2015 IEEE 56th Annual Symposium on Foundations of Computer Science

Hamiltonian simulation with nearly optimal dependence on all parameters

Dominic W. Berry*, Andrew M. Childs†‡ and Robin Kothari§¶
* *Department of Physics and Astronomy, Macquarie University, Sydney, Australia*
Email: dominic.berry@mq.edu.au
† *Department of Combinatorics & Optimization and Institute for Quantum Computing*
University of Waterloo, Waterloo, Ontario, Canada
‡ *Department of Computer Science, Institute for Advanced Computer Studies,*
and Joint Center for Quantum Information and Computer Science
University of Maryland, College Park, Maryland, USA
Email: amchilds@umd.edu
§ *David R. Cheriton School of Computer Science and Institute for Quantum Computing*
University of Waterloo, Waterloo, Ontario, Canada
¶ *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA*
Email: rkothari@mit.edu

Abstract

We present an algorithm for sparse Hamiltonian simulation whose complexity is optimal (up to log factors) as a function of all parameters of interest. Previous algorithms had optimal or near-optimal scaling in some parameters at the cost of poor scaling in others. Hamiltonian simulation via a quantum walk has optimal dependence on the sparsity at the expense of poor scaling in the allowed error. In contrast, an approach based on fractional-query simulation provides optimal scaling in the error at the expense of poor scaling in the sparsity. Here we combine the two approaches, achieving the best features of both. By implementing a linear combination of quantum walk steps with coefficients given by Bessel functions, our algorithm's complexity (as measured by the number of queries and 2-qubit gates) is logarithmic in the inverse error, and nearly linear in the product tau of the evolution time, the sparsity, and the magnitude of the largest entry of the Hamiltonian. Our dependence on the error is optimal, and we prove a new lower bound showing that no algorithm can have sublinear dependence on tau.

Keywords

Hamiltonian simulation; quantum algorithms

I. Introduction

The problem of simulating the dynamics of quantum systems was the original motivation for quantum computers [1] and remains one of their major potential applications. Although classical algorithms for this problem are inefficient, a significant fraction of the world's computing power today is spent in solving instances of this problem that arise in, e.g., quantum chemistry and materials science [2, 3]. Furthermore, efficient classical algorithms for this problem are unlikely to exist: since the simulation problem is BQP-complete [4], an efficient classical algorithm for quantum simulation would imply an efficient classical algorithm for any problem with an efficient quantum algorithm (e.g., integer factorization [5]).

The first explicit quantum simulation algorithm, due to Lloyd [6], gave a method for simulating Hamiltonians that are sums of local interaction terms. Aharonov and Ta-Shma gave an efficient simulation algorithm for the more general class of sparse Hamiltonians [7], and much subsequent work has given improved simulations [8, 9, 10, 11, 12, 13, 14, 15, 16]. Sparse Hamiltonians include most physically realistic Hamiltonians as a special case (making these algorithms potentially useful for simulating real-world systems). In addition, sparse Hamiltonian simulation can be used to design other quantum algorithms [17, 18, 19]. For example, it was used to convert the algorithm for evaluating a balanced binary NAND tree with n leaves [20] to the discrete-query model [19].

0272-5428/15 $31.00 © 2015 IEEE
DOI 10.1109/FOCS.2015.54

In the Hamiltonian simulation problem, we are given an n-qubit Hamiltonian H (a Hermitian matrix of size $2^n \times 2^n$), an evolution time t, and a precision $\epsilon > 0$, and are asked to implement the unitary operation e^{-iHt} up to error at most ϵ (as quantified by the diamond norm distance). That is, the task is to implement a unitary operation, rather than simply to generate [21] or convert [22] a quantum state. We say that H is d-sparse if it has at most d nonzero entries in any row. In the sparse Hamiltonian simulation problem, H is specified by a black box that takes input $(j, \ell) \in [2^n] \times [d]$ (where $[d] := \{1, \ldots, d\}$) and outputs the location and value of the ℓth nonzero entry in the jth row of H. Specifically, as in [13], we assume access to an oracle O_H acting as

$$O_H|j, k, z\rangle = |j, k, z \oplus H_{jk}\rangle \tag{1}$$

for $j, k \in [2^n]$ and bit strings z representing entries of H, and another oracle O_F acting as

$$O_F|j, \ell\rangle = |j, f(j, \ell)\rangle, \tag{2}$$

where $f(j, \ell)\colon [2^n] \times [d] \to [2^n]$ is a function giving the column index of the ℓth nonzero element in row j. Note that the form of O_F assumes that the locations of the nonzero entries of H can be computed in place. This is possible if we can efficiently compute both $(j, \ell) \mapsto f(j, \ell)$ and the reverse map $(j, f(j, \ell)) \mapsto \ell$, which holds in typical applications of sparse Hamiltonian simulation. Alternatively, if f provides the nonzero elements in order, we can compute the reverse map with only a $\log d$ overhead by binary search.

At present, the best algorithms for sparse Hamiltonian simulation, in terms of query complexity (i.e., the number of queries made to the oracles) and number of 2-qubit gates used, are one based on a Szegedy quantum walk [11, 13] and another based on simulating an unconventional model of query complexity called the fractional-query model [15]. An algorithm with similar complexity to [15] is based on implementing a Taylor series of the exponential [16]. The quantum walk approach has query complexity $O(d\|H\|_{\max}t/\sqrt{\epsilon})$, which is linear in both the sparsity d and the evolution time t. (Here $\|H\|_{\max}$ denotes the largest entry of H in absolute value.) However, this approach has poor dependence on the allowed error ϵ. In contrast, the fractional-query approach has query complexity $O\big(d\tau\frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\big)$, where $\tau := d\|H\|_{\max}t$. This approach gives exponentially better dependence on the error at the expense of quadratically worse dependence on the sparsity. Considering the fundamental importance of quantum simulation, it is desirable to have a method that achieves the best features of both approaches. In this work, we combine the two approaches, giving the following.

Theorem 1. *A d-sparse Hamiltonian H acting on n qubits can be simulated for time t within error ϵ with*

$$O\left(\tau\frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right) \tag{3}$$

queries and

$$O\left(\tau[n + \log^{5/2}(\tau/\epsilon)]\frac{\log(\tau/\epsilon)}{\log\log(\tau/\epsilon)}\right) \tag{4}$$

additional 2-qubit gates, where $\tau := d\|H\|_{\max}t$.

This result provides a strict improvement over the query complexity of [15, 16], removing a factor of d in τ, and thus providing near-linear instead of superquadratic dependence on d.

We also prove a lower bound showing that any algorithm must use $\Omega(\tau)$ queries. While a lower bound of $\Omega(t)$ was known previously [9], our new lower bound shows that the complexity must be at least linear in the product of the sparsity and the evolution time. Our proof is similar to a previous limitation on the ability of quantum computers to simulate non-sparse Hamiltonians [23]: by replacing each edge in the graph of the Hamiltonian by a complete bipartite graph $K_{d,d}$, we effectively boost the strength of the Hamiltonian by a factor of d at the cost of making the matrix less sparse by a factor of d. Combining this result with the error-dependent lower bound of [15], we find a lower bound as follows.

Theorem 2. *For any $\epsilon, t > 0$, integer $d \geq 2$, and fixed value of $\|H\|_{\max}$, there exists a d-sparse Hamiltonian H such that simulating H for time t with precision ϵ has query complexity*

$$\Omega\left(\tau + \frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right). \tag{5}$$

Thus our result is near-optimal for the scaling in either τ or ϵ on its own. However, our upper bound (3) has a product, whereas the lower bound (5) has a sum. It remains an open question how to close the gap between these bounds. Intriguingly, a slight modification of our technique gives another algorithm with the following complexity.

Theorem 3. *For any $\alpha \in (0, 1]$, a d-sparse Hamiltonian H acting on n qubits can be simulated for time t within error ϵ with query complexity*

$$O\big(\tau^{1+\alpha/2} + \tau^{1-\alpha/2}\log(1/\epsilon)\big). \tag{6}$$

This result provides a nontrivial tradeoff between the parameters t, d, and ϵ, and suggests that further improvements to such tradeoffs may be possible.

We now informally describe the key idea behind our algorithms. For simplicity, suppose that the entries of the Hamiltonian are small, satisfying $\|H\|_{\max} \leq 1/d$, and $t = 1$. Previous work on Hamiltonian simulation [11, 13] has shown that using a constant number of queries, we can construct a unitary U whose top-left block (in some basis) is exactly $e^{-i\arcsin(H)}$. Technical difficulties aside, the essential problem is to implement the unitary e^{-iH} given the ability to perform $e^{-i\arcsin(H)}$. While it is not clear how to express e^{-iH} as a product of easy-to-implement unitaries and $e^{-i\arcsin(H)}$, it can be approximated by a linear combination of powers of $e^{-i\arcsin(H)}$. Although such a decomposition may not seem natural, we show that nevertheless it leads to an efficient implementation.

