



# EXPONENTIAL IMPROVEMENT IN PRECISION FOR SIMULATING SPARSE HAMILTONIANS

DOMINIC W. BERRY<sup>1</sup>, ANDREW M. CHILDS<sup>2,3</sup>, RICHARD CLEVE<sup>4,5</sup>,  
ROBIN KOTHARI<sup>4,6</sup> and ROLANDO D. SOMMA<sup>7</sup>

<sup>1</sup> Department of Physics and Astronomy, Macquarie University, Sydney, NSW 2109, Australia;  
email: dominic.berry@mq.edu.au

<sup>2</sup> Department of Combinatorics & Optimization and Institute for Quantum Computing,  
University of Waterloo, Waterloo, ON N2L 3G1, Canada

<sup>3</sup> Department of Computer Science, Institute for Advanced Computer Studies, and Joint Center for  
Quantum Information and Computer Science, University of Maryland, College Park,  
MD 20742, USA;  
email: amchilds@umd.edu

<sup>4</sup> David R. Cheriton School of Computer Science and Institute for Quantum Computing,  
University of Waterloo, Waterloo, ON N2L 3G1, Canada;  
email: cleve@uwaterloo.ca

<sup>5</sup> Canadian Institute for Advanced Research, Toronto, ON M5G 1Z8, Canada

<sup>6</sup> Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge,  
MA 02139, USA;  
email: rkothari@mit.edu

<sup>7</sup> Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA;  
email: somma@lanl.gov

Received 1 July 2015; accepted 30 December 2016

## Abstract

We provide a quantum algorithm for simulating the dynamics of sparse Hamiltonians with complexity sublogarithmic in the inverse error, an exponential improvement over previous methods. Specifically, we show that a  $d$ -sparse Hamiltonian  $H$  acting on  $n$  qubits can be simulated for time  $t$  with precision  $\epsilon$  using  $O(\tau(\log(\tau/\epsilon)/\log\log(\tau/\epsilon)))$  queries and  $O(\tau(\log^2(\tau/\epsilon)/\log\log(\tau/\epsilon))n)$  additional 2-qubit gates, where  $\tau = d^2\|H\|_{\max}t$ . Unlike previous approaches based on product formulas, the query complexity is independent of the number of qubits acted on, and for time-varying Hamiltonians, the gate complexity is logarithmic in the norm of the derivative of the Hamiltonian. Our algorithm is based on a significantly improved simulation of the continuous- and fractional-query models using discrete quantum queries, showing that the former models are not much more powerful than the discrete model even for very small error. We also simplify the analysis

of this conversion, avoiding the need for a complex fault-correction procedure. Our simplification relies on a new form of ‘oblivious amplitude amplification’ that can be applied even though the reflection about the input state is unavailable. Finally, we prove new lower bounds showing that our algorithms are optimal as a function of the error.

2010 Mathematics Subject Classification: 68Q12 (primary); 81P68 (secondary)

## 1. Introduction

Simulation of quantum mechanical systems is a major potential application of quantum computers. Indeed, the problem of simulating Hamiltonian dynamics was the original motivation for the idea of quantum computation [27]. Lloyd provided an explicit algorithm for simulating many realistic quantum systems, namely those whose Hamiltonian is a sum of interactions acting nontrivially on a small number of subsystems of limited dimension [32]. If the interactions act on at most  $k$  subsystems, such a Hamiltonian is called  $k$ -local. Here we consider the more general problem of simulating sparse Hamiltonians, a natural class of systems for which quantum simulation has been widely studied. Note that  $k$ -local Hamiltonians are sparse, so algorithms for simulating sparse Hamiltonians can be used to simulate many physical systems. Sparse Hamiltonian simulation is also useful in quantum algorithms [1, 17, 22, 28].

A Hamiltonian is said to be  $d$ -sparse if it has at most  $d$  nonzero entries in any row or column. In the sparse Hamiltonian simulation problem, we are given access to a  $d$ -sparse Hamiltonian  $H$  acting on  $n$  qubits via a black box that accepts a row index  $i$  and a number  $j$  between 1 and  $d$ , and returns the position and value of the  $j$ th nonzero entry of  $H$  in row  $i$ . Given such a black box for  $H$ , a time  $t > 0$  (without loss of generality), and an error parameter  $\epsilon > 0$ , our task is to construct a circuit that performs the unitary operation  $e^{-iHt}$  with error at most  $\epsilon$  using as few queries to  $H$  as possible. To develop practical algorithms, we would also like to upper bound the number of additional 2-qubit gates. The *time complexity* of a simulation is the sum of the number of queries and additional 2-qubit gates.

The first efficient algorithm for sparse Hamiltonian simulation was due to Aharonov and Ta-Shma [1]. The key idea (also applied in [16]) is to use edge coloring to decompose the Hamiltonian  $H$  into a sum of Hamiltonians  $\sum_{j=1}^{\eta} H_j$ , where each  $H_j$  is easy to simulate. These terms are then recombined using the Lie product formula, which states that  $e^{-iHt} \approx (e^{-iH_1t/r} e^{-iH_2t/r} \dots e^{-iH_{\eta}t/r})^r$  for large  $r$ . This method gives query complexity  $O(\text{poly}(n, d)(\|H\|t)^2/\epsilon)$ , where  $\|\cdot\|$  denotes the spectral norm. This was later improved using high-order product formulas and more efficient decompositions of the Hamiltonian [8, 14, 19, 20, 40].

The best algorithms of this type [19, 20] have query complexity

$$d^2(d + \log^* n) \|H\|t \exp(O(\sqrt{\log(d\|H\|t/\epsilon)})). \quad (1)$$

This complexity is only slightly superlinear in  $\|H\|t$  in that  $\exp(O(\sqrt{\log(d\|H\|t/\epsilon)}))$  is asymptotically smaller than  $(d\|H\|t/\epsilon)^\delta$  for any constant  $\delta > 0$ ; however,  $\exp(O(\sqrt{\log(d\|H\|t/\epsilon)}))$  is not polylogarithmic in  $d\|H\|t/\epsilon$ .

In this work, we present an algorithm with query complexity  $O(\tau(\log(\tau/\epsilon)/\log\log(\tau/\epsilon)))$ . This algorithm has no query dependence on  $n$ , improved dependence on  $d$  and  $t$ , and exponentially improved dependence on  $1/\epsilon$ . The result for the number of queries is identical for time-dependent Hamiltonians. The number of 2-qubit gates required is slightly larger, but only by a logarithmic factor. Our Hamiltonian simulation algorithm is based on a connection to simulation of the so-called fractional-query model. We show that a continuous-query algorithm may be simulated with complexity polylogarithmic in  $1/\epsilon$ , and show that Hamiltonian simulation can be posed as a special case of a continuous-query algorithm.

*Subsequent work.* This article is an expanded version of work that appeared in STOC 2014 [10]. Subsequent to this, some related algorithms appeared, which build on the techniques introduced in this work (for example, oblivious amplitude amplification).

One algorithm [11] is based on directly implementing the Taylor series of a sum of certain basic Hamiltonians. The approach yields a simpler algorithm for the special case of time-independent Hamiltonians. Reference [11] also addresses the case of time-dependent Hamiltonians, but the approach is substantially more complicated than in the time-independent case. Reference [11] does not address the more general problem of simulating the fractional-query model, as we do here.

Another algorithm [12] uses a quantum walk to simulate time-independent sparse Hamiltonians. This yields a complexity that is nearly linear in the sparsity, whereas our present work has nearly quadratic complexity. However, for the notable special case of local Hamiltonians (or more generally, those with a given decomposition into 1-sparse terms), the present work already has complexity nearly linear in the sparsity, so [12] does not give a significant improvement. Moreover, it is unclear whether ideas from [12] can be applied to simulate time-dependent Hamiltonians. Also, it does not address the fractional-query model.

Our approach to Hamiltonian simulation has also been applied to the simulation of quantum chemistry [4, 5] and to the solution of linear systems [39]. The concept of oblivious amplitude amplification introduced in this paper has been applied to quantum circuit synthesis [37, 44].

## 2. Summary of results

We show the following (where  $\|H\|_{\max}$  denotes the largest entry of  $H$  in absolute value).

**THEOREM 2.1** (Sparse Hamiltonian simulation). *A  $d$ -sparse Hamiltonian  $H$  acting on  $n$  qubits can be simulated for time  $t$  within error  $\epsilon$  with  $O(\tau(\log(\tau/\epsilon)/\log\log(\tau/\epsilon)))$  queries and  $O(\tau(\log^2(\tau/\epsilon)/\log\log(\tau/\epsilon))n)$  additional 2-qubit gates, where  $\tau := d^2\|H\|_{\max}t$  (provided that  $\tau \geq 1$ ).*

Our algorithm has no query dependence on  $n$ , improved dependence on  $d$  and  $t$ , and exponentially improved dependence on  $1/\epsilon$ . Our new approach to Hamiltonian simulation strictly improves all previous approaches based on product formulas (for example, [1, 8, 14, 19, 32]). An alternative Hamiltonian simulation method based on a quantum walk [9, 15] is incomparable. The method has query complexity  $O(d\|H\|_{\max}t/\sqrt{\epsilon})$ , so its performance is better in terms of  $\|H\|_{\max}t$  and  $d$  but significantly worse in terms of  $\epsilon$ . Thus, while suboptimal for (say) constant-precision simulation, the results of Theorem 2.1 currently give the best known Hamiltonian simulations as a function of  $\epsilon$ .

Essentially the same approach used for Theorem 2.1 can be applied even when the Hamiltonian is time dependent. The query complexity is unaffected by any such time dependence, except that we take the largest max-norm of the Hamiltonian over all times (that is,  $\tau$  is redefined as  $\tau := d^2ht$  with  $h := \max_{s \in [0, t]} \|H(s)\|_{\max}$ ). The number of additional 2-qubit gates is  $O(\tau(\log(\tau/\epsilon) \log((\tau + \tau')/\epsilon)/\log\log(\tau/\epsilon))n)$ , where  $\tau' := d^2h't$  with  $h' := \max_{s \in [0, t]} \|(d/ds)H(s)\|$ . This dependence on  $h'$  is a dramatic improvement over previous methods for simulating time-dependent Hamiltonians using high-order product formulas [43]. Another previous simulation method [38] also improved the dependence on  $h'$ , but at the cost of substantially worse dependence on  $t$  and  $\epsilon$ .

While our approach applies to sparse Hamiltonians in general, it can sometimes be improved using additional structure. In particular, consider the case of a  $k$ -local Hamiltonian acting on a system of qubits. (A  $k$ -local Hamiltonian acting on subsystems of limited dimension is equivalent to a  $k$ -local Hamiltonian acting on qubits with an increased value of  $k$ .) Since a term acting only on  $k$  qubits is  $2^k$ -sparse, we can apply Theorem 2.1 with  $d = 2^k M$ , where  $M$  is the total number of local terms. However, by taking the structure of sparse Hamiltonians into account, we find an improved simulation with  $\tau$  replaced by  $\tilde{\tau} := 2^k M \|H\|_{\max}t$ .

The performance of our algorithm is optimal or nearly optimal as a function of some of its parameters. A lower bound of  $\Omega(\|H\|_{\max}t)$  follows from the no-fast-forwarding theorem of [8], showing that our algorithm's dependence on  $\|H\|_{\max}t$

is almost optimal. However, prior to our work, there was no known  $\epsilon$ -dependent lower bound, not even one ruling out algorithms with no dependence on  $\epsilon$ . We show that, surprisingly, our query dependence on  $\epsilon$  in Theorem 2.1 is optimal.

**THEOREM 2.2** ( $\epsilon$ -dependent lower bound for Hamiltonian simulation). *For any  $\epsilon > 0$ , there exists a 2-sparse Hamiltonian  $H$  with  $\|H\|_{\max} < 1$  such that simulating  $H$  with precision  $\epsilon$  for constant time requires  $\Omega(\log(1/\epsilon)/\log \log(1/\epsilon))$  queries.*

Our Hamiltonian simulation algorithm is based on a connection to the so-called fractional quantum query model. A result of Cleve *et al.* [23] shows that this model can be simulated with only small overhead using standard, discrete quantum queries. While this can be seen as a kind of Hamiltonian simulation, simulating the dynamics of a sparse Hamiltonian appears *a priori* unrelated. Here we relate these tasks, giving a simple reduction from Hamiltonian simulation to the problem of simulating (a slight generalization of) the fractional-query model, so that improved simulations of the fractional-query model directly yield improvements in Hamiltonian simulation.

To introduce the notion of fractional queries, recall that in the usual model of quantum query complexity, we wish to solve a problem whose input  $x \in \{0, 1\}^N$  is given by an oracle (or black box) that can be queried to learn the bits of  $x$ . The measure of complexity, called the query complexity, is the number of times we query the oracle. More precisely, we are given access to a unitary gate  $Q_x$  whose action on the basis states  $|j\rangle|b\rangle$  for all  $j \in [N] := \{1, 2, \dots, N\}$  and  $b \in \{0, 1\}$  is  $Q_x|j\rangle|b\rangle = (-1)^{b x_j}|j\rangle|b\rangle$ . A quantum query algorithm is a quantum circuit consisting of arbitrary  $x$ -independent unitaries and  $Q_x$  gates. The query complexity of such an algorithm is the total number of  $Q_x$  gates used in the circuit.

The query model is often used to study the complexity of evaluating a classical function of  $x$ . However, it is also natural to consider more general tasks. In order of increasing generality, such tasks include state generation [3], state conversion [31], and implementing unitary operations [7, 9]. Here we focus on the last of these tasks, where for each possible input  $x$  we must perform some unitary operation  $U_x$ . Considering this task leads to a strong notion of simulation: to simulate a given algorithm in the sense of unitary implementation, one must reproduce the entire correct output state for every possible input state, rather than simply (say) evaluating some predicate in one bit of the output with a fixed input state.

