QUANTUM TIME-SPACE TRADEOFFS FOR MATRIX PROBLEMS*

PAUL BEAME†, NIELS KORNERUP‡, AND MICHAEL WHITMEYER§

Abstract. We consider the time and space required for quantum computers to solve a wide variety of problems involving matrices, many of which have only been analyzed classically in prior work. Our main results show that for a range of linear algebra problems—including matrix-vector product, matrix inversion, matrix multiplication and powering—existing classical time-space tradeoffs, several of which are tight for every space bound, also apply to quantum algorithms with at most a constant factor loss. For example, for almost all fixed matrices A, including the discrete Fourier transform (DFT) matrix, we prove that quantum circuits with at most T input queries and S qubits of memory require $T = \Omega(n^2/S)$ to compute matrix-vector product Ax for $x \in \{0,1\}^n$. We similarly prove that matrix multiplication for $n \times n$ binary matrices requires $T = \Omega(n^3/\sqrt{S})$. Because many of our lower bounds are matched by deterministic algorithms with the same time and space complexity, our results show that quantum computers cannot provide any asymptotic advantage for these problems with any space bound.

We obtain matching lower bounds for the stronger notion of quantum cumulative memory complexity—the sum of the space per layer of a circuit.

We also consider Boolean (i.e. AND-OR) matrix multiplication and matrix-vector products, improving the previous quantum time-space tradeoff lower bounds for $n \times n$ Boolean matrix multiplication to $T = \Omega(n^{2.5}/S^{1/4})$ from $T = \Omega(n^{2.5}/S^{1/2})$.

Our improved lower bound for Boolean matrix multiplication is based on a new coloring argument that extracts more from the strong direct product theorem that was the basis for prior work. To obtain our tight lower bounds for linear algebra problems, we require much stronger bounds than strong direct product theorems. We obtain these bounds by adding a new bucketing method to the quantum recording-query technique of Zhandry that lets us apply classical arguments to upper bound the success probability of quantum circuits.

Key words. time-space tradeoffs, quantum query complexity, lower bounds

MSC codes. 68Q12, 68Q17, 68Q25

1. Introduction.

Matrix computations are among the most fundamental computational problems and are critically important in areas such as numerical and scientific computing, optimization, and machine learning. If quantum computers can be shown to have a significant advantage over classical computations for these types of problems then it would open up a wide range of applications for such devices.

Prior work has shown that non-standard versions of matrix problems may indeed admit exponential or large polynomial quantum advantage: For any efficiently implementable operator M, the HHL algorithm of Harrow, Hassidim, and Lloyd [28] (with the improvements of [21]) can efficiently ϵ-approximate the value of $x^\dagger Mx$ for the solution x of a well-conditioned linear system. However, it is important to note that this algorithm requires the input to be presented in an unconventional format.

*Submitted to the editors 11/15/2024. A preliminary version of these results appeared in the Proceedings of the 56th ACM Symposium on Theory of Computing (STOC 2024) [13]. This version is substantially revised and expanded.

Funding: Paul Beame's research was supported by NSF grants CCF-2006359 and CCF-2422205. Michael Whitmeyer's research was supported by NSF grants CCF-2006359, CCF-2422205, and Simons Foundation grant 928589. Niels Kornerup's research was supported by a Schmidt Sciences Polymath award to David Soloveichik and the LDRD Program at Sandia National Laboratories.

†Computer Science & Engineering, University of Washington, Seattle, WA (beame@cs.washington.edu, https://homes.cs.washington.edu/~beame/site/).

‡Sandia National Laboratories, Albuquerque, NM (nielskornerup@utexas.edu, https://nielskornerup.github.io).

§Computer Science & Engineering, University of Washington, Seattle, WA (mdwhit@cs.washington.edu, https://mwhitmeyer.github.io/).

Many extensions of the HHL algorithm have also been proposed that can be elegantly described in the quantum singular value transform (qSVT) framework first described in [33] and popularized by [24]. Despite initial hope of exponential speed-up, a series of papers by Tang and co-authors, and others (e.g. [44, 19, 20, 23, 8, 18]) has shown that, by providing classical algorithms a comparable input format to the HHL algorithm, these quantum algorithms can be replaced by classical ones with only a polynomial blowup in the running time, although this polynomial is not always small.

This body of work still begs the question: What is the conventional quantum complexity of standard classical problems like explicitly computing linear-system solutions, multiplying or inverting matrices, computing matrix-vector products, and computing the low rank approximation of a matrix?

By the polynomial method, we know that computing a single inner product (or parity) of n-bit vectors requires $\Omega(n)$ quantum queries [9], but linear algebra computations generally involve $\Omega(n)$ or $\Omega(n^2)$ such computations. Sherstov [40], generalizing results of Klauck, Špalek, and de Wolf [31] for the OR function, gave a strong direct product lower bound for quantum query complexity proved using the polynomial method, which yields strong lower bounds for inner products involving many *disjoint* input vectors. However, the matrix problems in linear algebra are very far from direct product problems: The vectors involved are highly correlated with each other, so this prior work does not shed light on the key question of whether quantum algorithms provide any advantage for general linear algebra.

In this paper, we resolve these questions for quantum computation of a wide array of linear algebra problems, proving lower bounds for quantum computation that are asymptotically the same as the best classical lower bounds. Since many of the problems also have deterministic algorithms whose resource usage matches the lower bounds, our results show that there is provably no asymptotic quantum advantage at all in solving these linear algebra problems!

As with the study of classical computation involving super-linear time lower bounds, we consider quantum algorithms in which we limit the number of qubits of memory and hence produce quantum time-space tradeoffs. That is, for each fixed bound on the amount of memory allowed, we derive asymptotically the same time lower bound for the quantum algorithm as one would get for the time lower bound on classical algorithms with the same number of classical bits. In many ways, quantum memory is an even more critical resource than classical memory since it is a measure of the maximum number of qubits that maintain coherence at any time during the algorithm's execution. For this reason the first general-purpose fault-tolerant quantum computers will likely have very limited memory and only be able to execute low depth quantum circuits. As such, it is crucial to consider both the time and space complexity for quantum algorithms.

We prove our lower bounds for quantum computation in a query model where algorithms are able to perform arbitrary input-independent unitary transformations on their state between quantum queries to their input. This is a sufficiently general model that our lower bounds also apply to any reasonable model of quantum computation—including quantum circuits where the (classical) input is stored in quantum-readable read only memory (QROM).

The keys to proving our time-space tradeoffs are new results that yield much stronger lower bounds than strong direct product theorems for matrix-vector products and matrix multiplication. While our bounds have the same form as strong direct product theorems (the success probability decays exponentially with the number of outputs), they also apply with almost completely overlapping sets of inputs, in contrast

to the disjoint inputs that are necessary to apply direct product theorems.

While there is a large body of work proving strong classical time-space tradeoffs (e.g. [45, 17, 46, 16, 3, 4, 10, 34]) and a large body of work analyzing unrestricted quantum query algorithms versus their classical randomized counterparts (e.g. [22, 15, 41, 9, 6, 43, 42, 39]), there are just a few previous papers that analyze the quantum memory required to make use of these quantum queries. Klauck, Špalek, and de Wolf [31] extended the classical method of Borodin and Cook [16] for proving time-space tradeoffs to quantum circuits using a new strong direct product theorem for quantum query algorithms computing the OR function. They showed that algorithms making T quantum queries and using S qubits of quantum memory require $T = \Theta(n^{1.5}/S^{1/2})$ to sort lists of length n, and require $T = \Omega(n^{2.5}/S^{1/2})$ to compute $n \times n$ Boolean matrix product. Ambainis, Špalek, and de Wolf [7] extended this direct product approach to 2-sided error algorithms computing k-threshold functions which allowed them to produce similar trade-off lower bounds for systems of linear inequalities/equalities (though these have the drawback, unlike the other results, that the hard function for space S depends on the space bound). This approach, based on an extension of the adversary method using eigenspace analysis, was very difficult to apply.

As a result, further study of quantum time-space tradeoff lower bounds languished until it was enabled by an idea of Zhandry [47] who, motivated by understanding quantum algorithms interacting with random function oracles, developed an approach to understanding quantum query algorithms using a *compressed oracle* and Fourier analysis. This views computations in a *recording query* basis that allow one to keep track of a quantum query algorithm as a superposition of basis states that have a natural classical query interpretation. It has been applied to finding multi-way collisions [32] and to inverting a random permutation [35]. This greatly simplifies the analysis of quantum query algorithms and can be applied to many lower bound methods that use randomly chosen inputs rather than being limited to cryptographic applications.

Extending Zhandry's approach, Hamoudi and Magniez [27] applied an even cleaner expression of the method, using phase oracles with the recording query basis rather than Fourier analysis, and extended it using biased random inputs to derive query lower bounds in a regime of exponentially small success probability. They used this to obtain time-space tradeoff lower bounds, proving that any quantum algorithm that finds K disjoint collisions in an input of length n with T quantum queries and S qubits of memory must have $T = \Omega(KN^{1/3}/S^{1/3})$. They also re-derived the earlier sorting lower bound using this method.

Our linear algebra lower bounds and methods. Time-space trade-off lower bounds for linear algebraic problems were among the first to be studied for classical computation [46] after the first bounds for sorting. The strongest classical results are due to Abrahamson [4] who developed a powerful general method based on matrix rigidity. This yields state-of-the-art lower bounds for computation of Fourier transforms, convolution, matrix-vector products, matrix multiplication, matrix inversion, matrix powering, and linear system solving. The lack of any analogous results for quantum computation has been a substantial gap in our understanding[1].

Our results show that all the linear algebraic time-space tradeoff lower bounds

[1]Over a field of more than n elements, one can reduce $n \times n$ Boolean matrix multiplication to ordinary multiplication of 0-1 matrices but the lower bound is inherently too weak because in the Boolean case each output bit is a disjointness function of its inputs and hence can be computed using only $O(\sqrt{n})$ quantum queries using Grover's algorithm ([25]).

shown by Abrahamson [4] also apply to quantum computation even when the quantum circuit can adaptively decide when to produce output based on the observed input[2]. Since many of these classical lower bounds are tight, our results directly imply that there is no hybrid classical-quantum algorithms with a polynomial advantage for these problems unlike the query bounds for search and collision finding in [26]. Using the generic results in [12], we also prove asymptotically equivalent lower bounds on the stronger notion of quantum cumulative memory complexity for these problems. We include a table of our time-space tradeoff lower bounds in Table 1.

As discussed already, we need a much stronger lower bound method than any derivable from strong direct product theorems. We do this by the adding new ideas to the compressed oracle/recording query approach of Zhandry [47] as extended and applied by Magniez and Hamoudi [27]. Thus far, the compressed oracle method has used a two-step pattern: First, identify a notion of unusual progress of a quantum algorithm towards a solution (i.e., the partial information so far is more determinative of the answer than one might expect) and show that the total amplitude of states where this occurs is small, Second, show that the total amplitude of the quantum states where many outputs are produced without unusual progress can be bounded; this latter part has used ideas with classical analogs that can be applied by breaking the algorithm's final state into mutually orthogonal components, each with small amplitude on the correct answers.

However, in our case with linear algebra problems, there is no form of unusual progress and also no clear way to break up the problem into mutually orthogonal basis states. Thus, neither part of the pattern seems to work. Instead, we can use the recording query framework to characterize how much a quantum circuit can know about its input. We use the triangle inequality to bucket amplitude from the algorithm's state into a small number of non-orthogonal components (or buckets) that share some set of inputs that they know nothing about. We can then apply a classical argument showing that each component must have small amplitude on the correct answers. By finding a way to divide the state into a small number of buckets that each have small amplitude on correct answers, we can obtain tight lower bounds. The properties required of this division become more subtle as we move to the problem of matrix multiplication, where in order to get small amplitude, we need to contend with a partition featuring significantly more parts.

Improved bounds for Boolean matrix operations. Here we improve the previous lower bound for quantum algorithms computing Boolean matrix multiplication given in [31] from $T = \Omega(n^{2.5}/S^{1/2})$ to $T = \Omega(n^{2.5}/S^{1/4})$. We do this using a more sophisticated embedding of the k-fold direct product of OR functions into an arbitrary subset of k outputs of Boolean matrix multiplication. The embedding hinges on the number of colors needed for a certain kind of partial coloring of subsets E of the $n \times n$ grid. The exponents of n and S in our lower bound are optimal for the general quantum circuit model to which it applies.

Our lower bounds also lead to improving the classical lower bound tradeoff of $T = \Omega(n^3/S)$ for circuits shown in [31] to $T = \Omega(n^3/S^{1/2})$. (In these bounds, T is circuit depth and S is circuit width.) Just as with our quantum lower bound, this has optimal exponents for n and S, achieving the goal of Klauck, Špalek, and de Wolf [31] who suggested that $T^2S = \Omega(n^6)$ was a likely tight tradeoff for classical computation of Boolean matrix multiplication. It is strictly larger almost everywhere than a classical

[2]Similar to [4], our bounds apply even when input entries are chosen from an arbitrary fixed subset D of a potentially larger field.

TABLE 1

Summary of our quantum lower bounds, along with prior work. Inputs are assumed to be of length n vectors or $n \times n$ matrices. Our linear algebra bounds apply for input elements coming from any fixed subset D of a field with $d = |D|$. These are the first quantum time-space lower bounds for all of these problems other than Boolean matrix multiplication. Problems with deterministic classical query algorithms given in [29] and [4] that match our quantum query lower bounds are denoted with Θ notation instead of Ω. Constructions of the matching query algorithms can be found in section A.

Problem	Quantum Lower Bound	Source
Matrix Multiplication $f(A, B) = AB$	$T = \Theta(n^3 (\log d)^{0.5}/S^{0.5})$	Theorem 5.1
Matrix Squaring $f(A) = A^2$	$T = \Theta(n^3 (\log d)^{0.5}/S^{0.5})$	Corollary 5.5
Matrix Triple Product $f(A, B, C) = ABC$	$T = \Theta(n^4 \log d\ /S)$	Corollary 4.12
Matrix Cubing $f(A) = A^3$	$T = \Theta(n^4 \log d\ /S)$	Corollary 4.13
Matrix Inversion $f(A) = A^{-1}$	$T = \Omega(n^4 \log d\ /S)$	Corollary 4.14
System of Linear Equations $f(A, y) = A^{-1}y$	$T = \Omega(n^3 \log d\ /S)$	Corollary 4.15
Matrix-Vector Product $f(x) = Ax$	$T = \Theta(n^2 \log d\ /S)$	Theorem 4.1
Discrete Fourier Transform $f(x) = Wx$	$T = \Theta(n^2 \log d\ /S)$	Corollary 4.6
Convolution $f(u, v) = u * v$	$T = \Theta(n^2 \log d\ /S)$	Corollary 4.8
Binary Integer Multiplication	$T = \Omega(n^2/(S \log^2 n))$	Corollary 4.9
Boolean Matrix Multiplication	$T = \Omega(n^{2.5}/S^{0.5})$	[31]
$f(A, B) = A \bullet B$	$T = \Omega(n^{2.5}/S^{0.25})$	Theorem 6.5
Classical	$T = \Omega(n^3/S)$	[31, 3]
Classical	$T = \Omega(n^{3.5}/S)$ for $S \geq cn$	[3]
Classical	$T = \Theta(n^3/S^{0.5})$	Theorem 6.15
Boolean Matrix Squaring $f(A) = A \bullet A$	$T = \Omega(n^{2.5}/S^{0.25})$	Corollary 6.17

lower bound of $T = \Omega(n^3/S)$ for $S \leq n^{0.5}$ and $T = \Omega(n^{3.5}/S)$ for $S \geq n$ for Boolean matrix multiplication on branching programs (a more general model than circuits) due to Abrahamson [3] that is tight almost surely for input matrices whose entries are 1 with probability $1/\sqrt{n}$ independently.

Finally, we make a small adjustment to convert the Boolean matrix-vector lower bounds and lower bounds for systems of inequalities given in [31] and [7], respectively, so that the problems that are shown hard for space S do not depend on S.

Organization. section 2 contains a formalization of space-bounded quantum computation and its relationship to the computation of multi-output functions like the

ones we consider, an overview of the Borodin-Cook method for proving time-space tradeoff lower bounds, and a review of the recording technique for quantum query algorithms. It also discusses some of the linear algebra concepts from prior work that we will need.

sections 4 and 5 contain our lower bounds for linear algebra problems. They use two different versions of our new bucketing methods for recording-query basis states, which we discuss earlier in general terms in section 3. Though there are some brief mentions in sections 4 and 5 of the connections to the general discussion of bucketing, these sections can also be read on their own without first reading section 3. In any case, section B should be skipped on first reading since it is not related to any of the specific lower bounds in this paper.

Our lower bounds for Boolean matrix problems are in section 6 and do not use bucketing or depend on any of our methods for linear algebra. section A contains a review of the known query algorithms that match our time-space tradeoff lower bounds for quantum computation. Finally, we discuss some directions and open problems in section 7.

2. Preliminaries. We define the binary entropy function $H_2 : [0,1] \to \mathbb{R}$, by $H_2(p) = -p \log_2 p - (1-p)\log_2(1-p)$.

PROPOSITION 2.1 (Shannon). *The number of subsets of $[k]$ of size at most αk is at most $2^{H_2(\alpha)\, k}$.*

DEFINITION 2.2. *An $m \times n$ matrix is (g,h,c)-*rigid *iff every $k \times w$ submatrix where $k \leq g$ and $w \geq n - h$ has rank at least ck. We call $(g,h,1)$-rigid matrices (g,h)-rigid.*

Matrix rigidity is a robust notion of rank and is an important property for proving time-space and cumulative complexity lower bounds for linear algebra. Fortunately, Yesha gives an explicit example of such a matrix and Abrahamson proved that there are many rigid square matrices.

PROPOSITION 2.3 (Lemma 3.2 in [46]). *The $n \times n$ Discrete Fourier Transform (DFT) matrix is $(n/4, n/4, 1/2)$ rigid.*

PROPOSITION 2.4 (Lemma 4.3 in [4]). *There is a constant $\gamma \in (0, \frac{1}{2})$ such that at least a $1 - d^{-1}(2/3)^{\gamma n}$ fraction of the matrices over $D^{n \times n}$ with $|D| = d$ are $(\gamma n, \gamma n)$-rigid.*

2.1. Time space tradeoffs for multi-output functions.

Unitary quantum circuits with oracle states. Throughout this paper, we consider quantum circuits that seek to compute target functions $f : D^n \to R^m$ (or functions $f : D^n \to \mathscr{P}(R)$, where $\mathscr{P}$ is powerset, and the requirement on each input x is to output at least m elements of $f(x)$ if they exist). Let $d = |D|$ and assume the existence of some canonical bijective map $\nu : D \to \{0, \ldots, d-1\}$ that gives us an ordering on the elements of D. A T-query quantum circuit $\mathcal{C}$ is specified using input independent unitaries $U_0, \ldots, U_T$. These unitaries define a sequence of quantum states $|\psi_1\rangle_{\mathcal{C}}, \ldots |\psi_T\rangle_{\mathcal{C}}$ that an algorithm enters during its execution. When it is ambiguous, we use the subscript $\mathcal{C}$ to denote the partial trace of $|\psi_t\rangle$ that keeps only the qubits involved in the state of the query algorithm. Note that even though $|\psi_t\rangle$ is always a pure state, $|\psi_t\rangle_{\mathcal{C}}$ is often a mixed state. We can think of each of these states $|\psi_t\rangle_{\mathcal{C}}$ as a linear combination of basis vectors $|i,p,w\rangle$ where i represents an index to query, p represents a phase for the query, and w contains all the remaining qubits of the state.

Similar to [6, 47, 27], we define a general oracle operator $\mathcal{O}$ that interacts with an input register that starts in a state $|\psi_0\rangle_{\mathcal{O}}$. When it is ambiguous, we use the subscript

$\mathcal{O}$ to denote the partial trace of $|\psi_t\rangle$ that keeps only the qubits involved in the state of the oracle containing the input. Given a distribution $\mathcal{D}$ over D^n, we can make $|\psi_0\rangle_{\mathcal{O}} = \sum_{X \in D^n} \sqrt{\Pr_{X' \sim \mathcal{D}}[X' = X]}\,|X\rangle$ to represent an input sampled from $\mathcal{D}$. We define our oracle operator $\mathcal{O}$ as

$$\mathcal{O}\,|i,p,w\rangle\,|X\rangle = \omega_d^{x_{ip}}\,|i,p,w\rangle\,|X\rangle\,.$$

Thus the joint state of the input and quantum circuit at the end of the computation is given by $|\psi_T\rangle = U_T\mathcal{O}\ldots\mathcal{O}U_0\,|\psi_0\rangle$ where $|\psi_0\rangle = |0\rangle_{\mathcal{C}} \otimes |\psi_0\rangle_{\mathcal{O}}$.

The output of the quantum circuit is determined by measuring the work register of $|\psi_T\rangle_{\mathcal{C}}$ in the standard basis and applying some input-independent post-processing function q to interpret the result as an output $\tau \in R^J$ where $J \subseteq [m]$. The correctness of these output values is then determined by measuring the input registers in the standard basis to obtain the input X and evaluating whether τ is consistent with $f(X)$, which we denote by writing $\tau \| f(X)$. In general we can define the projector Π_k where:

$$\Pi_k = \sum_{\substack{i,p,w,x_1,\ldots,x_n \\ \text{s.t. } q(w)\|f(x_1,\ldots,x_n) \\ \text{and } |q(w)|\geq k}} |i,p,w,x_1,\ldots,x_n\rangle\,\langle i,p,w,x_1,\ldots,x_n| \tag{2.1}$$

The probability that the circuit produces a correct partial assignment of at least k output values is given by $\|\Pi_k\,|\psi_T\rangle\|^2$. For a given partial assignment $q(w)$ to some outputs, we can define $\Pi_{q(w)}$ to be the projection onto the values of $|X\rangle$ where $q(w)\|f(X)$. More specifically we have that:

$$\Pi_{q(w)} = \sum_{\substack{x_1,\ldots,x_n \\ \text{s.t. } q(w)\|f(x_1,\ldots,x_n)}} |x_1,\ldots,x_n\rangle\,\langle x_1,\ldots,x_n| \tag{2.2}$$

By construction when q always produces a partial assignment of at least k elements we have that $\Pi_k = \sum_{i,p,w} |i,p,w\rangle\,\langle i,p,w| \otimes \Pi_{q(w)}$.

Space-bounded quantum computation. As described above, we think of space-bounded quantum circuits as starting in the all $|0\rangle$ state and cycling between applying input queries $\mathcal{O}$, and arbitrary input-independent computation U_t. Unlike in the unitary circuit model, we allow our space-bounded quantum circuits to make intermediate measurements after applying each U_t as shown in Figure 1. Adopting the notation of [12], we will consider the set of consecutive $\mathcal{O}$, U_t and measurement gates as layer L_t. As was done in [27], we will assume that the quantum query circuit has a dedicated register containing a boolean flag and a potential output $(i, y_i) \in [m] \times R$. After each query $\mathcal{O}$ and subsequent unitary operation U_t, the flag register is measured in the standard basis. Should the outcome 1 be obtained, the output register is measured in the standard basis and interpreted as an output pair (i, y_i) which is written to a write-only tape. Otherwise, the circuit produces no output during this layer. The space of layer L_t is the number of qubits that are passed from layer L_t to L_{t+1} and is denoted S_t. We define the space of a circuit as the maximum space of any layer, the time as the total number of layers, and the cumulative memory as the sum over all the S_t. Thus the space needed to store the input and output is not included in this model.

Intermediate measurements enable circuits to produce parts of their output early and discard unnecessary ancillary qubits. Similar to the disjoint collisions bound in [27], our results in sections 4 and 5 apply to quantum circuits without any required

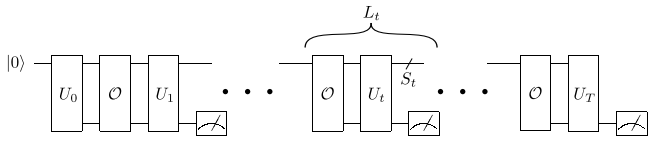

FIG. 1. *A general quantum circuit with T queries.*

structure on their output order. Thus, as long as the circuits produce the correct output value for each index i, they may do so during arbitrary layers of the circuit that may depend on the chosen input. However, as was the case in [31], our results for quantum Boolean matrix multiplication in section 6 apply to a more restricted model of computation where the choice of when to produce each output value is independent of the input. In this *output-oblivious* model, quantum circuits do not have a flag register. Instead, on predefined layers the quantum circuit measures the output register in the standard basis and interprets the result as an element of R corresponding to a fixed output index. This output-oblivious ordering restricts the set of allowed algorithms and is necessary to prove our key lemmas associated with Boolean matrix multiplication.