In the next section we present a more technical overview of this high-level idea. In Section III we analyze and prove the correctness of our algorithms. Section IV proves the lower bound presented in Theorem 2 and we conclude with some discussion in Section V.

II. Overview of algorithms

Our algorithm uses a Szegedy quantum walk as in [11, 13], but with a linear combination of different numbers of steps. Such an operation can be implemented using the techniques that were developed to simulate the fractional-query model [15]. This allows us to introduce a desired phase more accurately than with the phase estimation approach of [11, 13]. As in [15], we first implement the approximated evolution for some time interval with some amplitude and then use oblivious amplitude amplification to make the implementation deterministic, facilitating simulations for longer times. In the rest of this section, we describe the approach in more detail.

References [11, 13] define a quantum walk step U that depends on the Hamiltonian H to be simulated. In turn, this quantum walk step is based on a state preparation procedure that only requires one call to the sparse Hamiltonian oracle, avoiding the need to decompose H into a sum of terms as in product-formula approaches. Two copies of the Hilbert space acted on by H are used. First, the initial state is in one of these Hilbert spaces. Then, the state preparation procedure is used to map the initial state onto the joint Hilbert space. This state preparation acts on the second copy of the Hilbert space, controlled by the state in the first Hilbert space. The quantum walk steps take place in this joint Hilbert space. Finally, the controlled state preparation is inverted to map the final state back to the first Hilbert space.

In the controlled state preparation, each eigenstate of H is mapped onto a superposition of two eigenstates $|\mu_\pm\rangle$ of the quantum walk step U. The precise definition of U is not needed here; for our application, it suffices to observe that the eigenvalues $\mu_\pm$ of U are related to the eigenvalues λ of H via

$$\mu_\pm = \pm e^{\pm i\arcsin(\lambda/Xd)}, \tag{7}$$

where $X \geq \|H\|_{\max}$ is a parameter that can be increased to make the steps of the quantum walk closer to the identity. For small λ/Xd, the steps of the quantum walk yield a phase factor that is nearly proportional to that

for the Hamiltonian evolution. However, the phase deviates from the desired value since the function $\arcsin \nu$ is not precisely linear about $\nu = 0$. Also, there are two eigenvalues $\mu_\pm$, and in previous approaches it was necessary to distinguish between these to approximate Hamiltonian evolution [11, 13]. In contrast, for the new technique we present here it is not necessary to distinguish the eigenspaces.

An obvious way to increase the accuracy is to increase X above its minimum value of $\|H\|_{\max}$. However, the number of steps of the quantum walk is $O(tXd)$, so increasing X results in a less efficient simulation. Another approach is to use phase estimation to correct the phase factor [11, 13], but this approach still gives polynomial dependence on $1/\epsilon$.

Instead, we propose using a superposition of steps of the quantum walk to effectively linearize the $\arcsin$ function. Specifically, rather than applying U, we apply

$$V_k := \sum_{m=-k}^{k} a_m U^m \tag{8}$$

for some coefficients $a_{-k}, \ldots, a_k$. We show that the coefficients can be chosen by considering the generating function for the Bessel function [24, 9.1.41],

$$\sum_{m=-\infty}^{\infty} J_m(z)\mu_\pm^m = \exp\left[\frac{z}{2}\left(\mu_\pm - \frac{1}{\mu_\pm}\right)\right] = e^{i\lambda z/Xd}, \tag{9}$$

where the second equality follows from (7). Because the right-hand-side does not depend on whether the eigenvalue of U is μ_+ or μ_-, there is no need to distinguish the eigenspaces. Thus the ability to perform the operation

$$\sum_{m=-\infty}^{\infty} J_m(z)U^m \tag{10}$$

would allow us to exactly implement the evolution under H for time $-z/Xd$. Because of the minus sign, we will take z to be negative to obtain positive time. By truncating the sum in (10) to some finite range $\{-k, \ldots, k\}$, we obtain an expression in which each term can be performed using at most k queries. Because the Bessel function falls off exponentially for large $|m|$, we can obtain error at most ϵ with a cutoff k that is only logarithmic in $1/\epsilon$.

A linear combination of unitaries (LCU) such as (8) can be implemented using the LCU Lemma (Lemma 4) described in the next section. The high-level intuition for the procedure is as follows. We prepare ancilla qubits in a superposition encoding the coefficients of the linear combination and then perform the unitary operations of the linear combination in superposition, controlled by the ancilla. One could then obtain V_k by postselecting on an appropriate ancilla state. Instead, to obtain V_k deterministically, we apply the oblivious amplitude amplification procedure introduced in [15]. Rather than using V_k to implement evolution over the entire time, we break the time up into shorter time steps we call "segments" (named by analogy to the segments used in [15]) and use V_k to achieve the time evolution for each segment.

The complexity of our algorithm is the number of segments ($tXd/|z|$) times the complexity for each segment (k) times the number of steps needed for oblivious amplitude amplification (a). We have some freedom in choosing z, which controls the amount of evolution time simulated by each segment. To obtain near-linear dependence on the evolution time t, we choose $z = O(1)$. Then amplitude amplification requires $O(1)$ steps, and the number of segments needed is $O(\tau)$, giving the linear factor in (3). The value of k needed to achieve overall error at most ϵ is logarithmic in τ/ϵ, yielding the logarithmic factor in (3).

An alternative approach is to use a larger segment that scales with τ. Choosing $z = -\tau^\alpha$ for $\alpha \in (0, 1]$, we need $k = O(\tau^\alpha + \log(1/\epsilon))$. Then we require $O(\tau^{1-\alpha})$ segments and $O(\tau^{\alpha/2})$ steps of amplitude amplification, giving the scaling presented in Theorem 3.

III. Analysis of algorithms

A. A quantum walk for any Hamiltonian

We begin by reviewing the quantum walk defined in [11, 15]. Given a Hamiltonian H acting on $\mathbb{C}^N$ (where $N := 2^n$), the Hilbert space is expanded from $\mathbb{C}^N$ to $\mathbb{C}^{2N} \otimes \mathbb{C}^{2N}$. First, an ancilla qubit in the state $|0\rangle$ is appended, which expands the space from $\mathbb{C}^N$ to $\mathbb{C}^{2N}$. Then the entire Hilbert space is duplicated, giving $\mathbb{C}^{2N} \otimes \mathbb{C}^{2N}$. This is achieved using the isometry

$$T := \sum_{j=0}^{N-1} \sum_{b \in \{0,1\}} (|j\rangle\langle j| \otimes |b\rangle\langle b|) \otimes |\varphi_{jb}\rangle \tag{11}$$

with $|\varphi_{j1}\rangle = |0\rangle|1\rangle$ and

$$|\varphi_{j0}\rangle := \frac{1}{\sqrt{d}} \sum_{\ell \in F_j} |\ell\rangle \left(\sqrt{\frac{H_{j\ell}^*}{X}} |0\rangle + \sqrt{1 - \frac{|H_{j\ell}^*|}{X}} |1\rangle \right), \tag{12}$$

where $X \geq \|H\|_{\max}$ and F_j is the set of indices of nonzero elements in column j of H. Here we use the convention that the first subsystem is the original space, the next is the ancilla qubit, and the third and fourth subsystems are the duplicated space and duplicated ancilla qubit, respectively. This operation can be viewed as a controlled state preparation, creating state $|\varphi_{j0}\rangle$ on input $|j\rangle|0\rangle$. If the ancilla qubit is in the state $|1\rangle$, then $|0\rangle|1\rangle$ is prepared. Starting with the initial space, the controlled state preparation is performed, and then steps of the quantum walk are applied using the unitary

$$U := iS(2TT^\dagger - \mathbb{I}), \tag{13}$$

where S swaps the two registers (i.e., $S|j_1\rangle|j_2\rangle|\ell_1\rangle|\ell_2\rangle = |\ell_1\rangle|\ell_2\rangle|j_1\rangle|j_2\rangle$ for all $j_1, \ell_1 \in [N]$, $j_2, \ell_2 \in \{0,1\}$). Finally, the inverse state preparation $T^\dagger$ is performed. For a successful simulation, the output should lie in the original space, and the ancilla should be returned to the state $|0\rangle$.