Since quantum mechanics is fundamentally described by the continuous dynamics of the Schrödinger equation, it is natural to ask if the query model can be made less discrete. In particular, instead of using the gate  $Q_x$  for unit cost,

what if we can make half a query for half the cost? This perspective is motivated by the idea that if  $Q_x$  is performed by a Hamiltonian running for unit time, we can stop the evolution after half the time to obtain half a query. In general we could run this Hamiltonian for time  $\alpha \in (0, 1]$  at cost  $\alpha$ . This *fractional-query model* is at least as powerful as the standard (*discrete-query*) model. More formally, we define the model as follows.

**DEFINITION 1 (Fractional-query model).** For an  $n$ -bit string  $x$ , let  $Q_x^\alpha$  act as  $Q_x^\alpha |j\rangle|b\rangle = e^{-i\pi\alpha bx_j} |j\rangle|b\rangle$  for all  $j \in [N]$  and  $b \in \{0, 1\}$ . An algorithm in the fractional-query model is a sequence of unitary gates  $U_m Q_x^{\alpha_m} U_{m-1} \cdots U_1 Q_x^{\alpha_1} U_0$ , where  $U_i$  are arbitrary unitaries and  $\alpha_i \in (0, 1]$  for all  $i$ . The fractional-query complexity of this algorithm is  $\sum_{i=1}^m \alpha_i$  and the total number of fractional-query gates used is  $m$ .

This idea can be taken further by taking the limit as the sizes of the fractional queries approach zero to obtain a continuous variant of the model, called the *continuous-query model* [26]. In this model, we have access to a query Hamiltonian  $H_x$  acting as  $H_x |j\rangle|b\rangle = \pi bx_j |j\rangle|b\rangle$ . Unlike the fractional- and discrete-query models, this is not a circuit-based model of computation. In this model we are allowed to evolve for time  $T$  according to the Hamiltonian given by  $H_x + H_D(t)$  for an arbitrary time-dependent driving Hamiltonian  $H_D(t)$ , at cost  $T$ . More precisely, the model is defined as follows.

**DEFINITION 2 (Continuous-query model).** Let  $H_x$  act as  $H_x |j\rangle|b\rangle = \pi bx_j |j\rangle|b\rangle$  for all  $j \in [N]$  and  $b \in \{0, 1\}$ . An algorithm in the continuous-query model is specified by an arbitrary  $x$ -independent driving Hamiltonian  $H_D(t)$  for  $t \in [0, T]$ . The algorithm implements the unitary operation  $U(T)$  obtained by solving the Schrödinger equation

$$i \frac{d}{dt} U(t) = (H_x + H_D(t)) U(t) \quad (2)$$

with  $U(0) = \mathbb{1}$ . The continuous-query complexity of this algorithm is the total evolution time,  $T$ .

Because  $e^{-i\alpha H_x} = Q_x^\alpha$ , running the Hamiltonian  $H_x$  with no driving Hamiltonian for time  $T = \alpha$  is equivalent to an  $\alpha$ -fractional query. In the remainder of this work we omit the subscript  $x$  on  $Q$  for brevity.

While initial work on the continuous-query model focused on finding analogues of known algorithms [26, 34], it has also been studied with the aim of proving lower bounds on the discrete-query model [34]. Furthermore, the model has

led to the discovery of new quantum algorithms. In particular, Farhi *et al.* [24] discovered an algorithm with continuous-query complexity  $O(\sqrt{n})$  for evaluating a balanced binary NAND tree with  $n$  leaves, which is optimal. This result was later converted to the discrete-query model with the same query complexity [2, 17].

A similar conversion can be performed for any algorithm with a sufficiently well-behaved driving Hamiltonian [15]. However, this leaves open the question of whether continuous-query algorithms can be generically converted to discrete-query algorithms with the same query complexity. This was almost resolved by [23], which gave an algorithm that approximates a  $T$ -query continuous-query algorithm to bounded error with  $O(T(\log T/\log \log T))$  discrete queries. This algorithm can be made time efficient [13] (informally, the number of additional 2-qubit gates is close to the query complexity).

However, to approximate a continuous-query algorithm to precision  $\epsilon$ , the algorithm of [23] uses  $O((1/\epsilon)(T \log T/\log \log T))$  queries. Ideally we would like the dependence on  $\epsilon$  to be polylogarithmic, instead of polynomial, in  $1/\epsilon$ . For example, such behavior would be desirable when using a fractional-query algorithm as a subroutine. Here we present a significantly improved and simplified simulation of the continuous- and fractional-query models. In particular, we show the following.

**THEOREM 2.3 (Continuous-query simulation).** *An algorithm with continuous- or fractional-query complexity  $T \geq 1$  can be simulated with error at most  $\epsilon$  with  $O(T(\log(T/\epsilon)/\log \log(T/\epsilon)))$  queries. For continuous-query simulation, if there is a circuit using at most  $g$  gates that implements the time evolution due to  $H_D(t)$  with precision  $\epsilon/T$  between any two times  $t_1$  and  $t_2$  satisfying  $|t_2 - t_1| \leq T$ , given  $t_1$  and  $t_2$  encoded in quantum states, then the number of additional 2-qubit gates for the simulation is  $O(T(\log(T/\epsilon)/\log \log(T/\epsilon))[g + \log(\bar{h}T/\epsilon)])$ , where  $\bar{h} := (1/T) \int_0^T \|H_D(t)\| dt$ .*

Since the continuous-query model is at least as powerful as the discrete-query model, a discrete simulation must use  $\Omega(T)$  queries, showing our dependence on  $T$  is close to optimal. However, as for the problem of Hamiltonian simulation, there was previously no  $\epsilon$ -dependent lower bound. Along the lines of Theorem 2.2, we show a lower bound of  $\Omega(\log(1/\epsilon)/\log \log(1/\epsilon))$  queries for a continuous-query algorithm with  $T = O(1)$  (Theorem 7.1), so the dependence of our simulation on  $\epsilon$  is optimal.

For the problem of evaluating a classical function of a black-box input, an approach based on an invariant called the  $\gamma_2$  norm shows that the continuous-query complexity is at most a constant factor smaller than the discrete-query complexity for a bounded-error simulation [31]. However, it remains unclear

whether the algorithm can be made time efficient and whether the unitary dynamics of a continuous-query algorithm can be simulated (even with bounded error) using  $O(T)$  queries. Such a result does hold for state conversion, but its dependence on error is quadratic [31]. More generally, the optimal tradeoff between  $T$  and  $\epsilon$  for simulation of continuous-query algorithms using discrete queries—and for simulation of Hamiltonian dynamics—remains open (with or without conditions on the time complexity).

The remainder of this article is organized as follows. In Section 3, we give a high-level overview of the techniques used in our results. In Section 4, we describe our simulation of the continuous- and fractional-query models using discrete queries. In Section 5, we apply these results to Hamiltonian simulation. In Section 6, we analyze the time complexity of our algorithms, and in Section 7, we prove  $\epsilon$ -dependent lower bounds showing optimality of their error dependence. We conclude in Section 8 with a brief discussion of some open questions. In Appendix A, we provide some proofs of known results for the sake of completeness.

### 3. High-level overview of techniques

We begin by proving Theorem 2.3, our improved simulation of continuous- and fractional-query algorithms. Then we prove Theorem 2.1 by reducing an instance of a sparse Hamiltonian simulation problem to an instance of a fractional-query algorithm, which can then be simulated via Theorem 2.3. We prove Theorem 2.2 using ideas from the no-fast-forwarding theorem from [8] and properties of the unbounded-error quantum query complexity of the parity function.

We now sketch the approach for each of the main theorems, highlighting the novel ideas.

**3.1. Continuous-query simulation (Theorem 2.3).** First consider the simulation of fractional queries using discrete queries. We show that an algorithm with constant fractional-query complexity can be simulated in the discrete-query model using  $O(\log(1/\epsilon)/\log \log(1/\epsilon))$  queries (Lemma 4.2). The claimed upper bound for simulating a fractional-query algorithm with query complexity  $T$  follows easily by breaking the algorithm into pieces with constant fractional-query complexity. Since the continuous- and fractional-query models are equivalent (Theorem 4.1), the result for the continuous-query model (Theorem 2.3) follows.

We prove Lemma 4.2 in two steps. Let the unitary performed by the constant-query fractional-query algorithm be  $V$  and let the (unknown) state it acts on be  $|\psi\rangle$ . We would like to create the state  $V|\psi\rangle$  up to error  $\epsilon$ . First we construct a

circuit  $\tilde{U}$  that performs  $V$  with amplitude  $\sqrt{p}$  up to error  $\epsilon$ , in the sense that  $\tilde{U}$  is within error  $\epsilon$  of a unitary  $U$  that maps  $|0^m\rangle|\psi\rangle$  to  $\sqrt{p}|0^m\rangle V|\psi\rangle + \sqrt{1-p}|\Phi^\perp\rangle$  for some constant  $p$  and some state  $|\Phi^\perp\rangle$  with  $(|0^m\rangle\langle 0^m| \otimes \mathbb{1})|\Phi^\perp\rangle = 0$ . The existence of such a  $\tilde{U}$  that makes  $O(\log(1/\epsilon)/\log\log(1/\epsilon))$  queries was shown by [23]. Their strategy is to measure the first  $m$  qubits and obtain  $V|\psi\rangle$  with constant probability. If the measurement fails, they recover the original state  $|\psi\rangle$  from  $|\Phi^\perp\rangle$  using a fault-correction procedure, which is itself probabilistic and occasionally fails, requiring a recursive correction algorithm to remove all faults. The time-efficient implementation of this recursive fault-correction procedure [13] is cumbersome.

Our alternative approach uses  $\tilde{U}$  to deterministically create  $V|\psi\rangle$  without measurements. We show in general how to create  $V|\psi\rangle$  with a constant number of applications of  $\tilde{U}$  when  $p$  is a constant. To do this, we introduce a notion of ‘oblivious amplitude amplification’ that can have the same performance as standard amplitude amplification, but that can be applied even when the reflection about the input state is unavailable. This idea, which is inspired by the in-place QMA amplification procedure of Marriott and Watrous [33], is a general result that can potentially be applied in other contexts [37, 44].

Most of the algorithm is easily made time efficient, except the preparation of a certain quantum state. However, this state can be prepared efficiently [13] and the result follows.

**3.2. Hamiltonian simulation reduction (Theorem 2.1).** Next we describe the main ideas of our Hamiltonian simulation algorithm. We remove the dependence of the query cost on  $n$  with a simple trick involving local edge coloring of bipartite graphs. This strategy is quite general and can be used to remove  $n$ -dependence from several known Hamiltonian simulation algorithms. The improved dependence on  $\epsilon$  results from our algorithm for simulating the fractional-query model in the discrete-query model (Theorem 2.3).

As mentioned previously, we reduce Hamiltonian simulation to a generalization of the task of simulating the fractional-query model. Examining the basic Lie product formula  $e^{-iHt} \approx (e^{-iH_1 t/r} e^{-iH_2 t/r} \dots e^{-iH_\eta t/r})^r$ , we see that if  $Q_j := e^{-iH_j}$  were query oracles, this would be a fractional-query algorithm using multiple oracles  $Q_j$  for time  $t$  each. (Note that because the query complexity of the simulation depends only on the total time over which fractional queries are applied rather than the total number of fractional queries, there is no advantage to using higher-order product formulas.) We reduce a fractional-query algorithm that calls each of  $\eta$  different query oracles for time  $t$  to a fractional-query algorithm that uses query time  $\eta t$  with a single query oracle that can perform any  $Q_j$ . Thus it suffices to decompose the given Hamiltonian  $H$  into a sum of Hamiltonians

for which the matrices  $Q_j$  can be viewed as query oracles in Theorem 2.3. We show such a decomposition (Lemma 5.3) that yields that stated upper bound. This algorithm can be made time efficient since it is essentially a reduction to continuous-query simulation.

**3.3. Lower bounds (Theorems 2.2 and 7.1).** Finally, we prove lower bounds showing optimality of our algorithms as a function of  $\epsilon$  (Theorems 2.2 and 7.1). The main idea behind both lower bounds is to show a Hamiltonian whose exact simulation for any time  $t > 0$  allows us to compute the parity of a string with unbounded error, which is as hard as computing parity exactly, requiring  $\Omega(N)$  queries [6, 25]. The evolution for constant time only produces an overlap of  $\Theta(1/N!)$  with the state encoding the parity. To prevent the overlap with the state encoding the incorrect parity being as large, the state must be produced with accuracy  $\epsilon = O(1/N!)$ . Since simulating the state with this accuracy would enable the parity to be computed with unbounded error, it must require complexity  $\Omega(N)$ . That then gives the lower bound of  $\Omega(\log(1/\epsilon)/\log \log(1/\epsilon))$ .

#### 4. From continuous to discrete queries

In this section we present our improved simulation of continuous or fractional queries in the conventional discrete-query model. The main result of this section is Lemma 4.8, which establishes the query complexity claimed in Theorem 2.3. The time-complexity part of Theorem 2.3 is established in Section 6.

For concreteness, we quantify the distance between unitaries  $U$  and  $V$  with the function  $\|U - V\|$  and the distance between states  $|\psi\rangle$  and  $|\phi\rangle$  with the function  $\| |\psi\rangle - |\phi\rangle \|$ . As the error ultimately appears inside a logarithm, the precise choice of distance measure is not significant.

We begin by recalling the equivalence of the continuous- and fractional-query models for any error  $\epsilon > 0$ . An explicit simulation of the continuous-query model by the fractional-query model was provided by [23]; the proof is a straightforward application of a result of [29]. The other direction is apparently folklore (for example, both directions are implicitly assumed in [34]); we provide a short proof in Appendix A.1 for completeness.

