence the slice must produce fewer than k outputs. □

It is easy to see that the case $S \geq \frac{N}{t}$ (equivalently, $t \geq \frac{N}{S}$) is at least as hard as the $S = \frac{N}{t}$ case, for which we have the lower bound $T^2 S = \Omega(tN^3) = \Omega(N^4/S)$, hence $TS = \Omega(N^2)$. But that lower bound matches the *classical* deterministic upper bound up to a logarithmic factor and hence is essentially tight also for quantum. We thus have two different regimes for space: for small space, a quantum computer is faster than a classical one in evaluating solutions to systems of linear inequalities, while for large space it is not.

A similar slicing proof using Theorem 17 (with each slice of $Q = \alpha\sqrt{NS}$ queries producing at most $\frac{S}{t}$ outputs) gives the following lower bound on time-space tradeoffs for 1-sided error algorithms.

Theorem 20 *Let $t \leq S \leq \min(O(\frac{N}{t^2}), o(\frac{N}{\log N}))$. There exists an $N \times N$ Boolean matrix A such that every 1-sided error quantum algorithm for evaluating a system $Ax \geq \vec{t}$ of N inequalities that uses T queries and space S satisfies $T^2 S = \Omega(t^2 N^3)$.*

Note that our lower bound $\Omega(t^2 N^3)$ for 1-sided error algorithms is higher by a factor of t than the best upper bound for 2-sided error algorithms. This lower bound is probably not optimal. If $S > \frac{N}{t^2}$, then the essentially optimal classical tradeoff $TS = \Omega(N^2)$ takes over.

6 Summary

In this paper we described a new version of the adversary method for quantum query lower bounds, based on analyzing the eigenspace structure of the problem we want to lower bound. We proved two new quantum direct product theorems, the first using the new adversary method, the second using the polynomial method:

- For every symmetric function f, every 2-sided error quantum algorithm for $f^{(k)}$ using fewer than $\alpha k Q_2(f)$ queries has success probability at most $2^{-\Omega(k)}$.
- For every t-threshold function f, every 1-sided error quantum algorithm for $f^{(k)}$ using fewer than $\alpha k Q_2(f)$ queries has success probability at most $2^{-\Omega(kt)}$.

Both results are tight up to constant factors. From these results we derived the following time-space tradeoffs for quantum algorithms that evaluate a system $Ax \geq b$ of N linear inequalities (where A is a fixed $N \times N$ matrix of nonnegative integers, x, b are variable, $b_i \leq t$ for all i, and the algorithm has to determine the truth-value for each of the N inequalities):

- Every T-query, S-space 2-sided error quantum algorithm for evaluating $Ax \geq b$ satisfies $T^2 S = \Omega(tN^3)$ if $S \leq N/t$, and satisfies $TS = \Omega(N^2)$ if $S > N/t$. We gave an algorithm matching these bounds up to polylogarithmic factors.
- Every T-query, S-space 1-sided error quantum algorithm for evaluating $Ax \geq b$ satisfies $T^2 S = \Omega(t^2 N^3)$ if $t \leq S \leq N/t^2$, and satisfies $TS = \Omega(N^2)$ if $S > N/t^2$. We do not have a matching algorithm in the first case and conjecture that this bound is not tight.

Acknowledgments

We thanks the anonymous referees for some useful comments.

References

[Aar05] S. Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, 1:1–28, 2005. Earlier version in Complexity'04.

[ACR+07] A. Ambainis, A. Childs, B. Reichardt, R. Špalek, and S. Zhang. Any AND-OR formula of size n can be evaluated in time $n^{1/2+o(1)}$ on a quantum computer. In *Proc. of 48th IEEE FOCS*, 2007.

[Amb02] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. Earlier version in STOC'00.

[Amb04] A. Ambainis. Quantum walk algorithm for element distinctness. In *Proc. of 45th IEEE FOCS*, pages 22–31, 2004.

[Amb05] A. Ambainis. A new quantum lower bound method, with an application to strong direct product theorem for quantum search. quant-ph/0508200, 2005.