Let λ be the eigenvalue of H with eigenstate $|\lambda\rangle$, and let $\nu := \lambda/Xd$ be the corresponding scaled eigenvalue for the quantum walk. The steps of the quantum walk U satisfy $U|\mu_\pm\rangle = \mu_\pm|\mu_\pm\rangle$ [11] with

$$|\mu_\pm\rangle := (T + i\mu_\pm ST)|\lambda\rangle, \tag{14}$$

$$\mu_\pm := \pm\sqrt{1 - \nu^2} + i\nu = \pm e^{\pm i \arcsin \nu}. \tag{15}$$

To apply the steps of the quantum walk to approximate Hamiltonian evolution, there are two challenges: we must handle both the $|\mu_+\rangle$ and $|\mu_-\rangle$ sectors, and correct the applied phase. In this work we are able to solve both these challenges at once by using a superposition of steps of the quantum walk.

B. Linear combination of unitaries

We now describe how to perform a linear combination of unitary operations. Given an M-tuple of unitary operations $\vec{U} = (U_1, \ldots, U_M)$, we quantify the complexity of implementing a linear combination of the U_ms in terms of the number of invocations of

$$\mathrm{select}(\vec{U}) := \sum_{m=1}^{M} |m\rangle\langle m| \otimes U_m. \tag{16}$$

Such a result was previously given in [16, 25]. Here we formalize that result and generalize to allow more steps of oblivious amplitude amplification. The overall result is as given in the following lemma.

Lemma 4 (LCU Lemma)**.** *Let $\vec{U} = (U_1, \ldots, U_M)$ be unitary operations and let $\tilde{V} = \sum_{m=1}^{M} a_m U_m$ be δ-close to a unitary. We can approximate $\tilde{V}$ to within $O(\delta)$ using $O(a)$ $\mathrm{select}(\vec{U})$ and $\mathrm{select}(\vec{U}^\dagger)$ operations and $O(Ma)$ additional 2-qubit gates, where $a := \sum_{m=1}^{M} |a_m|$.*

To prove this result, we first consider an operation that would give $\tilde{V}$ with postselection, then apply oblivious amplitude amplification to achieve it deterministically. The operation that provides $\tilde{V}$ with postselection is described in the following lemma.

Lemma 5. *Let $\vec{U} = (U_1, \ldots, U_M)$ be unitary operations acting on a Hilbert space $\mathcal{H}_2$, let $\tilde{V} = \sum_{m=1}^{M} a_m U_m$, and let $s \geq \sum_{m=1}^{M} |a_m|$ be a real number. Define a Hilbert space $\mathcal{H}_1$ to be a tensor product of a qubit and a subspace of dimension M, and let $|\varsigma\rangle := |0\rangle|0\rangle \in \mathcal{H}_1$. Then there exists a unitary operation W acting on $\mathcal{H}_1 \otimes \mathcal{H}_2$ such that $Z = \frac{1}{s}(|\varsigma\rangle\langle\varsigma| \otimes \tilde{V})$, with $Z := PWP$, $P := |\varsigma\rangle\langle\varsigma| \otimes \mathbb{I}$. The operation W can be applied with $O(1)$ $\mathrm{select}(\vec{U})$ and $\mathrm{select}(\vec{U}^\dagger)$ operations and $O(M)$ additional 2-qubit gates.*

Proof: To perform W, we perform an operation that rotates the ancilla qubits from $|\varsigma\rangle$ to the state

$$|\chi\rangle = \left(\sqrt{\frac{a}{s}}|0\rangle + \sqrt{1 - \frac{a}{s}}|1\rangle\right) \otimes \frac{1}{\sqrt{a}} \sum_{m=1}^{M} \sqrt{a_m}|m\rangle, \tag{17}$$

where $a := \sum_{m=1}^{M} |a_m|$. This state is of dimension $2M$ and can be prepared from state $|\varsigma\rangle$ using $O(M)$ operations (which is trivial for $|m\rangle$ encoded in unary). Next we perform the controlled operation $\mathrm{select}(\vec{U})$. Finally, inverting the preparation of $|\chi\rangle$ and projecting onto $|\varsigma\rangle$ would effectively project the ancilla onto $|\chi\rangle$. Then the unnormalized operation on $\mathcal{H}_2$ is $\tilde{V}/s$, corresponding to Z. The action of applying the unitary operation to prepare $|\chi\rangle$, the controlled operation $\mathrm{select}(\vec{U})$, and the inverse preparation gives the desired operation W. ∎

Next we provide a multi-step version of robust amplitude amplification, generalizing the single-step version presented in [16]. In this lemma, and throughout this paper, $\|\cdot\|$ denotes the spectral norm.

Lemma 6 (Robust oblivious amplitude amplification)**.** *Let W be a unitary matrix acting on $\mathcal{H}_1 \otimes \mathcal{H}_2$ and let P be the projector onto the subspace whose first register is $|\varsigma\rangle := |0\rangle|0\rangle \in \mathcal{H}_1$, i.e., $P := |\varsigma\rangle\langle\varsigma| \otimes \mathbb{I}$. Furthermore let $Z := PWP$ satisfy $Z = \frac{1}{s}(|\varsigma\rangle\langle\varsigma| \otimes \tilde{V})$, where $\tilde{V}$ is δ-close to a unitary matrix and $\sin\left(\frac{\pi}{2(2\ell+1)}\right) = \frac{1}{s}$ for some $\ell \in \mathbb{N}$, and let $R := -W(\mathbb{I} - 2P)W^\dagger(\mathbb{I} - 2P)$. Then*

$$\|PR^\ell WP - (|\varsigma\rangle\langle\varsigma| \otimes \tilde{V})\| = O(\delta). \tag{18}$$

Proof: We start by considering a single iteration, as in [16]. Then we have

$$\begin{aligned} RWP &= -W(\mathbb{I} - 2P)W^\dagger(\mathbb{I} - 2P)WP \\ &= -WP + 2WP + 2PWP - 4WPW^\dagger PWP \\ &= WP + 2Z - 4WZ^\dagger Z. \end{aligned} \tag{19}$$

Multiplying by P on the left gives

$$PRWP = -PW(\mathbb{I} - 2P)W^\dagger(\mathbb{I} - 2P)WP = 3Z - 4ZZ^\dagger Z, \tag{20}$$

which matches the expression in [16].

The general solution after m iterations is

$$R^m WP = (WP - Z)\frac{T_{2m+1}(\sqrt{1 - Z^\dagger Z})}{\sqrt{1 - Z^\dagger Z}} + Z\frac{(-1)^m T_{2m+1}(\sqrt{Z^\dagger Z})}{\sqrt{Z^\dagger Z}}, \tag{21}$$

where T_{2m+1} are Chebyshev polynomials of the first kind. (Because Chebyshev polynomials for odd order only include odd powers, no square roots appear when (21) is expanded.)

We establish (21) by induction. First note that it holds for $m = 0$, because $T_1(x) = x$, so the right-hand side evaluates to WP. Next assume that it holds for a given m. It is straightforward to show that

$$\begin{aligned} R(WP - Z) &= (WP - Z)(1 - 2Z^\dagger Z) + 2Z(1 - Z^\dagger Z) \quad \text{and} \\ RZ &= (WP - Z)(-2Z^\dagger Z) + Z(1 - 2Z^\dagger Z). \end{aligned} \tag{22}$$

Hence, multiplying both sides of (21) by R, we get

$$\begin{aligned} R^{m+1}WP &= R\left[(WP-Z)\frac{T_{2m+1}(\sqrt{1-Z^\dagger Z})}{\sqrt{1-Z^\dagger Z}} + Z\frac{(-1)^m T_{2m+1}(\sqrt{Z^\dagger Z})}{\sqrt{Z^\dagger Z}}\right] \\ &= (WP-Z)\left[(1-2Z^\dagger Z)\frac{T_{2m+1}(\sqrt{1-Z^\dagger Z})}{\sqrt{1-Z^\dagger Z}} - 2Z^\dagger Z\frac{(-1)^m T_{2m+1}(\sqrt{Z^\dagger Z})}{\sqrt{Z^\dagger Z}}\right] \\ &\quad + Z\left[2(1-Z^\dagger Z)\frac{T_{2m+1}(\sqrt{1-Z^\dagger Z})}{\sqrt{1-Z^\dagger Z}} + (1-2Z^\dagger Z)\frac{(-1)^m T_{2m+1}(\sqrt{Z^\dagger Z})}{\sqrt{Z^\dagger Z}}\right]. \end{aligned} \tag{23}$$