Space-bounded classical computation. One can view our classical lower bounds in section 6 as applying to layered *branching programs* [16] where the space bound corresponds to the logarithm of the width of the program and the time corresponds to the number of layers. Output in a branching program is produced along the edges and written to a write-only output tape. Thus the space bound of a classical computation only considers the S bits of internal state maintained by the device and not the size of its read-only input or write-only output. Our results for classical Boolean matrix multiplication in section 6 apply to an output-oblivious model, which corresponds to branching programs that must produce outputs for the same output index regardless of which edge is taken between two layers.

The Borodin-Cook method. The Borodin-Cook method provides a general framework for proving time-space tradeoff lower bounds for multi-output problems, those for which every input vector in D^n is associated with some fixed set of possible output values from set R and the objective is to output at least m of these output values. As discussed earlier these can be functions $f : D^n \to R^m$, or $f : D^n \to \mathcal{P}(R)$ where the requirement is to produce at least m elements of $f(D^n)$, if they exist[3].

The property of the function f that enables the Borodin-Cook method to be used is the following[4] for some well-behaved function $h(k, n)$:

(*) Let $c = c(D) > 1$. Any classical query algorithm that makes at most $t \leq h = h(k, n)$ queries for an input distribution $\mathcal{D}$ on D^n, correctly produces k correct output values of f with probability at most c^{-k}.

With this property, Borodin and Cook showed that one directly obtains a classical time-space tradeoff for computing f of the form $T \cdot S = \Omega(m\, h(S/(\log c), n)\, \log c)$ for time T that is $n^{O(1)}$ and space S as follows:

Proof sketch. Choose k with $\log n \leq k \leq m$ such that $2^S \cdot T \cdot c^{-k} < 1$; then k is roughly $S/(\log c)$.

Divide the T query steps into disjoint blocks of $h = h(k, n)$ queries each and

[3]There is a more general version where the query algorithm is only required to produce these m outputs with some sufficiently high probability but we focus on the simpler form

[4]We do not specify an upper limit on the possible $k \leq m$ in this informal statement. The exact range for which it holds will impact the space bounds for which the tradeoff holds.

assume that T is a multiple of h, without loss of generality. Since m outputs must be produced on all inputs in D^n and there are T/h blocks, for $T < mh/k$, which is $\Theta(m\,h \log c\ /S)$, for every execution on every input there must some block where at least k correct outputs are produced.

However, since the space is at most S there are at most 2^S configurations of the states that the algorithm could have been in at the beginning of each time block. Since **(*)** says that any fixed block can produce at least k output values correctly with probability at most c^{-k} under $\mathcal{D}$, by a union bound the total probability that some fixed block produces at least k correct output values is at most $2^S c^{-k} < 1/T$ by our choice of k. Since there are only T/h blocks, the probability that there is one of them that produces k correct answers is < 1.

Therefore T must be $\Omega(mh \log c\ /S)$ as required. □

For quantum algorithms, Klauck et al. [31] observed that one could use a result by Aaronson in place of the union bound over the 2^S classical state configurations at the start of each block in the Borodin-Cook method.

PROPOSITION 2.5 ([1]). *Let $\mathcal{C}$ be a quantum circuit, ρ be an S-qubit (possibly mixed) state, and π_{mix} be the S-qubit maximally mixed state. If $\mathcal{C}$ starting in initial state ρ produces some output z with probability p, then $\mathcal{C}$ starting in state π_{mix} will produce z with probability q which is at least $p/2^S$.*

We include a stand-alone derivation here for completeness.

Proof. Without loss of generality we can assume $\mathcal{C}$ performs no measurements until the end of the circuit. Thus we can think of $\mathcal{C}$ as representing a unitary operator U. Let Π_z be the projection onto output states of $\mathcal{C}$ that cause the circuit to output the value z. Then $p_z = \mathrm{Tr}[\Pi_z U \rho U^\dagger]$. By the spectral decomposition theorem we can represent ρ as a convex combination of some set of orthogonal pure states $\rho = \sum_{i\in[2^S]} \lambda_i \left|\varphi_i\right\rangle \left\langle\varphi_i\right|$. Since the maximally mixed state can be represented as $\pi_{\mathrm{mix}} = \sum_{i\in[2^S]} (1/2^S) \left|\varphi_i\right\rangle \left\langle\varphi_i\right|$ we have that:

$$\begin{aligned}
q &= \mathrm{Tr}\left[\Pi_z U \pi_{\mathrm{mix}} U^\dagger\right] \\
&= \mathrm{Tr}\left[\Pi_z U \left(\sum_{i\in[2^S]} \frac{1}{2^S} \left|\varphi_i\right\rangle \left\langle\varphi_i\right|\right) U^\dagger\right] \\
&= \frac{1}{2^S} \mathrm{Tr}\left[\sum_{i\in 2^S} \left\langle\varphi_i\right| U^\dagger \Pi_z U \left|\varphi_i\right\rangle\right] \\
&\geq \frac{1}{2^S} \mathrm{Tr}\left[\sum_{i\in 2^S} \lambda_i \left\langle\varphi_i\right| U^\dagger \Pi_z U \left|\varphi_i\right\rangle\right] \\
&= \frac{1}{2^S} \mathrm{Tr}\left[\Pi_z U \left(\sum_{i\in[2^S]} \lambda_i \left|\varphi_i\right\rangle \left\langle\varphi_i\right|\right) U^\dagger\right] \\
&= \frac{1}{2^S} \mathrm{Tr}\left[\Pi_z U \rho U^\dagger\right] = p/2^S
\end{aligned}$$

Where the inequality comes from the fact that $\left\langle\varphi\right| U^\dagger \Pi_z U \left|\varphi\right\rangle \geq 0$ for any state $\left|\varphi\right\rangle$. □

With this they showed that essentially the same paradigm could be used to give similar time-space tradeoff lower bounds for quantum algorithms if one can prove a quantum analog of **(*)**. One subtlety that arises from the quantum version of the Borodin-Cook method is that often the quantum version of **(*)** is proven in a

non-space-bounded unitary circuit model without intermediate measurements. By using the deferred measurement principle, we can see that lower bounds on the success probability of short quantum circuits in this model imply equally tight lower bounds in the space-bounded model where we directly apply the Borodin-Cook method.

2.2. The quantum recording query technique. Here we review the methods developed in [47, 27] that allow us to analyze what a quantum circuit learns about its input by making quantum queries. We will assume that the input state $|\psi_0\rangle_{\mathcal{O}}$ is the equal superposition state over all inputs, although [47, 27, 35] generalize this method to other input distributions. We can exchange the general query operator $\mathcal{O}$ for the uniform input distribution with a recording query operator $\mathcal{R}$ that we define as follows:

DEFINITION 2.6 (adapted from [27]). *Let D be the input alphabet, $d = |D|$, and ν be our choice of canonical bijection between D and $\{0, \ldots, d-1\}$. We define $\mathcal{S}_1$ to be the unitary operator that maps*

$$\mathcal{S}_1 : \begin{cases} |\perp\rangle & \longrightarrow \frac{1}{\sqrt{d}} \sum_{y \in D} |y\rangle \\ \frac{1}{\sqrt{d}} \sum_{y \in D} |y\rangle & \longrightarrow |\perp\rangle \\ \frac{1}{\sqrt{d}} \sum_{y \in D} \omega_d^{p\,\nu(y)} |y\rangle & \longrightarrow \frac{1}{\sqrt{d}} \sum_{y \in D} \omega_d^{p\,\nu(y)} |y\rangle \;\; \forall p \in \{1, \ldots, d-1\}. \end{cases}$$

Let $\mathcal{S} = (I)_{i,p,w} \otimes (\mathcal{S}_1^{\otimes n})_{x_1,\ldots,x_n}$ and $\mathcal{O}$ be the standard oracle operator that maps the basis state

$$|i, p, w, x_1, \ldots, x_n\rangle \longrightarrow \omega_d^{p\,\nu(x_i)} |i, p, w, x_1, \ldots, x_n\rangle .$$

Then the recording query oracle operator *$\mathcal{R}$ is defined as $\mathcal{SOS}$.*

$\mathcal{S}_1$ introduces $\perp$ as a new value for the input registers. Intuitively, the $\perp$ symbol indicates that the algorithm does not know anything about that register of the oracle. Hence by adding and correctly manipulating the $\perp$ symbols in the oracle's registers, we can record what the algorithm knows about the input. Since $\mathcal{S}^2 = I$, we can exactly characterize how the states of quantum circuits with oracles $\mathcal{O}$ and $\mathcal{R}$ relate to one another.

PROPOSITION 2.7 (Theorem 3.3 in [27]). *Let $\mathcal{C}$ be a quantum circuit that for each $j \leq t$ applies unitary U_j after the j-th query. Let $\mathcal{S}$ be the unitary operation and $\mathcal{R}$ be the recording query oracle from Definition* 2.6*. Let*

$$|\psi_t\rangle = U_t \mathcal{O} U_{t-1} \ldots U_1 \mathcal{O} U_0 \left(|0\rangle_{i,p,w} \otimes \frac{1}{d^{n/2}} \sum_{x_1,\ldots,x_n \in D} |x_1, \ldots, x_n\rangle_{x_1,\ldots,x_n} \right)$$

$$|\phi_t\rangle = U_t \mathcal{R} U_{t-1} \ldots U_1 \mathcal{R} U_0 \left(|0\rangle_{i,p,w} \otimes |\perp\rangle_{x_1,\ldots,x_n} \right)$$

be the states of $\mathcal{C}$ with oracle $\mathcal{O}$ or $\mathcal{R}$ respectively. Then $|\psi_t\rangle = \mathcal{S} |\phi_t\rangle$.

In other words, it is impossible to distinguish the final state $|\psi_T\rangle$ of a circuit with standard oracle $\mathcal{O}$ from the output with recording oracle $\mathcal{R}$ if we apply $\mathcal{S}$ to the registers of $\mathcal{R}$ after the final query. Thus we can conclude that the success probability of a quantum circuit with T queries producing a partial assignment of k correct output values is given by $\|\Pi_k |\psi_T\rangle\|^2 = \|\Pi_k \mathcal{S} |\phi_T\rangle\|^2$. Note that while $|\phi_T\rangle$ may have inputs in the $\perp$ state, Proposition 2.7 tells us that $\mathcal{S} |\phi_T\rangle$ will never have an input in the $\perp$ state. This means that when considering recording query oracles, it is safe to keep our current definitions of Π_k and $\Pi_{q(w)}$ which will always project out any basis state

where an input is assigned to $\perp$. We will leverage the following property of $|\phi_T\rangle$ to bound the success probability of quantum circuits with at most T queries.

DEFINITION 2.8. *Let Γ_t be the set of all elements $(D \cup \{\perp\})^n$ with at most t non-$\perp$ elements. This is the set of indices for all recording query basis states associated with quantum algorithms that make at most t queries.*

PROPOSITION 2.9 (Fact 3.2 in [27]). *The state $|\phi_t\rangle$ from Proposition 2.7 is a linear combination of basis states $|i, p, w, x_1, \ldots, x_n\rangle$ where $(x_1, \ldots, x_n) \in \Gamma_t$.*

For the bounds in [27] it is essential to bound how the state of $|\phi\rangle_{\mathcal{O}}$ can change after each query. For our use of the recording query technique, this detailed analysis is not necessary. Nevertheless, we state the following proposition here for completeness.

PROPOSITION 2.10 (Lemma 4.1 in [27]). *Let D be the input alphabet, $d = |D|$, and ν be our choice of canonical bijection between D and $\{0, \ldots, d-1\}$. If the recording query operator $\mathcal{R}$ is applied to a basis state $|i, p, w, x_1, \ldots, x_n\rangle$ where $p \neq 0$ then the register $|x_i\rangle$ is mapped to*

$$\begin{cases} \sum_{y \in D} \frac{\omega_d^{p\,\nu(y)}}{\sqrt{d}} |y\rangle & \text{if } x_i = \perp \\ (1 - \frac{2}{d})\, \omega_d^{p\,\nu(x_i)} |x_i\rangle + \frac{1}{d} |x_i\rangle + \frac{\omega_d^{p\,\nu(x_i)}}{\sqrt{d}} |\perp\rangle + \sum_{y \in D \setminus \{x_i\}} \frac{1 - \omega_d^{p\,\nu(y)} - \omega_d^{p\,\nu(x_i)}}{d} |y\rangle & \text{otherwise.} \end{cases}$$

If $p = 0$ then the register remains unchanged.

3. Our bucketing methods.

The Borodin-Cook method with recording queries. Paraphrasing **(*)** from our earlier description of the Borodin-Cook method, to derive a time-space tradeoff lower bound for a function $f : D^n \to R^m$ or $f : D^n \to \mathcal{P}(R)$, we need to prove that any quantum query algorithm making at most some $h(k, n)$ queries can correctly produce at least k of the m output values only with a probability that decays exponentially in k (over the random choice of the input and the quantum measurements). In the recording query method, both the input and the state of the quantum algorithm are encoded in quantum states where measuring the local state of the algorithm determines the produced outputs (both indices and values) and measuring the local state of the input determines the classical input to the problem instance. Which k output positions are produced may depend on the input, so the paradigm proceeds by fixing both the quantum query algorithm and the k output values produced, and arguing that those k output values are correct for a fraction of the amplitude of the input state that is exponentially small in k.

Before discussing our bucketing method to do this, we first give some more detail about the method of Hamoudi and Magniez [27], as expressed in their lower bound for the m-disjoint collision problem. The method of Hamoudi and Magniez operates in two parts. They show that

- for any quantum query algorithm (making at most $\varepsilon k\sqrt{n}$ queries), only an exponentially small fraction in k of the total amplitude of the input is on recording query basis states with at least $k/2$ disjoint collisions explicitly represented in the state (and hence for which at least $k/2$ outputs would automatically be correct), and
- for any fixed partial assignment of k output values (disjoint collisions), the fraction of the total amplitude on recording query basis states that do not explicitly represent at least $k/2$ of those output values as collisions on which all k output values are correct is exponentially small in k.

The first part has most of the quantum flavor of the argument since the growth in the number of disjoint collisions observed is much faster in the quantum case than in the corresponding classical case. Because the k-disjoint collisions problem involves explicit local properties of the input, the second part involves many orthogonal components and hence is a rather straightforward adaptation of a classical argument, with a Cauchy-Schwartz calculation replacing a union bound.

In all of the matrix problems we consider, correctness of the output values depends on the input values in highly non-local ways that do not yield the kind of orthogonality properties that Hamoudi and Magniez are able to exploit. There also is no analog of the kind of progress argument from the first part available. We have to work simply from a bound on the total number of queries that the algorithm makes. To handle this we introduce a bucketing approach.

3.1. Bucketing. Throughout this section we assume a fixed function f defined on D^n with m output values; all of our definitions are implicitly defined relative to this fixed function. The bucketing processes we define apply to the state of a quantum query algorithm after it has made t queries to a recording query oracle. By Proposition 2.9, such a state is a linear combination of basis states $|i, p, w, x\rangle$ with $x \in \Gamma_t$. Then for any partial assignment q of k output values that the query algorithm could have produced, we wish to prove that the fraction of the total amplitude on which the recording query input leads to an output that agrees with q is exponentially small in k.

DEFINITION 3.1. *Let q be a partial assignment of k output values and Π_q be the projector defined in* (2.2) *for this partial assignment. For $c > 1$, we define a* c-admissible bucket *B for q to be a subset of $(D \cup \{\perp\})^n$ with the property that* any *quantum state over the inputs that is spanned by the elements of B (i.e. $|\phi\rangle = \sum_{x \in B} \alpha_x |x\rangle$) satisfies $\|\Pi_q \mathcal{S} |\phi\rangle\|^2 \leq c^{-k}$.*

In our definitions of admissible buckets, the exponentially small bound will follow from the fact that for some fixed set of a $k' \geq c'k$ of the k output values, any state spanned by the elements of B will have a squared amplitude for those k' outputs of q being correct of exactly $d^{-k'}$; i.e., that of a completely random guess. In this case, $c = d^{c'}$.

DEFINITION 3.2. *Let A be a subset of $(D \cup \{\perp\})^n$. Then $\Pi_A = \sum_{x \in A} |x\rangle \langle x|$. The* total amplitude *of a state $|\phi_t\rangle$ on recording query basis states in A is $\|\Pi_A |\phi_t\rangle\|$.*

In subsection 2.1 we defined projectors Π_k where $k \in \mathbb{N}$ and $\Pi_q(w)$ where $q(w) \in R^J$ for $J \subseteq [m]$ as ways to project onto basis states associated with the quantum circuit producing k correct output values or associated with a assignment $q(w)$ being correct for the value in the input register. Here, we use projectors Π_A to keep track of the contributions associated with various sets of recording query basis states in the analysis of our bucketing methods.

DEFINITION 3.3. *A* t-family of c-admissible buckets *with size ℓ for a partial assignment q of k output values is a collection $\mathcal{B}$ of subsets of $(D \cup \{\perp\})^n$ such that*

- *$|\mathcal{B}| \leq \ell$,*
- *each $B \in \mathcal{B}$ is a c-admissible bucket for q, and*
- *every element of Γ_t is in at least one c-admissible bucket $B \in \mathcal{B}$.*

The simple version of bucketing recording queries that we use to prove our lower bound for matrix-vector products works by showing that for $t \leq h(k, n)$ and each partial assignment of k output values q, there is a t-family of c-admissible buckets

$\mathcal{B}$ whose size is not too large. The amplitudes of the recording query basis states indexed by Γ_t can be partitioned arbitrarily by assigning each element of Γ_t to some c-admissible bucket in the family that contains it. We obtain that the total squared amplitude associated with successfully producing output q is at most $|\mathcal{B}|^2/c^k$ for $c > 1$. For matrix-vector product with suitable fixed matrices, we are able to show in this case that $|\mathcal{B}|^2$ is at most b^k for some $b < c$ and hence obtain an upper bound on the overall success probability that is exponentially small in k.

In the case of matrix multiplication, the families of admissible buckets we can produce are much too large. The basic approach above does not tailor the choice of buckets to the specific final state of the recording query input. We will instead need to produce an association of basis states with buckets that depends on the final state of the recording query input.

DEFINITION 3.4. *A* weighted t-scheme of c-admissible buckets *with total weight w for a partial assignment q of k output values is a mapping that takes any quantum state $|\phi_t\rangle$ defined over recording query basis state indexed by Γ_t to a t-family of c-admissible buckets $\mathcal{B}$ and an assignment of weights $w_B \in [0,1]$ to sets $B \in \mathcal{B}$ such that*

- *for every $B \in \mathcal{B}$ we have $\|\Pi_B |\phi_t\rangle\| \leq w_B$ and*
- *$\sum_{B\in\mathcal{B}} w_B \leq w$.*

When we want to emphasize the dependence of $\mathcal{B}$ on $|\phi_t\rangle$ we write it as $\mathcal{B}_{|\phi_t\rangle}$.

A t-family of c-admissible buckets $\mathcal{B}'$ with size ℓ can always be interpreted as a weighted t-scheme of c-admissible buckets with total weight ℓ by always setting $\mathcal{B}_{|\phi_t\rangle} = \mathcal{B}'$ and weight $w_B = 1$ for each $B \in \mathcal{B}'$.

LEMMA 3.5. *Let q be a partial assignment of k output values and Π_q be the projector defined in* (2.2) *for this partial assignment. If there is a weighted t-scheme of c-admissible buckets with total weight w for q then there is a constant $c > 0$ such that for any quantum state $|\phi_t\rangle$ defined over recording query basis states indexed by Γ_t, $\|\Pi_q \mathcal{S} |\phi_t\rangle\|^2 \leq w^2 \cdot c^{-k}$.*

Proof. Let $|\phi_t\rangle = \sum_{x\in\Gamma_t} \alpha_x |x\rangle$ be a quantum state defined over recording query basis elements indexed by Γ_t. Let $\mathcal{B}_{|\phi_t\rangle}$ with associated weights w_B for $B \in \mathcal{B}_{|\phi_t\rangle}$ be given by the weighted t-scheme of c-admissible buckets for state $|\phi_t\rangle$. For $B \in \mathcal{B}_{|\phi_t\rangle}$, by definition, we have $\Pi_B |\phi_t\rangle = \sum_{x\in B} \alpha_x |x\rangle$ and $\|\sum_{x\in B} \alpha_x |x\rangle\| = \|\Pi_B |\phi_t\rangle\| \leq w_B$. Then, since every $x \in \Gamma_t$ is contained in some $B \in \mathcal{B}_{|\phi_t\rangle}$, we have

$$\begin{aligned}
\|\Pi_q \mathcal{S} |\phi_t\rangle\|^2 &\leq \|\Pi_q \mathcal{S} \sum_{B\in\mathcal{B}_{|\phi_t\rangle}} \Pi_B |\phi_t\rangle\|^2 \\
&= \|\Pi_q \mathcal{S} \sum_{B\in\mathcal{B}_{|\phi_t\rangle}} \sum_{x\in B} \alpha_x |x\rangle\|^2 \\
&\leq \Big(\sum_{B\in\mathcal{B}_{|\phi_t\rangle}} \|\Pi_q \mathcal{S} \sum_{x\in B} \alpha_x |x\rangle\| \Big)^2 \\
&= \Big(\sum_{B\in\mathcal{B}_{|\phi_t\rangle}} w_B \|\Pi_q \mathcal{S} \sum_{x\in B} \frac{\alpha_x}{w_B} |x\rangle\| \Big)^2 .
\end{aligned}$$

By definition, $\sum_{x\in B} \frac{\alpha_x}{w_B} |x\rangle$ is a vector whose 2-norm is at most 1 and the fact that

B is a c-admissible bucket for q, implies that

$$\left(\sum_{B \in \mathcal{B}_{|\phi_t\rangle}} w_B \|\Pi_q \mathcal{S} \sum_{x \in B} \frac{\alpha_x}{w_B} |x\rangle\| \right)^2 \leq \left(\sum_{B \in \mathcal{B}_{|\phi_t\rangle}} w_B \cdot c^{-k/2} \right)^2 = w^2 \cdot c^{-k}$$

which yields our claimed bound on $\|\Pi_q \mathcal{S} |\phi_t\rangle\|^2$. □

Note that setting each w_B to 1 regardless of the quantum state gives us that $\|\Pi_q \mathcal{S} |\phi_t\rangle\| \leq |\mathcal{B}|^2 c^{-k}$, which is the simpler bound we use for the matrix-vector product case. Though weighted schemes are much more flexible than such simple families and can yield bounds where those may not, choosing good weights presents an additional challenge. The follow concept will allow us to define these weights implicitly and is what we use when we analyze matrix product algorithms.

DEFINITION 3.6. *A t-reduction scheme of c-admissible buckets with size ℓ for a partial assignment q of k output values is a mapping that takes any quantum state that involves a combination of recording query basis elements indexed by Γ_t, $|\phi_t\rangle = \sum_{z \in \Gamma_t} \delta_Z |z\rangle$, and outputs a collection of buckets $\mathcal{B}$ and a subset $\Gamma'_t \subseteq \Gamma_t$ such that*

- *$|\mathcal{B}| \leq \ell$,*
- *each $B \in \mathcal{B}$ is a c-admissible bucket for q,*
- *every element of Γ'_t is in at least one c-admissible bucket in $\mathcal{B}$, and*
- *the total amplitude of $|\phi_t\rangle$ on recording query basis states in $\Gamma_t \setminus \Gamma'_t$ is at most $1/2$. In other words, $\|\Pi_{\Gamma_t \setminus \Gamma'_t} |\phi_t\rangle\| \leq 1/2$.*

LEMMA 3.7. *The existence of a t-reduction scheme of c-admissible buckets with size ℓ implies the existence of a weighted t-scheme of c-admissible buckets with total weight 2ℓ.*

Proof. Fix q as the partial assignment to k output values and $|\phi_t\rangle$ as any input state over recording query basis states indexed by Γ_t. We apply the reduction scheme with the full state to begin with. This yields a collection of c-admissible buckets $\mathcal{B}$ of size ℓ and a set Γ'_t. Without loss of generality we can assume that $\Gamma'_t = \cup_{B \in \mathcal{B}} B$, as otherwise we could always define another t-reduction scheme of c-admissible buckets with size ℓ with this choice of Γ'_t to use instead. We assign weight 1 to all those buckets.

The remaining total amplitude of states in Γ_t is at most $1/2$. We apply the reduction scheme inductively to the state $|\phi'_t\rangle = \Pi_{\Gamma_t \setminus \Gamma'_t} |\phi_t\rangle / \|\Pi_{\Gamma_t \setminus \Gamma'_t} |\phi_t\rangle\|$ which is a renormalized state for the portion of $|\phi_t\rangle$ defined on $\Gamma_t \setminus \Gamma'_t$. This yields a new set of at most ℓ buckets, each of which we assign weight $1/2$. Each iteration of this procedure results in a renormalized state whose support is a strictly smaller subset of Γ_t. We repeat in this way until we have exhausted all of Γ_t and produced a weighted t-scheme of c-admissible buckets. The total weight of this scheme is at most $\sum_{i \geq 0} \ell/2^i \leq 2\ell$. □

COROLLARY 3.8. *Let q be a partial assignment of k output values and Π_q be the projector defined in (2.2) for this partial assignment. The existence of a t-reduction scheme of c-admissible buckets with size ℓ for q implies that for any quantum state $|\phi_t\rangle = \sum_{x \in \Gamma_t} \alpha_x |x\rangle$ there is a constant $c > 0$ such that $\|\Pi_q \mathcal{S} |\phi_t\rangle\|^2 \leq 4\ell^2 \cdot c^{-k}$.*

In sections 4 and 5 we use the above ideas to prove our lower bounds on matrix-vector products and matrix multiplication, respectively. However, it is also possible to produce reduction schemes of admissible buckets from a combinatorial property of the set of all admissible buckets without needing to reason about the amplitude of quantum states. We explore this idea further in section B.