[Amb06] A. Ambainis. Polynomial degree vs. quantum query complexity. *Journal of Computer and System Sciences*, 72(2):220–238, 2006. Earlier version in FOCS'03.

[AS04] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problem. *Journal of the ACM*, 51(4):595–605, 2004.

[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. Earlier version in FOCS'98.

[BBHT98] M. Boyer, G. Brassard, P. Høyer, and A. Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4–5):493–505, 1998. Earlier version in Physcomp'96.

[BCWZ99] H. Buhrman, R. Cleve, R. de Wolf, and Ch. Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proc. of 40th IEEE FOCS*, pages 358–368, 1999.

[BHMT02] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74. 2002.

[BŠ06] H. Buhrman and R. Špalek. Quantum verification of matrix products. In *Proc. of 17th ACM-SIAM SODA*, pages 880–889, 2006.

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

[BV97] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. Earlier version in STOC'93.

[BW02] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[CR92] D. Coppersmith and T. J. Rivlin. The growth of polynomials bounded at equally spaced points. *SIAM Journal on Mathematical Analysis*, 23(4):970–983, 1992.

[DJ92] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society, London*, A439:553–558, 1992.

[FGG07] E. Farhi, J. Goldstone, and S. Gutmann. A quantum algorithm for the Hamiltonian NAND tree. quant-ph/0702144, 2007.

[FGGS98] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. A limit on the speed of quantum computation in determining parity. *Physical Review Letters*, 81:5442–5444, 1998.

[FR99] L. Fortnow and J. D. Rogers. Complexity limitations on quantum computation. *Journal of Computer and System Sciences*, 59(2):240–252, 1999. Earlier version in Complexity'98.

[GKP98] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete mathematics*. Addison-Wesley, 1998.

[Gro96] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. of 28th ACM STOC*, pages 212–219, 1996.

[HLŠ07] P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proc. of 39th ACM STOC*, pages 526–535, 2007.

[HMW03] P. Høyer, M. Mosca, and R. de Wolf. Quantum search on bounded-error inputs. In *Proc. of 30th ICALP*, pages 291–299, 2003. LNCS 2719.

[HNS02] P. Høyer, J. Neerbek, and Y. Shi. Quantum complexities of ordered searching, sorting, and element distinctness. *Algorithmica*, 34(4):429–448, 2002. Special issue on Quantum Computation and Cryptography.

[KŠW07] H. Klauck, R. Špalek, and R. de Wolf. Quantum and classical strong direct product theorems and optimal time-space tradeoffs. *SIAM Journal on Computing*, 36(5):1472–1493, 2007. Earlier version in FOCS'04.

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

[MN05] F. Magniez and A. Nayak. Quantum complexity of testing group commutativity. In *Proc. of 32nd ICALP*, pages 1312–1324, 2005. LNCS 3580.

[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. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[NS94] N. Nisan and M. Szegedy. On the degree of Boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994. Earlier version in STOC'92.

[Pat92] R. Paturi. On the degree of polynomials that approximate symmetric Boolean functions. In *Proc. of 24th ACM STOC*, pages 468–474, 1992.

[Raz03] A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya of the Russian Academy of Science, mathematics*, 67(1):159–176, 2003. English version in quant-ph/0204025.

[Riv90] T. J. Rivlin. *Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory*. Wiley-Interscience, second edition, 1990.

[Sha01] R. Shaltiel. Towards proving strong direct product theorems. In *Proc. of 16th IEEE Complexity*, pages 107–119, 2001.

[Sho94] P. W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proc. of 35th IEEE FOCS*, pages 124–134, 1994.

[Sim97] D. R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. Earlier version in FOCS'94.

[Špa07] R. Špalek. The multiplicative quantum adversary. quant-ph/0703237, 2007.

[ŠS06] R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2(1):1–18, 2006. Earlier version in ICALP'05.

[Zha05] S. Zhang. On the power of Ambainis's lower bounds. *Theoretical Computer Science*, 339(2–3):241–256, 2005. Earlier version in ICALP'04.