To progress further, we use the relation [24, 22.3.15]

$$T_{2m+1}(x) = \cos[(2m+1)\arccos x] = (-1)^m \sin[(2m+1)\arcsin x]. \tag{24}$$

Using this we find that, with $x = \sin\theta$,

$$\begin{aligned} (1-2x^2)\frac{T_{2m+1}(\sqrt{1-x^2})}{\sqrt{1-x^2}} - 2x^2\frac{(-1)^m T_{2m+1}(x)}{x} &= (\cos^2\theta - \sin^2\theta)\frac{\cos[(2m+1)\theta]}{\cos\theta} - 2\sin\theta\sin[(2m+1)\theta] \\ &= \frac{\cos(2\theta)\cos[(2m+1)\theta] - \sin(2\theta)\sin[(2m+1)\theta]}{\cos\theta} \\ &= \frac{\cos[(2m+3)\theta]}{\cos\theta} \\ &= \frac{T_{2m+3}(\sqrt{1-x^2})}{\sqrt{1-x^2}}. \end{aligned} \tag{25}$$

Next put $x = \cos\phi$ to obtain

$$\begin{aligned} &2(1-x^2)\frac{T_{2m+1}(\sqrt{1-x^2})}{\sqrt{1-x^2}} + (1-2x^2)\frac{(-1)^m T_{2m+1}(x)}{x} \\ &= 2(\sin^2\phi)\frac{(-1)^m\sin[(2m+1)\phi]}{\sin\phi} - (\cos^2\phi - \sin^2\phi)\frac{(-1)^m\cos[(2m+1)\phi]}{\cos\phi} \\ &= (-1)^m\frac{\sin(2\phi)\sin[(2m+1)\phi] - \cos(2\phi)\cos[(2m+1)\phi]}{\cos\phi} \\ &= (-1)^{m+1}\frac{\cos[(2m+3)\phi]}{\cos\phi} \\ &= (-1)^{m+1}\frac{T_{2m+3}(x)}{x}. \end{aligned} \tag{26}$$

Using these relations, (23) simplifies to

$$R^{m+1}WP = (WP-Z)\frac{T_{2m+3}(\sqrt{1-Z^\dagger Z})}{\sqrt{1-Z^\dagger Z}} + Z\frac{(-1)^{m+1}T_{2m+3}(\sqrt{Z^\dagger Z})}{\sqrt{Z^\dagger Z}}. \tag{27}$$

Hence we find that, if (21) is correct for non-negative integer m, it is correct for $m+1$. Hence it is correct for all non-negative integers m by induction.

Thus we find that by multiplying on the left by P, we obtain

$$PR^mWP = Z\frac{(-1)^m T_{2m+1}(\sqrt{Z^\dagger Z})}{\sqrt{Z^\dagger Z}}. \tag{28}$$

Then, in the case that $\delta = 0$, i.e., if $\tilde{V}$ is equal to a unitary V, we would have

$$Z = \frac{1}{s}(|\varsigma\rangle\langle\varsigma| \otimes V). \tag{29}$$

Then $Z^\dagger Z = (|\varsigma\rangle\langle\varsigma| \otimes \mathbb{I})/s^2$, and we get

$$\begin{aligned} Z\frac{(-1)^m T_{2m+1}(\sqrt{Z^\dagger Z})}{\sqrt{Z^\dagger Z}} &= \frac{1}{s}(|\varsigma\rangle\langle\varsigma| \otimes V)\frac{(-1)^m T_{2m+1}(1/s)}{1/s} \\ &= (|\varsigma\rangle\langle\varsigma| \otimes V)(-1)^m T_{2m+1}(1/s) \\ &= (|\varsigma\rangle\langle\varsigma| \otimes V)\sin[(2m+1)\arcsin(1/s)]. \end{aligned} \tag{30}$$

In the case $m = \ell$, we can use $\sin\left(\frac{\pi}{2(2\ell+1)}\right) = \frac{1}{s}$ to obtain $\sin[(2\ell+1)\arcsin(1/s)] = 1$, which implies

$$PR^\ell WP = |\varsigma\rangle\langle\varsigma| \otimes V. \tag{31}$$

Next, consider the case where $\tilde{V}$ is only δ-close to being unitary. Let us define

$$\Delta := \sqrt{\tilde{V}^\dagger \tilde{V}} - \mathbb{I}. \tag{32}$$

We immediately obtain $\|\Delta\| = O(\delta)$, and

$$\frac{1}{s}(|\varsigma\rangle\langle\varsigma| \otimes \Delta) = \sqrt{Z^\dagger Z} - \frac{1}{s}(|\varsigma\rangle\langle\varsigma| \otimes \mathbb{I}). \tag{33}$$

We then get

$$\begin{aligned} Z\frac{(-1)^\ell T_{2\ell+1}(\sqrt{Z^\dagger Z})}{\sqrt{Z^\dagger Z}} &= \frac{1}{s}(|\varsigma\rangle\langle\varsigma| \otimes \tilde{V})\frac{(-1)^\ell T_{2\ell+1}((\mathbb{I}+\Delta)/s)}{(\mathbb{I}+\Delta)/s} \\ &= (|\varsigma\rangle\langle\varsigma| \otimes \tilde{V})\frac{(-1)^\ell T_{2\ell+1}((\mathbb{I}+\Delta)/s)}{\mathbb{I}+\Delta}. \end{aligned} \tag{34}$$

Using $(-1)^\ell T_{2\ell+1}(x) = \sin[(2\ell+1)\arcsin x]$ and $(-1)^\ell T_{2\ell+1}(1/s) = 1$, we obtain

$$\|(-1)^\ell T_{2\ell+1}((\mathbb{I}+\Delta)/s) - \mathbb{I}\| = O(\ell^2\delta^2/s^2). \tag{35}$$

Since $\ell = \Theta(s)$, $\ell^2/s^2 = O(1)$, which implies

$$\|(-1)^\ell T_{2\ell+1}((\mathbb{I}+\Delta)/s) - \mathbb{I}\| = O(\delta^2). \tag{36}$$

The contribution to the error from $\mathbb{I}+\Delta$ is $O(\delta)$, so we have

$$\left\| Z\frac{(-1)^\ell T_{2\ell+1}(\sqrt{Z^\dagger Z})}{\sqrt{Z^\dagger Z}} - (|\varsigma\rangle\langle\varsigma| \otimes \tilde{V}) \right\| = O(\delta). \tag{37}$$

Using (28) we then get (18) as required. ■

Lemma 6 is in terms of the spectral norm distance, but the diamond norm distance is at most a constant factor larger. The specific result, proven in the Appendix, is as follows.

Lemma 7. *Let U, V be operators satisfying $\|U\| \leq 1$ and $\|V\| \leq 1$, and let $\|\cdot\|_\diamond$ denote the diamond norm. Then $\|U - V\|_\diamond \leq 2\sqrt{2}\|U - V\|$. Furthermore, if $\mathcal{V}$ is a quantum channel with Kraus decomposition $\mathcal{V}(\rho) = V\rho V^\dagger + \sum_j V_j \rho V_j^\dagger$ and $\mathcal{U}(\rho) = U\rho U^\dagger$, then $\|\mathcal{U} - \mathcal{V}\|_\diamond \leq 4\sqrt{2}\|U - V\|$.*

The LCU Lemma follows by combining Lemma 5 and Lemma 6.

Proof of Lemma 4: Using Lemma 5 we can implement the operation W required for Lemma 6 using $O(1)$ $\mathrm{select}(\vec{U})$ and $\mathrm{select}(\vec{U}^\dagger)$ operations and $O(M)$ additional 2-qubit gates. We can choose $s \geq a$ such that $\sin\left(\frac{\pi}{2(2\ell+1)}\right) = \frac{1}{s}$ for some $\ell \in \mathbb{N}$. Then Lemma 6 shows that $\tilde{V}$ can be approximated to within $O(\delta)$ using $O(\ell)$ applications of W and the projection P. Since $\ell = O(s) = O(a)$, the total number of $\mathrm{select}(\vec{U})$ and $\mathrm{select}(\vec{U}^\dagger)$ operations is $O(a)$ and the number of additional 2-qubit gates is $O(Ma)$. ■

C. Main algorithm

The main problem with applying the quantum walk as presented in [11, 13] is that $\arcsin \nu$ is a nonlinear function of ν, so an imprecise phase is introduced. To solve this, we use a superposition of different numbers of applications of U. Define V_k as in (8), where the choice of $\{a_m\}_{m=-k}^{k}$ is considered below. The eigenvalues of V_k corresponding to the eigenvalues $\mu_\pm$ of U are

$$\mu_{\pm,k} := \sum_{m=-k}^{k} a_m \mu_\pm^m. \tag{38}$$

In general $\mu_{\pm,k}$ can depend on $\pm$. However, we will choose a_m satisfying $a_{-m} = (-1)^m a_m$, which yields $\mu_{\pm,k}$ independent of $\pm$.