4. Quantum matrix vector products. In this section, we consider the task of — for a fixed matrix $A \in \mathbb{F}^{m\times n}$ — computing the function $f_A(x) = Ax$ for inputs $x \in D^m$ (where D is a fixed subset of $\mathbb{F}$) using a quantum circuit. We note that this is a fundamentally harder task than is considered in many quantum machine learning papers (for example [28]) as we require the circuit to output a classical vector $y \in \mathbb{F}^n$ rather than either a quantum state encoding the entries of y in the amplitudes or an estimate of $y^\dagger My$. Also unlike many prior quantum time-space tradeoffs, including sorting [31, 27, 12] and boolean matrix multiplication [31] (and our Theorem 6.5), our matrix vector product and matrix multiplication lower bounds apply to circuits that can adaptively decide when to produce each output based on the observed inputs. Time-space lower bounds against such quantum circuits were first described in [27] for the multiple disjoint collisions problem, although they were not able to show such a result for sorting. Similar to [27] we are able to lower bound these circuits by identifying a single hard distribution over the inputs that applies to any set of outputs.

THEOREM 4.1. *Let $m \le n^r$ for some constant r and $2 \le d \le n^n$. There is a constant $C > 0$ such that the following holds: Let A be an $m\times n$ matrix over a field $\mathbb{F}$ that is (g, h, c)-rigid. Then any quantum circuit using time T and space $S < \frac{c}{6(r+6)} g \log_2 d$ that computes the function $f_A : D^n \to \mathbb{F}^m$ for $D \subseteq \mathbb{F}$ with $d = |D|$ given by $f_A(x) = Ax$ with success probability larger than 2^{-S} requires that $T \ge Cmh \log d \ /S$.*

When the fixed matrix A is sufficiently rigid, for example when both g and h are linear in n as is the case with the DFT matrix per Proposition 2.3 or a random matrix with high probability per Proposition 2.4, this lower bound becomes $\Omega(mn \log d)$ provided that S is at most some constant times $n \log d$ which is essentially a trivial constraint for the problem. This bound is tightly matched by a classical query algorithm in Proposition A.1.

This theorem follows from the following key lemma, proven in subsection 4.1, which lets us bound the number of correct output values produced by a shallow quantum circuit.

LEMMA 4.2. *Let A be any (k, h, c)-rigid $m \times n$ matrix over a finite field $\mathbb{F}$ and let $f_A : D^n \to \mathbb{F}^m$ for $D \subseteq \mathbb{F}$ be defined by $f_A(x) = Ax$. Then for $\alpha > 0$ and for input x sampled uniformly from D^n and any quantum circuit $\mathcal{C}$ with at most αh queries to x, the probability that $\mathcal{C}$ produces k correct output values of $f_A(x)$ is at most $\lceil h/(ck)\rceil^2 \, (4^{H_2(\alpha)}/|D|^{1-\alpha})^{ck}$.*

Note: For $\alpha \le 0.0737$ we have $1 - \alpha - 2H_2(\alpha) > 1/6$ and hence the bound is at most $\lceil h/(ck)\rceil^2 |D|^{-ck/6}$ for $d \ge 2$.

Proof of Theorem 4.1 *from Lemma* 4.2. Let $\mathcal{C}$ be a quantum circuit with T queries and space S that computes $f_A(x)$ with success probability larger than 2^{-S}. Since $h \le n$, $m \le n^r$ and $S \ge \log_2 n$ we only need to consider the case that $T \le n^{r+1} \log_n d \le n^{r+2}$.

Let $\alpha = 0.0737$. We partition $\mathcal{C}$ into $\lceil T/(\alpha h)\rceil$ sub-circuits that each have at most αh queries. By combining Proposition 2.5 and Lemma 4.2, we know that each sub-circuit can produce $k \le g$ correct output values with probability at most $2^S \lceil h/(ck)\rceil^2 \, d^{-ck/6} \le h^2 \, 2^S d^{-ck/6}$.

By assumption, we have $d^{-cg/6} \le 2^{-(r+6)S} \le n^{-(r+4)} 2^{-2S} \le h^{-2} 2^{-2S}/T$ since $S \ge \log_2 n$, $T \le n^{r+2}$, and $h \le n$. In particular, this implies that $h^2 d^{-cg/6} < 2^{-S}$ so we must have $T > \alpha h$ by Lemma 4.2. Set $k \le g$ to be the smallest integer such that $h^2 \, 2^S d^{-ck/6} \le 2^{-S}/T$. Then the probability that a sub-circuit produces k correct output values is at most $2^{-S}/T$. This gives $k = \lceil [6 \log_2(hT) + 12S]/(c \log_2 d)\rceil$. We note that k is at most $c^* S/\log_2 d$ for some constant $c^* > 0$ since $\log_2(hT) \le$

$(r+3)\log_2 n \leq (r+3)S$.

Taking a union bound over the sub-circuits, the probability that any of them produces k correct output values is at most 2^{-S}. Since f_A has m outputs, this means that

$$\lceil T/(\alpha h)\rceil\,(k-1) \geq m$$

Since $T \geq \alpha h$, we have

$$2Tk \geq \alpha m h.$$

Plugging in our upper bound on k we have that

$$2c^* TS/\log_2 d \geq \alpha m h$$

and hence $T \cdot S$ is at least $\frac{\alpha}{2c^*} mh \log d$ as claimed. □

4.1. Success probability of small depth quantum circuits. We first give an overview of the argument, which involves an initial uniform distribution over the inputs $x \in D^n$. This begins by decomposing the state after $t \leq \alpha h$ queries into orthogonal components based on the values of working qubits $|i, p, w\rangle$, which also determine the set of k output values produced. It then suffices to prove that, for any fixed set of k outputs, any input state spanned by recording query basis elements with at most t non-$\perp$ items can agree with the k outputs only on an exponentially small (in k) fraction of the amplitude.

If we knew which $t \leq \alpha h$ input indices were queried, as we would with classical algorithms in the analysis of [4], then things would be easy: Since the fixed matrix A is (k, h, c) rigid, the sub-matrix of A with rows corresponding to these k outputs, and with the $\geq n - \alpha h$ "unqueried" columns has rank at least ck, so any fixed output can be correct with probability at most d^{-ck} over the choice of inputs. However, the quantum state after t queries is a superposition of recording query basis states that could involve all possible subsets of $\leq t$ non-$\perp$ indices which is at least $\binom{n}{t}$ possibilities.

To handle this we use the basic version of our bucketing method for recording query basis states and find a relatively small collection of admissible buckets (whose size will be a sufficiently small exponential in k) that allows us to run the quantum analogue of the classical argument within each bucket. We now give the proof in detail.

Proof of Lemma 4.2. Let $d = |D|$. For simplicity we will assume that $q(w)$—the output as a function of the measured value of the work register—always produces k outputs.[5] Let A be a (k, h, c)-rigid matrix. By Proposition 2.9 after $t \leq \alpha h$ queries in the recording query oracle model, the state $|\phi_t\rangle$ is a linear combination of basis states $|i, p, w, x_1, \ldots, x_n\rangle$ where $(x_1, \ldots, x_n) \in \Gamma_t$. It will be useful to be more explicit in our discussion of Γ_t. Each element of Γ_t consists of an assignment $y \in D^I$ for some subset $I \subseteq [n]$ with $|I| \leq t$ and value $\perp$ on all coordinates in $[n] \setminus I$. Therefore, we can write the state as

$$|\phi_t\rangle = \sum_{\substack{i,p,w \\ I\subseteq[n],\ |I|\leq t \\ y\in D^I}} \alpha_{i,p,w,I,y}\,|i,p,w\rangle\,|y\rangle_I\,|\perp\rangle_{[n]\setminus I} \tag{4.1}$$

for some $\alpha_{i,p,w,I,y}$ with $\sum_{i,p,w,I,y} |\alpha_{i,p,w,I,y}|^2 = 1$. Thus by Proposition 2.7, the final state of the algorithm (after $t \leq \alpha h$ queries) in the non-recording query oracle setting

[5] If in general $q(w)$ produces more than k outputs, we only consider its first k outputs.

is given by

$$|\psi_t\rangle = \mathcal{S}\,|\phi_t\rangle = \mathcal{S} \sum_{\substack{i,p,w \\ I\subseteq[n],\ |I|\le t \\ y\in D^I}} \alpha_{i,p,w,I,y}\,|i,p,w\rangle\,|y\rangle_I\,|\perp\rangle_{[n]\setminus I}\,.$$

Since $\mathcal{S}$ behaves as the identity on $|\phi_t\rangle_{\mathcal{C}}$ and the $|i,p,w\rangle$ are orthogonal basis states, we can rewrite this as

$$\sum_{i,p,w} \beta_{i,p,w}\,|i,p,w\rangle \otimes \left[\mathcal{S}_1^{\otimes n} \sum_{\substack{I\subseteq[n],\ |I|\le t \\ y\in D^I}} \beta_{I,y}^{i,p,w}\,|y\rangle_I\,|\perp\rangle_{[n]\setminus I}\right]$$

for some $\beta_{i,p,w}$ and $\beta_{I,y}^{i,p,w}$ such that $\alpha_{i,p,w,I,y} = \beta_{i,p,w}\,\beta_{I,y}^{i,p,w}$, $\sum_{i,p,w}|\beta_{i,p,w}|^2 = 1$ and for each choice of i,p,w, we have that $\sum_{I,y}|\beta_{I,y}^{i,p,w}|^2 = 1$. With this decomposition, using the definition in (2.1), the success probability of producing k correct output values is given by

$$\begin{aligned}\left\|\Pi_k\mathcal{S}\,|\phi_t\rangle\right\|^2 &= \left\|\Pi_k \sum_{i,p,w} \beta_{i,p,w}\,|i,p,w\rangle \otimes \left[\mathcal{S}_1^{\otimes n} \sum_{\substack{I\subseteq[n],\ |I|\le t \\ y\in D^I}} \beta_{I,y}^{i,p,w}\,|y\rangle_I\,|\perp\rangle_{[n]\setminus I}\right]\right\|^2 \\ &= \left\|\sum_{i,p,w} \beta_{i,p,w}\,|i,p,w\rangle \otimes \left[\Pi_{q(w)}\mathcal{S}_1^{\otimes n} \sum_{\substack{I\subseteq[n],\ |I|\le t \\ y\in D^I}} \beta_{I,y}^{i,p,w}\,|y\rangle_I\,|\perp\rangle_{[n]\setminus I}\right]\right\|^2,\end{aligned}$$

where $\Pi_{q(w)}$ is defined as in (2.2) and is the projection of Π_k onto fixed values of $q(w)$. Since the basis states $|i,p,w\rangle$ are orthogonal and $\sum_{i,p,w}|\beta_{i,p,w}|^2 = 1$, we have

$$\left\|\Pi_k\mathcal{S}\,|\phi_t\rangle\right\|^2 \le \max_{i,p,w}\left\|\Pi_{q(w)}\mathcal{S}_1^{\otimes n} \sum_{\substack{I\subseteq[n],\ |I|\le t \\ y\in D^I}} \beta_{I,y}^{i,p,w}\,|y\rangle_I\,|\perp\rangle_{[n]\setminus I}\right\|^2. \tag{4.2}$$

We now fix i,p,w and let $A_{q(w)}$ be the submatrix of A restricted to the rows defined by the set of the k output values U associated with $q(w)$. We can describe $\Pi_{q(w)}$ as a projection onto basis states $|x_1,\ldots,x_n\rangle$ such that

$$A_{q(w)}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = q(w).$$

Since the basis states $|y\rangle_I\,|\perp\rangle_{[n]\setminus I}$ for distinct I are orthogonal in the recording query basis, they remain orthogonal in the standard basis after the $\mathcal{S}$ operator is applied. However, the subsequent application of the $\Pi_{q(w)}$ projector makes these vectors no longer orthogonal.

To handle this, we bucket the sets $I\subseteq[n]$ with $|I|\le t$ into a small number of buckets, $\mathcal{B}_1,\ldots$, so that for each bucket $\mathcal{B}_\ell$ we can bound

$$\mu_\ell = \left\|\Pi_{q(w)}\mathcal{S}_1^{\otimes n} \sum_{I\in\mathcal{B}_\ell, y\in D^I} \beta_{I,y}^{i,p,w}\,|y\rangle_I\,|\perp\rangle_{[n]\setminus I}\right\|^2$$

and then we can use the triangle inequality to bound the success probability as a sum of the μ_ℓ.

In particular, our key observation is that if a bucket of recording query basis states completely misses querying a fixed set of input variables that could completely scramble the value of a set of r output values, then one cannot do better than randomly guess those output values. More precisely, we show that the contribution to success from that bucket of basis states has amplitude at most $\frac{1}{\sqrt{d^r}}$.

LEMMA 4.3. *Let $U \subseteq [m]$ be a set of output indices and $V \subseteq [n]$ be a set of input indices with $|V| = |U| = r$ such that the submatrix $A_{U,V}$ is full rank. Fix $q \in \mathbb{F}^U$ and define Π_q to be the projection map onto the span of the set of basis states $|x_1, \ldots, x_n\rangle$ with $x_1 \ldots x_n \in D$ such that $A_U x = q$. Then for any collection $\mathcal{B}$ of sets $I \subseteq [n] \setminus V$ and any quantum state $\sum_{I \in \mathcal{B},\ y \in D^I} \eta_{I,y} |y\rangle_I |\perp\rangle_{[n]\setminus I}$ we have*

$$\left\| \Pi_q \mathcal{S}_1^{\otimes n} \sum_{I \in \mathcal{B},\ y \in D^I} \eta_{I,y} |y\rangle_I |\perp\rangle_{[n]\setminus I} \right\|^2 \leq \frac{1}{d^r}.$$

Proof. By definition each $I \in \mathcal{B}$ satisfies $I \cap V = \varnothing$, so

$$\begin{aligned}
&\Pi_q \mathcal{S}_1^{\otimes n} \sum_{I \in \mathcal{B},\ y \in D^I} \eta_{I,y} |y\rangle_I |\perp\rangle_{[n]\setminus I} \\
&= \Pi_q \mathcal{S}_1^{\otimes n} \left(|\perp\rangle_V \otimes \sum_{I \in \mathcal{B},\ y \in D^I} \eta_{I,y} |y\rangle_I |\perp\rangle_{[n]\setminus(I\cup V)} \right) \\
&= \Pi_q \left(\mathcal{S}_1^{\otimes r} |\perp\rangle_V \otimes \mathcal{S}_1^{\otimes (n-r)} \sum_{I \in \mathcal{B},\ y \in D^I} \eta_{I,y} |y\rangle_I |\perp\rangle_{[n]\setminus(I\cup V)} \right) \\
&= \Pi_q \left(\sum_{y' \in D^V} \frac{1}{\sqrt{d^r}} |y'\rangle_V \otimes \mathcal{S}_1^{\otimes (n-r)} \sum_{I \in \mathcal{B},\ y \in D^I} \eta_{I,y} |y\rangle_I |\perp\rangle_{[n]\setminus(I\cup V)} \right)
\end{aligned}$$

since $\mathcal{S}_1(|\perp\rangle) = \sum_{y' \in D} \frac{1}{\sqrt{d}} |y'\rangle$. Now

$$\mathcal{S}_1^{\otimes (n-r)} \sum_{I \in \mathcal{B},\ y \in D^I} \eta_{I,y} |y\rangle_I |\perp\rangle_{[n]\setminus(I\cup V)} = \sum_{z \in (D \cup \{\perp\})^{[n]\setminus V}} \delta_z |z\rangle_{n\setminus V}$$

for some amplitudes δ_z satisfying $\sum_{z \in (D\cup\{\perp\})^{[n]\setminus V}} |\delta_z|^2 = 1$.

For each value of $z \in D^{[n]\setminus V}$, since the sub-matrix $A_{U,V}$ is invertible, there is a unique value $y_z \in D^V$ such that $A_U(y_z \cup z) = q$ so we get that

$$\begin{aligned}
&\left\| \Pi_q \mathcal{S}_1^{\otimes n} \sum_{I \in \mathcal{B},\ y \in D^I} \eta_{I,y} |y\rangle_I |\perp\rangle_{[n]\setminus I} \right\|^2 \\
&= \left\| \Pi_q \left[\sum_{y' \in D^V} \frac{1}{\sqrt{d^r}} |y'\rangle_V \otimes \sum_{z \in (D\cup\{\perp\})^{n-r}} \delta_z |z\rangle_{[n]\setminus V} \right] \right\|^2 \\
&= \left\| \frac{1}{\sqrt{d^r}} \cdot \Pi_q \left[\sum_{y' \in D^V} |y'\rangle_V \sum_{z \in D^{n-r}} \delta_z |z\rangle_{n\setminus V} \right] \right\|^2 \\
&= \left\| \frac{1}{\sqrt{d^r}} \cdot \Pi_q \sum_{z \in D^{[n]\setminus V}} \delta_z \sum_{y' \in D^V} |y'\rangle_V |z\rangle_{n\setminus V}) \right\|^2
\end{aligned}$$

$$= \left\| \frac{1}{\sqrt{d^r}} \sum_{z \in D^{[n]\setminus V}} \delta_z \left|y_z\right\rangle_V \left|z\right\rangle_{n\setminus V} \right\|^2 \leq \frac{1}{d^r}$$

since $\sum_{z \in D^{[n]\setminus V}} |\delta_z|^2 \leq 1$. ∎

Next we decompose the set of all I with $|I| \leq t$ into buckets where we can apply the above with r equal to a constant fraction of k. (This decomposition of the sets I into buckets automatically implies a decomposition of Γ_t into buckets, each of which will be a c'-admissible bucket for some constant c' by Lemma 4.3 and the value of r, yielding a c'-admissible bucket t-family corresponding to the basic version of our bucketing methods as discussed in section 3.)

LEMMA 4.4. *Let A be a (k,h,c)-rigid matrix and let $k' = \lceil ck \rceil$. Then for every subset U of k rows of A, there is a collection of disjoint k'-subsets of columns from $[n]$, $V_1, \ldots, V_\ell$ for $\ell = \lceil h/k' \rceil \leq \lceil h/(ck) \rceil$ and corresponding sets of rows $U_1, \ldots, U_\ell \subseteq U$ such that for each $j \in [\ell]$, the $k' \times k'$ submatrix A_{U_j,V_j} is full rank. (In particular the union, W, of the sets V_j has size at least h.) If $c = 1$ then all $U_j = U$.*

Proof. Fix $U \in [m]$ with $|U| = k$. The following procedure constructs such a collection, one set at a time. We maintain a subset of W columns that is the union of the V_j constructed so far. Suppose that $|W| < h$. Then, by the (k,h,c)-rigidity of A, the submatrix $A_{U,[n]\setminus W}$ has rank at least k'. Hence there is a $k' \times k'$ submatrix A_{U_j,V_j} of $A_{U,[n]\setminus W}$ that has full rank k'. We now add V_j to the collection of k'-sets of columns, record its corresponding row set U_j, and set $W \leftarrow W \cup V_j$. This produces exactly $\lceil h/k' \rceil$ subsets. ∎

Fix the collection of sets $V_1, \ldots, V_\ell$ given by Lemma 4.4. Let $k'' = \lfloor \alpha k' \rfloor$. Suppose that $V_j = \{i_1, \ldots, i_{k'}\} \subseteq [n]$ with $i_1 \leq \cdots \leq i_{k'}$. For each $\lambda \in \binom{[k']}{k''}$, define the set V_j^λ to be the subset of V_j that has the k'' elements of V_j indexed by λ removed. (That is, $i_{j'} \notin V_j^\lambda$ iff $j' \in \lambda$.) Then $|V_j^\lambda| = k' - k'' \geq c(1-\alpha)k$. There are a total of $\binom{k'}{k''} \leq 2^{H_2(\alpha)\,k'}$ possible values of λ and hence $\lceil h/k' \rceil \cdot 2^{H_2(\alpha)\,k'}$ sets of the form V_j^λ. These sets have two useful properties: first any subset of $[n]$ with size at most αh must miss some V_j^λ and second if the entries of x corresponding to some V_j^λ are uniformly random, then for any set of k indices in Ax, at least $c(1-\alpha)k$ of these values are also uniformly random.

LEMMA 4.5. *For $t \leq \alpha h$ and every $I \subseteq [n]$ with $|I| \leq t$, there is some $j \leq \lceil h/k' \rceil$ and $\lambda \in \binom{[k']}{k''}$ such that $I \subseteq [n] \setminus V_j^\lambda$.*

Proof. Fix such a set I with $|I| \leq t$. Since $t \leq \alpha h$, $|\bigcup_{j \in [\ell]} V_j| \geq h$, and the sets V_j are disjoint, by averaging there is some set V_j that has at most an α fraction of its elements in I. Hence V_j has at most $k'' \leq \alpha k'$ elements of I. Choose a set $\lambda \in \binom{[k']}{k''}$ that contains the indices within V_j of all of the elements of $V_j \cap I$. Then by construction $I \cap V_j^\lambda = \varnothing$. ∎

(Lemmas 4.3 and 4.5 together give us all the ingredients we need to yield a c'-admissible bucket t-family with $c' = d^{c(1-\alpha)}$ as defined in section 3; each element of the family is determined by a pair (j,λ) as follows:) By applying Lemma 4.5 we can associate each $I \subseteq [n]$ with $|I| \leq t$ with a pair (j,λ) such that $I \in [n] \setminus V_j^\lambda$ and define bucket $\mathcal{B}_j^\lambda$ to consist of all such sets I associated with pair (j,λ).[6] Further,

[6]Note that while some sets I could be associated with multiple pairs (j,λ) in the admissible bucket family, since we only require one bucket per recording query basis element for the analysis, we will choose only one such pair for each I.

define a set $U_j^\lambda \subseteq U_j \subseteq [m]$ of the rows of $A_{q(w)}$ with $|U_j^\lambda| = k' - k''$ such that the submatrix $A_{U_j^\lambda, V_j^\lambda}$ is full rank. Such a subset of rows must exist since A_{U_j, V_j^λ} is a full rank matrix. Then let $q_j^\lambda = q(w)|_{U_j^\lambda}$ be the portion of the assignment $q(w)$ on the rows of U_j^λ.

We are now ready to provide an upper bound on the success probability from (4.2) using our admissible bucket family. We have

$$\begin{aligned}
&\left\| \Pi_{q(w)} \mathcal{S}_1^{\otimes n} \sum_{\substack{I \subseteq [n],\ |I| \le t \\ y \in D^I}} \beta_{I,y}^{i,p,w} \left|y\right\rangle_I \left|\perp\right\rangle_{[n]\setminus I} \right\|^2 \\
&= \left\| \Pi_{q(w)} \mathcal{S}_1^{\otimes n} \sum_{j \in [\ell]} \sum_{\lambda \in \binom{[k']}{k''}} \sum_{I \in \mathcal{B}_j^\lambda,\ y \in D^I} \beta_{I,y}^{i,p,w} \left|y\right\rangle_I \left|\perp\right\rangle_{[n]\setminus I} \right\|^2 \\
&\le \left\| \sum_{j \in [\ell]} \sum_{\lambda \in \binom{[k']}{k''}} \Pi_{q_j^\lambda}\ \mathcal{S}_1^{\otimes n} \sum_{I \in \mathcal{B}_j^\lambda,\ y \in D^I} \beta_{I,y}^{i,p,w} \left|y\right\rangle_I \left|\perp\right\rangle_{[n]\setminus I} \right\|^2 .
\end{aligned} \tag{4.3}$$

Applying Lemma 4.3 with $r = k' - k''$, $q = q_j^\lambda$, $U = U_j^\lambda$, $V = V_j^\lambda$, and $\mathcal{B} = \mathcal{B}_j^\lambda$, we have that

$$\left\| \Pi_{q_j^\lambda}\ \mathcal{S}_1^{\otimes n} \sum_{I \in \mathcal{B}_j^\lambda,\ y \in D^I} \beta_{I,y}^{i,p,w} \left|y\right\rangle_I \left|\perp\right\rangle_{[n]\setminus I} \right\|^2 \le 1/d^{k'-k''} \le 1/d^{(1-\alpha)\,k'}.$$

and hence using (4.3) we obtain that the success probability of producing k correct output values,

$$\begin{aligned}
\left\| \Pi_k \mathcal{S} \left|\phi_t\right\rangle \right\|^2 &\le \ell^2 \binom{k'}{k''}^2 / d^{(1-\alpha)\,k'} \le \lceil h/k' \rceil^2\ 4^{H_2(\alpha)\,k'} / d^{(1-\alpha)\,k'} \\
&= \lceil h/k' \rceil\ (4^{H_2(\alpha)} / d^{(1-\alpha)})^{k'}.
\end{aligned}$$

Without loss of generality in our desired bound we can assume that $4^{H_2(\alpha)}/d^{(1-\alpha)} < 1$. Therefore the bound still applies when we replace k' by the potentially smaller ck which is what we needed to show. ∎

4.2. Related time-space tradeoff and cumulative memory lower bounds. Following the same arguments as for classical computation [4], we use Theorem 4.1 to obtain a collection of time-space lower bounds for problems that are closely related to matrix vector products. Our proofs are identical to their classical counterparts proven in [4, sections 5-6] and are duplicated here for completeness. Many of these lower bounds are tightly matched by classical query algorithms. Constructions of matching upper bounds can be found in section A.