**THEOREM 4.1** (Equivalence of continuous- and fractional-query models). *For any  $\epsilon > 0$ , any algorithm with continuous-query complexity  $T$  can be implemented with fractional-query complexity  $T$  with error at most  $\epsilon$  and  $m = O(\bar{h}T^2/\epsilon)$  fractional-query gates, where  $\bar{h} := (1/T) \int_0^T \|H_D(t)\| dt$  is the average norm of the driving Hamiltonian. Conversely, any algorithm with fractional-query*

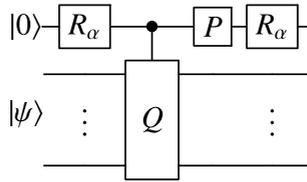


Figure 1. The fractional-query gadget. After performing the controlled- $Q$  operation on the target state  $|\psi\rangle$ , the operation  $Q^\alpha$  is performed with amplitude depending on  $\alpha$ .

complexity  $T$  can be implemented with continuous-query complexity  $T$  with error at most  $\epsilon$ .

Since the two models are equivalent, it suffices to convert a fractional-query algorithm to a discrete-query algorithm. We start with a fractional-query algorithm that makes at most one query. The result for multiple queries (Lemma 4.8) follows straightforwardly.

**Lemma 4.2.** *Any algorithm in the fractional-query model with query complexity at most 1 can be implemented with  $O(\log(1/\epsilon)/\log \log(1/\epsilon))$  queries in the discrete-query model with error at most  $\epsilon$ .*

The construction of the algorithm in this main lemma can be viewed in two steps. First, we show how to unitarily construct a superposition of the required state along with a label in state  $|0^{m+1}\rangle$  and another state whose label is orthogonal. The construction is similar to that in [13, 23]; the main difference is that we do not measure the state of the label. (This step is shown in the sequence Lemmas 4.3, 4.4, and 4.5.) Then, in the second step, rather than performing a fault-correction procedure upon seeing a measurement outcome other than  $0^{m+1}$ , we perform the underlying unitary operation in the first step three times (one of which is backwards) in conjunction with certain reflections to arrive at the required state. This step can be viewed as applying a generalization of amplitude amplification that is shown in Lemma 4.6.

The first step of the construction uses the fractional-query gadget [23, Section II.B] shown in Figure 1. This gadget behaves as follows, as we show in Appendix A.2.

**Lemma 4.3** (Gadget Lemma [23]). *Let  $Q$  be a unitary matrix with eigenvalues  $\pm 1$ ; let  $\alpha \in [0, 1]$ . The circuit in Figure 1, with  $R_\alpha := (1/\sqrt{c+s}) \begin{pmatrix} \sqrt{c} & \sqrt{s} \\ \sqrt{s} & -\sqrt{c} \end{pmatrix}$  and*

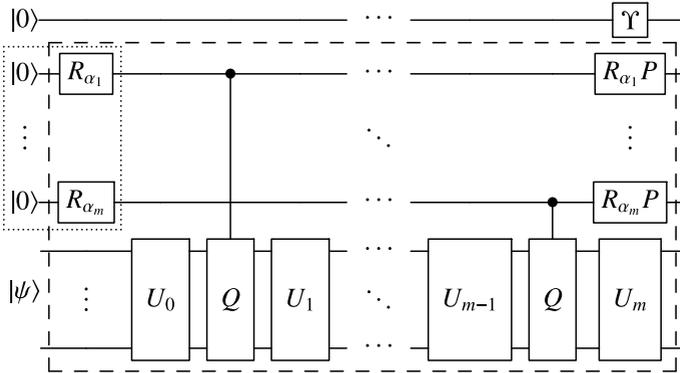


Figure 2. A segment to implement the fractional-query algorithm. The segment consists of many concatenated applications of the fractional-query gadget, interspersed with  $x$ -independent unitaries  $U_i$ . The state preparation is indicated in the dotted box, and the main operation is performed by the circuit in the dashed box. The additional ancilla at the top is introduced to reduce the amplitude for performing the correct operation to exactly  $1/2$ .

$P := \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ , performs the map

$$|0\rangle|\psi\rangle \mapsto \sqrt{q_\alpha}|0\rangle e^{i\pi\alpha/2} Q^\alpha |\psi\rangle + \sqrt{1-q_\alpha}|1\rangle|\phi\rangle \tag{3}$$

for some state  $|\phi\rangle$ , where  $c := \cos(\pi\alpha/2)$ ,  $s := \sin(\pi\alpha/2)$ ,  $q_\alpha := 1/(c+s)^2 = 1/(1+\sin(\pi\alpha))$ , and  $Q^\alpha = \frac{1}{2}(\mathbb{1} + Q) + e^{-i\pi\alpha} \frac{1}{2}(\mathbb{1} - Q) = e^{-i\pi\alpha/2}(c\mathbb{1} + isQ)$ .

While the proof in Appendix A.2 shows that  $|\phi\rangle = e^{-i\pi/4} Q^{-1/2}|\psi\rangle$ , we do not use this fact in our analysis, in contrast to previous approaches [13, 23].

Note that while we have defined the fractional-query model to use fractions  $\alpha \in (0, 1]$ , a similar simulation could be applied if we allowed negative fractional-time evolutions with  $\alpha \in [-1, 1]$ . In particular, we could define  $s = \sin(\pi|\alpha|/2)$ ,  $P = \begin{pmatrix} 1 & 0 \\ 0 & i \operatorname{sgn}(\alpha) \end{pmatrix}$  and carry through an analogous analysis. However, for simplicity, we restrict our attention to the model with only positive fractional-time evolutions.

We now collect the gadgets into segments as shown in Figure 2 and show that, with an appropriate choice of parameters, a segment implements a fractional-query algorithm with constant query complexity with amplitude  $1/2$ . This specific choice facilitates one-step exact oblivious amplitude amplification. Other than this choice of constant, this lemma is the same as in [23]. For completeness, we provide a proof in Appendix A.2.

**Lemma 4.4** (Segment Lemma). *Let  $V$  be a unitary implementable by a fractional-query algorithm with query complexity at most  $1/5$ , that is, there exists an  $m$  such*

that  $V = U_m Q^{\alpha_m} U_{m-1} \cdots U_1 Q^{\alpha_1} U_0$  with  $\alpha_i \geq 0$  for all  $i$  and  $\sum_{i=1}^m \alpha_i \leq 1/5$ . Let  $P$  and  $R_\alpha$  be as in Lemma 4.3. Then there exists a unitary  $\Upsilon$  on the additional ancilla such that the circuit in Figure 2 performs the map

$$|0^{m+1}\rangle|\psi\rangle \mapsto \frac{1}{2}|0^{m+1}\rangle e^{i\vartheta} V|\psi\rangle + \frac{\sqrt{3}}{2}|\Phi^\perp\rangle \quad (4)$$

for some state  $|\Phi^\perp\rangle$  satisfying  $(|0^{m+1}\rangle\langle 0^{m+1}| \otimes \mathbb{1})|\Phi^\perp\rangle = 0$  and some  $\vartheta \in [0, 2\pi)$ .

Although the segment in Figure 2 makes  $m$  queries, it is possible to approximate this segment within precision  $\epsilon$  using only  $O(\log(1/\epsilon)/\log\log(1/\epsilon))$  queries. To get some intuition for why this is possible, note that the state on the control registers decides how many queries are performed. For example, if all the control registers were set to  $|0\rangle$  when the controlled- $Q$  gates act, then no queries would be performed, even though the circuit contains  $m$  query gates. In general, the number of queries performed when the control registers are set to  $|b_1, b_2, \dots, b_m\rangle$  is the Hamming weight of  $b$ . In Figure 2, the state of the control registers has very little overlap with high-weight states, so we can approximate that state with one that has no overlap with high-weight states. We then show how to rearrange such a circuit to obtain a new circuit that uses very few query gates.

This lemma follows the same proof structure as [23, Section II.C], but is more general since we do not restrict all the fractional queries to have the same value of  $\alpha$ . This change requires us to use a version of the Chernoff bound for independent (but not necessarily identically distributed) random variables instead of the one used in [23]. The lemma is proved in Appendix A.2.

**Lemma 4.5** (Approximate Segment Lemma). *Let  $V$  be a unitary implementable by a fractional-query algorithm with query complexity at most  $1/5$ . Then for any  $\epsilon > 0$ , there exists a unitary quantum circuit that makes  $O(\log(1/\epsilon)/\log\log(1/\epsilon))$  discrete queries and, within error  $\epsilon$ , performs a unitary  $U$  acting as*

$$U|0^{m+1}\rangle|\psi\rangle = \frac{1}{2}|0^{m+1}\rangle e^{i\vartheta} V|\psi\rangle + \frac{\sqrt{3}}{2}|\Phi^\perp\rangle \quad (5)$$

for some state  $|\Phi^\perp\rangle$  satisfying  $(|0^{m+1}\rangle\langle 0^{m+1}| \otimes \mathbb{1})|\Phi^\perp\rangle = 0$  and some  $\vartheta \in [0, 2\pi)$ .

Up to this point our proof is similar to previous approaches [13, 23]. In those previous approaches, the map of Lemma 4.5 was used to probabilistically create the desired state by measuring the first  $m + 1$  qubits. With constant probability we obtain the desired state, but in the other case we have a fault and have to recover the original input state. This recovery stage required a fault-correction procedure that is difficult to analyze and considerably harder to make time efficient.

We avoid these difficulties by introducing oblivious amplitude amplification. Given a unitary  $U$  that implements another unitary  $V$  with some amplitude (in a certain precise sense), this idea allows one to use a version of amplitude amplification to give a better implementation of  $V$ . In particular, as in amplitude amplification, if the amplitude for implementing  $V$  is known, we can exactly perform  $V$ .

In standard amplitude amplification, to amplify the ‘good’ part of a state, we need to be able to reflect about the state itself and the projector onto the good subspace. While the latter is easy in our application, we cannot reflect about the unknown input state. Nevertheless, we show the following.

**Lemma 4.6** (Oblivious amplitude amplification). *Let  $U$  and  $V$  be unitary matrices on  $\mu + n$  qubits and  $n$  qubits, respectively, and let  $\theta \in (0, \pi/2)$ . Suppose that for any  $n$ -qubit state  $|\psi\rangle$ ,*

$$U|0^\mu\rangle|\psi\rangle = \sin(\theta)|0^\mu\rangle V|\psi\rangle + \cos(\theta)|\Phi^\perp\rangle, \tag{6}$$

where  $|\Phi^\perp\rangle$  is an  $(\mu + n)$ -qubit state that depends on  $|\psi\rangle$  and satisfies  $\Pi|\Phi^\perp\rangle = 0$ , where  $\Pi := |0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1}$ . Let  $R := 2\Pi - \mathbb{1}$  and  $S := -URU^\dagger R$ . Then for any  $\ell \in \mathbb{Z}$ ,

$$S^\ell U|0^\mu\rangle|\psi\rangle = \sin((2\ell + 1)\theta)|0^\mu\rangle V|\psi\rangle + \cos((2\ell + 1)\theta)|\Phi^\perp\rangle. \tag{7}$$

Note that  $R$  is not the reflection about the initial state, so Lemma 4.6 does not follow from amplitude amplification alone. However, in the context described in the lemma, it suffices to use a different reflection.

The motivation for oblivious amplitude amplification comes from work of Marriott and Watrous on in-place amplification of QMA [33] (see also related work on quantum rewinding for zero-knowledge proofs [41] and on using amplitude amplification to obtain a quadratic improvement [36]). Specifically, the following technical lemma shows that amplitude amplification remains in a certain 2-dimensional subspace in which it is possible to perform the appropriate reflections.

**Lemma 4.7** (2D Subspace Lemma). *Let  $U$  and  $V$  be unitary matrices on  $\mu + n$  qubits and  $n$  qubits, respectively, and let  $p \in (0, 1)$ . Suppose that for any  $n$ -qubit state  $|\psi\rangle$ ,*

$$U|0^\mu\rangle|\psi\rangle = \sqrt{p}|0^\mu\rangle V|\psi\rangle + \sqrt{1 - p}|\Phi^\perp\rangle, \tag{8}$$

where  $|\Phi^\perp\rangle$  is an  $(\mu + n)$ -qubit state that depends on  $|\psi\rangle$  and satisfies  $\Pi|\Phi^\perp\rangle = 0$ , where  $\Pi := |0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1}$ . Then the state  $|\Psi^\perp\rangle$  defined by the equation

$$U|\Psi^\perp\rangle := \sqrt{1 - p}|0^\mu\rangle V|\psi\rangle - \sqrt{p}|\Phi^\perp\rangle \tag{9}$$

is orthogonal to  $|\Psi\rangle := |0^\mu\rangle|\psi\rangle$  and satisfies  $\Pi|\Psi^\perp\rangle = 0$ .

*Proof.* For any  $|\psi\rangle$ , let  $|\Phi\rangle := |0^\mu\rangle V|\psi\rangle$ . Then for all  $|\psi\rangle$ , we have

$$U|\Psi\rangle = \sqrt{p}|\Phi\rangle + \sqrt{1-p}|\Phi^\perp\rangle \quad (10)$$

$$U|\Psi^\perp\rangle = \sqrt{1-p}|\Phi\rangle - \sqrt{p}|\Phi^\perp\rangle, \quad (11)$$

where  $\Pi|\Phi^\perp\rangle = 0$ . By taking the inner product of these two equations, we get  $\langle\Psi|\Psi^\perp\rangle = 0$ . The lemma asserts that not only is  $|\Psi^\perp\rangle$  orthogonal to  $|\Psi\rangle$ , but also  $\Pi|\Psi^\perp\rangle = 0$ .

To show this, consider the operator

$$Q := (|0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1})U^\dagger\Pi U(|0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1}). \quad (12)$$

For any state  $|\psi\rangle$ ,

$$\langle\psi|Q|\psi\rangle = \|\Pi U|0^\mu\rangle|\psi\rangle\|^2 = \|\Pi(\sqrt{p}|\Phi\rangle + \sqrt{1-p}|\Phi^\perp\rangle)\|^2 = \|\sqrt{p}|\Phi\rangle\|^2 = p. \quad (13)$$

In particular, this holds for a basis of eigenvectors of  $Q$ , so  $Q = p\mathbb{1}$ .