To see how to choose the coefficients a_m, solve (15) for ν to give

$$\nu = -\frac{i}{2}\left(\mu_\pm - \frac{1}{\mu_\pm}\right). \tag{39}$$

This implies that, for any z,

$$e^{i\nu z} = \exp\left[\frac{z}{2}\left(\mu_\pm - \frac{1}{\mu_\pm}\right)\right]. \tag{40}$$

This corresponds to the standard generating function for the Bessel function [24, 9.1.41], so

$$e^{i\nu z} = \exp\left[\frac{z}{2}\left(\mu_\pm - \frac{1}{\mu_\pm}\right)\right] = \sum_{m=-\infty}^{\infty} J_m(z)\mu_\pm^m. \tag{41}$$

Thus we can take $a_m \approx J_m(z)$. Because there are efficient classical algorithms to calculate Bessel functions, the circuit to prepare $|\chi_k\rangle$ can be designed efficiently. Note that for large m, we have $|J_m(z)| \sim \frac{1}{m!}|z/2|^m$ [24, 9.3.1], so the values of a_m are similar to the coefficients in the expansion of the exponential function. Thus the segments used here are analogous to the segments used in [16].

To determine the complexity of this approach, we primarily need to bound the error in approximating $e^{i\nu z}$. To optimize the result, we use the coefficients

$$a_m := \frac{J_m(z)}{\sum_{j=-k}^{k} J_j(z)}. \tag{42}$$

We make this choice because the most accurate results are obtained when the values of a_m sum to 1. Note also that this yields the symmetry $a_{-m} = (-1)^m a_m$, because $J_{-m}(z) = (-1)^m J_m(z)$ [24, 9.1.5]. The sum of $J_m(z)$ over all integers m is equal to 1 (which can be shown by putting $t = 1$ in [24, 9.1.41]), but because k is finite, we normalize the values as in (42). With this choice, we have the following error bound, proved in the Appendix.

Lemma 8. *With the values a_m as in (42), for $|z| \leq k$ we have*

$$\|V_k - V_\infty\| = O\left(\frac{\|H\|}{Xd}\frac{(z/2)^{k+1}}{k!}\right). \tag{43}$$

Note that V_∞ is the exact unitary operation desired. We now determine the query complexity of this approach. In fact, we prove a result that is slightly tighter than the query complexity stated in Theorem 1.

Lemma 9. *A d-sparse Hamiltonian H acting on n qubits can be simulated for time t within error ϵ with complexity (quantified by the number of 2-qubit operations and controlled-U and controlled-$U^\dagger$ operations)*

$$O\left(\tau \frac{\log(\|H\|t/\epsilon)}{\log\log(\|H\|t/\epsilon)}\right). \tag{44}$$

Proof: Our main goal is to determine the value of k needed to bound the error by ϵ. This depends on the length of time for the segments, which we can adjust by choosing the value of z. We wish to perform each step deterministically with one step of oblivious amplitude amplification, so we should have $s = 2$ in Lemma 6. Using the values of a_m given in (42), this means that we should take $z = O(1)$, and for concreteness $z = -1/2$ yields $a < 2$, so we can take $s = 2$. Then, using Lemma 4 with $U_m = U^m$ and $\tilde{V} = V_k$, we can approximate the operation V_k to within $O(\delta)$.

Given an allowable error in a segment of $\delta > 0$, let us take

$$k = O\left(\frac{\log(\|H\|/Xd\delta)}{\log\log(\|H\|/Xd\delta)}\right). \tag{45}$$

Then, using Lemma 8 and the inequality $k! > (k/e)^k$, it is straightforward to show that the error in each segment is no more than δ. Using Lemma 7, this bound on the error in terms of the spectral norm distance implies a bound on the diamond norm distance that is at most a constant factor larger. For the total error to be no more than ϵ, the value of δ can be no more than ϵ divided by the number of segments. The number of segments is $O(tXd)$, which gives

$$k = O\left(\frac{\log(\|H\|t/\epsilon)}{\log\log(\|H\|t/\epsilon)}\right). \tag{46}$$

Using Lemma 4, the complexity of each segment is $O(k)$ since a $\mathrm{select}(\vec{U})$ operation can be implemented with complexity $O(M)$, and $M = 2k + 1$. It is straightforward to apply $\mathrm{select}(\vec{U})$ using $O(k)$ controlled-U and controlled-$U^\dagger$ operations. If $|m\rangle$ is encoded in unary, then each controlled operation may be just controlled on one qubit of $|m\rangle$.

The number of segments required is $O(tXd)$. It is most efficient to take the minimum value of X, which is $\|H\|_{\max}$. Because each segment uses $O(k)$ controlled-U and controlled-$U^\dagger$ operations, as well as $O(M) = O(k)$ additional 2-qubit gates, the complexity for the simulation over time t is $O(\tau k)$, Using the value of k from (46) gives the overall complexity stated in (44). ∎

Next we determine the gate complexity of this approach. Again we give a slightly tighter result than presented in Theorem 1.

Lemma 10. *A d-sparse Hamiltonian H acting on n qubits can be simulated for time t within error ϵ using*

$$O\left(\tau[n + F(\log(\|H\|t/\epsilon))]\frac{\log(\|H\|t/\epsilon)}{\log\log(\|H\|t/\epsilon)}\right) \tag{47}$$

2-qubit gates, where $F(m)$ is the complexity of performing elementary functions with m bits.

Proof: To obtain the gate complexity, we need to consider the procedure for performing the step U in detail. We can perform T by first performing $\log d$ Hadamard gates to prepare the superposition state

$$\frac{1}{\sqrt{d}}\sum_{\ell=0}^{d-1}|\ell\rangle|0\rangle. \tag{48}$$

Here we take d to be a power of 2 without loss of generality. (The value of d can always be rounded up to the nearest power of two.) Then the oracle O_F (from (2)) can be used to produce the state

$$\frac{1}{\sqrt{d}}\sum_{\ell\in F_j}|\ell\rangle|0\rangle. \tag{49}$$

A call to the oracle O_H for the value of an element of the Hamiltonian (from (1)) then gives the value of $H_{j\ell}$ in an ancilla space. Another ancilla qubit is rotated from $|0\rangle$ to

$$\sqrt{\frac{H^*_{j\ell}}{X}}|0\rangle + \sqrt{1 - \frac{|H^*_{j\ell}|}{X}}|1\rangle \tag{50}$$

based on the value of $H_{j\ell}$. Then inverting the oracle O_H erases the value of $H_{j\ell}$ from the ancilla space. Note that there is a sign ambiguity for the square root when $H_{j\ell}$ takes negative real values. This is addressed in [13] and does not affect the complexity.

To perform the step U, we also require the swap operation S, which has complexity $O(n)$ due to the number of qubits. The gate complexity is $O(n)$ from S, plus $O(\log d) = O(n)$ from the Hadamard gates, plus the complexity of performing the rotations to obtain the state (50). The complexity of the rotations depends on the number of bits of precision used for the entries of H. To obtain overall error $O(\epsilon)$, the number of bits must be $\log(\|H\|t/\epsilon)$. To determine the rotations needed, we must also compute a square root and trigonometric functions on the output of the oracle. If these functions can be computed with complexity $F(m)$ for m-bit inputs, the contribution to the overall complexity is $F(\log(\|H\|t/\epsilon))$.

In Lemma 9 the complexity is quantified in terms of the number of controlled-U and controlled-$U^\dagger$ operations, so to obtain the overall gate complexity we just need to multiply that complexity by the cost of U. There is also a cost in terms of additional 2-qubit gates in Lemma 9, but that is smaller than the gate cost of performing U. Therefore, the gate complexity is equal to the complexity from Lemma 9 times $O(n + F(\log(\|H\|t/\epsilon)))$, which gives a gate complexity as in (47). ∎

This result depends on the complexity of elementary functions, $F(m)$, needed to calculate the rotations. Using advanced techniques, $F(m)$ may be made close to linear in m [26], though such advanced techniques only give an improvement for extremely high precision. Using simple techniques based on Taylor series and long multiplication, $F(m) = O(m^{5/2})$.

The classical complexity of determining the coefficients $\{a_m\}_{m=-k}^{k}$ is also potentially significant. A set of values of the Bessel function can be efficiently computed using Miller's recurrence algorithm [27, 28]. The complexity scales as k (the number of entries) times $\log(\|H\|t/\epsilon)$ (the bits of precision needed for each $J_m(z)$). This is no larger than the quantum gate complexity.

Note that the gate complexity in Lemma 10 depends linearly on n, whereas the query complexity in Theorem 1 does not. This is because performing an operation on a target state with n qubits must require at least $\Omega(n)$ gates. In contrast, the number of queries need not scale with n, because the queries are used to determine which gates to perform. There is an implicit complexity of $\Omega(n)$ for the queries, because the input to a query is of size at least n.

The proof of Theorem 1 then follows immediately.