COROLLARY 4.6. *Let $\mathbb{F}$ be a field and $D \subseteq \mathbb{F}$ such that $d = |D|$. Any quantum circuit that computes the discrete Fourier transform (DFT) of vectors in D^n in time T and space S with probability at least 2^{-S} requires T to be $\Omega(n^2 \log(d)\ /S)$.*

Proof. Applying Theorem 4.1 with the rigidity of the DFT from Proposition 2.3 directly gives us the lower bound. ∎

PROPOSITION 4.7 ([4]). *There is a constant $\gamma \in (0, 1/2)$ such that at least a $1 - |D|^{-1}(2/3)^{\gamma n}$ fraction of the Toeplitz (diagonal constant) matrices over $D^{n \times n}$ are $(\gamma n, \gamma n)$-rigid.*

Recall that the convolution of two vectors $w = u * v$ is $w_k = \sum_{i\in[n]} u_i v_{k-i}$ where the indices are reduced modulo n, where we identify n with 0.

COROLLARY 4.8. *Let $\mathbb{F}$ be a field and $D \subseteq \mathbb{F}$ such that $d = |D|$. Any quantum query algorithm computing the convolution of two vectors in D^n in time T and space S with probability at least 2^{-S} requires T to be $\Omega(n^2 \log(d)\ /S)$.*

Proof. For simplicity assume that n is even. Let

$$U = \begin{bmatrix} u_n & u_{n-1} & \dots & u_2 & u_1 \\ u_1 & u_n & \dots & u_3 & u_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{n-2} & u_{n-3} & \dots & u_n & u_{n-1} \\ u_{n-1} & u_{n-2} & \dots & u_1 & u_n \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

Where A, B, C and D are $n/2 \times n/2$ submatrices. Then Uv is the convolution between vectors u and v. Observe that U is a Toeplitz matrix and by picking u to be a uniform vector over D, Proposition 4.7 tells us that for sufficiently large n, there is a constant $\gamma \in (0, 1/2)$ such that both A and B are $(\gamma n, \gamma n/2)$-rigid with probability at least $1/2$. This lets us restrict our input to such choices for u and observe that the matrix $U' = \begin{bmatrix} A & B \end{bmatrix}$ is $(\gamma n, \gamma n/2)$-rigid, so Theorem 4.1 gives us that computing $U'v$ requires T that is $\Omega(n^2 \log(d)\ /S)$. Since U' is a subfunction of U, convolution also requires T that is $\Omega(n^2 \log(d)\ /S)$. ∎

COROLLARY 4.9. *A quantum circuit that multiplies two n bit binary numbers in time T and space S with probability at least 2^{-S} requires T to be $\Omega(n^2/(S \log^2 n))$.*

Proof. Let u, v be arbitrary vectors over $\mathbb{F}_2$. Define the binary number

$$u' = 0^{\lceil \log_2 n \rceil - 1} u_n \dots 0^{\lceil \log_2 n \rceil - 1} u_1 0^{\lceil \log_2 n \rceil - 1} u_n \dots 0^{\lceil \log_2 n \rceil - 1} u_1$$

and similarly define v'. Then observe that the product $u' \cdot v'$ contains all entries of the convolution between u and v encoded in blocks of $\lceil \log_2 n \rceil$ bits each. By Corollary 4.8 this requires T to be $\Omega(n^2/(S \log^2 n))$. ∎

PROPOSITION 4.10 ([4]). *Let $A, B, C, Y \in D^{n\times n}$. Let $\mathcal{B}$ (and $\mathcal{Y}$) be the vectors in D^{n^2} formed by stacking the transposes of the rows of B (and Y) into a column vector. If D is a commutative ring, then the following conditions are equivalent:*

- $Y = ABC$
- $\mathcal{Y} = (A \otimes C^T)\mathcal{B}$

where $\otimes$ is the standard tensor (Kronecker) product.

PROPOSITION 4.11 ([4]). *Let $\gamma \in (0, 1/2)$. If A and B are $(\gamma n, \gamma n)$-rigid, then $A \otimes B$ is $(\gamma^2 n^2, \gamma^2 n^2, \gamma^2)$-rigid.*

COROLLARY 4.12. *Let $\mathbb{F}$ be a field and $D \subseteq \mathbb{F}$ such that $d = |D|$. Any quantum circuit that computes the product ABC on inputs $A, B, C \in D^{n\times n}$ in time T and space S with probability at least 2^{-S} requires T that is $\Omega(n^4 \log(d)\ /S)$.*

Proof. We use Proposition 4.10 to view this as a matrix-vector product problem where $\mathcal{B}$ is the input and $\mathcal{Y}$ is the output. By Proposition 2.4 there is a constant $\gamma \in (0, 1/2)$ such that both A and C are γ rigid with constant probability, so we can assume such without increasing the expected cost by more than a constant factor. Then Proposition 4.11 gives us that $A \otimes C$ is $(\gamma^2 n^2, \gamma^2 n^2, \gamma^2)$-rigid and we can apply Theorem 4.1 to get that T must be $\Omega(n^4 \log(d)\ /S)$ as desired. ∎

COROLLARY 4.13. *Let $\mathbb{F}$ be a field and $D \subseteq \mathbb{F}$ such that $d = |D|$. Any quantum circuit that computes A^3 on inputs in $D^{n\times n}$ in time T and space S with probability at least 2^{-S} requires T that is $\Omega(n^4 \log(d)\ /S)$.*

Proof. Let $A, B, C \in D^{n\times n}$. Then construct the $4n \times 4n$ matrix

$$M = \begin{bmatrix} 0 & A & 0 & 0 \\ 0 & 0 & B & 0 \\ 0 & 0 & 0 & C \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Observe that the top right $n \times n$ sub-matrix of M^3 is equal to the product ABC. Thus we get a reduction to matrix-matrix-matrix product and can apply Corollary 4.12 to get our lower bound. □

COROLLARY 4.14. *Let $\mathbb{F}$ be a field and $D \subseteq \mathbb{F}$ such that $d = |D|$. Any quantum circuit that computes A^{-1} on unit upper triangular inputs in $D^{n\times n}$ in time T and space S with probability at least 2^{-S} requires T that is $\Omega(n^4 \log(d)/S)$.*

Proof. Let $A, B, C \in D^{n\times n}$. Construct the $4n \times 4n$ matrix

$$M = \begin{bmatrix} I & -A & 0 & 0 \\ 0 & I & -B & 0 \\ 0 & 0 & I & -C \\ 0 & 0 & 0 & I \end{bmatrix}$$

where I is the $n \times n$ identity submatrix. Then observe that M^{-1} has the product ABC as its top right $n \times n$ submatrix. We can again use Theorem 4.1 to get our lower bound. □

COROLLARY 4.15. *Let $\mathbb{F}$ be a field and $D \subseteq \mathbb{F}$ such that $d = |D|$. Any quantum circuit that solves any $n \times n$ system of linear equations over D in time T and space S with probability at least 2^{-S} requires T that is $\Omega(n^3 \log(d)\ /S)$.*

Proof. It is possible to invert a matrix by solving n systems of n linear equations. By a reduction Corollary 4.14 gives us that solving these equations requires T that is $\Omega(n^4 \log(d)\ /S)$. Thus least one of these equations must require T that is $\Omega(n^3 \log(d)\ /S)$ to solve. □

In [12] the authors showed that the kinds of quantum time-space product lower bounds we proved in this section can be extended to asymptotically equivalent lower bounds on the stronger notion of cumulative memory complexity. We restate a simplified version of their main theorem for quantum circuits and classical query algorithms here.

PROPOSITION 4.16 ([12]). *Let $f : D^n \to R^m$ be a function such that there exists constant C, functions $m'(n) \in \omega(\log n), h(k,n) = k^{\Delta} h_1(n), K(n)$, and a distribution μ over D^n where when $x \sim \mu$ the probability that - for any $k \leq m'(n)$ - any quantum circuit (or classical query algorithm) with at most $h(k,n)$ queries to x produces k correct output values of $f(x)$ with probability at most $C \cdot K(n)^{-k}$. Then for any constant $c > 0$, any quantum circuit (or classical query algorithm) that computes f with T queries and error $\epsilon \leq (1 - 1/(2T^c))$ must have cumulative memory that is $\Omega\left(\min\left(\left[(mh_1(n))^{1/(1-\Delta)} \log K(n)\right]/T^{\Delta/(1-\Delta)}, m'(n)^{1+\Delta} h_1(n) \log K(n)\right)\right)$.*

Using the above result, we can extend the quantum time-space product lower bound for matrix vector products to a matching quantum cumulative memory lower bound.

THEOREM 4.17. *Let $\gamma > 0$ and $c \in (0, 1/2]$ be fixed. If A is a $(\gamma n, \gamma n, c)$-rigid $n \times n$ matrix over a field $\mathbb{F}$ then any quantum circuit using time T and space S that computes the function $f_A : D^n \to \mathbb{F}^n$ for $D \subseteq \mathbb{F}$ with $d = |D|$ given by $f_A(x) = Ax$ with success probability larger than $1/T$ requires cumulative memory that is $\Omega(n^2 \log d)$.*

Proof. By Lemma 4.2 we can apply Proposition 4.16 where $C = \lceil 1/c \rceil$, $m'(n) = \gamma n$, $\Delta = 0$, $h_1(n) = \alpha n$, $K(n) = d^{1/6}$, and μ is the uniform distribution. This give us that any quantum circuit computing f_A with T queries and error at most $1 - 1/(2T)$ requires cumulative memory $\Omega(n^2 \log d)$ as desired. ∎

Directly applying this in place of Theorem 6.5 gives us matching cumulative (CM) memory lower bounds for Corollary 4.6 through Corollary 4.15.

COROLLARY 4.18. *Let $\mathbb{F}$ be a field and $D \subseteq \mathbb{F}$ such that $d = |D|$. Any quantum circuit with inputs over D that computes the DFT or vector convolution requires CM that is $\Omega(n^2 \log d)$. Any quantum circuit that computes the product of three matrices, matrix cubing, or matrix inversion requires CM that is $\Omega(n^4 \log d)$. Any quantum circuit that solves $n \times n$ systems of linear equations requires CM that is $\Omega(n^3 \log d)$. Additionally any quantum circuit that multiplies two n bit binary numbers requires CM that is $\Omega(n^2 / \log^2 n)$.*

5. Quantum matrix multiplication. While many of the applications so far, including the matrix triple product lower bound discussed in the previous section, are derived from the matrix-vector product lower bound, our matrix multiplication lower bound requires a separate argument using ideas from the classical lower bound for the problem in [4]. Implementing this requires a much more subtle way of applying our bucketing method for states that allows us to concentrate on just a subset of the buckets containing most of the total amplitude and ignore the others. As in section 4, our lower bounds in this section apply to a more general model of quantum circuits that can decide which outputs they want to produce in a given layer based on the inputs that they have queried.

Here we consider the matrix multiplication problem $f(A, B) = AB$ where both A and B are considered input. If we could fix a choice of A, we would be able to make our proof somewhat simpler. However, as Abrahamson pointed out in [4], there is a classical algorithm that can compute the function $f(B) = AB$ for any fixed matrix A in $O(n^2)$ queries and $O(n \log d)$ space. Thus our lower bound requires both A and B to be inputs to the function.

THEOREM 5.1. *Let $\mathbb{F}$ be a field and $D \subseteq \mathbb{F}$ with $d = |D|$. Then any quantum circuit $\mathcal{C}$ that uses time T and space S and computes the function $f : D^{2n^2} \to \mathbb{F}^{n^2}$ given by $f(A, B) = AB$ with success probability larger than $1/T$ must have T that is $\Omega(n^3 \sqrt{\log d \ / S})$.*

Again this theorem follows from the following key lemma, proven in subsection 5.1, which lets us bound the number of correct output values produced by a shallow quantum circuit.

LEMMA 5.2. *Let $\gamma \in (0, 1/2)$ and $f : D^{n^2} \times D^{n^2} \to \mathbb{F}^{n^2}$ for $D \subseteq \mathbb{F}$ with $|D| = d$ be defined by $f(A, B) = AB$. Then for any constant $\beta > 0$ and quantum circuit $\mathcal{C}$ with at most $h = \beta\gamma n\sqrt{k/2}$ queries to input matrices A, B sampled uniformly from D^{n^2}, the probability that A and B are $(\gamma n, \gamma n)$-rigid and $\mathcal{C}$ produces k correct output values of $f(A, B)$ is at most $16 \min(k, n)^{\sqrt{2k}} (4^{H_2(4\beta)} / d^{1-4\beta})^{k/4}$.*

Note that for $\beta \leq 0.0184$ we have $1 - 4\beta - 2H_2(4\beta) > 1/6$ so the bound is at most

$16\min(k,n)^{\sqrt{2k}}d^{-k/24}$.

Proof of Theorem 5.1 *from Lemma* 5.2. Let $\gamma \in (0,1/2)$ be the constant given by Proposition 2.4. By that proposition, the probability that either of two matrices A and B chosen uniformly randomly from D^{n^2} is not $(\gamma n, \gamma n)$-rigid is at most $2d^{-1}(2/3)^{\gamma n}$. Let $\mathcal{C}$ be a quantum circuit with T queries and space S. Let $\beta = 0.0429$, $d = |D|$, and set $k = \lceil 48(6S+4)/\log_2 d \rceil$. We partition $\mathcal{C}$ into $\left\lceil T/(\beta\gamma n\sqrt{k/2})\right\rceil$ sub-circuits that each have at most $\beta\gamma n\sqrt{k/2}$ queries. Without loss of generalities there are at most n^2 such sub-circuits. By combining Proposition 2.5 with Lemma 5.2, we know that for a uniformly random input, the probability that A and B are $(\gamma n, \gamma n)$-rigid matrices and a fixed sub-circuit can produce k outputs is at most $16\min(k,n)^{\sqrt{2k}}2^S d^{-k/24} \leq 16k^{\sqrt{2k}}2^S d^{-k/24}$. Therefore the probability that A and B are $(\gamma n, \gamma n)$-rigid matrices and one of the sub-circuits produces k correct output values is at most $16k^{\sqrt{2k}}2^S d^{-k/24}n^2$. Combining this with the probability that one of A or B is not $(\gamma n, \gamma n)$-rigid, the probability that there is a sub-circuit that correctly produces k output values is at most $16k^{\sqrt{2k}}2^S d^{-k/24}n^2 + 2d^{-1}(2/3)^{2\gamma n}$.

Since we can assume without loss of generality that $T \leq n^3$, for sufficiently large n, $2d^{-1}(2/3)^{2\gamma n} \leq 1/(2T)$ and $k^{\sqrt{2k}} \leq 2^{k/48} \leq d^{k/48}$. Plugging in our value of k and the fact that $S \geq \log_2 n$ without loss of generality gives a probability of at most

$$\begin{aligned} 16k^{\sqrt{2k}}2^{2S}d^{-k/24}n^2 + 2d^{-1}(2/3)^{2\gamma n} &\leq 16^2{}^S d^{-k/48}n^2 + 1/(2T) \\ &\leq 1/(2T) + 1/(2T) = 1/T. \end{aligned}$$

Since $\mathcal{C}$ must be correct with probability larger than $1/T$, this implies that

$$(k-1)\left\lceil T/(\beta\gamma n\sqrt{k/2})\right\rceil \geq n^2.$$

Plugging in our value of k gives us that

$$T \text{ is } \Omega(n^3\sqrt{\log d}/\sqrt{S+\log T}).$$

Since $S \geq \log_2 n$ and our bound trivially holds when T is $\omega(n^3\sqrt{\log d})$ there is a constant $c > 0$ such that $cS \geq \log_2 T$. So T is $\Omega(n^3\sqrt{\log d/S})$ as desired. □

Our quantum lower bound is tightly matched by a classical query algorithm in Proposition A.5.

5.1. The success probability of small depth quantum circuits. We first give an overview of the argument which assumes a uniform distribution over all input matrices A and B in $D^{n\times n}$. Unlike the matrix-vector product proof, in addition to the requirement of k correct output values, for success we also include the extra condition that both matrices must be $(\gamma n, \gamma n)$ rigid. As in the case of the matrix-vector product proof, we decompose the state after $t \leq h = \beta\gamma n\sqrt{k/2}$ steps into orthogonal components based on different values $|i,p,w\rangle$ which determines the k output values produced, though this now can be up to quadratic in n. However, unlike that proof, we need to use the weighted version of our bucketing method. We show that for each such $|i,p,w\rangle$ the total fraction of the squared amplitude for any recording query state spanned by basis elements with at most t non-$\perp$ items where the k output values produced are correct is exponentially small in k.

The output values produced determine a set of rows of the matrix A and columns of the matrix B that are relevant. For classical algorithms, where we can determine a

set of input locations queried, the lower bound of [4] shows that either at least $k/4$ of the output values lie in rows where few elements of A are queried or $k/4$ lie in columns where few elements of B are queried. For each of these cases ("light" rows or "light" columns) the corresponding output values in those rows or columns are hard to produce in that the requirement that the other matrix is rigid means that the algorithm is exponentially unlikely in k to be correct on those entries.

In the quantum case, when viewed in the recording query basis, the state involves a superposition over all possible assignments to subsets of indices for the relevant rows of A and columns of B with at most t non-$\perp$ entries. For convenience, we first split these basis states depending on whether there are many outputs in light rows or many in light columns; and then on which rows/columns those are; each determines a set of $k/4$ output values to consider hard and whether to focus on matrix A or B. The number of such possibilities is not too large so the total is not too much larger than the maximum over all such choices. We further consider a fixed choice of the other rigid matrix that maximizes the resulting probability that the hard outputs produced have correct values. The number of consistent recording query basis states in each such superposition is still enormous.

We need to apply bucketing where either A or B is fixed as a rigid matrix and the other can be interpreted as a having a collection of light columns (or rows) such that the output values are the results of a matrix-vector products involving vectors with few queries. However, repeatedly applying the basic bucketing method for basis states we used for matrix-vector products fails because the total number of buckets would be too large since it would end up being the product over the number of choices for each row or column.

Instead, we show that among these potential buckets we can find a small number of admissible buckets that together capture a large portion of the amplitude associated with the state, yielding an t-reduction scheme of admissible buckets that lets us derive the final lower bound. We now give the details of this argument.

Proof of Lemma 5.2. Let $C = AB$, $\Pi_{\text{rigid}(A)}$ (and $\Pi_{\text{rigid}(B)}$) be the projection onto inputs where A (and B) are $(\gamma n, \gamma n)$-rigid matrices, and define $\Pi_{\text{rigid}} = \Pi_{\text{rigid }A}\Pi_{\text{rigid }B}$. Assume that $q(w)$ — the output as a function of the measured value of the work register — produces exactly k outputs; we ignore anything it produces after the first k. We will use $[A]$ to denote the set of indices of elements in A and likewise for $[B]$ and $[C]$. By Proposition 2.9, after $t \leq h$ queries in the recording query basis, the state $|\phi_t\rangle$ is a linear combination of basis states $|i, p, w, x_1, \ldots, x_n\rangle$ where $(x_1, \ldots, x_n) \in \Gamma_t$. As in our analysis of the case of matrix-vector products, it will be necessary to be more explicit in our discussion of Γ_t. Each element of Γ_t consists of an assignment $x \in D^E$ and $y \in D^F$ for some subsets $E \subseteq [A]$ and $F \subseteq [B]$ with $|E| + |F| \leq t$ and value $\perp$ on all coordinates in $[A] \setminus E$ and $[B] \setminus F$. Therefore, our state can be written as

$$|\phi_t\rangle = \sum_{\substack{i,p,w \\ E\subseteq[A],F\subseteq[B] \\ |E|+|F|\leq t \\ x\in D^E, y\in D^F}} \alpha_{i,p,w,E,F,x,y} \, |i,p,w\rangle \, |x\rangle_E \, |\perp\rangle_{[A]\setminus E} \, |y\rangle_F \, |\perp\rangle_{[B]\setminus F}$$

for some $\alpha_{i,p,w,E,F,x,y}$ with $\sum_{i,p,w,E,F,x,y} |\alpha_{i,p,w,E,F,x,y}|^2 = 1$. We first apply an analogous series of observations and decompositions to those that allowed us to derive (4.2) from (4.1) in the case of matrix-vector product. By Proposition 2.7, we note that

the final state of the algorithm in the standard oracle setting is given by

$$|\psi_t\rangle = \mathcal{S}\,|\phi_t\rangle = \mathcal{S} \sum_{\substack{i,p,w \\ E\subseteq[A],F\subseteq[B] \\ |E|+|F|\le t \\ x\in D^E, y\in D^F}} \alpha_{i,p,w,E,F,x,y}\,|i,p,w\rangle\,|x\rangle_E\,|\perp\rangle_{[A]\setminus E}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F}$$

Because $\mathcal{S}$ behaves as the identity on $|\phi_t\rangle_{\mathcal{C}}$ and each distinct choice of $|i,p,w\rangle$ gives an orthogonal basis state, this equals

$$\sum_{i,p,w} \beta_{i,p,w}\,|i,p,w\rangle \otimes \left[S_1^{\otimes 2n^2} \sum_{\substack{E\subseteq[A],F\subseteq[B] \\ |E|+|F|\le t \\ x\in D^E, y\in D^F}} \beta^{i,p,w}_{E,F,x,y}\,|x\rangle_E\,|\perp\rangle_{[A]\setminus E}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F} \right]$$

for some $\beta_{i,p,w}$ and $\beta^{i,p,w}_{E,F,x,y}$ such that $\sum_{i,p,w} |\beta_{i,p,w}|^2 = 1$ and $\sum_{E,F,x,y} |\beta^{i,p,w}_{E,F,x,y}|^2 = 1$ for each i,p,w. Now the probability over the choices of the input matrices and the result of the quantum algorithm making t queries that the matrices A and B are both $(\gamma n,\gamma n)$-rigid and the algorithm produces k correct output values from $C = AB$ is $\left\|\Pi_k \Pi_{\text{rigid}} \mathcal{S}\,|\phi_t\rangle\right\|^2$. Expanding this using the value of $\mathcal{S}\,|\phi_t\rangle$ yields

$$\left\| \Pi_k \Pi_{\text{rigid}} \sum_{i,p,w} \beta_{i,p,w}\,|i,p,w\rangle \otimes \left[S_1^{\otimes 2n^2} \sum_{\substack{E\subseteq[A],F\subseteq[B] \\ |E|+|F|\le t \\ x\in D^E, y\in D^F}} \beta^{i,p,w}_{E,F,x,y}\,|x\rangle_E\,|\perp\rangle_{[A]\setminus E}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F} \right] \right\|^2.$$

Factoring out the coefficients related to choices of $|i,p,w\rangle$ and replacing Π_k with $\Pi_{q(w)}$ for the corresponding w yields that $\left\|\Pi_k \Pi_{\text{rigid}} \mathcal{S}\,|\phi_t\rangle\right\|^2$ is

$$\sum_{i,p,w} |\beta_{i,p,w}|^2 \left\| \left[\Pi_{q(w)} \Pi_{\text{rigid}} S_1^{\otimes 2n^2} \sum_{\substack{E\subseteq[A],F\subseteq[B] \\ |E|+|F|\le t \\ x\in D^E, y\in D^F}} \beta^{i,p,w}_{E,F,x,y}\,|x\rangle_E\,|\perp\rangle_{[A]\setminus E}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F} \right] \right\|^2.$$

Next, since $\sum_{i,p,w} |\beta_{i,p,w}|^2 = 1$, this value is a convex combination of the squared norm terms. Thus, the probability that both the input matrices are rigid and the quantum algorithm produces k correct outputs, $\left\|\Pi_k \Pi_{\text{rigid}} \mathcal{S}\,|\phi_t\rangle\right\|^2$, is at most of the largest squared norm term,

$$\max_{i,p,w} \left\| \Pi_{q(w)} \Pi_{\text{rigid}} \mathcal{S}_1^{\otimes 2n^2} \sum_{\substack{E\subseteq[A],F\subseteq[B] \\ |E|+|F|\le t \\ x\in D^E, y\in D^F}} \beta^{i,p,w}_{E,F,x,y}\,|x\rangle_E\,|\perp\rangle_{[A]\setminus E}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F} \right\|^2. \tag{5.1}$$

For the rest of the proof we fix an i,p,w to achieve the maximum value in (5.1) and prove an upper bound on the resulting probability. This fixes the output values $q(w)$; we write $G \subseteq [C]$ with $|G| = k$ for the set of indices of the outputs given by $q(w)$. To keep notations simpler in the remainder of the proof we observe that (5.1) is upper bounded by the maximum of

$$\left\| \Pi_{q(G)} \Pi_{\text{rigid}} \mathcal{S}_1^{\otimes 2n^2} \sum_{\substack{E\subseteq[A],F\subseteq[B] \\ |E|,|F|\le t \\ x\in D^E, y\in D^F}} \beta_{E,F,x,y}\,|x\rangle_E\,|\perp\rangle_{[A]\setminus E}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F} \right\|^2 \tag{5.2}$$

over all $\beta_{E,F,x,y}$ with $\sum_{E,F,x,y} |\beta_{E,F,x,y}|^2 = 1$, all sets $G \subseteq [C]$ with $|G| = k$ and all assignments $q(G)$ to G.