Thus for any  $|\psi\rangle$ , we have

$$\begin{aligned} p|\psi\rangle &= Q|\psi\rangle = (|0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1})U^\dagger\Pi U(|0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1})|\psi\rangle = (|0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1})U^\dagger\Pi U|\Psi\rangle \\ &= \sqrt{p}(|0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1})U^\dagger|\Phi\rangle. \end{aligned} \quad (14)$$

From (10) and (11) we get  $U^\dagger|\Phi\rangle = \sqrt{p}|\Psi\rangle + \sqrt{1-p}|\Psi^\perp\rangle$ . Plugging this into the previous equation, we get

$$\begin{aligned} p|\psi\rangle &= \sqrt{p}(|0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1})(\sqrt{p}|\Psi\rangle + \sqrt{1-p}|\Psi^\perp\rangle) \\ &= p|\psi\rangle + \sqrt{p(1-p)}(|0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1})|\Psi^\perp\rangle. \end{aligned} \quad (15)$$

This gives us  $\sqrt{p(1-p)}(|0^\mu\rangle\langle 0^\mu| \otimes \mathbb{1})|\Psi^\perp\rangle = 0$ . Since  $p \in (0, 1)$ , this implies  $\Pi|\Psi^\perp\rangle = 0$ .  $\square$

Note that this fact can also be viewed as a consequence of Jordan's Lemma [30], which decomposes the space into a direct sum of 1- and 2-dimensional subspaces that are invariant under the projectors  $\Pi$  and  $U^\dagger\Pi U$ . In this decomposition,  $\Pi$  and  $U^\dagger\Pi U$  are rank-1 projectors within each 2-dimensional subspace. Let  $|0\rangle|\psi_i\rangle$  denote the eigenvalue-1 eigenvector of  $\Pi$  within the  $i$ th 2-dimensional subspace  $S_i$ . Since  $S_i$  is invariant under  $U^\dagger\Pi U$ , the state  $U^\dagger\Pi U|0\rangle|\psi_i\rangle = \sqrt{p}U^\dagger|0\rangle V|\psi_i\rangle$  belongs to  $S_i$ . Let  $|\Phi_i^\perp\rangle$  be such that  $|0\rangle|\psi_i\rangle = U^\dagger(\sqrt{p}|0\rangle V|\psi_i\rangle + \sqrt{1-p}|\Phi_i^\perp\rangle)$ . Then  $|\Psi_i^\perp\rangle := U^\dagger(\sqrt{1-p}|0\rangle V|\psi_i\rangle - \sqrt{p}|\Phi_i^\perp\rangle)$  is in  $S_i$ , since it is a linear combination of  $|0\rangle|\psi_i\rangle$  and  $U^\dagger\Pi U|0\rangle|\psi_i\rangle$ . However,  $|\Psi_i^\perp\rangle$  is orthogonal to  $|0\rangle|\psi_i\rangle$  and is therefore an eigenvalue-0 eigenvector of  $\Pi$ , since  $\Pi$  is a rank-1

projector in  $S_i$ . Thus for each  $i$ ,  $|\psi_i\rangle$  and  $|\Psi_i^\perp\rangle$  satisfy the conditions of the lemma. We claim that the number of 2-dimensional subspaces (and hence the number of states  $|\psi_i\rangle$ ) is  $2^n$ . There are at most  $2^n$  such subspaces since  $\Pi$  has rank  $2^n$  and is rank-1 in each subspace. There also must be at least  $2^n$  2-dimensional subspaces, since otherwise there would be a state  $|0\rangle|\psi\rangle$  that is in a 1-dimensional subspace, that is, is invariant under both  $\Pi$  and  $U^\dagger \Pi U$ . This is not possible because  $U^\dagger \Pi U$  acting on  $|0\rangle|\psi\rangle$  yields  $\sqrt{p}U^\dagger|0\rangle V|\psi\rangle$ , which is a subnormalized state since  $p < 1$ . Finally, since there are  $2^n$  linearly independent  $|\psi_i\rangle$ , an arbitrary state  $|\psi\rangle$  can be written as a linear combination of  $|\psi_i\rangle$ , and the result follows.

With the help of Lemma 4.7 we can prove Lemma 4.6.

*Proof of Lemma 4.6.* Since Lemma 4.7 shows that the evolution occurs within a 2-dimensional subspace (or its image under  $U$ ), the remaining analysis is essentially the same as in standard amplitude amplification. For any  $|\psi\rangle$ , we define  $|\Psi\rangle := |0^\mu\rangle|\psi\rangle$  and  $|\Phi\rangle := |0^\mu\rangle V|\psi\rangle$ , so that

$$U|\Psi\rangle = \sin(\theta)|\Phi\rangle + \cos(\theta)|\Phi^\perp\rangle, \tag{16}$$

where  $\theta \in (0, \pi/2)$  is such that  $\sqrt{p} = \sin(\theta)$ . We also define  $|\Psi^\perp\rangle$  through the equation

$$U|\Psi^\perp\rangle := \cos(\theta)|\Phi\rangle - \sin(\theta)|\Phi^\perp\rangle. \tag{17}$$

By Lemma 4.7, we know that  $\Pi|\Psi^\perp\rangle = 0$ . Using these two equations, we have

$$U^\dagger|\Phi\rangle = \sin(\theta)|\Psi\rangle + \cos(\theta)|\Psi^\perp\rangle \tag{18}$$

$$U^\dagger|\Phi^\perp\rangle = \cos(\theta)|\Psi\rangle - \sin(\theta)|\Psi^\perp\rangle. \tag{19}$$

Then a straightforward calculation gives

$$\begin{aligned} S|\Phi\rangle &= -URU^\dagger|\Phi\rangle \\ &= -UR(\sin(\theta)|\Psi\rangle + \cos(\theta)|\Psi^\perp\rangle) \\ &= -U(\sin(\theta)|\Psi\rangle - \cos(\theta)|\Psi^\perp\rangle) \\ &= (\cos^2(\theta) - \sin^2(\theta))|\Phi\rangle - 2\cos(\theta)\sin(\theta)|\Phi^\perp\rangle \\ &= \cos(2\theta)|\Phi\rangle - \sin(2\theta)|\Phi^\perp\rangle. \end{aligned} \tag{20}$$

Similarly,

$$\begin{aligned} S|\Phi^\perp\rangle &= URU^\dagger|\Phi^\perp\rangle \\ &= UR(\cos(\theta)|\Psi\rangle - \sin(\theta)|\Psi^\perp\rangle) \\ &= U(\cos(\theta)|\Psi\rangle + \sin(\theta)|\Psi^\perp\rangle) \end{aligned}$$

$$\begin{aligned}
&= 2 \cos(\theta) \sin(\theta) |\Phi\rangle + (\cos^2(\theta) - \sin^2(\theta)) |\Phi^\perp\rangle \\
&= \sin(2\theta) |\Phi\rangle + \cos(2\theta) |\Phi^\perp\rangle.
\end{aligned} \tag{21}$$

Thus we see that  $S$  acts as a rotation by  $2\theta$  in the subspace  $\text{span}\{|\Phi\rangle, |\Phi^\perp\rangle\}$ , and the result follows.  $\square$

We are now ready to complete the proof of Lemma 4.2 using Lemmas 4.5 and 4.6.

*Proof of Lemma 4.2.* We are given a fractional-query algorithm that makes at most one query. This can be split into 5 steps that make at most  $1/5$  queries each in the fractional-query model. We perform the analysis for these steps of size  $1/5$ ; the difference is only a constant factor that does not affect the asymptotics. We convert this fractional-query algorithm into a discrete-query algorithm with some error.

From Lemma 4.5, we know that for any such fractional-query algorithm  $V$ , there is an algorithm that makes  $O(\log(1/\epsilon)/\log \log(1/\epsilon))$  discrete queries and maps the state  $|0^{m+1}\rangle|\psi\rangle$  to a state that is at most  $\epsilon$  far from  $\frac{1}{2}|0^{m+1}\rangle e^{i\vartheta} V|\psi\rangle + \frac{\sqrt{3}}{2}|\Phi\rangle$ , for some state  $|\Phi\rangle$  that satisfies  $(|0^{m+1}\rangle\langle 0^{m+1}| \otimes \mathbb{1})|\Phi\rangle = 0$  and some  $\vartheta \in [0, 2\pi)$ . We wish to perform the unitary  $V$  on the input state  $|\psi\rangle$  approximately.

The unitary operation  $U$  defined in Lemma 4.5 maps  $|0^{m+1}\rangle|\psi\rangle \mapsto \frac{1}{2}|0^{m+1}\rangle e^{i\vartheta} V|\psi\rangle + \frac{\sqrt{3}}{2}|\Phi\rangle$ . The operation  $U$  satisfies the conditions of Lemma 4.6 with  $\mu = m + 1$  and  $\sin^2(\theta) = 1/4$ . Thus a single application of  $S$  (using three applications of  $U$ ) would produce the state  $V|\psi\rangle$  exactly.

While we cannot necessarily perform  $U$ , using Lemma 4.5 we can perform another unitary operation  $\tilde{U}$  that is within error  $\epsilon/3$  of  $U$ . Since we only perform the unitary three times, we obtain a state  $\epsilon$ -close to  $V|\psi\rangle$  when we use  $\tilde{U}$  instead of  $U$ .  $\square$

By straightforwardly concatenating such simulations with sufficiently small error, we obtain simulations for longer times. This establishes the following lemma, which is the query-complexity part of Theorem 2.3.

**Lemma 4.8.** *An algorithm with continuous- or fractional-query complexity  $T \geq 1$  can be simulated with error at most  $\epsilon$  with  $O(T(\log(T/\epsilon)/\log \log(T/\epsilon)))$  queries.*

*Proof.* Given an algorithm that runs for time  $T$  in the continuous-query model, we can convert it to an algorithm with fractional-query complexity  $T$  with error at most  $\epsilon/2$  using Theorem 4.1. Given a fractional-query algorithm that makes  $T$  queries, we can divide it into  $\lceil T \rceil$  pieces that make at most one query each and invoke Lemma 4.2 with error  $\epsilon/2\lceil T \rceil$  to obtain  $\lceil T \rceil$  discrete-query

algorithms, each of which makes  $O(\log(\lceil T \rceil/\epsilon)/\log \log(\lceil T \rceil/\epsilon))$  queries. When run sequentially on the input state, they yield an output that is  $\epsilon/2$ -close to the correct output (by subadditivity of error). Thus the final state has error at most  $\epsilon$ .  $\square$

## 5. Hamiltonian simulation

We now apply the results of the previous section to give improved algorithms for simulating sparse Hamiltonians. The main result of this section is the reduction from an instance of the sparse Hamiltonian simulation problem to a fractional-query algorithm, which establishes Lemma 5.5, the query-complexity part of Theorem 2.1. The time-complexity part of Theorem 2.1 is established in Section 6.

To see the connection between the fractional-query model and Hamiltonian simulation, consider the example of a Hamiltonian  $H = H_1 + H_2$ , where  $H_1$  and  $H_2$  have eigenvalues 0 and  $\pi$ , so that  $e^{-iH_1}$  and  $e^{-iH_2}$  have eigenvalues  $\pm 1$ . From the Lie product formula, we have  $e^{-i(H_1+H_2)T} \approx (e^{-iH_1T/r}e^{-iH_2T/r})^r$  for large  $r$ . If we think of  $H_1$  and  $H_2$  as query Hamiltonians, this is a fractional-query algorithm that makes  $T$  queries to each Hamiltonian. We might therefore expect that  $O(T(\log(T/\epsilon)/\log \log(T/\epsilon)))$  discrete queries to  $e^{-iH_1}$  and  $e^{-iH_2}$  suffice to implement  $e^{-i(H_1+H_2)T}$  to precision  $\epsilon$ . Here we do this by generalizing the results of the previous section to allow multiple fractional-query oracles.

For a set  $\mathcal{Q} = \{Q_1, \dots, Q_\eta\}$  of unitary matrices with eigenvalues  $\pm 1$ , we say  $U$  is a fractional-query algorithm over  $\mathcal{Q}$  with cost  $T$  if  $U$  can be written as  $U_\lambda Q_{i_\lambda}^{\alpha_\lambda} U_{\lambda-1} \dots U_1 Q_{i_1}^{\alpha_1} U_0$ , where  $0 < \alpha_i \leq 1$ ,  $\sum_{i=1}^\lambda \alpha_i = T$ , and  $i_j \in [\eta]$  for all  $j \in [\lambda]$ .

**THEOREM 5.1 (Multiple-query model).** *Let  $\mathcal{Q} = \{Q_1, \dots, Q_\eta\}$  be a set of unitaries with eigenvalues  $\pm 1$ . Let  $U$  be a fractional-query algorithm over  $\mathcal{Q}$  with cost  $T$ . Let  $Q := \sum_{j=1}^\eta |j\rangle\langle j| \otimes Q_j$ . Then  $U$  can be implemented by a circuit that makes  $O(T(\log(T/\epsilon)/\log \log(T/\epsilon)))$  queries to  $Q$  with error at most  $\epsilon$ .*

*Proof.* We prove this by reduction to Theorem 2.3. We know that  $U$  can be written in the form  $U = U_\lambda Q_{i_\lambda}^{\alpha_\lambda} U_{\lambda-1} \dots U_1 Q_{i_1}^{\alpha_1} U_0$ , where  $0 < \alpha_i \leq 1$ ,  $\sum_{i=1}^\lambda \alpha_i = T$ , and  $i_j \in [\eta]$  for all  $j \in [\lambda]$ .

We first express  $U$  as a fractional-query algorithm over  $\mathcal{Q}$  with cost  $T$ . To do this, we add an extra control register to the original circuit for  $U$ . This register holds the index  $i_j$  of the next query to be performed. We start with this register initialized to  $|0\rangle$ . Let  $V_0$  be any unitary that maps  $|0\rangle$  to  $|i_1\rangle$ . The action of  $Q_{i_1}^{\alpha_1} U_0$  on any state  $|\psi\rangle$  is the same as the action of  $Q^{\alpha_1}(V_0 \otimes U_0)$  on the second register

of  $|0\rangle|\psi\rangle$ . Similarly, for all  $j \in [\lambda]$ , let  $V_j$  be any unitary that maps  $|i_j\rangle$  to  $|i_{j+1}\rangle$ , where  $i_{\lambda+1} := 0$ . Thus the circuit  $(V_\lambda \otimes U_\lambda)Q^{\alpha_\lambda}(V_{\lambda-1} \otimes U_{\lambda-1}) \cdots (V_1 \otimes U_1)Q^{\alpha_1}(V_0 \otimes U_0)$  maps  $|0\rangle|\psi\rangle$  to  $|0\rangle U|\psi\rangle$ .