Proof of Theorem 1: The implementation of U uses $O(1)$ oracle calls, which means that the query complexity is the same as the number of controlled applications of U. Noting that $\|H\| \leq d\|H\|_{\max}$, Lemma 9 implies the query complexity in Theorem 1, and Lemma 10 with $F(m) = O(m^{5/2})$ implies the gate complexity in Theorem 1. ∎

D. A tradeoff between τ and ϵ

The alternative algorithm characterized by Theorem 3 uses larger segments with $z \propto -\tau^\alpha$ for $\alpha \in (0, 1]$. The case $\alpha = 0$ corresponds to the case considered above, whereas $\alpha = 1$ corresponds to a *single* segment. The analysis of this section assumes $\alpha > 0$.

To analyze this algorithm, we first need to bound the absolute sum of Bessel functions.

Lemma 11. *The quantity*

$$\mathcal{S}(z) := \sum_{m=-\infty}^{\infty} |J_m(z)| \tag{51}$$

is $O(\sqrt{|z|})$.

We prove this in the Appendix. Using the robust version of amplitude amplification given in Lemma 6, we obtain the bound in Theorem 3.

Proof of Theorem 3: Using Lemma 8, Stirling's formula, and the fact that $\|H\| \le d\|H\|_{\max} \le Xd$, we find that for $|z| \le k$ the error is bounded as

$$\|V_k - V_\infty\| = O\big((e/k)^k (z/2)^{k+1}\big). \tag{52}$$

By Lemma 6, the error in a segment after amplitude amplification is of the same order. Therefore, to ensure that the error in a segment is at most δ, it suffices to take

$$k = O(|z| + \log(1/\delta)) = O(\tau^\alpha + \log(1/\delta)). \tag{53}$$

With this value of k, we have $\sum_{m=-k}^{k} J_m(z) = 1 + O(\delta)$, so Lemma 11 gives

$$\sum_{m=-k}^{k} |a_m| = O(\mathcal{S}(z)) = O(\sqrt{|z|}). \tag{54}$$

This corresponds to the number of steps of oblivious amplitude amplification. The overall complexity is therefore $O(k\sqrt{|z|}) = O(k\tau^{\alpha/2})$ for a single segment.

The number of segments is $\tau/|z| \propto \tau^{1-\alpha}$. This means that the complexity is $O(k\tau^{1-\alpha/2})$. The value of k also depends on the number of segments. We can take $\delta = \epsilon/\tau^{1-\alpha}$, which gives $k = O(\tau^\alpha + \log(1/\epsilon))$, implying the result in Theorem 3. ∎

In this proof we have ignored $\log \tau$ in comparison to τ^α, which would not be valid for $\alpha = 0$. For the gate complexity, we again have a multiplying factor of $n + F(\log(\|H\|t/\epsilon))$, yielding a number of gates scaling as

$$O\big([n + F(\log(\|H\|t/\epsilon))]\big[\tau^{1+\alpha/2} + \tau^{1-\alpha/2}\log(1/\epsilon)\big]\big). \tag{55}$$

IV. Lower bound

We now present a lower bound showing that the dependence of our algorithm on $\tau := d\|H\|_{\max} t$ is nearly optimal (and that the dependence of [13] on τ is optimal). The main idea of the proof is the same as in Theorem 3 of [23], but we slightly adapt that argument to let t vary independently of d. Note that this is stronger than proving separate lower bounds of $\Omega(t)$ and $\Omega(d)$, since that would only show a lower bound of $\Omega(d+t)$, which is weaker than our $\Omega(td)$ lower bound.

Lemma 12. *For any positive integer d and any $t > 0$, there exists a $2d$-sparse Hamiltonian H with $\|H\|_{\max} = \Theta(1)$ such that simulating H with constant precision for time t requires $\Omega(td)$ queries.*

Proof: Similarly to the $\Omega(t)$ lower bound from [9], we construct a sparse Hamiltonian whose dynamics compute the parity of a bit string, and we use the fact that at least $N/2$ quantum queries are needed to compute the parity of N bits [29, 30].

First consider a Hamiltonian H_1 whose graph is a path with $N+1$ vertices. (Here the *graph of H* has a vertex for each basis state and an edge between two vertices if the corresponding entry of H is nonzero.) The Hamiltonian acts on vectors $|i\rangle$ with $i \in \{0, \ldots, N\}$ and has nonzero matrix elements

$$\langle i-1|H_1|i\rangle = \langle i|H_1|i-1\rangle = \sqrt{i(N-i+1)} \tag{56}$$

for $i \in [N]$. Simulating H_1 for time $\pi/2$ starting with the state $|0\rangle$ gives the state $|N\rangle$ (i.e., $e^{-iH_1\pi/2}|0\rangle = |N\rangle$).

Next, consider a Hamiltonian H_2 generated from a string $x \in \{0,1\}^N$ as in [9]. H_2 acts on vertices $|i,j\rangle$ with $i \in \{0, \ldots, N\}$ and $j \in \{0,1\}$ and has nonzero matrix elements

$$\langle i-1, j|H_2|i, j \oplus x_i\rangle = \langle i, j \oplus x_i|H_2|i-1, j\rangle = \sqrt{i(N-i+1)} \tag{57}$$

for all $i \in [N]$ and $j \in \{0,1\}$. By construction, $|0,0\rangle$ is connected to $|i,j\rangle$ if and only if $j = x_1 \oplus \cdots \oplus x_i$. In particular, $|0,0\rangle$ is connected to $|N, x_1 \oplus \cdots \oplus x_N\rangle$, and determining whether it is connected to $|N,0\rangle$ or $|N,1\rangle$ determines the parity of x. The graph of H_2 consists of two disjoint paths, one containing $|0,0\rangle$ and

$|N, x_1 \oplus \cdots \oplus x_N\rangle$. Thus we have $e^{-iH_2\pi/2}|0,0\rangle = |N, x_1 \oplus \cdots \oplus x_N\rangle$, so evolution for time $\pi/2$ computes the parity of x.

Finally, we construct the Hamiltonian H claimed in the lemma. As before, H is generated from a string $x \in \{0,1\}^N$. H acts on vertices $|i,j,\ell\rangle$ with $i \in \{0,\ldots,N\}$, $j \in \{0,1\}$, and $\ell \in [d]$. The nonzero entries of H are

$$\langle i-1,j,\ell|H|i,j\oplus x_i,\ell'\rangle = \langle i,j\oplus x_i,\ell'|H|i-1,j,\ell\rangle = \sqrt{i(N-i+1)}/N \tag{58}$$

for all $i \in [N]$, $j \in \{0,1\}$, and $\ell,\ell' \in [d]$. The graph of H is similar to that of H_2, except that for each vertex in H_2, there are now d copies of it in H. Each vertex is connected to all d copies of its neighboring vertices, so the graph has maximum degree $2d$. Observe that, having divided the matrix elements by N, we have $\|H\|_{\max} = \Theta(1)$.

Now we simulate the Hamiltonian starting from the state $|0,0,*\rangle$, where $|i,j,*\rangle := \frac{1}{\sqrt{d}}\sum_\ell |i,j,\ell\rangle$ denotes a uniform superposition over the third register. The subspace $\operatorname{span}\{|i,j,*\rangle : i \in \{0,\ldots,N\},\ j \in \{0,1\}\}$ is an invariant subspace of H. Since the initial state lies in this subspace, the quantum walk remains in this subspace. The nonzero matrix elements of H in this invariant subspace are

$$\langle i-1,j,*|H|i,j\oplus x_i,*\rangle = \langle i,j\oplus x_i,*|H|i-1,j,*\rangle = d\sqrt{i(N-i+1)}/N, \tag{59}$$

so we have $e^{-iHt}|0,0,*\rangle = |N, x_1 \oplus \cdots \oplus x_N, *\rangle$ for $t = N\pi/2d$. Since this determines the parity of x, we find a lower bound of $\Omega(N) = \Omega(td)$ as claimed. ∎

It is now straightforward to use this result to prove Theorem 2.

Proof of Theorem 2: We choose one of two Hamiltonians depending on whether the first or second term in (5) is larger. If τ is larger, then we use Lemma 12. The value of d used in Lemma 12 is denoted d' here, to distinguish it from the d given in Theorem 2. Taking $d' = \lfloor d/2 \rfloor$, we ensure that d' is a positive integer, because $d \geq 2$. Then Lemma 12 shows that there is a $2d'$-sparse Hamiltonian; given this value of d', this Hamiltonian is also d-sparse, as required for Theorem 2.

For Theorem 2, we are also given a required value for $\|H\|_{\max}$. The Hamiltonian used in Lemma 12 has $\|H\|_{\max} = \Theta(1)$. By multiplying that Hamiltonian by a scaling factor, we obtain a Hamiltonian with the required value of $\|H\|_{\max}$. Dividing the time used in Lemma 12 by the same factor, the simulation requires time $\Omega(\tau)$ for constant precision. In Theorem 2 we require precision ϵ, which can only increase the complexity.