We will split the sum in (5.2) over the different sets E and F of queried input indices depending on how they relate to the set of output indices given by G. Let $r(G)$ be the set of rows containing elements of G and $c(G)$ be the set of columns containing elements of G.[7]

Recall our bound $h = \beta\gamma n\sqrt{k/2}$ on the number of queries. We define a *light row of* E to be an element of $r(G)$ that contains at most $\beta\gamma n$ elements of E and define a *light column of* F to be an element of $c(G)$ that contains at most $\beta\gamma n$ elements of F. Since $|E| + |F| \le t \le \beta\gamma n\sqrt{k/2}$ we have $\le \sqrt{k/2}$ rows of E in $r(G)$ and $\le \sqrt{k/2}$ columns of F in $c(G)$ that are not light. We define $\mathcal{L}(E) \subseteq r(G)$, to be the set of light rows of E and $\mathcal{L}'(F) \subseteq c(G)$ to be the set of light columns of F. Therefore $|\{(i', j') \in G \mid i' \notin \mathcal{L}(E),\ j' \notin \mathcal{L}'(F)\}| \le k/2$ so at least $k/2$ elements of G are in light rows of E or in light columns of F. Therefore for every pair (E, F) at least one of the sets of outputs $G^r_{\mathcal{L}(E)} = \{(i', j') \in G \mid i' \in \mathcal{L}(E)\}$ or $G^c_{\mathcal{L}'(F)} = \{(i', j') \in G \mid j' \in \mathcal{L}'(F)\}$ has size $\ge k/4$.

Let $\mathcal{E}$ be the set of all $E \subseteq [A]$ with $|E| \le t$ such that G has at least $k/4$ outputs in light rows and $\mathcal{F}$ be the set of all $F \subseteq [B]$ with $|F| \le t$ such that G has at least $k/4$ outputs in light columns. We separately bound the contribution to (5.2) from pairs (E, F) with $E \in \mathcal{E}$ or $F \in \mathcal{F}$. The analyses of the two cases are completely symmetric up to matrix transposition. It will be convenient to focus on the case $F \in \mathcal{F}$ representing basis states where there are many outputs of G in light columns and compute an upper bound on

$$\left\| \Pi_{q(G)} \Pi_{\text{rigid}} \mathcal{S}_1^{\otimes 2n^2} \sum_{\substack{E \subseteq [A] \\ |E| \le t \\ x \in D^E}} \sum_{\substack{F \in \mathcal{F} \\ y \in D^F}} \beta_{E,F,x,y} \left|x\right\rangle_E \left|\perp\right\rangle_{[A]\setminus E} \left|y\right\rangle_F \left|\perp\right\rangle_{[B]\setminus F} \right\|^2. \tag{5.3}$$

Basis states where $E \in \mathcal{E}$ give exactly the same upper bound as (5.3) by applying the argument to the transposed product $B^T A^T$ and corresponding transposed sets F^T, E^T, and G^T. Hence, the quantity in (5.2) is at most 4 times that of (5.3).

To upper bound (5.3), we first remove the projection operator $\Pi_{\text{rigid } B}$ from $\Pi_{q(G)}\Pi_{\text{rigid}} = \Pi_{q(G)}\Pi_{\text{rigid } A}\Pi_{\text{rigid } B}$ to get $\Pi_{q(G)}\Pi_{\text{rigid } A}$. We then rewrite this combined projection operator as $\Pi_{q(G)}\Pi_{\text{rigid } A} = \sum_{A\ (\gamma n, \gamma n)\text{-rigid}} \Pi_A \otimes \Pi^A_{q(G)}$ where Π_A is the projection onto the specific matrix A and for each A, $\Pi^A_{q(G)}$ is the projection onto the choices for matrix B such that $C = AB$ agrees with $q(w)$. We therefore obtain that (5.3) is at most

$$\left\| \sum_{A\ (\gamma n, \gamma n)\text{-rigid}} (\Pi_A \otimes \Pi^A_{q(G)}) \mathcal{S}_1^{\otimes 2n^2} \sum_{\substack{E \subseteq [A] \\ |E| \le t \\ x \in D^E}} \sum_{\substack{F \in \mathcal{F} \\ y \in D^F}} \beta_{E,F,x,y} \left|x\right\rangle_E \left|\perp\right\rangle_{[A]\setminus E} \left|y\right\rangle_F \left|\perp\right\rangle_{[B]\setminus F} \right\|^2 \tag{5.4}$$

$$= \left\| \sum_{A\ (\gamma n, \gamma n)\text{-rigid}} (\Pi_A \otimes \Pi^A_{q(G)} \mathcal{S}_1^{\otimes n^2}) \sum_{A' \in (D \cup \{\perp\})^{[A]}} \sum_{\substack{F \in \mathcal{F} \\ y \in D^F}} \beta_{A'} \beta^{A'}_{F,y} \left|A'\right\rangle_{[A]} \left|y\right\rangle_F \left|\perp\right\rangle_{[B]\setminus F} \right\|^2$$

[7]We will think of $r(G)$ and $c(G)$ as being subsets of indices in $[n]$ that correspond to rows in A and columns of B, respectively, that are relevant for the outputs in G.

for some $\beta_{A'}$ and $\beta^{A'}_{F,y}$ such that $\sum_{A' \in (D \cup \{\perp\})^{n^2}} |\beta_{A'}|^2 = 1$ and $\sum_{F \in \mathcal{F}, y \in D^F} |\beta^{A'}_{F,y}|^2 = 1$ for each A'. Applying projector Π_A through this term gives that (5.4) equals

$$\left\| \sum_{A \ (\gamma n, \gamma n)\text{-rigid}} \beta_A |A\rangle_{[A]} \otimes [\Pi^A_{q(G)} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F \in \mathcal{F} \\ |F| \le t \\ y \in D^F}} \beta^A_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F}] \right\|^2 \tag{5.5}$$

Since $\Pi^A_{q(G)}$ only projects onto the $[B]$ input registers, each distinct choice of $|A\rangle_{[A]}$ gives orthogonal states so (5.5) equals

$$\begin{aligned} &\sum_{A \ (\gamma n, \gamma n)\text{-rigid}} |\beta_A|^2 \left\| \Pi^A_{q(G)} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F \in \mathcal{F} \\ |F| \le t \\ y \in D^F}} \beta^A_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2 \qquad (5.6) \\ &\qquad \le \max_{A \ (\gamma n, \gamma n)\text{-rigid}} \left\| \Pi^A_{q(G)} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F \in \mathcal{F} \\ y \in D^F}} \beta^A_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2. \end{aligned}$$

We fix a $(\gamma n, \gamma n)$-rigid matrix A that maximizes (5.6) and partition the set $\mathcal{F}$ based on the set $\mathcal{L}'(F)$ which contains all but at most $\left\lfloor \sqrt{k/2} \right\rfloor$ columns in $c(G)$. Therefore we can rewrite (5.6) as

$$\left\| \sum_{\substack{H \subseteq c(G) \\ |H| \le \left\lfloor \sqrt{k/2} \right\rfloor}} \Pi^A_{q(G)} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F \in \mathcal{F} \\ \mathcal{L}'(F) = c(G) \setminus H \\ y \in D^F}} \beta^A_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2. \tag{5.7}$$

Since $|c(G)| \le \min(k, n)$ we can upper bound (5.7) by

$$\min(k,n)^{\sqrt{2k}} \max_{\substack{H \subseteq c(G) \\ |H| \le \left\lfloor \sqrt{k/2} \right\rfloor}} \left\| \Pi^A_{q(G)} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F \in \mathcal{F} \\ \mathcal{L}'(F) = c(G) \setminus H \\ y \in D^F}} \beta^A_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2. \tag{5.8}$$

We fix the set H achieving the maximum value in (5.8), which fixes the value of $\mathcal{L}'(F) = c(G) \setminus H$. This fixes the set $G^c_{\mathcal{L}'(F)}$ of elements in G that are in light columns of F (equivalently, not in H) which, since $F \in \mathcal{F}$, contains at least $k/4$ elements of G. Let G' be a fixed subset of $k/4$ of the elements of $G^c_{\mathcal{L}'(F)}$. By construction we have $c(G') \subseteq \mathcal{L}'(F)$. By only requiring that the outputs in G' are correct, we therefore can upper bound the probability that both the input matrices are rigid and the quantum algorithm produces k correct outputs, $\left\| \Pi_k \Pi_{\text{rigid}} \mathcal{S} |\phi_t\rangle \right\|^2$, by the maximum value of

$$4 \min(k,n)^{\sqrt{2k}} \left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F \subseteq [B] \\ c(G') \subseteq \mathcal{L}'(F) \\ y \in D^F}} \beta'_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2 \tag{5.9}$$

over all $G' \subseteq [C]$ with $|G'| = k/4$ and $\beta'_{F,y}$ with $\sum_{F,y} |\beta'_{F,y}|^2 = 1$.

For each $j \in c(G')$, let k_j be the number of elements of G' in column j. Our overall strategy is to consider the $j \in c(G')$ one by one, and show that the total amplitude on states where these k_j outputs are correct conditioned on the success for previous values of j is of the form $d^{-\delta k_j}$ for some fixed constant $\delta > 0$. These are k_j outputs of the matrix-vector product Ay^j where y^j is the j-th column of B and the fact that $c(G') \subseteq \mathcal{L}'(F)$ implies that F has made at most $\beta\gamma n$ queries to y^j. This is very similar to the situation with the matrix-vector problem from Lemma 4.2. In analogy with Lemma 4.2, define U^j to be the set of k_j rows containing outputs of G' in column j.

Applying Lemma 4.4 with $c = 1$, for each $j \in c(G')$ there is a collection $V_1^j, \ldots, V_{\ell_j}^j$ of $\ell_j = \lceil \gamma n/k_j \rceil$ k_j-subsets of $[n]$ such that the $k_j \times k_j$ sub-matrix $A_{U^j V_i^j}$ has full rank.

Using the ideas of Lemma 4.2 we could bucket the possible basis states into one bucket for each large subset of the set associated with the tuple $(V_{i_j}^j)_{j \in q(G')}$ using Lemmas 4.3 and 4.4 and bound each bucket separately. However, unlike its use in the proof of Lemma 4.2, the value of many of the k_j can be very small, as low as 1, in which case the upper bounds using Lemmas 4.3 and 4.4 would be meaningless.

Instead, we need a stronger argument that depends on the amplitudes $\beta'_{F,y}$ in (5.9). The large subsets of the sets associated with tuples $(V_{i_j}^j)_{j \in q(G')}$ yield candidate buckets but there are too many of them to be used. However, we will see in the following lemma that a relatively small collection of them can capture all but a constant fraction of the total amplitude given by the $\beta'_{F,y}$. We will then see, in Corollary 5.4, how this can be applied inductively with the portion of the total amplitude that is left over to yield a good upper bound on the total probability of producing the output values in $q(G')$, which is what we need to prove. (In the terminology of section 3, Lemma 5.3 describes an t-reduction scheme of admissible buckets for $q(G')$, deriving some of its implications in parallel with its construction. On the other hand, Corollary 5.4 describes how that yields the overall bound; this is essentially a combination of the ideas of Lemmas 3.5 and 3.7.)

LEMMA 5.3. *Let $G' \subseteq [C]$ with $|G'| = k/4$ and $\mathcal{F}'$ be a set of $F \subseteq [B]$ such that $c(G') \subseteq \mathcal{L}'(F)$. Suppose that $\sum_{F \in \mathcal{F}', y \in D^F} |\delta_{F,y}|^2 = 1$ for some $\delta_{F,y}$. Define $\alpha = 4\beta$. Then there is an $\mathcal{F}'' \subseteq \mathcal{F}'$ and coefficients $\delta'_{F,y}$ where $\sum_{F \in \mathcal{F}'', y \in D^F} |\delta'_{F,y}|^2 = 1$ and*

$$\left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F \in \mathcal{F}' \\ y \in D^F}} \delta_{F,y} \, |y\rangle_F \, |\perp\rangle_{[B]\setminus F} \right\|^2 \tag{5.10}$$

$$\leq \frac{2^{1+H_2(\alpha)\,k/2}}{d^{(1-\alpha)\,k/4}} + \frac{1}{2} \left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F \in \mathcal{F}'' \\ y \in D^F}} \delta'_{F,y} \, |y\rangle_F \, |\perp\rangle_{[B]\setminus F} \right\|^2.$$

Proof. We first recall the definitions in our discussion preceding the lemma statement. For each $j \in c(G')$, define U^j to be the set of row indices of G' in column j and let $k_j = |U_j|$. Define $\ell_j = \lceil \gamma n/k_j \rceil$, apply Lemma 4.4 for each j, and let $V_1^j, \ldots, V_{\ell_j}^j$ be the collection of disjoint subsets of $[n]$ of size k_j found for each j such that each $k_j \times k_j$ sub-matrix $A_{U^j V_i^j}$ has full rank.

For each $F \in \mathcal{F}'$ and $i \in c(G')$, define F^j to be the set of row indices of elements of F in column j; since $c(G') \subseteq \mathcal{L}'(F)$, we have $|F^j| \leq \beta\gamma n$. For each $i \in [\ell_j]$ define

$$m_i^j = \sum_{F \in \mathcal{F}',\ y \in D^F} |\delta_{F,y}|^2 \cdot |F^j \cap V_i^j|.$$

Since $\sum_{F,y} |\delta_{F,y}|^2 = 1$, m_i^j can be viewed as the expected size of the overlap between the recorded queries in the j-th column of the matrix B and each V_i^j. Since for each j, the sets V_i^j are disjoint and $|F^j| \le \beta\gamma n$ we have $\sum_{i\in[\ell_j]} m_i^j \le \beta\gamma n$. Therefore, for each j, we have some index $i_j \in [\ell_j]$ such that $m_{i_j}^j \le \beta\gamma n/\ell_j \le \beta k_j$.

Since $\sum_{j\in c(G')} k_j = |G'| = k/4$, the expected total overlap between the recorded queries in the columns of G' and the chosen sets $V_{i_j}^j$ for those columns is $\sum_j m_{i_j}^j \le \sum_j \beta k_j = \beta k/4$. Define $\mathcal{F}''$ to be the set of $F \in \mathcal{F}'$ such that $\sum_j |F^j \cap V_{i_j}^j| \ge \alpha k/4 = \beta k$. By Markov's inequality we have

$$\sum_{F\in\mathcal{F}'',\ y\in D^F} |\delta_{F,y}|^2 \le \frac{\sum_j m_{i_j}^j}{\beta k} \le 1/4. \tag{5.11}$$

We split our analysis for $\mathcal{F}'$ into two parts due to sets F in $\mathcal{F}''$ and $\mathcal{F}' \setminus \mathcal{F}''$, respectively.

We begin with $F \in \mathcal{F}''$. Write $\kappa = \sum_{F\in\mathcal{F}'',\ y\in D^F} |\delta_{F,y}|^2 \le 1/4$. For $F \in \mathcal{F}''$, define $\delta'_{F,y} = \frac{1}{\sqrt{\kappa}}\delta_{F,y}$. Then $\sum_{F\in\mathcal{F}'', y\in D^F} |\delta'_{F,y}|^2 = 1$ and

$$\begin{aligned}
&\left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F\in\mathcal{F}''\\ y\in D^F}} \delta_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2 \\
&\quad = \kappa \left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F\in\mathcal{F}''\\ y\in D^F}} \delta'_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2 \\
&\quad \le \frac{1}{4} \left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F\in\mathcal{F}''\\ y\in D^F}} \delta'_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2 .
\end{aligned} \tag{5.12}$$

We now consider $\mathcal{F}' \setminus \mathcal{F}''$. By definition, for $F \in \mathcal{F}' \setminus \mathcal{F}''$, we have $\sum_j |F^j \cap V_{i_j}^j| < \alpha k/4$. By definition we have $\sum_j |V_{i_j}^j| = \sum_j k_j = k/4$ so F must miss more than $(1-\alpha)k/4$ elements of the set $V = \bigcup_j (V_{i_j}^j \times \{\, j \,\})$ of size $k/4$. For each subset V' of V of size $k/4 - \lfloor \alpha k/4 \rfloor$ we define a bucket $\mathcal{B}_{V'}$ that contains sets F that must miss the elements of V' and assign each $F \in \mathcal{F}' \setminus \mathcal{F}''$ to a unique bucket in an arbitrary fixed way. There are at most $2^{H_2(\alpha)k/4}$ such buckets. Then

$$\begin{aligned}
&\left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F\in\mathcal{F}'\setminus\mathcal{F}''\\ y\in D^F}} \delta_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2 \\
&\quad \le \left(\sum_{\substack{V'\subseteq V\\ |V'|=k/4-\lfloor\alpha k/4\rfloor}} \left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F\in\mathcal{B}_{V'}\\ y\in D^F}} \delta_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\| \right)^2 \\
&\quad \le 2^{H_2(\alpha)\,k/2} \sum_{\substack{V'\subseteq V\\ |V'|=k/4-\lfloor\alpha k/4\rfloor}} \left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \sum_{\substack{F\in\mathcal{B}_{V'}\\ y\in D^F}} \delta_{F,y} |y\rangle_F |\perp\rangle_{[B]\setminus F} \right\|^2
\end{aligned} \tag{5.13}$$

where we first used the triangle inequality followed by Jensen's inequality. (5.13) can be rewritten as

$$2^{H_2(\alpha)\,k/2} \sum_{\substack{V'\subseteq V\\ |V'|=k/4-\lfloor \alpha k/4\rfloor}} \left\| \Pi^A_{q(G')} \mathcal{S}_1^{\otimes n^2} \left|\perp\right\rangle_{V'} \sum_{\substack{F\in\mathcal{B}_{V'}\\ y\in D^F}} \delta_{F,y} \left|y\right\rangle_F \left|\perp\right\rangle_{[B]\setminus(F\cup V')} \right\|^2. \tag{5.14}$$

Now, applying the $\mathcal{S}_1^{\otimes n^2}$ operator in (5.14) will convert the $\left|\perp\right\rangle_{V'}$ to a uniform superposition of all $\left|y'\right\rangle_{V'}$ for all $y' \in D^{V'}$ and convert $\sum_{\substack{F\in\mathcal{B}_{V'}\\ y\in D^F}} \delta_{F,y} \left|y\right\rangle_F \left|\perp\right\rangle_{[B]\setminus(F\cup V')}$ to some superposition of $\left|y''\right\rangle \in D^{[B]\setminus V'}$ with amplitudes some $\delta_{V',y''}$ such that $\sum_{y''} |\delta_{V',y''}|^2 = \sum_{F\in\mathcal{B}_{V'},y\in D^F} |\delta_{F,y}|^2$. Therefore, we can rewrite (5.14) as

$$2^{H_2(\alpha)\,k/2} \sum_{\substack{V'\subseteq V\\ |V'|=k/4-\lfloor \alpha k/4\rfloor}} \left\| \Pi^A_{q(G')} \left[\sum_{y'\in D^{V'}} \frac{1}{\sqrt{d^{|V'|}}} \left|y'\right\rangle_{V'} \right] \otimes \sum_{y''\in D^{[n]\setminus V'}} \delta_{V',y} \left|y\right\rangle_{[B]\setminus V'} \right\|^2. \tag{5.15}$$

We now consider the application of $\Pi^A_{q(G')}$. Let $V'_j \subseteq V^j_{i_j}$ be the set of row indices in column j of $V' \subseteq [B]$ and consider the corresponding set of columns in A. Since $A_{U^j V^j_{i_j}}$ has full rank, there is a subset $U^j_0 \subseteq U^j$ with $|U^j_0| = |V'_j|$ so that $A_{U^j_0 V'_j}$ also has full rank. Now define $G'_0 \subseteq G'$ to be $\bigcup_{j\in c(G)} (U_j \times \{\, j \,\})$ which has size $|V'|$.

For each j, the outputs in $U_j \times \{\, j \,\} \subset [C]$ can be expressed as the matrix-vector product $A_{U^j_0 V'_j} y^j_{V'_j} + M'$ for some $|V'_j| \times |V'_j|$ matrix M' defined by the product of the $U^j_0 \times ([n] \setminus V'_j)$ submatrix of the fixed matrix A and $y^j_{[n]\setminus V'_j}$. Since $A_{U^j_0 V'_j}$ is full rank, for each value of M' given by $y^j_{[n]\setminus V'_j}$, there is precisely one value of $y^j_{V'_j}$ that will yield the output values $q(U_j \times \{\, j \,\})$. Therefore, putting the properties for the columns of $c(G')$ together, there is precisely one value $y' \in D^{V'}$ that will yield the output values $q(G'_0)$.

(In the terminology of section 3, this says that each of the $2^{H_2(\alpha)k/4}$ buckets $\mathcal{B}_{V'}$ corresponds to a c-admissible bucket for $q(G')$ with $c = d^{(1-\alpha)/4}$. (5.11) means that the squared amplitude of the projection on the set $\mathcal{F}''$ corresponding to recording query basis states not associated with these buckets has total squared amplitude at most $1/4$ and hence total amplitude at most $1/2$. Thus, this construction produces a t-reduction scheme of c-admissible buckets with size $\ell = 2^{H_2(\alpha)k/4}$.) It follows that (5.15) is at most

$$\begin{aligned}
&2^{H_2(\alpha)\,k/2} \sum_{\substack{V'\subseteq V\\ |V'|=k/4-\lfloor \alpha k/4\rfloor}} \left\| \Pi^A_{q(G'_0)} \left[\sum_{y'\in D^{V'}} \frac{1}{\sqrt{d^{|V'|}}} \left|y'\right\rangle_{V'} \right] \otimes \sum_{y''\in D^{[n]\setminus V'}} \delta_{V',y} \left|y\right\rangle_{[B]\setminus V'} \right\|^2 \qquad (5.16)\\
&\quad = 2^{H_2(\alpha)\,k/2} \sum_{\substack{V'\subseteq V\\ |V'|=k/4-\lfloor \alpha k/4\rfloor}} \left\| \frac{1}{\sqrt{d^{|V'|}}} \sum_{y''\in D^{[n]\setminus V'}} \delta_{V',y} \left|y\right\rangle_{[B]\setminus V'} \right\|^2\\
&\quad = 2^{H_2(\alpha)\,k/2} \sum_{\substack{V'\subseteq V\\ |V'|=k/4-\lfloor \alpha k/4\rfloor}} \frac{1}{d^{|V'|}} \sum_{y''\in D^{[n]\setminus V'}} |\delta_{V',y}|^2\\
&\quad = 2^{H_2(\alpha)\,k/2} \sum_{\substack{V'\subseteq V\\ |V'|=k/4-\lfloor \alpha k/4\rfloor}} \frac{1}{d^{|V'|}} \sum_{F\in\mathcal{B}_{V'},y\in D^F} |\delta_{F,y}|^2
\end{aligned}$$

$$= 2^{H_2(\alpha)\,k/2}\frac{1}{d^{|V'|}}\sum_{F\in\mathcal{F}'\setminus\mathcal{F}'',y\in D^F}|\delta_{F,y}|^2$$
$$\leq 2^{H_2(\alpha)\,k/2}/d^{(1-\alpha)\,k/4}$$

where the last equality follows since the buckets $\mathcal{B}_{V'}$ partition $\mathcal{F}'\setminus\mathcal{F}''$.