This construction gives a fractional-query algorithm with fractional-query complexity  $T$  given oracle access to  $Q$ . Since  $Q$  has eigenvalues  $\pm 1$ , we can invoke Theorem 2.3 to give a discrete-query algorithm that makes  $O(T(\log(T/\epsilon)/\log\log(T/\epsilon)))$  queries to  $Q$  and performs  $U$  up to error  $\epsilon$ . Theorem 2.3 assumes the queries are diagonal in the computational basis, whereas here we assume only that  $Q$  has eigenvalues  $\pm 1$ . However, these two scenarios are equivalent since the target system can be considered in a basis where  $Q$  is diagonal. Therefore, Theorem 2.3 applies to the slightly more general scenario considered here.  $\square$

This theorem allows us to simulate a Hamiltonian  $H = H_1 + \cdots + H_\eta$  for time  $t$  using resources that scale only slightly superlinearly in  $\eta t$ , provided each  $H_j$  has eigenvalues 0 and  $\pi$  (or more generally, by rescaling, provided each  $H_j$  has the same two eigenvalues). For any  $\epsilon > 0$ , there is a sufficiently large  $r$  so that  $e^{-iHt}$  is  $\epsilon$ -close to  $(e^{-iH_1 t/r} \cdots e^{-iH_\eta t/r})^r$ , which is of the form required by Theorem 5.1 if  $e^{-iH_j}$  has eigenvalues  $\pm 1$ . Since  $\|e^{-iHt} - (e^{-iH_1 t/r} \cdots e^{-iH_\eta t/r})^r\| = O((\eta \bar{h} t)^2/r)$ , where  $\bar{h} := \max_j \|H_j\|$  [8], choosing  $r = \Omega((\eta \bar{h} t)^2/\epsilon)$  is sufficient to achieve an  $\epsilon$ -approximation. Since our Hamiltonians  $H_j$  have constant norm, we have  $\bar{h} = O(1)$  and get the following corollary.

**Corollary 5.2.** *For a Hamiltonian  $H = \sum_{j=1}^\eta H_j$ , where  $H_j$  has eigenvalues 0 and  $\pi$  for all  $j \in [\eta]$ , define  $Q := \sum_j |j\rangle\langle j| \otimes e^{-iH_j}$ . The unitary  $e^{-iHt}$  can be implemented by a fractional-query algorithm over  $Q$ , up to error  $\epsilon$ , with query complexity  $\tau = \eta t$  and  $O(\eta^3 t^2/\epsilon)$  fractional-query gates. Thus  $e^{-iHt}$  can be implemented up to error  $\epsilon$  by a circuit with  $O(\tau(\log(\tau/\epsilon)/\log\log(\tau/\epsilon)))$  invocations of  $Q$ .*

To simulate arbitrary sparse Hamiltonians, we decompose them into Hamiltonians with this property. To do this we first decompose the Hamiltonian into a sum of 1-sparse Hamiltonians (with at most 1 nonzero entry in any row or column). Second, we decompose 1-sparse Hamiltonians into Hamiltonians in the required form.

**Lemma 5.3.** *For any 1-sparse Hamiltonian  $G$  and precision  $\gamma > 0$ , there exist  $O(\|G\|_{\max}/\gamma)$  Hamiltonians  $G_j$  with eigenvalues  $\pm 1$  such that  $\|G - \gamma \sum_j G_j\|_{\max} \leq \sqrt{2}\gamma$ .*

*Proof.* First we decompose the Hamiltonian  $G$  as  $G = G_X + iG_Y + G_Z$ , where  $G_X$  contains the off-diagonal real terms,  $iG_Y$  contains the off-diagonal imaginary

terms, and  $G_Z$  contains the on-diagonal real terms. Next, for each of  $G_\xi$  for  $\xi \in \{X, Y, Z\}$ , we construct an approximation  $\tilde{G}_\xi$  with each entry rounded off to the closest multiple of  $2\gamma$ . Since each entry of  $\tilde{G}_\xi$  is at most  $\gamma$  away from the corresponding entry in  $G_\xi$ , we have  $\|G_\xi - \tilde{G}_\xi\|_{\max} \leq \gamma$ . Denoting  $\tilde{G} = \tilde{G}_X + i\tilde{G}_Y + \tilde{G}_Z$ , this implies  $\|G - \tilde{G}\|_{\max} \leq \sqrt{2}\gamma$ .

Next, we take  $C^\xi := \tilde{G}_\xi/\gamma$ , so  $\|C^\xi\|_{\max} \leq 2\lceil\|G_\xi\|_{\max}/2\gamma\rceil \leq 2\lceil\|G\|_{\max}/2\gamma\rceil$ . We can then decompose each 1-sparse matrix  $C^\xi$  into  $\|C^\xi\|_{\max}$  matrices, each of which is 1-sparse and has entries from  $\{-2, 0, 2\}$ . If  $C_{jk}^\xi$  is  $2p$ , then the first  $|p|$  matrices in the decomposition have a 2 for  $p > 0$  (or  $-2$  if  $p < 0$ ) at the  $(j, k)$  entry, and the rest have 0. More explicitly, we define

$$C_{jk}^{\xi,\ell} := \begin{cases} 2 & \text{if } C_{jk}^\xi \geq 2\ell > 0 \\ -2 & \text{if } C_{jk}^\xi \leq -2\ell < 0 \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

for  $\xi \in \{X, Y, Z\}$  and  $\ell \in [\|C^\xi\|_{\max}/2]$ . This gives a decomposition into at most  $3\lceil\|G\|_{\max}/2\gamma\rceil$  terms with eigenvalues in  $\{-2, 0, 2\}$ .

To obtain matrices with eigenvalues  $\pm 1$ , we perform one more step to remove the 0 eigenvalues. We divide each  $C^{\xi,\ell}$  into two copies,  $C^{\xi,\ell,+}$  and  $C^{\xi,\ell,-}$ . For any column where  $C^{\xi,\ell}$  is all zero, the corresponding diagonal element of  $C^{\xi,\ell,+}$  is  $+1$  (if  $\xi \in \{X, Z\}$ ) or  $+i$  (if  $\xi = Y$ ) and the diagonal element of  $C^{\xi,\ell,-}$  is  $-1$  (if  $\xi \in \{X, Z\}$ ) or  $-i$  (if  $\xi = Y$ ). Otherwise, we let  $C_{jk}^{\xi,\ell,+} = C_{jk}^{\xi,\ell,-} = C_{jk}^{\xi,\ell}/2$ . Thus  $C^{\xi,\ell} = C^{\xi,\ell,+} + C^{\xi,\ell,-}$ . Moreover, each column of  $C^{\xi,\ell,\pm}$  has exactly one nonzero entry, which is  $\pm 1$  (or  $\pm i$  on the diagonal of  $C^{Y,\ell,\pm}$ ).

This gives a decomposition  $\tilde{G}/\gamma = \sum_{\ell,\pm} (C^{X,\ell,\pm} + iC^{Y,\ell,\pm} + C^{Z,\ell,\pm})$  in which each term has eigenvalues  $\pm 1$ . The decomposition contains at most  $6\lceil\|G\|_{\max}/2\gamma\rceil = O(\|G\|_{\max}/\gamma)$  terms.  $\square$

Lemma 5.3 gives a decomposition of the required form as the eigenvalues can be adjusted to 0 and  $\pi$  by adding the identity matrix and multiplying by  $\pi/2$ .

It remains to decompose a sparse Hamiltonian into 1-sparse Hamiltonians. Known results decompose a  $d$ -sparse Hamiltonian  $H$  into a sum of  $O(d^2)$  1-sparse Hamiltonians [8], but simulating one query to a 1-sparse Hamiltonian requires  $O(\log^* n)$  queries to the oracle for  $H$ . We present a simplified decomposition theorem that decomposes a  $d$ -sparse Hamiltonian into  $d^2$  1-sparse Hamiltonians. A query to the individual 1-sparse Hamiltonians can be performed using  $O(1)$  queries to the original Hamiltonian, removing the  $\log^* n$  factor.

**Lemma 5.4.** *If  $H$  is a  $d$ -sparse Hamiltonian, there exists a decomposition  $H = \sum_{j=1}^{d^2} H_j$  where each  $H_j$  is 1-sparse and a query to any  $H_j$  can be simulated with  $O(1)$  queries to  $H$ .*

*Proof.* The new ingredient in our proof is to assume that the graph of  $H$  is bipartite. (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.) This is without loss of generality because we can simulate the Hamiltonian  $\sigma_x \otimes H$  instead, which is indeed bipartite and has the same sparsity as  $H$ . From a simulation of  $\sigma_x \otimes H$ , we can recover a simulation of  $H$  using the identity  $e^{-i(\sigma_x \otimes H)t} |+\rangle |\psi\rangle = |+\rangle e^{-iHt} |\psi\rangle$ .

Now we decompose a bipartite  $d$ -sparse Hamiltonian into a sum of  $d^2$  terms. To do this, we give an edge coloring of the graph of  $H$  (that is, an assignment of colors to the edges so that no two edges incident on the same vertex have the same color). Given such a coloring with  $d^2$  colors, the Hamiltonian  $H_j$  formed by only considering edges with color  $j$  is 1-sparse.

We use the following simple coloring. For any pair of adjacent vertices  $u$  and  $v$ , let  $r(u, v)$  denote the rank of  $v$  in  $u$ 's neighbor list, that is, the position occupied by  $v$  in a sorted list of  $u$ 's neighbors. This is a number between 1 and  $d$ . Let the color of the edge  $(u, v)$ , where  $u$  comes from the left part of the bipartition and  $v$  comes from the right, be the ordered pair  $(r(u, v), r(v, u))$ . This is a valid coloring since if  $(u, v)$  and  $(u, w)$  have the same color, then in particular the first component of the ordered pair is the same, so  $r(u, v) = r(u, w)$  implies  $v = w$ . A similar argument handles the case where the common vertex is on the right.

Given a color  $(a, b)$ , it is easy to simulate queries to the Hamiltonian corresponding to that color. To compute the nonzero entries of the  $j$ th row for this color, if  $j$  is in the left partition, then we find the neighbor of  $j$  that has rank  $a$ ; let us call this  $\ell$ . Then we find the neighbor of  $\ell$  that has rank  $b$ . If this neighbor is  $j$ , then  $\ell$  is the position of the nonzero entry in row  $j$ ; otherwise there is no nonzero entry. If  $j$  is in the right partition, the procedure is the same, except with the roles of  $a$  and  $b$  reversed. This procedure uses two queries.  $\square$

Observe that the simple trick of making the Hamiltonian bipartite suffices to remove the  $O(\log^* n)$  term present in previous decompositions of this form. This trick is quite general and can be applied to remove a factor of  $O(\log^* n)$  wherever such a factor appears in a known Hamiltonian simulation algorithm (for example, [8, 19, 43]).

**Lemma 5.5.** *A  $d$ -sparse Hamiltonian  $H$  can be simulated for time  $t$  with error at most  $\epsilon$  using  $O(\tau(\log(\tau/\epsilon)/\log \log(\tau/\epsilon)))$  queries, where  $\tau := d^2 \|H\|_{\max} t \geq 1$ .*

*Proof.* Lemma 5.4 decomposes our Hamiltonian  $H$  into  $d^2$  1-sparse Hamiltonians. We further decompose  $H$  using Lemma 5.3 into a sum of  $\eta = O(d^2 \|H\|_{\max}/\gamma)$  Hamiltonians  $G_j$  such that  $\|H - \gamma \sum_{j=1}^{\eta} G_j\|_{\max} \leq \sqrt{2}\gamma d^2$ , since each 1-sparse Hamiltonian is approximated with precision  $\sqrt{2}\gamma$  and there are  $d^2$  approximations in this sum. To upper bound the simulation error, we have  $\|e^{-iHt} - e^{-i\gamma \sum_{j=1}^{\eta} G_j t}\| \leq \|(H - \gamma \sum_{j=1}^{\eta} G_j)t\| \leq \sqrt{2}\gamma d^3 t$ , where we used the fact that  $\|e^{iA} - e^{iB}\| \leq \|A - B\|$  (as explained in the proof of Theorem 4.1) and  $\|A\| \leq d\|A\|_{\max}$  for a  $d$ -sparse matrix  $A$ . Choosing  $\gamma = \epsilon/\sqrt{2}d^3 t$  gives the required precision. We now invoke Corollary 5.2 with number of Hamiltonians  $\eta = O(d^2 \|H\|_{\max}/\gamma)$  and simulation time  $\gamma t$  to get  $\tau = d^2 \|H\|_{\max} t$ . Plugging this value of  $\tau$  into Corollary 5.2 gives the result in Lemma 5.5, which is the query-complexity part of Theorem 2.1.  $\square$

Note that above we have determined the values of  $r$  and  $\gamma$  to use, but these values do not affect the query complexity (although they do affect the time complexity). This is because  $r$  and  $\gamma$  affect the value of  $m$  in Section 4, but the analysis in Section 4 is independent of  $m$ . This enables a simple generalization to time-dependent Hamiltonians. We can approximate the true evolution by a product of evolutions under time-independent Hamiltonians for each of the  $r$  time intervals of length  $t/r$ . Provided the derivative of the Hamiltonian is bounded, this approximation can be made arbitrarily accurate by choosing  $r$  large enough. As the query complexity does not depend on  $r$ , it is independent of  $h'$  (defined as  $\max_{s \in [0, t]} \|(d/ds)H(s)\|$  in Section 2), similar to [38].