In the case where the second term is larger, we use Theorem 6.1 of [15]. There it is shown that performing a simulation of a 2-sparse Hamiltonian with precision ϵ and $\|H\|_{\max}t = O(1)$ requires

$$\Omega\left(\frac{\log(1/\epsilon)}{\log\log(1/\epsilon)}\right) \tag{60}$$

queries. Because $d \geq 2$, this Hamiltonian is also d-sparse. As using larger values of $\|H\|_{\max}t$ can only increase the complexity, we also have this lower bound in the more general case. Therefore, regardless of whether the first or second term in (5) is larger, this expression provides a lower bound on the complexity. ∎

It is also possible to combine our lower bound with the lower bound of [15] to obtain a combined lower bound in terms of d, t, and ϵ, that is stronger than Theorem 2. This yields a lower bound of $\Omega(N)$ for any N that satisfies $\epsilon < \frac{1}{2}|\sin(td/N)|^N$. Note that when ϵ is a constant, we recover Lemma 12 and when td is constant, we recover (60). However, for intermediate values this lower bound can be strictly larger than the expression in Theorem 2.

V. Conclusion

Our technique for Hamiltonian simulation combines ideas from quantum walks and fractional-query simulation to provide improved performance over both previous techniques. As a result, it provides near-optimal scaling with respect to all parameters of interest. In particular, the scaling is only slightly superlinear in $\tau = d\|H\|_{\max}t$, whereas

we have proven that linear scaling is optimal. Furthermore, the method has query complexity sublogarithmic in the allowed error, which was proven to be optimal in [15].

Nevertheless, there is still a gap between the complexity of our algorithm and the lower bound in (5), as they involve different tradeoffs between the parameters τ and ϵ. It remains open whether the performance can be further improved, perhaps to give performance similar to (5), although as observed at the end of Section IV, we can rule out scaling strictly as in (5).

Our technique can potentially be used for the more general task of operation conversion, in which we use one quantum operation to implement another. In our work, we convert a step of a quantum walk to Hamiltonian evolution, whereas in [17] the task is to convert Hamiltonian evolution to an inverse. One approach to operation conversion is to use phase estimation. Here we have shown that a superposition of operations can provide far better performance.

ACKNOWLEDGMENT

D.W.B. is funded by an ARC Future Fellowship (FT100100761). This work was also supported in part by CIFAR, NSERC, the Ontario Ministry of Research and Innovation, and the US ARO under ARO grant Contract Numbers W911NF-12-1-0482 and W911NF-12-1-0486.

REFERENCES

[1] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, no. 6–7, pp. 467–488, June 1982.

[2] Oak Ridge Leadership Computing Facility, "OLCF Annual Report 2012–2013," https://www.olcf.ornl.gov/media-center/center-reports/.

[3] National Energy Research Scientific Computing Center, "NERSC Annual Report 2013," https://www.nersc.gov/news-publications/publications-reports/nersc-annual-reports/.

[4] R. P. Feynman, "Quantum mechanical computers," *Optics News*, vol. 11, no. 2, pp. 11–20, February 1985.

[5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, October 1997.

[6] S. Lloyd, "Universal quantum simulators," *Science*, vol. 273, no. 5278, pp. 1073–1078, August 1996.

[7] D. Aharonov and A. Ta-Shma, "Adiabatic quantum state generation and statistical zero knowledge," in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, June 2003, pp. 20–29.

[8] A. M. Childs, "Quantum information processing in continuous time," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

[9] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, "Efficient quantum algorithms for simulating sparse Hamiltonians," *Communications in Mathematical Physics*, vol. 270, no. 2, pp. 359–371, March 2007.

[10] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, "Simulating quantum dynamics on a quantum computer," *Journal of Physics A*, vol. 44, no. 44, p. 445308, October 2011.

[11] A. M. Childs, "On the relationship between continuous- and discrete-time quantum walk," *Communications in Mathematical Physics*, vol. 294, no. 2, pp. 581–603, March 2010.

[12] D. Poulin, A. Qarry, R. D. Somma, and F. Verstraete, "Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space," *Physical Review Letters*, vol. 106, no. 17, p. 170501, April 2011.

[13] D. W. Berry and A. M. Childs, "Black-box Hamiltonian simulation and unitary implementation," *Quantum Information and Computation*, vol. 12, no. 1–2, pp. 29–62, January 2012.

[14] A. M. Childs and N. Wiebe, "Hamiltonian simulation using linear combinations of unitary operations," *Quantum Information and Computation*, vol. 12, no. 11–12, pp. 901–924, November 2012.

[15] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, "Exponential improvement in precision for simulating sparse Hamiltonians," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, May 2014, pp. 283–292.

[16] ——, "Simulating Hamiltonian dynamics with a truncated Taylor series," *Physical Review Letters*, vol. 114, no. 9, p. 090502, March 2015.
[17] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Physical Review Letters*, vol. 103, no. 15, p. 150502, October 2009.
[18] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, "Exponential algorithmic speedup by quantum walk," in *Proceedings of the 35th ACM Symposium on Theory of Computing*, June 2003, pp. 59–68.
[19] A. M. Childs, R. Cleve, S. P. Jordan, and D. Yonge-Mallo, "Discrete-query quantum algorithm for NAND trees," *Theory of Computing*, vol. 5, no. 5, pp. 119–123, July 2009.
[20] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum algorithm for the Hamiltonian NAND tree," *Theory of Computing*, vol. 4, no. 8, pp. 169–190, December 2008.
[21] A. Ambainis, L. Magnin, M. Roetteler, and J. Roland, "Symmetry-assisted adversaries for quantum state generation," in *Proceedings of the 26th IEEE Conference on Computational Complexity*, June 2011, pp. 167–177.
[22] T. Lee, R. Mittal, B. W. Reichardt, R. Špalek, and M. Szegedy, "Quantum query complexity of state conversion," in *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science*, October 2011, pp. 344–353.
[23] A. M. Childs and R. Kothari, "Limitations on the simulation of non-sparse Hamiltonians," *Quantum Information and Computation*, vol. 10, no. 7–8, pp. 669–684, July 2010.
[24] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions*. National Bureau of Standards, 1964.
[25] R. Kothari, "Efficient algorithms in quantum query complexity," Ph.D. dissertation, University of Waterloo, 2014.
[26] R. P. Brent, "Fast multiple-precision evaluation of elementary functions," *Journal of the Association for Computing Machinery*, vol. 23, no. 2, pp. 242–251, April 1976.
[27] W. G. Bickley, L. J. Comrie, J. C. P. Miller, D. H. Sadler, and A. J. Thompson, *Bessel Functions, Part II, Functions of Positive Integer Order*. Cambridge University Press, 1952.
[28] F. W. J. Olver, "Error analysis of Miller's recurrence algorithm," *Mathematics of Computation*, vol. 18, pp. 65–74, January 1964.
[29] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf, "Quantum lower bounds by polynomials," *Journal of the ACM*, vol. 48, no. 4, pp. 778–797, July 2001.
[30] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Limit on the speed of quantum computation in determining parity," *Physical Review Letters*, vol. 81, no. 24, pp. 5442–5444, December 1998.

APPENDIX

We now present proofs of some of the more technical results.

Proof of Lemma 8: For $|m| \leq k$, the values of a_m differ from $J_m(z)$ only due to the normalization factor in (42). The Bessel function $J_m(z)$ is bounded, for real z and integer m, by [24, 9.1.62]

$$|J_m(z)| \leq \frac{1}{|m|!} \left|\frac{z}{2}\right|^{|m|} \tag{61}$$

(here we use the fact that $J_{-m}(z) = (-1)^m J_m(z)$ [24, 9.1.5]). Using this bound, together with the condition that $|z| \leq k$,

$$2 \sum_{m=k+1}^{\infty} |J_m(z)| \leq 2 \sum_{m=k+1}^{\infty} \frac{|z/2|^m}{m!} < 2\frac{|z/2|^{k+1}}{(k+1)!} \sum_{m=k+1}^{\infty} (1/2)^{m-(k+1)} = 4\frac{|z/2|^{k+1}}{(k+1)!}. \tag{62}$$

As a result, the normalization factor in (42) satisfies

$$\sum_{m=-k}^{k} J_m(z) \geq 1 - 4\frac{|z/2|^{k+1}}{(k+1)!}. \tag{63}$$

This means that a_m closely approximates $J_m(z)$, in the sense that

$$a_m = J_m(z)\left[1 + O\left(\frac{(z/2)^{k+1}}{(k+1)!}\right)\right]. \tag{64}$$