We now combine the contributions from $\mathcal{F}''$ and $\mathcal{F}'\setminus\mathcal{F}''$. Applying Jensen's inequality together with the bounds in (5.12) and (5.16) we obtain that

$$\left\|\Pi^A_{q(G')}\mathcal{S}_1^{\otimes n^2}\sum_{\substack{F\in\mathcal{F}'\\ y\in D^F}}\delta_{F,y}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F}\right\|^2$$
$$\leq 2\left[\left\|\Pi^A_{q(G')}\mathcal{S}_1^{\otimes n^2}\sum_{\substack{F\in\mathcal{F}'\setminus\mathcal{F}''\\ y\in D^F}}\delta_{F,y}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F}\right\|^2+\left\|\Pi^A_{q(G')}\mathcal{S}_1^{\otimes n^2}\sum_{\substack{F\in\mathcal{F}''\\ y\in D^F}}\delta_{F,y}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F}\right\|^2\right]$$
$$\leq \frac{2^{1+H_2(\alpha)\,k/2}}{d^{(1-\alpha)\,k/4}}+\frac{1}{2}\left\|\Pi^A_{q(G')}\mathcal{S}_1^{\otimes n^2}\sum_{\substack{F\in\mathcal{F}''\\ y\in D^F}}\delta'_{F,y}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F}\right\|^2$$

as required. ∎

Corollary 5.4. *Let $G'\subseteq[C]$ with $|G'|=k/4$, $\mathcal{F}'$ be a set of $F\subseteq[B]$ such that $c(G')\subseteq\mathcal{L}'(F)$, and $\sum_{F\in\mathcal{F}',y\in D^F}|\delta_{F,y}|^2=1$ for some $\delta_{F,y}$. Then*

$$\left\|\Pi^A_{q(G')}\mathcal{S}_1^{\otimes n^2}\sum_{\substack{F\in\mathcal{F}'\\ y\in D^F}}\delta_{F,y}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F}\right\|^2\leq 2^{2+H_2(4\beta)\,k/2}/d^{(1-4\beta)\,k/4}.$$

Proof. Let M be the maximum value of

$$\left\|\Pi^A_{q(G')}\mathcal{S}_1^{\otimes n^2}\sum_{F\in\mathcal{F}',\ y\in D^F}\delta_{F,y}\,|y\rangle_F\,|\perp\rangle_{[B]\setminus F}\right\|^2$$

over all choices of $\mathcal{F}'$ and $\delta_{F,y}$ with the required properties. This corollary follows from Lemma 5.3 by observing that the right-hand term in (5.10) multiplied by $1/2$ is also upper bounded by M and hence $M\leq 2^{1+H_2(4\beta)\,k/2}/d^{(1-4\beta)\,k/4}+M/2$. ∎

Finally, plugging the bound from Corollary 5.4 into (5.9), we obtain that the probability that A and B are both $(\gamma n,\gamma n)$-rigid and $\mathcal{C}$ produces k correct output values for $C=AB$, $\left\|\Pi_k\Pi_{\text{rigid}}\mathcal{S}\,|\phi_t\rangle\right\|^2$, is at most

$$16\ \min(k,n)^{\sqrt{2k}}\left(\frac{4^{H_2(4\beta)}}{d^{(1-4\beta)}}\right)^{k/4}$$

as desired. ∎

5.2. Related time-space tradeoff and cumulative memory lower bounds. Now we use Theorem 5.1 to prove some related quantum linear algebra lower bounds. Constructions of matching upper bounds can be found in section A.

Corollary 5.5. *Let $\mathbb{F}$ be a field and $D\subseteq\mathbb{F}$ with $d=|D|$. If $\mathcal{C}$ is a quantum circuit that computes the function $f:D^{n^2}\to\mathbb{F}^{n^2}$ where $f(A)=A^2$ on all upper triangular inputs in time T and space S with success probability at least $1/T$, then T must be $\Omega(n^3\sqrt{\log d\ /S})$.*

Proof. Let $A, B \in D^{n^2}$ and construct the $3n \times 3n$ matrix

$$M = \begin{bmatrix} 0 & A & 0 \\ 0 & 0 & B \\ 0 & 0 & 0 \end{bmatrix}.$$

Since the top right $n \times n$ sub-matrix of M^2 is equal to the product AB, we get a reduction from matrix multiplication and can apply Theorem 5.1 to derive the lower bound. ∎

Using Proposition 4.16 we can also bound the cumulative memory complexity for these problems.

COROLLARY 5.6. *Let $\mathbb{F}$ be a field and $D \subseteq \mathbb{F}$ with $d = |D|$. If $\mathcal{C}$ is a quantum circuit that computes the function $f : D^{2n^2} \to \mathbb{F}^{n^2}$ given by $f(A, B) = AB$ or the function $g : D^{n^2} \to \mathbb{F}^{n^2}$ given by $f(A) = A^2$, then $\mathcal{C}$ must have cumulative memory complexity $\Omega(n^6 \log(d)\ /T)$.*

Proof. For f, we apply Proposition 4.16 with Lemma 5.2 where m' is $\Theta(n^2)$, Δ is $1/2$, $h_1(n)$ is $\Theta(n)$, $K(n) = d^{-1/48}$, $C = 16$. This gives us that the cumulative memory complexity is $\Omega(n^6 \log(d)\ /T)$. Using the same reduction as in Corollary 5.5, this same lower bound applies to computing g. ∎

6. Quantum tradeoffs for Boolean matrix operations. In this section we focus on Boolean matrix operations, which use (AND, OR) inner product of vectors rather than the usual $(+, \times)$ inner product. We denote this Boolean inner product of vectors u and v by $u \bullet v$ and extend this notation to Boolean matrix-vector product and Boolean matrix multiplication. For $u, v \in \{0,1\}^n$, $u \bullet v = 1$ if and only if the subsets of $[n]$ encoded by u and v intersect, so the problems of computing Boolean matrix multiplication and Boolean matrix-vector product can be seen as computing many correlated copies of the set disjointness problem.

6.1. Tradeoffs for Boolean matrix multiplication. Unlike what we have shown for algebraic problems, as noted in [31], quantum algorithms for Boolean matrix multiplication have better time-space tradeoff properties than their classical counterparts.

PROPOSITION 6.1. *For any $c > 0$, there are quantum circuits computing $n \times n$ Boolean matrix multiplication $A \bullet B$ with error at most n^{-c} using space $O(\log n)$ and a number of queries T that is $O(n^{2.5} \log n)$.*

Proof. Fix $c > 0$. Each of the n^2 entries in the product is a disjointness function of length n that can be computed with error at most n^{-c-2} and space $O(\log n)$ using Grover's algorithm in time $O(\sqrt{n} \log n)$ for error at most n^{-c} overall. ∎

This is in contrast to the following result of Abrahamson which shows that classical algorithms as fast as this quantum algorithm require space $\tilde{\Omega}(n^{0.5})$ rather than $O(\log n)$.

PROPOSITION 6.2 ([3]). *There is a probability distribution on input matrices and constants $0 < c_1 < c_2$ under which the best classical algorithms (branching programs) for Boolean matrix multiplication $A \bullet B$ using time T and space S require that*

$$T \cdot S \text{ is } \begin{cases} \Theta(n^{3.5}) & \text{for } T \le c_1 n^{2.5} \\ \Theta(n^3) & \text{for } T \ge c_2 n^{2.5}. \end{cases}$$

For quantum circuits, Klauck, Špalek, and de Wolf [31] proved the following time-space tradeoff lower bound which proves that the quantum algorithm in Proposition 6.1 is nearly optimal when the space S is $O(\log n)$.

PROPOSITION 6.3 ([31]). *Any bounded error quantum circuit that computes the $n \times n$ Boolean matrix multiplication $A \bullet B$ with T queries and space S requires T to be $\Omega(n^{2.5}/S^{0.5})$.*

A key difference between the methods used in Abrahamson's bounds and our results for linear algebra versus those in this proof is that we require that the set of output values produced in each part of the computation is fixed independent of the input. (See our discussion of such output-oblivious computation in subsection 2.1.) Such an assumption was essential for the quantum time-space lower bounds in [31, 7], although the bound for multiple disjoint collision pairs in [27] and our results in sections 4 and 5 apply to quantum query algorithms without such a restriction on output production. Fixing the output values produced in each part of the computation allows one to go beyond using a single hard distribution on inputs, and instead choose hard distributions for each part of the computation depending on the target outputs. To give a sense of how this works we sketch the lower bound method of [31] for Boolean matrix multiplication, which relies on a strong direct product lemma for the function OR_n^k (i.e. k independent copies of the OR function each on inputs of size n):

PROPOSITION 6.4 (Strong Direct Product Theorem for OR_n^k [31]). *There are positive constants ε and γ such that the following hold:*

(a) Any randomized algorithm making at most εkn queries has success probability at most $2^{-\gamma k}$ in computing OR_n^k.

(b) Any quantum algorithm making at most $\varepsilon k\sqrt{n}$ queries has success probability at most $2^{-\gamma k}$ in computing OR_n^k.

Proof sketch for Proposition 6.3. For any integer $k \leq n/2$, the function $OR^k_{\lfloor n/k \rfloor}$ can be embedded in any set $E \subseteq [n] \times [n]$ of k outputs of the $n \times n$ Boolean matrix product $A \bullet B$ as follows: Begin by dividing $[n]$ into k blocks $b_1, \ldots, b_k$ each of size $\lfloor n/k \rfloor$ (together with at most $k-1$ other elements) and associate each $(i,j) \in E$, with a distinct index $\ell = \ell(i,j) \in [k]$. For each $(i,j) \in E$, for $\ell = \ell(i,j)$ set every entry in A_{i,b_ℓ} to 1 and set the vector of inputs in $B_{b_\ell,j}$ to the ℓ-th block of the input to $OR^k_{\lfloor n/k \rfloor}$. Set all other bits in A and B to 0. It is easy to see that the k outputs indexed by E will be the outputs for k disjoint OR functions on $\lfloor n/k \rfloor$ bits.

Without loss of generality one can assume that the space bound S is at most αn for some small constant $\alpha > 0$ since the number of queries must be $\Omega(n^2)$ in the worst case[8]. Choose $k = cS$ for some suitably large constant c that depends on the constant γ in Proposition 6.4. Begin by slicing the circuit into layers of $\varepsilon\sqrt{kn}$ queries each. There are $\Theta(T/\sqrt{kn})$ such layers. By Proposition 6.4 and the embedding, any circuit of depth $\varepsilon\sqrt{kn} = \varepsilon k\sqrt{n/k}$ queries can produce k correct output values with probability only $2^{-\gamma k}$ for some $\gamma > 0$. This is the same depth as each of the layers but each layer also gets an S qubit input-dependent state to begin. By Proposition 2.5, the probability that the resulting layer can produce k correct output values is at most $2^S 2^{-\gamma k}$ which is at most 2^{-S} if the constant c used in defining k is sufficiently large.

Therefore, the total number of correct output values that can be produced with probability larger than 2^{-S} must be $O(T/\sqrt{kn}) \cdot k$ which is $O(T\sqrt{S/n})$. On the other

[8]Note that this is not completely obvious since quantum algorithms for some problems may have a sublinear number of queries.

hand this number of outputs produced must be at least n^2. It follows that T must be $\Omega(n^{2.5}/\sqrt{S})$. □

Our improved lower bound.

THEOREM 6.5. *Any quantum circuit computing $n \times n$ Boolean matrix multiplication $A \bullet B$ with T queries and space S and success probability more than 2^{-S} must have T that is $\Omega(n^{2.5}/S^{1/4})$.*

Though the form of our lower bound may seem somewhat unusual, both the exponent of n and that of S are optimal: The algorithm of Proposition 6.1 shows that exponent of n is optimal since there is only a gap of $O(\log^{5/4} n)$ for space $\Theta(\log n)$. In our quantum query model, at the other end of the scale, an algorithm with space $3n^2$ can query and completely remember both matrices in $2n^2$ time and $2n^2$ space, after which a single global unitary transformation will produce the n^2 bits of output needed in the remaining n^2 qubits of working memory; hence the exponent of $1/4$ on S cannot be reduced.

Theorem 6.5 follows from the following key lemma which improves on the corresponding bound in [31] by a factor of $\Theta(k^{1/4})$.

LEMMA 6.6. *There are constants $\varepsilon, \gamma > 0$ such that the following holds. Let $k < n^2/100$ be an integer. For any quantum circuit $\mathcal{C}$ with at most $\varepsilon k^{3/4} n^{1/2}$ queries to x, the probability that $\mathcal{C}$ produces k correct output values of $n \times n$ Boolean matrix multiplication $A \bullet B$ is at most $2^{-\gamma k}$.*

We first see how this lemma suffices for the theorem:

Proof of Theorem 6.5 *via Lemma* 6.6. Since there are n^2 outputs, it seems that $T \geq n^2$ queries are required, but that isn't quite obvious. Nonetheless, we can, for example, derive a $T = \Omega(n^2)$ lower bound by applying Lemma 6.6 with $k = n^2/101$ which shows that a circuit with at most some βn^2 queries can only achieve exponentially small success probability for producing a small fraction of the output. Therefore without loss of generality we can assume that $\sqrt{S} < \alpha n$ for some arbitrarily small constant $\alpha > 0$. Let ε and γ be the constants from Lemma 6.6. Let $c = 2/\gamma$ and define $k = cS$. Therefore for $\alpha \leq 1/(10\sqrt{c})$ we obtain that $5\sqrt{k} = 5\sqrt{cS} < n/2$. By Lemma 6.6, since $k < n^2/100$, any quantum query algorithm with at most $\varepsilon k^{3/4} n^{1/2}$ queries has success probability at most $2^{-\gamma k} = 2^{-2S}$ of producing k correct output values.

We prove the contrapositive of the theorem statement: Suppose that $T \leq \varepsilon n^{2.5}/(cS)^{1/4} = \varepsilon n^{2.5}/k^{1/4}$. When we divide $\mathcal{C}$ into layers with $\varepsilon k^{3/4} n^{1/2}$ quantum queries each, there are at most n^2/k layers. Since there are a total of n^2 outputs, there must be some layer i during which at least k outputs are produced. Let E be the set of the first k outputs produced in layer i. By the argument above since the space is at most S, by Proposition 2.5 the probability that these k outputs are correct given the S qubits of input-dependent initial state at the beginning of layer i is at most 2^S times larger than that of a circuit without them and the same number of queries, which is at most $2^S \cdot 2^{-2S} = 2^{-S}$ which is what we needed to show. □

The main idea behind the proof of this key lemma is an improved method for embedding the direct product of OR functions into outputs of the Boolean matrix multiplication problem; this uses the following definition of an L-coloring of subsets of $[n] \times [n]$.

DEFINITION 6.7. *For $E \subseteq [n] \times [n]$ an* L-coloring *of E is a map $\chi : E \to [L]$ such that*

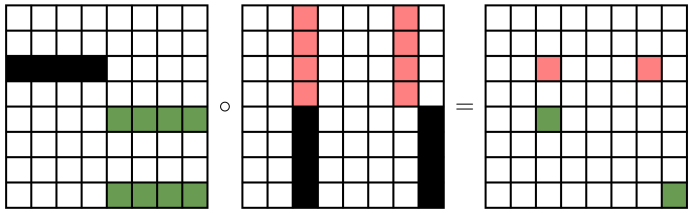

FIG. 2. *An example of a valid 3-coloring (as in Definition* 6.7*), where the pink and green squares on the right matrix correspond to the colored outputs. For the left two matrices, the black squares are fixed to the input* 1 *while the white square are fixed to the input* 0*. The pink and green squares in the left two matrices encode an input to* OR_4^4 *whose outputs are the colored entries of the right matrix.*

- *within each color class either all rows are distinct or all columns are distinct, and*
- *for each color* ℓ *there is a rectangle given by sets* $R_\ell \subseteq [n]$ *of rows and* $C_\ell \subseteq [n]$ *of columns such that the set of points of color* ℓ *is precisely* $E \cap (R_\ell \times C_\ell)$*.*

(Note that the rectangles $R_\ell \times C_\ell$ *may overlap, but their overlap must not contain any points in* E*, see Figure* 2*.)*

We say that a rectangle $R \times C \in [n] \times [n]$ *is* colorable *iff* $E \cap (R \times C)$ *either has all its elements in different rows or all its elements in different columns.*

The motivation for this definition is given by the following lemma.

LEMMA 6.8. *Let* $E \subseteq [n] \times [n]$ *with* $|E| = k$ *and* L *be an integer with* $L \leq n/2$*. If* E *has an* L*-coloring then* $OR^k_{\lfloor n/L \rfloor}$ *is a sub-function of the function that produces the* k *outputs of* $A \bullet B$ *indexed by* E *for* $n \times n$ *Boolean matrices* A *and* B*.*

Proof. Write $E = \dot{\bigcup}_{\ell=1}^{L} E_\ell$ where E_ℓ is the set of (i,j) in E in color class ℓ. We now divide $[n]$ into L disjoint blocks $b_1, \ldots, b_L$ of at least $\lfloor n/L \rfloor \geq 2$ elements each. Given the coloring and division into blocks, we define a partial assignment to the matrices A and B as follows:

- If color class ℓ consists of points that do not share a column, for each $(i,j) \in E_\ell$, we set all entries of A_{i,b_ℓ} to 1 and leave all entries of $B_{b_\ell,j}$ unset.
- If color class ℓ consists of points that do not share a row, for each $(i,j) \in E_\ell$, we set all entries of $B_{b_\ell,j}$ to 1 and leave all the entries of A_{i,b_ℓ} unset.
- All entries of A and B that are not defined by the above two cases are set to 0.

In particular, this means that if E_ℓ does not contain any element of the form $(i,\cdot)$ then the submatrix A_{i,b_ℓ} is all 0 and if E_ℓ does not contain any element of the form $(\cdot,j)$ then the submatrix $B_{b_\ell,j}$ is all 0.

It remains to show that the outputs in E of this matrix product are k disjoint ORs on at least $\lfloor n/L \rfloor$ bits each.

Observe that if the color of (i,j) is ℓ, there cannot be another color $\ell' \neq \ell$ and $i' \neq i,\ j' \neq j$ such that $(i,j'),(i',j) \in E$ both have color ℓ', as this would violate the rectangle condition for color ℓ'. This implies that either all entries of $A_{i,b_{\ell'}}$ are 0 or all entries of $B_{b_{\ell'},j}$ are 0 for all $\ell' \neq \ell$. Therefore, assuming that (i,j) is colored ℓ, the (i,j) entry of the product must equal $A_{i,b_\ell} \bullet B_{b_\ell,j}$.

If color class E_ℓ consists of points that do not share a column then the output for each $(i,j) \in E_\ell$ is the OR of the $\geq \lfloor n/L \rfloor$ unrestricted input bits of $B_{b_\ell,j}$; the inputs for different (i,j) are disjoint since no two points of E_ℓ share a column. The analogous property holds for each color class E_ℓ whose points do not share rows. In that case,

each output $(i,j) \in E_\ell$ is the OR of $\geq \lfloor n/L \rfloor$ unrestricted input bits of A_{i,b_ℓ} and input bits of A_{i,b_ℓ} are disjoint from each other. Finally, the disjointness of the inputs to the OR functions associated with different color classes is inherited from the disjointness of $b_1, \ldots, b_L$, and the lemma follows since $|E| = k$. □

The lower bound of [31] in Proposition 6.3 embedded $OR^k_{\lfloor n/k \rfloor}$ into any set E of k outputs of $A \bullet B$. Their argument corresponds to the trivial k-coloring that assigns each element of E to its own color class.

DEFINITION 6.9. *For integer $k > 0$ define $L_\alpha(k)$ to be the minimum number of colors L such that for all subsets $E \subseteq [n] \times [n]$ with $|E| \leq k$, there is an L-coloring of a subset $E' \subseteq E$ with $|E'| \geq \alpha|E|$.*

LEMMA 6.10. *There are constants $c, c' > 0$ such that the following holds. Let $\alpha > 0$ and k be an integer such that $L_\alpha(k) \leq n/2$. For any quantum circuit $\mathcal{C}$ with at most $ckn^{1/2}/L_\alpha(k)^{1/2}$ queries to x, the probability that $\mathcal{C}$ produces k correct output values of $n \times n$ Boolean matrix product $A \bullet B$ is at most $2^{-c'\alpha k}$.*

Proof. Let E be any fixed set of k output positions in $A \bullet B$. We show that for each fixed value of E the probability that $\mathcal{C}$ can correctly guess the output values at these indices is exponentially small in k. Let $L \leq L_\alpha(k)$ be such that there is an L-coloring of a subset $E' \subseteq E$ with $|E'| \geq \alpha|E|$. By Lemma 6.8, $OR^{\lceil \alpha k \rceil}_{\lfloor n/L \rfloor}$ is a sub-function of the $\lceil \alpha k \rceil$ outputs indexed by the set E'. Since $L \leq n/2$, $\lfloor n/L \rfloor \geq 2n/(3L)$ and $\sqrt{\lfloor n/L \rfloor} \geq 4\sqrt{n/L}/5$. Choose $c = 4\varepsilon\alpha/5$ and $c' = \gamma$ for ε and γ given in Proposition 6.4. By that proposition, the probability that $\mathcal{C}$ produces the values of these k outputs correctly is at most the probability that $\mathcal{C}$ produces the $\lceil \alpha k \rceil$ outputs in E' correctly which is $2^{-\gamma \lceil \alpha k \rceil} \leq 2^{-c'\alpha k}$. □

Then Lemma 6.6 is an immediate corollary of Lemma 6.10 and the following bound on $L_{1/2}(k)$.

LEMMA 6.11 (Coloring Lemma[9]). $L_{1/2}(k) \leq 2\sqrt{6k} < 5\sqrt{k}$.

Proof. Without loss of generality, E is contained in a grid with side lengths at least $n > 2\sqrt{6k}$, as otherwise we could just use a single color for each row (or column). For a given subset $A \subseteq [n]$ or rows or columns, we use $\overline{A}$ to denote $[n] \setminus A$.

Our strategy is as follows: for some constant c to be determined we show that either

1. there is a row containing at least $c\sqrt{k}$ points of E, or
2. there is a rectangle $R \times C$ such that there are at least $c\sqrt{k}$ points in the rectangle, all of which can be colored with a single color. Moreover, in this case, we show that $|(\overline{R} \times C) \cap E| \leq |(R \times C) \cap E|$.

We now argue why the above two conditions are enough to prove that $L_{1/2}(k) \leq \frac{2}{c}\sqrt{k}$.

If we colored a single row or column, then we can inductively color the remaining points of $E' \subseteq E$ outside that row/column with no issue. However, if we colored the points in $R \times C$, inductively coloring the remaining points could cause an issue because of the rectangle requirement for colors. To address this, we discard the points of $(\overline{R} \times C) \cap E$ and proceed inductively on $E' := E \cap ([n] \times \overline{C})$. At the end of the procedure, since we always color at least the number of points we discard, we will have discarded at most $k/2$ points, as desired.

[9] In a preliminary version of this paper [13] there was an error in this lemma, which claimed to show that $L_1(k) \leq 2\sqrt{6k}$. We thank the anonymous reviewers for asking the question that led to us find and address this error.

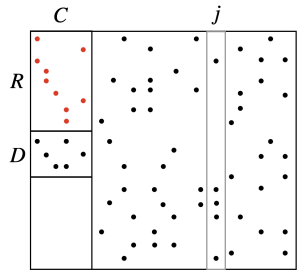

FIG. 3. *Visualization of a single iteration of Algorithm* 6.1.

It remains to show that this such a coloring would always use at most $\frac{2}{c}\sqrt{k}$ colors. We prove this using induction. Indeed, applying induction to color at least $1/2$ of the remaining $k' \leq k - c\sqrt{k}$ elements of E' in $[n] \times \overline{C}$ will require at most $\frac{2}{c}\sqrt{k'} \leq \frac{2}{c}\sqrt{k - c\sqrt{k}} \leq \frac{2}{c}\sqrt{k}(1 - \frac{c}{2\sqrt{k}}) = \frac{2}{c}\sqrt{k} - 1$ colors. It follows that at most $\frac{2}{c}\sqrt{k}$ colors are needed to color at least $1/2$ the points in E, as required.

We now show that we can execute this strategy with the constant $c = 1/\sqrt{6}$, which will prove the lemma. That is, we show how to find either a row containing at least $\sqrt{k/6}$ points of E or a colorable rectangle $R \times C$ with at least $\sqrt{k/6}$ points of E such that $|E \cap (\overline{R} \times C)| \leq |E \cap (R \times C)|$.

For any column j we write E^j for the set of i such that $(i, j) \in E$. Build $R \times C$ in the following way:[10]

Algorithm 6.1 Finding a colorable rectangle with many points.

1: Initialize $R \leftarrow \varnothing$; $C \leftarrow \varnothing$; $D \leftarrow \varnothing$
2: **while** there is a j such that $|E^j \setminus (R \cup D)| \geq \frac{3}{4}|E^j|$ **do**
3: $\quad C \leftarrow C \cup \{j\}$
4: $\quad D \leftarrow D \cup (R \cap E^j)$
5: $\quad R \leftarrow (R \setminus E^j) \cup (E^j \setminus D)$
6: **end while**
7: **return** $R \times C$

First, observe that at the end of the procedure (and indeed at the end of every iteration) the rectangle $R \times C$ contains exactly one element of E in every row, every row of $D \times C$ contains at least two elements of E, and there are no elements of E in $\overline{(R \cup D)} \times C$ – see Figure 3 for a visualization of these observations.

Our first simple claim lets us bound the number of points in $\overline{R} \times C$.

CLAIM 6.12. *$|E \cap (\overline{R} \times C)| \leq |E \cap (R \times C)|$, and $|D| \leq |R|/2$.*

Proof of Claim. The claim is true initially. Suppose that it is true at the beginning of an iteration. When we add j to C on line 3, we have $|E^j \setminus (R \cup D)| \geq 3|E^j|/4$, and therefore have $|R \cap E^j| \leq |E^j|/4$.