Finally, consider simulating a  $k$ -local Hamiltonian. A term acting nontrivially on at most  $k$  qubits is  $2^k$ -sparse: two states  $x, y \in \{0, 1\}^n$  are adjacent if the only bits on which  $x$  and  $y$  differ are among the  $k$  bits involved in the local term. Using this structure, we can give an explicit  $2^k$ -coloring, improving over the  $4^k$ -coloring provided by Lemma 5.4: we simply color an edge between states  $x$  and  $y$  by indicating which of the  $k$  bits are flipped. Thus we can decompose a  $k$ -local Hamiltonian with  $M$  terms as a sum of  $2^k M$  1-sparse Hamiltonians. Using this decomposition in place of Lemma 5.4, we find a simulation as in Theorem 2.1 but with  $\tau$  replaced by  $\tilde{\tau} := 2^k M \|H\|_{\max} t$ .

## 6. Time complexity

We now consider the time complexities of the algorithms described in Theorems 2.3 and 2.1 (recall that time complexity refers to the sum of the number of queries and additional 2-qubit gates used in the algorithm). Our approach considerably simplifies this analysis over previous work and gives improved upper bounds.

The basic algorithm as described in Section 4 is inefficient as it relies on creating a state of  $m = \text{poly}(h, T, 1/\epsilon)$  qubits. Instead, as in previous work [13], we create a compressed version of this state that allows us to perform the necessary controlled operations and to reflect about the zero state. Our simplified approach does not require measuring the control qubits, an operation that accounts for much of the technical complexity of [13].

We now prove Theorem 2.3 from Section 1, which we restate for convenience.

**THEOREM 2.3 (Continuous-query simulation).** *An algorithm with continuous- or fractional-query complexity  $T \geq 1$  can be simulated with error at most  $\epsilon$  with  $O(T(\log(T/\epsilon)/\log \log(T/\epsilon)))$  queries. For continuous-query simulation, if there is a circuit using at most  $g$  gates that implements the time evolution due to  $H_D(t)$  with precision  $\epsilon/T$  between any two times  $t_1$  and  $t_2$  satisfying  $|t_2 - t_1| \leq T$ , given  $t_1$  and  $t_2$  encoded in quantum states, then the number of additional 2-qubit gates for the simulation is  $O(T(\log(T/\epsilon)/\log \log(T/\epsilon))[g + \log(\bar{h}T/\epsilon)])$ , where  $\bar{h} := (1/T) \int_0^T \|H_D(t)\| dt$ .*

*Proof.* The query complexity of this theorem was established in Lemma 4.8. As in the analysis of query complexity, it suffices to simulate a segment implementing evolution for time  $1/5$  with precision  $\epsilon/5T$ . To simulate the continuous-query model, we can assume without loss of generality that query evolutions are approximated (as in Theorem 4.1) by  $m$  fractional evolutions of equal length  $1/5m$ . Thus we can assume that in each segment, as defined in Lemma 4.4,  $\alpha := \alpha_i = 1/5m$  for all  $i \in [m]$ . Let  $c := \cos(\pi/10m)$  and  $s := \sin(\pi/10m)$ .

The idealized initial state of the ancilla qubits (that is, the state in the dotted box of Figure 2) is

$$\left( \frac{\sqrt{c}|0\rangle + \sqrt{s}|1\rangle}{\sqrt{c+s}} \right)^{\otimes m} = \sum_{b \in \{0,1\}^m} \kappa^{m-|b|} \sigma^{|b|} |b\rangle, \quad (23)$$

where  $\kappa := \sqrt{c}/\sqrt{c+s}$  and  $\sigma := \sqrt{s}/\sqrt{c+s}$ . We truncate this state to the subspace of those  $b$  with Hamming weight  $|b| \leq k$ . Specifically, we prepare the encoded state

$$\sum_{\ell \in L} \kappa^{m-|\ell|} \sigma^{|\ell|} |\ell\rangle + \delta |\perp\rangle, \quad (24)$$

where  $L := \{(\ell_1, \dots, \ell_h) : 1 \leq h \leq k, \ell_1 + \dots + \ell_h \leq m - h\}$ ,  $|\perp\rangle$  is a special state orthogonal to all terms in the first sum, and the coefficient  $\delta$  was shown to be small in Lemma 4.5. Observe that there is a natural bijection between  $L$  and the set of strings  $b$  with  $|b| \leq k$ , given by  $b \leftrightarrow 0^{\ell_1} 10^{\ell_2} 10^{\ell_3} \dots 0^{\ell_h} 10^{m-h-\ell_1-\dots-\ell_h}$ .

It is straightforward to perform the operation (A.13) from the proof of Lemma 4.5, conditioning on  $b$  as represented by  $\ell$ . Recall that  $W_i(b)$  represents

the evolution under the driving Hamiltonian from time  $\sum_{j=1}^i \ell_j/5m$  to time  $\sum_{j=1}^{i+1} \ell_j/5m$  (where we define  $\ell_{k+1} := m$ ). By assumption, any such evolution can be performed with precision  $O(\epsilon/T)$  using  $g$  gates. Also, recall that  $Q_i(b)$  is simply  $Q$  if  $i \leq |b|$  or  $\mathbb{1}$  otherwise, so it can be applied in time  $O(\log k)$ . Thus the operation (A.13) can be applied in time  $O(k(g + \log k))$ .

At the end of the segment we must effectively apply the final  $P$  and  $R$  gates to the encoded state before reflecting about the encoding of  $|0^m\rangle$ . (That is, we jointly reflect about this state and  $|0\rangle$  for the additional ancilla in Figure 2.) The  $P$  gates are straightforward to apply in the given encoding. Rather than apply the encoded  $R$  gates directly, reflect about the encoding of  $|0^m\rangle$ , and then apply the encoded  $R$  gates for the next segment, it suffices to reflect about the encoding of  $R_\alpha^{\otimes m}|0^m\rangle$  (note that  $R_\alpha^\dagger = R_\alpha$ ). This can be done by applying the inverse of the procedure for preparing (24), reflecting about the initial state, and applying the preparation procedure. Overall, we see that the segment can be applied to the encoded initial state with suitable accuracy using  $O(k(g + \log m))$  gates, plus the cost of preparing the encoded ancillas.

The encoded initial state (24) can be prepared in time  $O(k(\log m + \log \log(1/\epsilon))) = O(k \log m)$ , as described in [13, Sections 4.2–4.4] (see in particular equation (22)). Since  $k = O(\log(T/\epsilon)/\log \log(T/\epsilon))$  (from the proof of Lemma 4.5 with error at most  $\epsilon/5T$ ) and  $m = \text{poly}(T, \hbar, 1/\epsilon)$  (from Theorem 4.1), the overall complexity of making the encoded ancilla state is  $O(\log(T/\epsilon) \log(\hbar T/\epsilon)/\log \log(T/\epsilon))$ . Thus the cost of implementing a constant-query algorithm to precision  $\epsilon/5T$  is

$$O(k(g + \log m)) = O\left(\frac{\log(T/\epsilon)}{\log \log(T/\epsilon)}[g + \log(\hbar T/\epsilon)]\right). \tag{25}$$

Implementing  $O(T)$  segments, each with this complexity, gives the stated time complexity. With error bounded by  $\epsilon/5T$  for each segment, the overall error is at most  $\epsilon$ . □

Using this approach we can similarly prove Theorem 2.1 from Section 1, which we restate for convenience.

**THEOREM 2.1** (Sparse Hamiltonian simulation). *A  $d$ -sparse Hamiltonian  $H$  acting on  $n$  qubits can be simulated for time  $t$  within error  $\epsilon$  with  $O(\tau(\log(\tau/\epsilon)/\log \log(\tau/\epsilon)))$  queries and  $O(\tau(\log^2(\tau/\epsilon)/\log \log(\tau/\epsilon))n)$  additional 2-qubit gates, where  $\tau := d^2\|H\|_{\max}t$  (provided that  $\tau \geq 1$ ).*

*Proof.* The query complexity of this theorem was established in Lemma 5.5, so we only need to determine the time complexity (number of additional 2-qubit

gates) for this proof. The query complexity given in Lemma 5.5 is proved by encoding Hamiltonian simulation as a fractional-query algorithm, then using Theorem 2.3 for the complexity of discrete simulations of continuous-query algorithms. In this approach,  $\tau$  plays the role of  $T$ . We adopt the same approach to obtain the time complexity, except we cannot directly use the result as stated in Theorem 2.3, where the time complexity is for the continuous-query case. Nevertheless, we can obtain a similar result if  $g$  is taken to represent the cost of performing any sequence of consecutive nonquery operations in the fractional-query algorithm. The term  $\log(\hbar T/\epsilon)$  in Theorem 2.3 results from discretizing a continuous-query algorithm with a driving Hamiltonian and does not arise here.

The nonquery operations  $V_j$  for  $j \in [m]$  described in the proof of Theorem 5.1 are straightforward to implement. In the application to Hamiltonian simulation, we simply cycle through all  $\eta$  terms in order, so all the  $V_j$ s can simply add 1 modulo  $\eta$ , and a sequence  $V_{j'} \cdots V_j$  adds  $j' - j \bmod \eta$ . Without loss of generality, we can assume  $\eta$  is a power of 2, so addition modulo  $\eta$  can be performed by standard binary addition, keeping only the  $\log_2 \eta$  least significant bits. Thus any operation to be performed between queries can be applied using  $g = O(\log \eta) = O(\log(d\|H\|_{\max}t/\epsilon))$  operations (where the value of  $\eta$  is discussed following the proof of Lemma 5.4). Next, observe that it suffices to decompose the evolution into  $m = \eta^3 t^2/\epsilon = \text{poly}(t, \|H\|_{\max}, d, 1/\epsilon)$  terms (as stated in Corollary 5.2). In the proof of Theorem 2.3, the time complexity for a constant-query algorithm is  $O(k(g + \log m))$ . This upper bounds the number of additional gates required to perform the nonquery operations. Using  $g = O(\log(d\|H\|_{\max}t/\epsilon))$  and  $\log m = O(\log(d\|H\|_{\max}t/\epsilon))$ , we see that this is  $O(\tau(\log^2(\tau/\epsilon)/\log \log(\tau/\epsilon)))$ .

This only accounts for the operations performed between applications of the unitary  $Q$  defined in Corollary 5.2. It remains to implement  $Q := \sum_{j=1}^{\eta} |j\rangle\langle j| \otimes e^{-iH_j}$  using the oracle, where  $H = \sum_{j=1}^{\eta} H_j$  and  $H_j$  are Hamiltonians with eigenvalues 0 and  $\pi$ . To implement  $Q$  we need to read the first register to learn which 1-sparse Hamiltonian is to be simulated and then simulate the 1-sparse Hamiltonian  $H_j$ .

To achieve this task, given  $j$  we need to be able to determine the position and value of the nonzero entry in any row of  $H_j$ . The method to perform that task is determined by the decomposition technique. The index  $j$  includes a value from 1 to  $d^2$  from the decomposition in Lemma 5.4 into 1-sparse Hamiltonians, and the values  $\xi \in \{X, Y, Z\}$ ,  $\ell \in [\|C^\xi\|_{\max}/2]$ , and  $\pm$  for  $C^{\xi, \ell, \pm}$  in the decomposition in Lemma 5.3. The procedure to simulate queries to the 1-sparse Hamiltonian is detailed in the proof of Lemma 5.4, and involves inputting part of the index of size  $\log d$  into the oracle for the positions of nonzero entries of the Hamiltonian and obtaining results of size  $n$ . The complexity of this part is therefore  $O(\log d + n)$ . The value of  $\ell$  is compared to the value of the corresponding entry in the

Hamiltonian. The complexity of this step is only linear in the number of qubits encoding  $\ell$ , which is  $O(\log \eta)$ . There are only three values of  $\xi$  and two of  $\pm$ , so they do not contribute to the complexity. The procedure in Lemma 5.3 also involves setting the diagonal of  $C^{\xi, \ell, +}$  to be nonzero if  $C^{\xi, \ell} = 0$  for an entire column. This involves setting values in  $n$  qubits for the position of the nonzero element, which has complexity  $O(n)$ . Overall the complexity of this part is  $O(\log \eta + n)$ .

Now we need to implement the 1-sparse Hamiltonian on an  $n$ -qubit register. This can be done with  $O(n)$  gates using the constructions in [1, 16]. For example, consider a basis state  $|v\rangle$  and a Hamiltonian  $C^{X, \ell, \pm}$  in the decomposition (such that the neighbor of  $v$  is not  $v$ ). In the subspace formed by  $v$  and its neighbor, evolution under  $C^{X, \ell, \pm}$  just acts as a Pauli  $X$  operator (or  $-X$ ). Via the decomposition procedure we can calculate the index of  $v$ 's neighbor in another register, then swap the two registers, and uncompute the second register. In the case where the operator is  $-X$ , we can simply apply a phase shift of  $-1$  as well. Thus we can implement  $Q$  using  $O(\log \eta + n)$  gates. Since the number of uses of  $Q$  is the query complexity, the total number of gates used for all invocations of  $Q$  and the nonquery operations is  $O(\tau(\log(\tau/\epsilon)/\log \log(\tau/\epsilon))[\log(\tau/\epsilon) + n])$ , which is  $O(\tau(\log^2(\tau/\epsilon)/\log \log(\tau/\epsilon))n)$ .  $\square$

The same techniques can be straightforwardly applied to simulate time-dependent sparse Hamiltonians. We divide the evolution into intervals of length  $t/r$ , so the Hamiltonian can change by no more than  $h't/r$  over such an interval, where  $h' := \max_{s \in [0, t]} \|(d/ds)H(s)\|$ . Thus the error for each interval is  $O(h't^2/r^2)$ , and the error in the overall simulation is  $O(h't^2/r)$ . Therefore, it suffices to take  $r = \Omega(h't^2/\epsilon)$ . Then  $m = \text{poly}(t, h, h', d, 1/\epsilon)$ , and the complexity is  $O(\tau(\log(\tau/\epsilon) \log((\tau + \tau')/\epsilon)/\log \log(\tau/\epsilon))n)$  as stated.