Similarly, using $|\mu_\pm^m - 1| \leq |\nu m|$ and $|z| \leq k$ gives

$$\begin{aligned}\left|2\sum_{m=k+1}^{\infty} J_m(z)(\mu_\pm^m - 1)\right| &\leq 2|\nu| \sum_{m=k+1}^{\infty} m\frac{|z/2|^m}{m!} \\ &< 2|\nu|\frac{|z/2|^{k+1}}{(k+1)!}\sum_{m=k+1}^{\infty} m(1/2)^{m-(k+1)} \\ &= 4|\nu|(k+2)\frac{|z/2|^{k+1}}{(k+1)!}.\end{aligned} \tag{65}$$

Using (41) gives

$$e^{i\nu z} - 1 = \sum_{m=-\infty}^{\infty} J_m(z)(\mu_\pm^m - 1). \tag{66}$$

Therefore, with (65), we obtain

$$\sum_{m=-k}^{k} J_m(z)(\mu_\pm^m - 1) = e^{i\nu z} - 1 + O\left(\nu\frac{(z/2)^{k+1}}{k!}\right). \tag{67}$$

Using this expression together with (64) then gives

$$\sum_{m=-k}^{k} a_m(\mu_\pm^m - 1) = \left[e^{i\nu z} - 1 + O\left(\nu\frac{(z/2)^{k+1}}{k!}\right)\right]\left[1 + O\left(\frac{(z/2)^{k+1}}{(k+1)!}\right)\right]. \tag{68}$$

Using $|e^{i\nu z} - 1| \leq |\nu z|$ and $|z| \leq k$ gives

$$\sum_{m=-k}^{k} a_m(\mu_\pm^m - 1) = e^{i\nu z} - 1 + O\left(\nu\frac{(z/2)^{k+1}}{k!}\right). \tag{69}$$

Because $\sum_m a_m = 1$, we obtain

$$\left|\sum_{m=-k}^{k} a_m\mu_\pm^m - e^{i\nu z}\right| = O\left(\nu\frac{(z/2)^{k+1}}{k!}\right). \tag{70}$$

Now $\sum_{m=-k}^{k} a_m\mu_\pm^m$ are the eigenvalues of V_k, and $e^{i\nu z}$ are the eigenvalues of the desired unitary operation V_∞. Using $\nu = \lambda/Xd$, we have $|\nu| \leq \|H\|/Xd$. Hence the norm of the difference of these operators is bounded as in (43). ∎

Proof of Lemma 11: First we bound the sum over small values of m. The Bessel functions satisfy [24, 9.1.76]

$$\sum_{m=-\infty}^{\infty} [J_m(z)]^2 = 1. \tag{71}$$

For any positive integer k we have

$$\sum_{m=-k}^{k} |J_m(z)|^2 < 1. \tag{72}$$

Using the Cauchy-Schwarz inequality, we find

$$\sum_{m=-k}^{k} |J_m(z)| \leq \sqrt{\sum_{m=-k}^{k} 1} \sqrt{\sum_{m=-k}^{k} |J_m(z)|^2} < \sqrt{2k+1}. \tag{73}$$

Provided $|z| \leq k$, using (62), together with $\ell! > (\ell/e)^\ell$, we obtain

$$2 \sum_{m=k+1}^{\infty} |J_m(z)| < 4 \left| \frac{ez}{2(k+1)} \right|^{k+1}. \tag{74}$$

Combining (73) and (74), we find

$$\mathcal{S}(z) < \sqrt{2k+1} + 4 \left| \frac{ez}{2(k+1)} \right|^{k+1}. \tag{75}$$

Finally, taking $k = \lceil ez/2 \rceil$, the second term is $O(1)$, which gives $\mathcal{S}(z) = O(\sqrt{|z|})$ as claimed. ∎

Proof of Lemma 7: Consider two operators U and V acting on a pure state $|\psi\rangle$. The states $U|\psi\rangle$ and $V|\psi\rangle$ span a 2-dimensional space, and therefore

$$\|U|\psi\rangle\langle\psi|U^\dagger - V|\psi\rangle\langle\psi|V^\dagger\|_1 \leq \sqrt{2}\|U|\psi\rangle\langle\psi|U^\dagger - V|\psi\rangle\langle\psi|V^\dagger\|. \tag{76}$$

We have

$$\|U|\psi\rangle\langle\psi|U^\dagger - V|\psi\rangle\langle\psi|V^\dagger\| = \|(U-V)|\psi\rangle\langle\psi|U^\dagger + V|\psi\rangle\langle\psi|(U-V)^\dagger\| \tag{77}$$

so, provided $\|U\| \leq 1$ and $\|V\| \leq 1$,

$$\|U|\psi\rangle\langle\psi|U^\dagger - V|\psi\rangle\langle\psi|V^\dagger\|_1 \leq 2\sqrt{2}\|U - V\|. \tag{78}$$

Given a mixed state $\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j|$, strong convexity implies that

$$\begin{aligned} \|U\rho U^\dagger - V\rho V^\dagger\|_1 &\leq \sum_j p_j \|U|\psi_j\rangle\langle\psi_j|U^\dagger - V|\psi_j\rangle\langle\psi_j|V^\dagger\|_1 \\ &\leq 2\sqrt{2} \sum_j p_j \|U - V\| \\ &= 2\sqrt{2}\|U - V\|. \end{aligned} \tag{79}$$

Similarly, tensoring with the identity gives

$$\begin{aligned} \|(U \otimes \mathbb{I})\rho(U^\dagger \otimes \mathbb{I}) - (V \otimes \mathbb{I})\rho(V^\dagger \otimes \mathbb{I})\|_1 &\leq 2\sqrt{2}\|U \otimes \mathbb{I} - V \otimes \mathbb{I}\| \\ &= 2\sqrt{2}\|U - V\|. \end{aligned} \tag{80}$$

The equality is obvious, because tensoring with the identity only gives repeated eigenvalues, which leaves the spectral norm unchanged. Hence, maximizing over ρ, the diamond norm satisfies

$$\|U - V\|_\diamond \leq 2\sqrt{2}\|U - V\|. \tag{81}$$

Now consider the case that U is some desired unitary, and $V = V_0$ is an operation that happens in the case of success of a measurement, but the operation will otherwise be V_k. That is, the overall channel is

$$C(\rho) = V\rho V^\dagger + \sum_{j>0} V_j \rho V_j^\dagger. \tag{82}$$

The trace is bounded by

$$\begin{aligned}\mathrm{Tr}(V\rho V^\dagger) &= \mathrm{Tr}(U\rho U^\dagger) - \mathrm{Tr}(V\rho V^\dagger - U\rho U^\dagger) \\ &\geq 1 - \|U\rho U^\dagger - V\rho V^\dagger\|_1 \\ &\geq 1 - 2\sqrt{2}\|U - V\|. \end{aligned} \tag{83}$$

Hence, the trace of the remaining part is bounded as

$$\begin{aligned}\mathrm{Tr}\left(\sum_{j>0} V_j \rho V_j^\dagger\right) &= \mathrm{Tr}(C(\rho)) - \mathrm{Tr}(V\rho V^\dagger) \\ &\leq 1 - (1 - 2\sqrt{2}\|U - V\|) \\ &\leq 2\sqrt{2}\|U - V\|. \end{aligned} \tag{84}$$

The 1-norm is equal to the trace for non-negative Hermitian operators, which means

$$\left\|\sum_{j>0} V_j \rho V_j^\dagger\right\|_1 \leq 2\sqrt{2}\|U - V\|. \tag{85}$$

The trace is unchanged tensoring with the identity, so

$$\left\|\sum_{j>0} (V_j \otimes \mathbb{I})\rho(V_j \otimes \mathbb{I})^\dagger\right\|_1 \leq 2\sqrt{2}\|U - V\|. \tag{86}$$

Hence

$$\begin{aligned}\|C - U\|_\diamond &= \max_\rho \left\|(V \otimes \mathbb{I})\rho(V \otimes \mathbb{I})^\dagger + \sum_{j>0}(V_j \otimes \mathbb{I})\rho(V_j \otimes \mathbb{I})^\dagger - (U \otimes \mathbb{I})\rho(U \otimes \mathbb{I})^\dagger\right\|_1 \\ &\leq \max_\rho \left[\|(V \otimes \mathbb{I})\rho(V \otimes \mathbb{I})^\dagger - (U \otimes \mathbb{I})\rho(U \otimes \mathbb{I})^\dagger\|_1 + \left\|\sum_{j>0}(V_j \otimes \mathbb{I})\rho(V_j \otimes \mathbb{I})^\dagger\right\|_1\right] \\ &\leq 4\sqrt{2}\|U - V\| \end{aligned} \tag{87}$$

as claimed. ■