Line 4 therefore adds at most $|E^j|/4$ row indices to D. Since each element of $R \times C$ contained exactly one element of E at the end of the previous iteration, each row added to D by line 4 has exactly two points of E in the columns of C and there

[10] In Algorithm 6.1, instead of the constant $3/4$ in line 2, we could have chosen any $(1 - \gamma)$ instead. In this case, we would achieve a bound for $L_{1-2\gamma}(k) \leq 2\sqrt{\frac{1-\gamma}{\gamma(1-2\gamma)}k}$. For simplicity, we have chosen $\gamma = 1/4$, which is quite close to optimal and has a larger value of $\alpha = 1 - 2\gamma$.

are no points of E in $\overline{(R \cup D)} \times C$, the iteration adds at most $2|E^j|/4 = |E^j|/2$ points of E to $\overline{R} \times C$.

On the other hand, line 5 adds at least $3|E^j|/4$ elements of E^j to R and only removes the at most $|E^j|/4$ elements of $R \cap E^j$, so R grows by at least $|E^j|/2$ rows in total. Since each row of $R \times C$ has exactly one point in the columns of C, at least $|E^j|/2$ points of E get added to $R \times C$.

Counting rows, we have added at most $|E^j|/4$ rows to D and at least $|E^j|/2$ rows to R, which maintains that $|D| \leq |R|/2$.

Counting points, the increase in size of $E \cap (\overline{R} \times C)$ is at most $|E^j|/2$ which lower bounds the net gain for $E \cap (R \times C)$. This maintains $|E \cap (\overline{R} \times C)| \leq |E \cap (R \times C)|$ as required. □

We define s to be the larger of $|R|$ and the maximal number of points in E of any row. For convenience, write $Z = R \cup D$.

When Algorithm 6.1 finishes, for every column $j \in \overline{C}$, fewer than $3/4$ of its points are in rows of $\overline{Z}$ and hence more than $1/4$ of its points are in rows of Z. So we must have that

$$|E \cap (Z \times \overline{C})| > |E \cap (\overline{Z} \times \overline{C})|/3.$$

As $\overline{Z} \times C$ has no points of E and each row has at most s points of E, the total number of points is

$$\begin{aligned} k &= |E \cap (Z \times [n])| + |E \cap (\overline{Z} \times [n])| \\ &= |E \cap (Z \times [n])| + |E \cap (\overline{Z} \times \overline{C})| \\ &\leq |E \cap (Z \times [n])| + 3|E \cap (Z \times \overline{C})| \\ &\leq 4|Z|\, s \leq 4 \cdot (3|R|/2)\, s \leq 6s^2. \end{aligned}$$

Therefore $s \geq \sqrt{k/6}$. □

Lemma 6.6 is an immediate corollary of Lemmas 6.10 and 6.11 which completes the proof of Theorem 6.5.

Theorem 6.5 can be directly extended to an equivalent lower bound on the quantum cumulative memory complexity for Boolean matrix multiplication.

COROLLARY 6.13. *Any quantum circuit computing $n \times n$ Boolean matrix multiplication $A \bullet B$ with T queries, space S, and success probability more than $1/(2T)$ must have cumulative memory that is $\Omega(n^{10}/T^3)$*

Proof. Using Lemmas 6.10 and 6.11, we can apply Proposition 4.16 with $C = 1$, $m'(n) = n^2/8$, $h(k,n) = ck^{3/4}n^{1/2}/2^{1/4}$, $K(n) = 2^{c'}$ where constants $c, c' > 0$. This gives us a cumulative memory lower bound of: $\Omega(\min(n^{10}/T^3, n^4)) = \Omega(n^{10}/T^3)$ as T must be $\Omega(n^2)$. □

We also obtain a general classical lower bound from these arguments. We start by showing a classical analogue of Lemma 6.10.

LEMMA 6.14. *Let $\varepsilon, \gamma > 0$ be the constants from Proposition 6.4. Let k be an integer such that $L(k) \leq n/2$. Any randomized algorithm with at most $(2\varepsilon/3)kn/L(k)$ queries to x can only produce k correct output values of $n \times n$ Boolean matrix product $A \bullet B$ with probability at most $2^{-\gamma k}$.*

Proof. Let E be any fixed set of k output indices in $A \bullet B$. Let $L \leq L(k)$ be the smallest number such that E can be colored with L colors. By Lemma 6.8 we know that $OR^k_{\lfloor n/L \rfloor}$ is a sub-function of the outputs indexed by E. Thus, by Proposition 6.4

any randomized algorithm making at most $\varepsilon k \lfloor n/L \rfloor \geq (2\varepsilon/3)kn/L(k)$ queries can compute these outputs with probability at most $2^{-\gamma k}$. □

THEOREM 6.15. *Any output-oblivious classical query algorithm computing $n \times n$ Boolean matrix-multiplication with T queries and space S with success probability more than 2^{-S} must have T that is $\Omega(n^3/\sqrt{S})$.*

Proof. Since there are n^2 outputs, which is a trivial time lower bound for sequential algorithms, we can assume that $\sqrt{S}$ is at most αn for some arbitrarily small constant $\alpha > 0$. Let $c = 2/\gamma$ for γ given by Proposition 6.4 and let $k = cS$. Our assumption with $\alpha < 1/(10\sqrt{c})$ implies, by Lemma 6.11 that $L(k) < 5\sqrt{k} = 5\sqrt{cS} < n/2$. The main difference in parameters from the quantum case is that we need to apply Lemma 6.14 instead of Lemma 6.10 to say that classical output-oblivious branching programs of width 2^S have success probability at most $2^{-\gamma k} = 2^{-2S}$ of computing k correct output values of $A \bullet B$. There are at most 2^S nodes at a layer boundary and hence the probability that a layer of height $(2\varepsilon/3)kn/L(k)$ correctly produces k output values is at most 2^{-S}. Rewriting using $L(k) < 5\sqrt{k}$, we obtain that a layer of height $(2\varepsilon/15)\sqrt{k}\,n$ correctly produces outputs with probability at most 2^{-S}. Since there are n^2 outputs, for any circuit of depth T at most $(2\varepsilon/15)n^3/\sqrt{k}$ must have some layer of depth $2\varepsilon/15)\sqrt{k}\,n$ during which at most k outputs are produced and each output value must be correct for the algorithm to be correct, so the overall success probability is at most 2^{-S}. □

This achieves the goal suggested by Klauck, Špalek, and de Wolf [31] who ventured that the likely tight tradeoff for classical computation of Boolean matrix multiplication is $T^2S = \Omega(n^6)$. Note that our quantitative bound asymptotically dominates the bounds of Abrahamson Proposition 6.2 for all values of S; it always is at least as large (up to a constant factor) and the only regimes where our quantitative bound does not strictly dominate that of Abrahamson are when S is $\Theta(1)$ and when S is $\Theta(n)$. Of course, Abrahamson's lower bounds are for the branching program model which allows for the timing of each output bit to depend on the input. (The classical lower bound of [31] for output-oblivious query algorithms is exactly the same as that of Abrahamson for space $O(\sqrt{n})$.) Abrahamson's bound on the number of queries becomes the trivial $\Theta(n^2)$ when $S = \Theta(n^{3/2})$ which is tight for the distribution used in Abrahamson's paper, whereas the lower bound of Theorem 6.15 remains non-trivial so long as S is $o(n^2)$. In fact, just as with our quantum lower bound in Theorem 6.5, the exponents of n and S in Theorem 6.15 are optimal for a circuit model that allows arbitrary gates between queries since that would allow the circuit to simulate a decision tree of height $2n^2$ that reads and remembers the entire input and produces all of the outputs at its leaves; our lower bounds also apply to such a model. See Figure 4 for a comparison of our lower bounds with those of prior work for both classical and quantum computation.

We can extend the above to get a matching lower bound on the classical cumulative memory complexity.

COROLLARY 6.16. *Any output-oblivious classical query algorithm computing $n \times n$ Boolean matrix-multiplication with T queries and space S with success probability more than $1/(2T)$ must have cumulative memory that is $\Omega(n^6/T)$.*

Proof. Using Lemma 6.14 we can apply Proposition 4.16 with $m'(n) = n^2$, $h(k, n) = (2\epsilon/15)\sqrt{k}n$, and $K(n) = 2^{\gamma/2}$ to get that the cumulative memory must be $\Omega(\min(n^6/T, n^4)) = \Omega(n^6/T)$ as T must be $\Omega(n^2)$. □

Using the same proof idea as in Corollary 5.5, the bounds in Theorems 6.5 and 6.15 immediately imply lower bounds for Boolean matrix squaring.

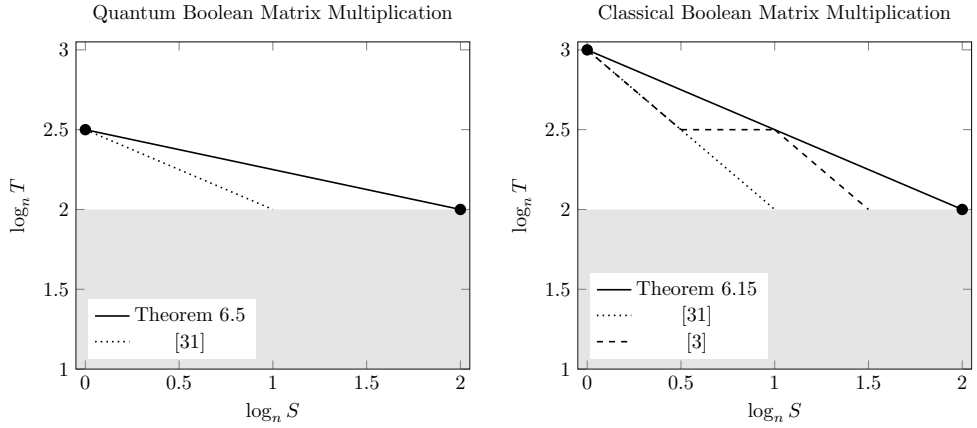

FIG. 4. *Comparison of our lower bounds for Boolean matrix multiplication with those of prior work for both quantum and classical computation. The shaded region comes from the fact that the time must always be $\Omega(n^2)$. The endpoints mark choices of parameters where the upper and lower bounds match.*

COROLLARY 6.17. *Any quantum circuit computing $n \times n$ Boolean matrix squaring on all inputs with T queries, space S, and success probability more than 2^{-S} must have T that is $\Omega(n^{2.5}/S^{1/4})$. Any such output-oblivious classical query algorithm must have T that is $\Omega(n^3/S^{1/2})$. Quantum and classical circuits for Boolean matrix squaring with success probability larger than $1/(2T)$ must have cumulative memories $\Omega(n^{10}/T^3)$ or $\Omega(n^6/T)$ respectively.*

6.2. Boolean matrix-vector product. Finally, we discuss the problem of quantum computation of Boolean matrix-vector product and the closely-associated problem of systems of linear inequalities. Here, rather than producing quantitative improvements which seem unlikely, we focus on a qualitative improvement in existing results.

Though [3] does not contain an explicit theorem statement on time-space tradeoffs for Boolean matrix-vector products that is the analog of the linear algebra bound in [4] or our Theorem 4.1, [3] contains the claim that analogous results do indeed hold for this problem using the same ideas. (The lower bound would be a factor n smaller than the lower bound for linear algebra.)

For quantum circuits, Klauck, Špalek, and de Wolf [31] prove the following results for computing Boolean matrix-vector products. (They also prove a similar result for the case of output-oblivious classical query algorithms, though that does not apply to unconstrained branching programs.)

PROPOSITION 6.18 (Theorem 23 in [31]). *For every S in $o(n/\log n)$, there is an $n \times n$ Boolean matrix $A^{(S)}$ such that every bounded-error quantum circuit with space at most S that computes Boolean matrix-vector product $A^{(S)} \bullet x$ in T queries requires that T is $\Omega(\sqrt{n^3/S}) = \Omega(n^{1.5}/S^{0.5})$.*

This result is weaker than a standard time-space tradeoff since the function involved is not independent of the circuits that might compute it. In particular, [31] does not find a single function that is hard for all space bounds, as the matrix $A^{(S)}$ that they use changes depending on the value of S. Because [31] does not express

this dependence in the statement of their results, we provide a detailed discussion of their arguments to make the need for that dependence clear. We will also need their definitions in our results.

For $S = o(n/\log n)$, the matrix $A^{(S)}$ is produced via the probabilistic method using the following distribution: Choose k to be a sufficiently large constant multiple of S. This distribution chooses matrices $A \subseteq \{0,1\}^{n\times n}$ by selecting a uniformly random subset of $n/(2k)$ positions in each row to set to 1, with the remainder of the entries in each row being 0. They show that with positive probability over the choice of A, for all sets $I \subseteq [n]$ of size k, at least $k/2$ of the rows of A_I contain at least $n/(6k)$ 1's that are unique in their column of A_I; that is, those columns are 0 in all of the $k-1$ other rows of A_I. $A^{(S)}$ is then some fixed matrix for which this property is true.

More precisely, when we fix a row $j \in I$ and the $n/(2k)$ columns where it is 1, the expected number of the $(k-1)n/(2k) < n/2$ 1's among the rows in $I \setminus \{j\}$ that land in those $n/(2k)$ columns is less than $n/(4k)$. By a Hoeffding bound, the number of those 1's is at most $n/(3k)$ except with probability exponentially small in n/k, which is $n^{-\omega(1)}$ since $k = O(S) = o(n/\log n)$. Hence, except with probability $n^{-\omega(1)}$, a row $j \in I$ is *good for* I in that at least $n/(2k) - n/(3k) = n/(6k)$ of the 1's in row j are unique in their respective columns in A_I. For a fixed I, the probability that there is no $J \subseteq I$ of size $k/2$ all of whose rows are good for I is less than the probability that there are $k/2$ rows of I that are not good for I. This happens with probability at most $n^{-\omega(k)}$ since are at most $\binom{k}{k/2}$ such subsets of rows of size $k/2$, each of which is not good for I with probability $n^{-\omega(k)}$ (and the probabilities are negatively associated). Since there are only $\binom{n}{k}$ choices of I, the total probability that A does not have desired properties is only $n^{-\omega(k)}$.

The proof of Proposition 6.18 follows from the usual time-space lower bound methodology and the following lemma:

LEMMA 6.19. *There is an $\alpha > 0$ such that for every quantum circuit $\mathcal{C}$ that makes at most $\alpha\sqrt{kn}$ queries to $x \in \{0,1\}^n$, the probability that $\mathcal{C}$ produces at least k correct output values of $A^{(S)} \bullet x$ is at most $2^{-\Omega(k)}$.*

Proof. Let $I \subseteq [n]$ be the set of indices of the first k outputs of $A^{(S)} \bullet x$ produced by $\mathcal{C}$. Let $J \subseteq I$ be the set of size $k/2$ rows that are good for I guaranteed by the properties of $A^{(S)}$. We show that the probability that $\mathcal{C}$ produces all outputs even for the rows in J is exponentially small in k: For each row $j \in J$ there is a set C_j of $n/(6k)$ columns of $A^{(S)}_I$ where the unique 1 is in row j. Consider the restriction to input vectors $x \in \{0,1\}^n$ that are 0 outside of $\bigcup_{j\in J} C_j$. Then the outputs for $j \in J$ are a direct product of $k/2$ OR functions of size $n/(6k)$ on the bits of $\bigcup_{j\in J} C_j$. By a strong direct product theorem for OR (Theorem 14 of [31]), for ε a sufficiently small constant, any circuit of height at most $\varepsilon(k/2)\sqrt{n/(6k)} = \varepsilon\sqrt{kn/24}$ is correct with probability at most $2^{-\gamma k}$ for some constant $\gamma > 0$. □

On the algorithmic side, we have the following:

PROPOSITION 6.20. *For every $c > 0$ and every Boolean matrix $A \in \{0,1\}^{m\times n}$ there is a quantum circuit using space $O(\log n)$ and time $O(mn^{1/2}\log m)$ that computes Boolean matrix-vector product $A \bullet x$ with error at most m^{-c}. More precisely, the algorithm runs in time $O(|A|_{1/2}\log m)$ where $|A|_{1/2} = \sum_{i=1}^{m}\sqrt{|A_i|_1}$.*

Proof. For each row in turn, run Grover's algorithm to compute the OR of the bits indexed by the 1's of A_i, the i-th row of A with probability of error at most m^{-c-1} per row for a total error of at most m^{-c}. □

We note that for the fixed matrix $A^{(S)}$, each row has $\Theta(n/S)$ 1's so $|A^{(S)}|_{1/2} = \Theta(n^{3/2}/S^{1/2})$. This is an odd situation in that the matrix $A^{(S)}$ designed to require large time for space S algorithms can be solved in nearly the same time bound by space $O(\log n)$ algorithms.

Systems of linear inequalities. The same space-dependent matrix $A^{(S)}$ in Proposition 6.18 was also used in [7] for systems of inequalities.

PROPOSITION 6.21 (Theorem 11 in [7]). *Let $\vec{b}$ be the length n all-b vector. For every S in $\min(O(n/b), o(n/\log n))$ there exists an $n \times n$ Boolean matrix $A^{(S)}$ such that every bounded error quantum circuit with space at most S that decides the system $A^{(S)}x \geq \vec{b}$ of n inequalities requires that T is $\Omega(\sqrt{bn^3/S})$.*

Similar to [31] this matrix is used so that any quantum circuit that computes $A^{(S)}x \geq \vec{b}$ can be broken down into slices that solve independent instances of the b-threshold function.

Our results. Using Proposition 6.18, we can obtain a time-space tradeoff lower bound for quantum computation of Boolean matrix-vector product that has an only slightly weaker lower bound in terms of the matrix dimensions but, unlike the previous bound, defines a fixed computational problem whose definition is independent of the space bound allowed.

THEOREM 6.22. *There is a fixed $m \times n$ Boolean matrix A with $m \leq n \log_2 n$ such that for every S that is $o(n/\log n)$ every bounded-error quantum circuit with space at most S that computes Boolean matrix-vector product $A \bullet x$ in T queries requires that T is $\Omega(\sqrt{n^3/S})$.*

Proof. The matrix A consists of a stacked version of the matrices $A_{(S_i)}$ from Proposition 6.18 for each choice of $S_i = 2^i \log_2 n$ and $0 \leq i \leq \log_2 n - 2\log_2 \log_2 n - \omega(1)$. Any quantum circuit computing $A \bullet x$ using space S must compute $A^{(S_i)} \bullet x$ for some S_i where $S_i \leq S$ is within factor of 2 of S. It is easy to see that the construction of $A_{(S)}$ for Proposition 6.18 is flexible in terms of the constant factor by which k exceeds S and hence computing matrix $A^{(S_i)} \bullet x$ also requires time T that is $\Omega(\sqrt{n^3/S})$ as required. ∎

Systems of linear inequalities. This same matrix A can be substituted into Proposition 6.21 to obtain a time-space tradeoff for systems of inequalities.

COROLLARY 6.23. *Let $\vec{b}$ be the length n all-b vector. There is a fixed $m \times n$ Boolean matrix A with $m \leq n \log_2 n$ such that for every S in $\min(O(n/b), o(n/\log n))$ every bounded error quantum circuit with space at most S that decides the system $Ax \geq \vec{b}$ requires T that is $\Omega(\sqrt{bn^3/S})$.*

7. Directions and open problems.

Boolean matrix multiplication. The quantum time-space tradeoff lower bounds for Boolean matrix problems, both our improved bounds and prior work, apply only to output-oblivious algorithms, unlike our lower bounds for algebraic problems. This is primarily due to the fact that all the lower bounds only use the strong direct product theorem given in Proposition 6.4 as a black box. To obtain a more general lower bound tradeoff, one would need a much more flexible kind of exponential probability decay bound that works for any individual sequence of k output values, even with partial information that is the result of a bounded number of quantum queries.

Further, to extend the quantitative lower bounds we prove in Theorems 6.5 and 6.15 to arbitrary input-independent output orders, one would also need a single fixed input

distribution for the entire branching program, rather than one that depends on the outputs being produced. The probability distribution with independent $n^{-1/2}$-biased coins used in Abrahamson's unrestricted classical lower bounds in Proposition 6.2 cannot yield such bounds since his weaker bounds are essentially optimal in expectation for this distribution. It seems likely that any such distribution will need some sort of dependence between the rows A and columns of B; otherwise, for example, relatively inexpensive estimates for the density of 1's in the rows of A and columns of B could be used to provide good predictions for all output values.

We also note that, though our lower bounds in Theorems 6.5 and 6.15 are optimal at the extremes of time and space, and hence for any fixed power relationship between time, space and input size (see Figure 4) we still do not know whether our lower bounds are optimal between the extremes: Can algorithms match our lower bounds when the space is in a middle range such as $S \in [n^{\varepsilon}, n^{2-\varepsilon}]$.

Bucketing for other multi-output problems. There are a number of other classical time-space tradeoff lower bounds for multi-output functions such as those in [2, 10, 34, 36] but no quantum time-space tradeoff lower bound is known. Can one use recording queries together with the bucketing methods that we introduce to prove analogous quantum lower bounds for these problems?

Single-output functions. Though there are some methods for classical time-space tradeoff lower bounds for single-output functions such as [11, 5, 14, 37, 30], there are no time-space tradeoff lower bounds known for quantum algorithms computing any single-output function. For example, is there any natural single-output problem where (1) quantum algorithms with unrestricted space require only a substantially sublinear number of queries, or otherwise beat the best classical algorithms, while (2) space-restricted quantum algorithms cannot? Problems such as collision-finding and element distinctness seem natural candidates for such tradeoffs.

Acknowledgments. A majority of this research was done while Niels Kornerup was attending the University of Texas at Austin.

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC (NTESS), a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration (DOE/NNSA) under contract DE-NA0003525.

This written work is authored by an employee of NTESS. The employee, not NTESS, owns the right, title and interest in and to the written work and is responsible for its contents. Any subjective views or opinions that might be expressed in the written work do not necessarily represent the views of the U.S. Government. The publisher acknowledges that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this written work or allow others to do so, for U.S. Government purposes. The DOE will provide public access to results of federally sponsored research in accordance with the DOE Public Access Plan https://www.energy.gov/doe-public-access-plan.

We would like to thank the anonymous reviewers for many helpful comments and suggestions.

REFERENCES

[1] S. Aaronson, *Limitations of quantum advice and one-way communication*, Theory Comput., 1 (2005), pp. 1–28, https://doi.org/10.4086/toc.2005.v001a001, https://theoryofcomputing.org/articles/v001a001.

[2] K. R. Abrahamson, *Generalized string matching*, SIAM J. Comput., 16 (1987), pp. 1039–1051, https://doi.org/10.1137/0216067.

[3] K. R. Abrahamson, *A time-space tradeoff for Boolean matrix multiplication*, in 31st Annual Symposium on Foundations of Computer Science, Volume I, Washington DC, United States, 1990, IEEE Computer Society, pp. 412–419, https://doi.org/10.1109/FSCS.1990.89561.

[4] K. R. Abrahamson, *Time-space tradeoffs for algebraic problems on general sequential machines*, J. Comput. System Sci., 43 (1991), pp. 269–289, https://doi.org/10.1016/0022-0000(91)90014-v.

[5] M. Ajtai, *A non-linear time lower bound for Boolean branching programs*, Theory Comput., 1 (2005), pp. 149–176, https://doi.org/10.4086/toc.2005.v001a008.

[6] A. Ambainis, *Quantum lower bounds by quantum arguments*, Journal of Computer and System Sciences, 64 (2002), pp. 750–767, https://doi.org/10.1006/jcss.2002.1826, https://www.sciencedirect.com/science/article/pii/S002200000291826X.

[7] A. Ambainis, R. Špalek, and R. de Wolf, *A new quantum lower bound method, with applications to direct product theorems and time-space tradeoffs*, Algorithmica, 55 (2009), pp. 422–461, https://doi.org/10.1007/s00453-007-9022-9.

[8] A. Bakshi and E. Tang, *An improved classical singular value transformation for quantum machine learning*, in Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '24, SIAM, 2024, pp. 2398–2453, https://doi.org/10.1137/1.9781611977912.86.

[9] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf, *Quantum lower bounds by polynomials*, J. ACM, 48 (2001), pp. 778–797, https://doi.org/10.1145/502090.502097.

[10] P. Beame, *A general sequential time-space tradeoff for finding unique elements*, SIAM J. Comput., 20 (1991), pp. 270–277, https://doi.org/10.1137/0220017.

[11] P. Beame, T. S. Jayram, and M. E. Saks, *Time-space tradeoffs for branching programs*, J. Comput. Syst. Sci., 63 (2001), pp. 542–572, https://doi.org/10.1006/jcss.2001.1778.