## 7. Lower bounds

We now show that in general, any sparse Hamiltonian simulation method must use  $\Omega(\log(1/\epsilon)/\log \log(1/\epsilon))$  discrete queries to obtain error at most  $\epsilon$ , so dependence of the query complexity in Theorem 2.1 on  $\epsilon$  is tight up to constant factors. To show this, we use ideas from the proof of the no-fast-forwarding theorem [8, Theorem 3], which says that generic Hamiltonians cannot be simulated in time sublinear in the evolution time. The Hamiltonian used in the proof of that theorem has the property that simulating it for time  $t = \pi n/2$  determines the parity of  $n$  bits exactly. We observe that simulating this Hamiltonian (with sufficiently high precision) for any time  $t > 0$  gives an unbounded-error algorithm for the parity of  $n$  bits, which also requires  $\Omega(n)$  queries [6, 25].

We now prove Theorem 2.2 from Section 1, which we restate for convenience.

**THEOREM 2.2** ( $\epsilon$ -dependent lower bound for Hamiltonian simulation). *For any  $\epsilon > 0$ , there exists a 2-sparse Hamiltonian  $H$  with  $\|H\|_{\max} < 1$  such that simulating  $H$  with precision  $\epsilon$  for constant time requires  $\Omega(\log(1/\epsilon)/\log \log(1/\epsilon))$  queries.*

*Proof.* To construct the Hamiltonian, we begin with a simpler Hamiltonian  $H'$  that acts on vectors  $|i\rangle$  with  $i \in \{0, 1, \dots, N\}$  [21]. The nonzero matrix entries of  $H'$  are  $\langle i|H'|i+1\rangle = \langle i+1|H'|i\rangle = \sqrt{(N-i)(i+1)}/N$  for  $i \in \{0, 1, \dots, N-1\}$ . We have  $\|H'\|_{\max} < 1$ , and simulating  $H'$  for  $t = \pi N/2$  starting with the state  $|0\rangle$  gives the state  $|N\rangle$  (that is,  $e^{-iH'\pi N/2}|0\rangle = |N\rangle$ ). More generally, for  $t \in [0, \pi N/2]$ , we claim that  $|\langle N|e^{-iH't}|0\rangle| = |\sin(t/N)|^N$ .

To see this, consider the Hamiltonian  $\bar{X} := \sum_{j=1}^N X^{(j)}$ , where  $X := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$  and the superscript  $(j)$  indicates that the operator acts nontrivially on the  $j$ th qubit. Since  $e^{-iXt} = \cos(t)\mathbb{1} - i\sin(t)X$ , we have  $|\langle 11 \dots 1|e^{-i\bar{X}t}|00 \dots 0\rangle| = |\sin(t)|^N$ . Defining  $|\text{wt}_k\rangle := \binom{N}{k}^{-1/2} \sum_{|x|=k} |x\rangle$ , we have

$$\bar{X}|\text{wt}_k\rangle = \sqrt{(N-k+1)k}|\text{wt}_{k-1}\rangle + \sqrt{(N-k)(k+1)}|\text{wt}_{k+1}\rangle. \quad (26)$$

This is precisely the behavior of  $NH'$  with  $|k\rangle$  playing the role of  $|\text{wt}_k\rangle$ , so the claim follows.

Now, as in [8], consider a Hamiltonian  $H$  generated from an  $N$ -bit string  $x_1x_2 \dots x_N$ .  $H$  acts on vertices  $|i, j\rangle$  with  $i \in \{0, \dots, N\}$  and  $j \in \{0, 1\}$ . The nonzero matrix entries of this Hamiltonian are

$$\langle i, j|H|i-1, j \oplus x_i\rangle = \langle i-1, j \oplus x_i|H|i, j\rangle = \sqrt{(N-i+1)i}/N \quad (27)$$

for all  $i$  and  $j$ . By construction,  $|0, 0\rangle$  is connected to either  $|i, 0\rangle$  or  $|i, 1\rangle$  (but not both) for any  $i$ ; it is connected to  $|i, j\rangle$  if and only if  $j = x_1 \oplus x_2 \oplus \dots \oplus x_i$ . Thus  $|0, 0\rangle$  is connected to either  $|N, 0\rangle$  or  $|N, 1\rangle$ , and determining which is the case determines the parity of  $x$ . The graph of this Hamiltonian contains two disjoint paths, one containing  $|0, 0\rangle$  and  $|N, \text{PARITY}(x)\rangle$  and the other containing  $|0, 1\rangle$  and  $|N, 1 \oplus \text{PARITY}(x)\rangle$ . Restricted to the connected component of  $|0, 0\rangle$ , this Hamiltonian is the same as  $H'$ . Thus, starting with the state  $|0, 0\rangle$  and simulating  $H$  for time  $t$  gives  $|\langle N, \text{PARITY}(x)|e^{-iHt}|0, 0\rangle| = |\sin(t/N)|^N$ . Furthermore, for any  $t$ , we have  $\langle N, 1 \oplus \text{PARITY}(x)|e^{-iHt}|0, 0\rangle = 0$  since the two states lie in disconnected components.

Simulating this Hamiltonian exactly for any time  $t > 0$  starting with  $|0, 0\rangle$  yields an unbounded-error algorithm for computing the parity of  $x$ , as follows. First we measure  $e^{-iHt}|0, 0\rangle$  in the computational basis. We know that for any  $t > 0$ , the state  $e^{-iHt}|0, 0\rangle$  has some nonzero overlap on  $|N, \text{PARITY}(x)\rangle$  and

zero overlap on  $|N, 1 \oplus \text{PARITY}(x)\rangle$ . If the first register is not  $N$ , we output 0 or 1 with equal probability. If the first register is  $N$ , we output the value of the second register. This is an unbounded-error algorithm for the parity of  $x$ , and thus requires  $\Omega(N)$  queries.

Since the unbounded-error query complexity of parity is  $\Omega(N)$  [6, 25], this shows that exactly simulating  $H$  for any time  $t > 0$  needs  $\Omega(N)$  queries. However, even if we only have an approximate simulation, the previous algorithm still works as long as the error in the output state is smaller than the overlap  $|\langle N, \text{PARITY}(x) | e^{-iHt} | 0, 0 \rangle|$ . If we ensure that the overlap is larger than  $\epsilon$  by a constant factor, then even with error  $\epsilon$ , the overlap on that state will be larger than  $\epsilon$ . On the other hand, the overlap on  $|N, 1 \oplus \text{PARITY}(x)\rangle$  is at most  $\epsilon$ , since the output state is  $\epsilon$  close to the ideal output state which has no overlap.

To achieve an overlap larger than  $\epsilon$  by at least a constant factor, we need  $|\sin(t/N)|^N$  to be larger than  $\epsilon$  by a constant factor. There is some value of  $N$  in  $\Theta(\log(1/\epsilon)/\log \log(1/\epsilon))$  that achieves this.  $\square$

A similar construction shows that any  $\epsilon$ -error simulation of the continuous-query model must use  $\Omega(\log(1/\epsilon)/\log \log(1/\epsilon))$  discrete queries, so Lemma 4.2 is tight up to constant factors. Again we show that a sufficiently high-precision simulation of a certain Hamiltonian could be used to compute parity with unbounded error. However, in the fractional-query model, the form of the Hamiltonian is restricted and it is unclear how to implement the weights that simplify the analysis of the dynamics in Theorem 2.2. Instead, we consider a quantum walk on an infinite unweighted path that also solves parity with unbounded error, and we show that this still holds if the path is long but finite.

**THEOREM 7.1** ( $\epsilon$ -dependent lower bound for continuous-query simulation). *For any  $\epsilon > 0$ , given a query Hamiltonian  $H_x$  for a string of  $N = \Theta((\log(1/\epsilon)/\log \log(1/\epsilon)))$  bits, simulating  $H_x + H_D(t)$  for constant time with precision  $\epsilon$  requires  $\Omega(N)$  queries.*

*Proof.* We prove a lower bound for simulating a Hamiltonian of the form  $H' = \sum_{a=1}^{\eta} c_a U_a^\dagger H_x U_a$  with coefficients  $c_1, \dots, c_\eta \in \mathbb{R}$ . The Hamiltonian  $H_x$  can be used to simulate  $H'$  to any given accuracy with overhead  $\sum_a |c_a|$ , so this implies a lower bound for simulating  $H_x$ . In particular, by taking  $r$  sufficiently large, the evolution under  $H'$  can be approximated arbitrarily closely as

$$e^{-iH't} \approx \left( \prod_{a=1}^{\eta} U_a^\dagger e^{-iH_x c_a t/r} U_a \right)^r. \tag{28}$$

This corresponds to a fractional-query algorithm with cost  $t \sum_{a=1}^{\eta} |c_a|$ . By Theorem 4.1, this fractional-query algorithm can be simulated with arbitrarily small error by a continuous-query algorithm with the same cost. This continuous-query algorithm uses the query Hamiltonian  $H_x$ , and its driving Hamiltonian  $H_D(t)$  implements the unitaries  $\{U_a, U_a^\dagger\}_{a=1}^{\eta}$  at appropriate times.

Viewing the Hamiltonian in terms of the graph of its nonzero entries, the oracle Hamiltonian  $H_x$  provides input-dependent self-loops. First we modify it to give input-dependent edges. Observe that

$$\text{Had} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \text{Had} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad (29)$$

where  $\text{Had} := 1/\sqrt{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$  is the Hadamard gate. Thus we can include a term in the Hamiltonian that has an edge between two vertices associated with the input index  $i$  (and self-loops on those vertices) if  $x_i = 1$ , and is zero otherwise.

Now consider a space with basis states  $|i, j, k\rangle$  where  $i \in \mathbb{Z}$  and  $j, k \in \{0, 1\}$ . The label  $j$  plays the same role as in Theorem 2.2, whereas the new label  $k$  indexes two positions for each value of  $i$ . These new positions are needed because the pairs of vertices associated with each input index must be disjoint.

To specify the Hamiltonian, we define unitaries  $U_1, U_2, U_3, U_4$  so that the nonzero matrix elements of  $U_a^\dagger H_x U_a$  for  $a \in \{1, 2, 3, 4\}$  are

$$\langle i, 0, k | U_1^\dagger H_x U_1 | i, 0, \bar{k} \rangle = \langle i, 0, k | U_1^\dagger H_x U_1 | i, 0, k \rangle = x_i/2 \quad (30)$$

$$\langle i, 1, k | U_2^\dagger H_x U_2 | i, 1, \bar{k} \rangle = \langle i, 1, k | U_2^\dagger H_x U_2 | i, 1, k \rangle = x_i/2 \quad (31)$$

$$\langle i, k, k | U_3^\dagger H_x U_3 | i, \bar{k}, \bar{k} \rangle = \langle i, k, k | U_3^\dagger H_x U_3 | i, k, k \rangle = x_i/2 \quad (32)$$

$$\langle i, k, \bar{k} | U_4^\dagger H_x U_4 | i, \bar{k}, k \rangle = \langle i, \bar{k}, k | U_4^\dagger H_x U_4 | i, \bar{k}, k \rangle = x_i/2 \quad (33)$$

for all  $i \in [N]$  and  $k \in \{0, 1\}$ . Combining these four contributions to obtain a Hamiltonian  $-U_1^\dagger H_x U_1 - U_2^\dagger H_x U_2 + U_3^\dagger H_x U_3 + U_4^\dagger H_x U_4$  and observing that the self-loops cancel, these matrix elements can be summarized in terms of the gadget shown in Figure 3.

We add a driving Hamiltonian to connect these gadgets to form two paths encoding the parity similarly as in Theorem 2.2, and we extend the paths infinitely in both directions. Specifically, the driving Hamiltonian  $H_D$  has nonzero matrix elements

$$\langle i, j, k | H_D | i, j, \bar{k} \rangle = 1/2 \quad (34)$$

for all  $i \in \mathbb{Z}$  and  $j, k \in \{0, 1\}$  (corresponding to the dashed edges in Figure 3, but with positive weight), and

$$\langle i+1, j, 0 | H_D | i, j, 1 \rangle = \langle i, j, 1 | H_D | i+1, j, 0 \rangle = 1/2 \quad (35)$$

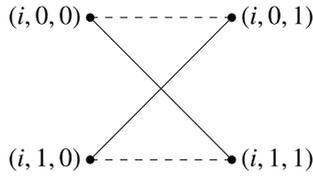


Figure 3. The gadget for querying  $x_i$ . If  $x_i = 0$ , no edges are present. If  $x_i = 1$ , the solid edges have weight  $1/2$  and the dashed edges have weight  $-1/2$ .

for all  $i \in \mathbb{Z}$  and  $j \in \{0, 1\}$  (corresponding to edges that join sectors with adjacent values of  $i$ ). Then the total Hamiltonian

$$H = -U_1^\dagger H_x U_1 - U_2^\dagger H_x U_2 + U_3^\dagger H_x U_3 + U_4^\dagger H_x U_4 + H_D \tag{36}$$

is  $1/2$  times the adjacency matrix of the disjoint union of two infinite paths, one with vertices

$$\begin{aligned} &\dots, (0, 0, 0), (0, 0, 1), (1, 0, 0), (1, x_1, 1), (2, x_1, 0), (2, x_1 \oplus x_2, 1), \dots, \\ &(N, x_1 \oplus \dots \oplus x_N, 1), (N + 1, x_1 \oplus \dots \oplus x_N, 0), \\ &(N + 1, x_1 \oplus \dots \oplus x_N, 1), \dots \end{aligned} \tag{37}$$

and the other with vertices

$$\begin{aligned} &\dots, (0, 1, 0), (0, 1, 1), (1, 1, 0), (1, 1 \oplus x_1, 1), (2, 1 \oplus x_1, 0), \\ &(2, 1 \oplus x_1 \oplus x_2, 1), \dots, (N, 1 \oplus x_1 \oplus \dots \oplus x_N, 1), \\ &(N + 1, 1 \oplus x_1 \oplus \dots \oplus x_N, 0), (N + 1, 1 \oplus x_1 \oplus \dots \oplus x_N, 1), \dots \end{aligned} \tag{38}$$

Analogous to the Hamiltonian  $H$  in the proof of Theorem 2.2,  $(0, 0, 1)$  is in the same component as  $(N, b, 1)$  if and only if  $b = \text{PARITY}(x)$ .