[12] P. Beame and N. Kornerup, *Cumulative Memory Lower Bounds for Randomized and Quantum Computation*, in 50th International Colloquium on Automata, Languages, and Programming (ICALP 2023), vol. 261, Dagstuhl, Germany, 2023, LIPIcs, pp. 17:1–17:20, https://doi.org/10.4230/LIPIcs.ICALP.2023.17, https://drops.dagstuhl.de/opus/volltexte/2023/18069.

[13] P. Beame, N. Kornerup, and M. Whitmeyer, *Quantum time-space tradeoffs for matrix problems*, in Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024, New York, NY, USA, 2024, ACM, pp. 596–607, https://doi.org/10.1145/3618260.3649700.

[14] P. Beame, M. E. Saks, X. Sun, and E. Vee, *Time-space trade-off lower bounds for randomized computation of decision problems*, J. ACM, 50 (2003), pp. 154–195, https://doi.org/10.1145/636865.636867.

[15] E. Bernstein and U. V. Vazirani, *Quantum complexity theory*, SIAM J. Comput., 26 (1997), pp. 1411–1473, https://doi.org/10.1137/S0097539796300921.

[16] A. Borodin and S. A. Cook, *A time-space tradeoff for sorting on a general sequential model of computation*, SIAM J. Comput., 11 (1982), pp. 287–297, https://doi.org/10.1137/0211022.

[17] A. Borodin, M. J. Fischer, D. G. Kirkpatrick, N. A. Lynch, and M. Tompa, *A time-space tradeoff for sorting on non-oblivious machines*, in 20th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, 1979, pp. 319–327, https://doi.org/10.1109/SFCS.1979.4.

[18] N. Chepurko, K. L. Clarkson, L. Horesh, H. Lin, and D. P. Woodruff, *Quantum-inspired algorithms from randomized numerical linear algebra*, in International Conference on Machine Learning, ICML 2022, vol. 162 of Proceedings of Machine Learning Research, PMLR, 2022, pp. 3879–3900, https://proceedings.mlr.press/v162/chepurko22a.html.

[19] N.-H. Chia, A. Gilyén, T. Li, H.-H. Lin, E. Tang, and C. Wang, *Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning*, in Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, New York, NY, USA, 2020, ACM, pp. 387–400, https://doi.org/10.1145/3357713.3384314.

[20] N.-H. Chia, A. Gilyén, H.-H. Lin, S. Lloyd, E. Tang, and C. Wang, *Quantum-Inspired Algorithms for Solving Low-Rank Linear Equation Systems with Logarithmic Dependence on the Dimension*, in 31st International Symposium on Algorithms and Computation (ISAAC 2020), vol. 181, Dagstuhl, Germany, 2020, LIPIcs, pp. 47:1–47:17, https://doi.org/10.4230/LIPIcs.ISAAC.2020.47, https://drops.dagstuhl.de/opus/volltexte/2020/13391.

[21] A. M. Childs, R. Kothari, and R. D. Somma, *Quantum algorithm for systems of linear equations with exponentially improved dependence on precision*, SIAM J. Comput., 46 (2015), pp. 1920–1950, https://api.semanticscholar.org/CorpusID:3834959.

[22] D. Deutsch and R. Jozsa, *Rapid solution of problems by quantum computation*, Proceedings of the Royal Society of London A, 439 (1992), pp. 553–558, https://doi.org/10.1098/rspa.1992.0167.

[23] A. Gilyén, Z. Song, and E. Tang, *An improved quantum-inspired algorithm for linear regression*, Quantum, 6 (2022), p. 754, https://doi.org/10.22331/q-2022-06-30-754.

[24] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, *Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics*, in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, New York, NY, USA, 2019, ACM, pp. 193–204, https://doi.org/10.1145/3313276.3316366.

[25] L. K. Grover, *A fast quantum mechanical algorithm for database search*, in Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC '96, New York, NY, USA, 1996, ACM, pp. 212–219, https://doi.org/10.1145/237814.237866.

[26] Y. Hamoudi, Q. Liu, and M. Sinha, *The NISQ complexity of collision finding*, in Advances in Cryptology - EUROCRYPT 2024 - 43rd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, May 26-30, 2024, Proceedings, Part IV, M. Joye and G. Leander, eds., vol. 14654 of Lecture Notes in Computer Science, Springer, 2024, pp. 3–32, https://doi.org/10.1007/978-3-031-58737-5_1.

[27] Y. Hamoudi and F. Magniez, *Quantum time-space tradeoff for finding multiple collision pairs*, ACM Trans. Comput. Theory, 15 (2023), pp. 1–22, https://doi.org/10.1145/3589986.

[28] A. W. Harrow, A. Hassidim, and S. Lloyd, *Quantum algorithm for linear systems of equations*, Phys. Rev. Lett., 103 (2009), https://doi.org/10.1103/physrevlett.103.150502.

[29] J. F. JáJá and J. Simon, *Space efficient algorithms for some graph theoretical problems*, Acta Informatica, 17 (1982), pp. 411–423, https://doi.org/10.1007/BF00264160.

[30] S. Jukna, *A nondeterministic space-time tradeoff for linear codes*, Inf. Process. Lett., 109 (2009), pp. 286–289, https://doi.org/10.1016/j.ipl.2008.11.001.

[31] 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 (2007), pp. 1472–1493, https://doi.org/10.1137/05063235x.

[32] Q. Liu and M. Zhandry, *On finding quantum multi-collisions*, in Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part III, vol. 11478 of Lecture Notes in Computer Science, Darmstadt, Germany, 2019, Springer, pp. 189–218, https://doi.org/10.1007/978-3-030-17659-4_7.

[33] G. H. Low and I. L. Chuang, *Hamiltonian simulation by qubitization*, Quantum, 3 (2019), p. 163, https://doi.org/10.22331/q-2019-07-12-163.

[34] Y. Mansour, N. Nisan, and P. Tiwari, *The computational complexity of universal hashing*, Theor. Comput. Sci., 107 (1993), pp. 121–133, https://doi.org/10.1016/0304-3975(93)90257-T.

[35] A. Rosmanis, *Tight bounds for inverting permutations via compressed oracle arguments*, CoRR, abs/2103.08975 (2021), https://doi.org/10.48550/arXiv.2103.08975, https://arxiv.org/abs/2103.08975, https://arxiv.org/abs/arXiv:2103.08975v2.

[36] N. Santhi and A. Vardy, *Minimum distance of codes and their branching program complexity*, in Proceedings 2006 IEEE International Symposium on Information Theory, ISIT 2006, IEEE, 2006, pp. 1490–1494, https://doi.org/10.1109/ISIT.2006.262116.

[37] M. Sauerhoff and P. Woelfel, *Time-space tradeoff lower bounds for integer multiplication and graphs of arithmetic functions*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, New York, NY, USA, 2003, ACM, pp. 186–195, https://doi.org/10.1145/780542.780571.

[38] J. E. Savage and S. Swamy, *Space-time trade-offs on the FFT algorithm*, IEEE Trans. Inf. Theory, 24 (1978), pp. 563–568, https://doi.org/10.1109/TIT.1978.1055938.

[39] A. A. Sherstov, *Strong direct product theorems for quantum communication and query complexity*, in Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, ACM, 2011, pp. 41–50, https://doi.org/10.1145/1993636.1993643.

[40] A. A. Sherstov, *Strong direct product theorems for quantum communication and query complexity*, SIAM J. Comput., 41 (2012), pp. 1122–1165, https://doi.org/10.1137/110842661.

[41] D. Simon, *On the power of quantum computation*, SIAM J. Comput., 26 (1997), pp. 1474–1483, https://doi.org/10.1137/S0097539796298637.

[42] R. Spalek, *The multiplicative quantum adversary*, in Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC 2008, IEEE Computer Society, 2008, pp. 237–248, https://doi.org/10.1109/CCC.2008.9.

[43] R. Spalek and M. Szegedy, *All quantum adversary methods are equivalent*, Theory Comput., 2 (2006), pp. 1–18, https://doi.org/10.4086/TOC.2006.V002A001.

[44] E. Tang, *A quantum-inspired classical algorithm for recommendation systems*, in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, ACM, 2019, pp. 217–228, https://doi.org/10.1145/3313276.3316310.

[45] M. Tompa, *Time-space tradeoffs for computing functions, using connectivity properties of their circuits*, in Proceedings of the Tenth Annual ACM Symposium on Theory of Computing, STOC '78, New York, NY, USA, 1978, ACM, pp. 196–204, https://doi.org/10.1145/800133.804348.

[46] Y. Yesha, *Time-space tradeoffs for matrix multiplication and the discrete Fourier transform on any general sequential random-access computer*, J. Comput. System Sci., 29 (1984), pp. 183–197, https://doi.org/10.1016/0022-0000(84)90029-1, https://www.sciencedirect.com/science/article/pii/0022000084900291.

[47] M. Zhandry, *How to record quantum queries, and applications to quantum indifferentiability*, in Advances in Cryptology – CRYPTO 2019, Cham, 2019, Springer International Publishing, pp. 239–268.

Appendix A. Deterministic query algorithms.

Here we review the matching time-space space tradeoffs that match our quantum and classical lower bounds. Most of these results were mentioned in [4] but are more fully sketched here. In the following, for simplicity, we describe versions of several of these algorithms over finite fields rather than finite subsets of size d over arbitrary fields. For the more general case, the output values are sums of products of input values and may take more bits to represent; because of this the $\log p$ in our bounds below can be replaced by $O(\max(\log d, \log n))$.

The first gives classical algorithms for matrix-vector products matching Theorem 4.1.

Proposition A.1. *Let A be any $n \times n$ matrix over a finite field $\mathbb{F}_p$. For any $S \in [\log_2 n, n \log_2 p]$ there is a deterministic classical query algorithm computing the matrix vector product $f(x) = Ax$ for all inputs $x \in \mathbb{F}_p$ that uses space S and only $O(n^2 \log p \ /S)$ queries to the input.*

Proof. Let $s = S/\log_2 p$. The query algorithm (which has the matrix A encoded in it) reads one entry of the input x at a time and maintains a block of s different partial sums (using $s \log_2 p$ space). This algorithm produces S outputs every n queries and thus produces all outputs with $n^2/s = n^2 \log_2 p \ /S$ queries. ∎

Note that in the special case of computing the Discrete Fourier Transform (Corollary 4.6), this deterministic query bound can be made explicit using standard operations:

Proposition A.2 ([38]). *There is a deterministic classical algorithm computing the Discrete Fourier Transform (DFT) $DFT_n(x) = Wx$ using space $S \geq \log_2 n$ and time $O(n^2/S + n \log S)$.*

Proof. Assume without loss of generality that S and n are powers of 2 and we have $O(S)$ space. This follows by evaluating the graph of the fast Fourier transform (FFT) algorithm for computing the DFT as shown in Figure 5. In a single pass over the input x in $O(n + S \log S)$ time the algorithm can compute the values of S of the outputs using space $O(S)$ as follows: while maintaining $\log_2(n/S) \leq S$ entries for the depth-first evaluation of each subproblem at depth $\log_2 S$ and uses space $2S$ to iterate through the top $\log_2 S$ levels which are evaluated together in a size S FFT computation. This pass is repeated for each of the n/S such blocks in turn. ∎

The following deterministic algorithms for convolution match Corollary 4.8.

Proposition A.3. *For any $S \in [\log_2 n, n \log_2 p]$ there is a deterministic classical query algorithm that computes the convolution $f(u, v) = u * v$ where $u, v \in \mathbb{F}_p^n$ that uses S space and only $O(n^2 \log p \ /S)$ queries.*

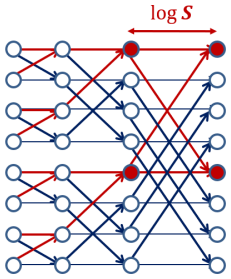

FIG. 5. *The FFT graph with the space-efficient evaluations on one pass highlighted.*

Proof. Let $s = S/(2\log_2 p)$. The indices of u, v and $w = u * v$ are reduced modulo n. The query algorithm computes outputs $w_i, \ldots w_{i+s}$ of the convolution as follows: Initialize $w_i, \ldots w_{i+s}$ to the value zero. First query and record the values of $v_{i-1}, \ldots v_{i+s-1}$. Then query values of u one at a time in increasing order $(u_1, u_2, \ldots u_n)$. After reading u_j, for each $k \in \{i, \ldots, i+s\}$, add $u_j \cdot v_{k-j}$ to the value of w_k. Then forget the value of v_{i+s-j} and query the value of v_{i-j-1}, remembering this value. After all of u has been queried, we have that $w_k = \sum_{j\in[n]} u_j v_{k-j}$ which is the correct value for these outputs. Repeating this procedure n/s times gives the convolution of u and v using only S space and $2n$ queries per iteration. Since there are n/s iterations, we get $O(n^2 \log p\ /S)$ queries. ∎

The algorithms below show that our matrix-inversion lower bound for upper-triangular matrices in Corollary 4.14 cannot be improved for large space bounds, even for deterministic query algorithms. This is open for small space bounds.

PROPOSITION A.4. *For any $S \in [n\log_2 p, n^2 \log_2 p]$ there is a deterministic classical query algorithm computing the inverse $f(A) = A^{-1}$ where $A \in \mathbb{F}_p^{n\times n}$ is a unit upper triangular matrix that uses S space and only $O(n^4 \log p\ /S)$ queries.*

Proof. Let $s = S/(2n \log p)$. We will produce columns $j_1, \ldots j_s$ of A^{-1} as follows: Let e_j be the column vector with entry 1 at index j and 0 everywhere else. We use back substitution to solve the systems $Ax_1 = e_{j_1}, \ldots, Ax_s = e_{j_s}$ by querying each entry of A exactly once. In particular, the i-th entry of x_k is $1 - \sum_{\ell\in[n-i]} A_{i,n-\ell+1}x_{n-\ell+1}$ when $i = k$ and $-\sum_{\ell\in[n-i]} A_{i,n-\ell+1}x_{n-\ell+1}$ otherwise. We start by computing the n-th entry of each x_k and work backward toward the first entry. We record each entry of each x_k as is it computed for use in the subsequent computational steps. Note that the i-th entries of all the x_k only require making queries to the i-th row of A and so all the x_k can be computed with only $O(n^2)$ queries. Finally, each x_k is output as the j_k-th column of A^{-1}. This procedure uses $O(n^2)$ queries and at most S space to produce s columns of the output. Thus the procedure must be repeated $n/s = 2n^2 \log p\ /S$ times to produce all n columns of output. This gives a total query complexity of $O(n^4 \log p\ /S)$. ∎

The following give the deterministic algorithms matching our matrix-multiplication, Boolean matrix-multiplication (Theorems 5.1 and 6.5) and squaring lower bounds (Corollaries 5.5 and 6.17).

PROPOSITION A.5. *There are deterministic query algorithms for $n \times n$ Matrix Multiplication over $\mathbb{F}_p$ using space S that make $O(n^3\sqrt{\log p}/\sqrt{S})$ queries. Further, $O(n^3/\sqrt{S})$ queries suffice for deterministic algorithms using space S to compute $n \times n$*

Boolean Matrix Multiplication.

Proof. Let $s = S/(3\log p)$. We partition each input matrix A and B into $\sqrt{s} \times \sqrt{s}$ blocks A_{ij} and B_{ij} for $i, j \in [\ell]$ where $\ell = n/\sqrt{s}$. We compute the $\sqrt{s} \times \sqrt{s}$ blocks C_{ij} of the product as follows: Initialize the block C_{ij} to 0. For $k = 1$ to ℓ, query all entries of A_{ik} and B_{kj} and add their product $A_{ik}B_{kj}$ to C_{ij}. The 3 matrices A_{ik}, B_{kj}, and C_{ij} together require space S since each entry can be expressed using $\log p$ bits. The total number of queries to compute C_{ij} is $n\sqrt{s}$ and there are $\ell^2 = n^2/s$ blocks to compute for a total of $n^3/\sqrt{s} = O(n^3\sqrt{\log p}/\sqrt{S})$ queries as claimed.

The query algorithm for Boolean Matrix Multiplication is analogous with $s = S/3$ and entry-wise $\vee$ instead of addition. ∎

Finally, we see that the matrix triple-product and cubing lower bounds in Corollaries 4.12 and 4.13 have matching deterministic query algorithms.

PROPOSITION A.6. *For any $S \in [\log_2 n, n^2 \log_2 p]$ there is a deterministic classical query algorithm computing the Matrix Triple Product $f(A, B, C) = ABC$ where $A, B, C \in \mathbb{F}_p^{n\times n}$ that uses S space and only $O(n^4 \log p\ /S)$ queries.*

Proof. Let $s = S/(4\log p)$. We view the product ABC as $(AB)C$ and use the same strategy as in Proposition A.5 to compute partial products of (AB) and then ABC. We partition the input, partial product, and output matrices into blocks $A_{ij}, B_{ij}, C_{ij}, (AB)_{ij}$, and $(ABC)_{ij}$ for $i, j \in [\ell]$ where $\ell = n/\sqrt{s}$. To compute $(AB)_{ij}$ we initialize the values in the block to zero. Then, for each $k \in [\ell]$, we query each A_{ik} and B_{kj} and then perform the multiplication of these submatrices, adding the result into $(AB)_{ij}$. After iterating over all k, we have computed the value of $(AB)_{ij}$. Now to compute $(ABC)_{ij}$ we start by initializing the values in $(ABC)_{ij}$ to zero. For each $k \in [\ell]$, we first compute $(AB)_{ik}$ as a subroutine and then query C_{kj} and add the partial product $(AB)_{ik}C_{kj}$ into $(ABC)_{ij}$. After iterating over all k, we have computed the block $(ABC)_{ij}$. This query algorithm stores at most 4 different $\sqrt{s} \times \sqrt{s}$ blocks at any time step. It requires $\sqrt{s}n$ queries to compute each $(AB)_{ij}$ and needs to compute $n/\sqrt{s}$ such blocks for each $(ABC)_{ij}$. Adding the $\sqrt{s}$ queries to C needed to compute $(ABC)_{ij}$ gives $n\sqrt{s}(1 + n/\sqrt{s})$ total queries to compute each block $(ABC)_{ij}$. Since there are n^2/s such blocks, we get $O(n^4/s)$ or $O(n^4 \log p\ /S)$ queries. ∎

Appendix B. When do good bucket reduction schemes exist?.

Our definition of t-reduction schemes for c-admissible buckets requires that there is a way to select c-admissible buckets for any quantum state $|\phi_t\rangle$ defined over Γ_t. In this section we show that a much simpler combinatorial property of the admissible buckets is sufficient to yield such schemes. As noted in section 3, none of the lower bounds for specific functions that we prove require the methods in this section.

To motivate this property, for any $G \subseteq \Gamma_t$, we can consider a state $|\phi_t^G\rangle = \sum_{x\in G} \frac{1}{\sqrt{|G|}} |x\rangle$, the uniform superposition over G. A t-reduction scheme of c-admissible buckets for q with size ℓ must yield a set $\mathcal{B} = \mathcal{B}_G$ of c-admissible buckets for q of size ℓ such that $\|\Pi_{\bigcup_{B\in\mathcal{B}_G} B} |\phi_t^G\rangle\| \geq \sqrt{3}/2$. In particular, since $|\phi_t^G\rangle$ is a uniform superposition, $\bigcup_{B\in\mathcal{B}_G} B$ must contain at least a 3/4 fraction of the elements of G. This naturally leads us to the following definition:

DEFINITION B.1. *Let q be a partial assignment of k output values. We say that q satisfies the* (c, ℓ, t) admissible bucket covering property *for q if, for every subset G of Γ_t, there is a set of at most ℓ c-admissible buckets for q that contain at least a $1/\sqrt{2}$*

fraction[11] *of the elements of* G.

We will see that merely having such a property is sufficient to yield a reduction scheme that is not much larger and works for any state $|\phi_t\rangle$ defined over Γ_t.

LEMMA B.2. *Let f be a function defined on D^n. Let $c > 1$ and q be a partial assignment of k output values of f. If the (c, ℓ, t) admissible bucket covering property holds for q then there is a t-reduction scheme of c-admissible buckets for q with size $O(t\ell \log(ne|D|/t))$.*

Proof. Let $|\phi_t\rangle = \sum_{x\in\Gamma_t} \alpha_x |x\rangle$ be a quantum state. For $\lambda = 2^{1/12}$, partition Γ_t into subsets $\Gamma_t^1, \Gamma_t^2, \ldots$ such that Γ_t^i contains the $x \in \Gamma_t$ such that $|\alpha_x| \in (\lambda^{-i}, \lambda^{-(i-1)}]$.

Let $\kappa = 24 + \lceil 6t \ \log_2(ne|D|/t)\rceil$ and let $E = \bigcup_{i>\kappa} \Gamma_t^i$ be the portion of $|\phi_t\rangle$ not associated with the first κ sets of elements of Γ_t. The norm of the projection of $|\phi_t\rangle$ on E (in other words $\|\Pi_E |\phi_t\rangle\|$) is at most

$$\sqrt{|E|} \cdot \lambda^{-\kappa} \le \sqrt{|\Gamma_t|} \cdot \lambda^{-\kappa} = \left(\sum_{j=0}^{t} \binom{n}{j} |D|^j \right)^{1/2} 2^{-\kappa/12} \tag{B.1}$$

$$\le \left(\frac{ne|D|}{t}\right)^{t/2} 2^{-\kappa/12} \le 1/4$$

by our definition of κ. The reduction we construct will definitely leave this "error" portion of $|\phi_t\rangle$ uncovered.

Associated with each Γ_t^i for $i \in [\kappa]$, we apply the (c, ℓ, t) admissible covering bucket property for q twice. The first yields at set of ℓ c-admissible buckets for q that together contain at least $1/\sqrt{2}$ fraction of the elements of Γ_t^i; we then apply the property to the $\le 1 - 1/\sqrt{2}$ fraction of elements of Γ_t^i that were not covered by the first application. Together we obtain a family $\mathcal{B}_i$, of size at most 2ℓ that contains a subset G_i consisting of at least a $\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}}(1 - \frac{1}{\sqrt{2}}) = \sqrt{2} - 1/2$ fraction of the elements of Γ_t^i. The set of c-admissible buckets for q associated with $|\phi_t\rangle$ in the t-reduction scheme is $\mathcal{B} = \bigcup_{i\in[\kappa]} \mathcal{B}_i$. The size of this family is at most $2\kappa\ell$ which is $O(t\ell \log(ne|D|/t))$ as claimed.

It remains to prove that at most amplitude $1/2$ is left after removing the portion of $|\phi_t\rangle$ covered by $\mathcal{B}$. For each $\mathcal{B}_i$, which contains the elements of G_i for $i \le \kappa$, we have

$$\sum_{x\in G_i} |\alpha_x|^2 \ge (\sqrt{2} - 1/2) \cdot |\Gamma_t^i| \lambda^{-2i}.$$

As we know that for all $x \in G_i$, $|\alpha_x| \ge \lambda^{-i}$ and $|G_i| \ge (\sqrt{2} - 1/2)|\Gamma_t^i|$. Next we can use that for all $x \in G_i$, $\lambda^{i-1} \ge |\alpha_x|$ to get

$$\sum_{x\in G_i} |\alpha_x|^2 \ge \lambda^{-2}(\sqrt{2} - 1/2) \cdot \sum_{x\in\Gamma_t^i} |\alpha_x|^2$$

$$= (2^{1/3} - 2^{-7/6}) \cdot \sum_{x\in\Gamma_t^i} |\alpha_x|^2.$$

Now let $\Gamma_t' = \bigcup_{B\in\mathcal{B}} B$. Since $|\phi_t\rangle$ is a normalized vector supported by basis state in

[11] While a 3/4 fraction of the elements are covered by a t-reduction scheme of c-admissible buckets, we do not need to cover the same fraction of elements in this definition. We choose a fraction that is more convenient here.

Γ_t,

$$\begin{aligned}\|\Pi_{\Gamma'_t}|\phi_t\rangle\|^2 &= \sum_{\substack{i\in[\kappa]\\ x\in G_i}} |\alpha_x|^2 \\ &\geq (2^{1/3}-2^{-7/6}) \sum_{\substack{i\in[\kappa]\\ x\in \Gamma^i_t}} |\alpha_x|^2 \\ &= (2^{1/3}-2^{-7/6})\,\|\Pi_{\Gamma_t\setminus E}|\phi_t\rangle\|^2.\end{aligned}$$

Now since $|\phi_t\rangle$ is a quantum state with a 2-norm of 1 and all of its amplitude is on basis elements in Γ_t, we have that $\|\Pi_{\Gamma_t\setminus E}|\phi_t\rangle\|^2 + \|\Pi_E|\phi_t\rangle\|^2 = 1$ and so the above can be rewritten as

$$\begin{aligned}\|\Pi_{\Gamma'_t}|\phi_t\rangle\| &= (2^{1/3}-2^{-7/6})(1-\|\Pi_E|\phi_t\rangle\|^2) \\ &\geq (2^{1/3}-2^{-7/6})\,15/16 \\ &> 3/4\end{aligned}$$

where the middle inequality follows directly from (B.1). This directly implies that $\|\Pi_{\Gamma_t\setminus\Gamma'_t}|\phi_t\rangle\| < 1/2$ as required. □