To compute the probability of reaching  $(n, \text{PARITY}(x), 1)$  starting from  $(0, 0, 1)$  after evolving with the Hamiltonian (36) for time  $t$ , we can use the expression for the propagator on an infinite path in terms of a Bessel function (see for example [16]). Specifically, we have

$$|\langle N, \text{PARITY}(x), 1 | e^{-iHt} | 0, 0, 1 \rangle| = |J_{2N}(t)|. \tag{39}$$

For large  $N$  and for any fixed  $t \neq 0$ , we have  $|J_N(t)| = e^{-\Theta(N \log N)}$  [42, Section 8.1]. Thus, as in the proof of Theorem 2.2, even a simulation with error  $\epsilon$  gives the result with nonzero probability provided  $N = \Theta(\log(1/\epsilon)/\log \log(1/\epsilon))$ .

The preceding argument uses a Hamiltonian acting on an infinite-dimensional space. However, we can truncate it to act on a finite space with essentially the same effect. Specifically, we apply the Truncation Lemma of [18], which we repeat here.

**Lemma 7.2** (Truncation Lemma [18]). *Let  $H$  be a Hamiltonian acting on a Hilbert space  $\mathcal{H}$  and let  $|\Phi\rangle \in \mathcal{H}$  be a normalized state. Let  $\mathcal{K}$  be a subspace of  $\mathcal{H}$ , let  $P$  be the projector onto  $\mathcal{K}$ , and let  $\tilde{H} = PHP$  be the Hamiltonian within this subspace. Suppose that, for some  $T > 0$ ,  $W \in \{H, \tilde{H}\}$ ,  $N_0 \in \mathbb{N}$ , and  $\delta > 0$ , we have, for all  $0 \leq t \leq T$ ,*

$$e^{-iWt}|\Phi\rangle = |\gamma(t)\rangle + |\epsilon(t)\rangle \quad \text{with } \|\epsilon(t)\| \leq \delta$$

and

$$(\mathbb{1} - P)H^r|\gamma(t)\rangle = 0 \quad \text{for all } r \in \{0, 1, \dots, N_0 - 1\}.$$

Then, for all  $0 \leq t \leq T$ ,

$$\|(e^{-iHt} - e^{-i\tilde{H}t})|\Phi\rangle\| \leq \left(\frac{4e\|H\|t}{N_0} + 2\right)(\delta + 2^{-N_0}(1 + \delta)).$$

We use this lemma with  $\mathcal{K} = \text{span}\{|i, j, k\rangle : -N^3 - N^2 \leq i \leq N^3 + N^2, j, k \in \{0, 1\}\}$  and  $W = H$ . Let  $P$  project onto  $\mathcal{K}$  and let  $P'$  project onto  $\text{span}\{|i, j, k\rangle : -N^2 \leq i \leq N^2, j, k \in \{0, 1\}\}$ . Finally, let  $|\gamma(t)\rangle = P'e^{-iHt}|0, 0, 1\rangle$ . Then  $\delta^2 := \|e^{-iHt}|0, 0, 1\rangle - |\gamma(t)\rangle\|^2 = |J_{2N^2+1}(t)|^2 + 2\sum_{j=2}^{\infty} |J_{2N^2+j}(t)|^2 \leq e^{-\Omega(N^2 \log N)}$ . Furthermore,  $(\mathbb{1} - P)H^r|\gamma(t)\rangle = 0$  for all  $r \in \{0, 1, \dots, N^3\}$ . Also observe that  $\|H\| = 1$ . Thus the Truncation Lemma shows that

$$\|(e^{-iHt} - e^{-iPHPt})|0, 0, 1\rangle\| \leq \left(\frac{4et}{N^3} + 2\right)(\delta + 2^{-N^3}(1 + \delta)) \leq e^{-\Omega(N^2 \log N)}, \quad (40)$$

so the error incurred by truncating  $H$  to the Hamiltonian  $PHP$  acting on the finite-dimensional space  $\mathcal{K}$  is asymptotically negligible compared to  $\epsilon$ .  $\square$

## 8. Open questions

While our algorithm for continuous-query simulation is optimal as a function of  $\epsilon$  alone, it is suboptimal as a function of  $T$ , and it is unclear what tradeoffs might exist between these two parameters. The best known lower bound as a function of both  $\epsilon$  and  $T$  is  $\Omega(T + \log(1/\epsilon)/\log \log(1/\epsilon))$ . It would be surprising if this bound were achievable, but it remains open to find such an algorithm or to prove

a better lower bound. In general, any improvement to the tradeoff between  $\epsilon$  and  $T$  could be of interest.

In the context of time-independent sparse Hamiltonian simulation, the quantum walk-based simulation of [9, 15] achieves linear dependence on  $d$ , but polynomial dependence on  $\epsilon$ . In contrast, our algorithm here obtains polylogarithmic dependence on  $\epsilon$ , but scales as  $d^2$ . In more recent work, it has been shown that quantum walks can yield scaling close to linear in  $d$ , as well as polylogarithmic in  $\epsilon$ , but only for the time-independent case [12]. It is still an open question how to obtain such scaling for time-dependent Hamiltonians.

### Acknowledgments

We thank Sevag Gharibian and Nathan Wiebe for valuable discussions. This work was supported in part by ARC grants FT100100761 and DP160102426, Canada's NSERC, CIFAR, the Ontario Ministry of Research and Innovation, and the US ARO. RDS acknowledges support from the Laboratory Directed Research and Development Program at Los Alamos National Laboratory and from the AFOSR through Grant No. FA9550-12-1-0057.

### Appendix A. Proofs of known results

In this appendix, for the sake of completeness we provide proofs of claims that are known or essentially follow from known results.

#### A.1. Equivalence of continuous- and fractional-query models.

**THEOREM 4.1** (Equivalence of continuous- and fractional-query models). *For any  $\epsilon > 0$ , any algorithm with continuous-query complexity  $T$  can be implemented with fractional-query complexity  $T$  with error at most  $\epsilon$  and  $m = O(\bar{h}T^2/\epsilon)$  fractional-query gates, where  $\bar{h} := (1/T) \int_0^T \|H_D(t)\| dt$  is the average norm of the driving Hamiltonian. Conversely, any algorithm with fractional-query complexity  $T$  can be implemented with continuous-query complexity  $T$  with error at most  $\epsilon$ .*

*Proof.* A simulation of the continuous-query model by the fractional-query model with the stated properties appears in [23, Section II.A]. We present their proof for completeness.

We wish to implement the unitary  $U(T)$  satisfying the Schrödinger equation (2) with  $U(0) = \mathbb{1}$ . To refer to the solutions of this equation for arbitrary

Hamiltonians and time intervals, we define  $U_H(t_2, t_1)$  to be the solution of the Schrödinger equation with Hamiltonian  $H$  from time  $t_1$  to time  $t_2$  where  $U(t_1) = \mathbb{1}$ . In this notation,  $U(T) = U_{H_x+H_D}(T, 0)$ .

Let  $m$  be an integer and  $\theta = T/m$ . We have

$$U_{H_x+H_D}(T, 0) = U_{H_x+H_D}(m\theta, (m-1)\theta) \cdots U_{H_x+H_D}(2\theta, \theta)U_{H_x+H_D}(\theta, 0). \quad (\text{A.1})$$

If we can approximate each of these  $m$  terms, we can use the subadditivity of error in implementing unitaries (that is,  $\|UV - \tilde{U}\tilde{V}\| \leq \|U - \tilde{U}\| + \|V - \tilde{V}\|$  for unitaries  $U, \tilde{U}, V, \tilde{V}$ ) to obtain an approximation of  $U(T)$ .

Reference [29] shows that for small  $\theta$ , the evolution according to Hamiltonians  $A$  and  $B$  over an interval of length  $\theta$  approximates the evolution according to  $A + B$  over the same interval. Specifically, from [29, equation (A8b)] we have

$$\begin{aligned} & \|U_{A+B}((j+1)\theta, j\theta) - U_A((j+1)\theta, j\theta)U_B((j+1)\theta, j\theta)\| \\ & \leq \int_{j\theta}^{(j+1)\theta} dv \int_{j\theta}^v du \| [A(u), B(v)] \|. \end{aligned} \quad (\text{A.2})$$

In our application,  $A(t) = H_D(t)$  and  $B = H_x$ . Since  $\|H_x\| = 1$ , the right-hand side is at most

$$\begin{aligned} 2 \int_{j\theta}^{(j+1)\theta} dv \int_{j\theta}^v du \|H_D(u)\| & \leq 2 \int_{j\theta}^{(j+1)\theta} dv \int_{j\theta}^{(j+1)\theta} du \|H_D(u)\| \\ & = 2\theta \int_{j\theta}^{(j+1)\theta} \|H_D(u)\| du. \end{aligned} \quad (\text{A.3})$$

By subadditivity, the error in implementing  $U(T)$  is at most

$$2\theta \sum_{j=0}^{m-1} \int_{j\theta}^{(j+1)\theta} \|H_D(u)\| du = 2\theta \int_0^T \|H_D(u)\| du = 2\theta \bar{h}T = \frac{2\bar{h}T^2}{m}. \quad (\text{A.4})$$

This error is smaller than  $\epsilon$  when  $m \geq 2\bar{h}T^2/\epsilon$ , which proves this direction of the equivalence.

For the other direction, consider a fractional-query algorithm

$$U_{\text{fq}} := U_m Q^{\alpha_m} U_{m-1} \cdots Q^{\alpha_2} U_1 Q^{\alpha_1} U_0 \quad (\text{A.5})$$

(recall that  $Q$  depends on  $x$ ), where  $\alpha_i \in (0, 1]$  for all  $i \in [m]$ , with complexity  $T = \sum_{i=1}^m \alpha_i$ . Let  $A_i := \sum_{j=1}^i \alpha_j$  for all  $i \in [m]$  and let  $U_j := e^{-iH_D^{(j)}}$  for all  $j \in \{0, 1, \dots, m\}$ . Consider the piecewise constant Hamiltonian

$$H(t) = H_x + \frac{1}{\epsilon_1} \left( \delta_{t \in [0, \epsilon_1]} H_D^{(0)} + \sum_{i=1}^m \delta_{t \in [A_i - \epsilon_1, A_i]} H_D^{(i)} \right), \quad (\text{A.6})$$

where  $\delta_B$  is 0 if  $B$  is false and 1 if  $B$  is true. Provided  $\epsilon_1 \leq \min\{\alpha_1/2, \alpha_2, \dots, \alpha_m\}$ , evolving with  $H(t)$  from  $t = 0$  to  $T$  implements a unitary close to our fractional-query algorithm. More precisely, it implements

$$U(T) = e^{-i(H_D^{(m)} + \epsilon_1 H_x)} e^{-i(\alpha_m - \epsilon_1) H_x} e^{-i(H_D^{(m-1)} + \epsilon_1 H_x)} \dots e^{-i(\alpha_2 - \epsilon_1) H_x} e^{-i(H_D^{(1)} + \epsilon_1 H_x)} e^{-i(\alpha_1 - 2\epsilon_1) H_x} e^{-i(H_D^0 + \epsilon_1 H_x)}, \tag{A.7}$$

which satisfies  $\|U(T) - U_{\text{fg}}\| = O(m\epsilon_1)$ . This follows from the fact that each exponential in (A.7) approximates the corresponding unitary of (A.5) within error  $\epsilon_1$  (for example,  $\|e^{-i(H_D^{(m)} + \epsilon_1 H_x)} - U_m\| = O(\epsilon_1)$  and  $\|e^{-i(\alpha_m - \epsilon_1) H_x} - Q^{\alpha_m}\| = O(\epsilon_1)$ ) and the subadditivity of error when implementing unitaries. The fact that each exponential has error  $O(\epsilon_1)$  follows from the inequality  $\|e^{iA} - e^{iB}\| \leq \|A - B\|$ . This can be proved by observing that  $\|e^{iA} - e^{iB}\| = \|(e^{iA/n})^n - (e^{iB/n})^n\| \leq n\|e^{iA/n} - e^{iB/n}\| \leq \|A - B\| + O(1/n)$ , where the first inequality uses subadditivity of error and the second inequality follows by Taylor expansion. Since the statement is true for all  $n$ , the claim follows.

This simulation has continuous-query complexity  $T$ . Its error can be made less than  $\epsilon$  by choosing  $\epsilon_1$  sufficiently small (in particular, it suffices to take some  $\epsilon_1 = \Theta(\epsilon/m)$ ). □

**A.2. The Approximate Segment Lemma.** In this section, we establish the Approximate Segment Lemma (Lemma 4.5). This lemma essentially follows from [23] with minor modification. We start by proving the following Gadget Lemma, which follows from [23, Section II.B].

**Lemma 4.3** (Gadget Lemma [23]). *Let  $Q$  be a unitary matrix with eigenvalues  $\pm 1$ ; let  $\alpha \in [0, 1]$ . The circuit in Figure 1, with  $R_\alpha := (1/\sqrt{c+s}) \begin{pmatrix} \sqrt{c} & \sqrt{s} \\ \sqrt{s} & -\sqrt{c} \end{pmatrix}$  and  $P := \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ , performs the map*

$$|0\rangle|\psi\rangle \mapsto \sqrt{q_\alpha}|0\rangle e^{i\pi\alpha/2} Q^\alpha |\psi\rangle + \sqrt{1-q_\alpha}|1\rangle|\phi\rangle \tag{3}$$

for some state  $|\phi\rangle$ , where  $c := \cos(\pi\alpha/2)$ ,  $s := \sin(\pi\alpha/2)$ ,  $q_\alpha := 1/(c+s)^2 = 1/(1+\sin(\pi\alpha))$ , and  $Q^\alpha = \frac{1}{2}(\mathbb{1} + Q) + e^{-i\pi\alpha/2} \frac{1}{2}(\mathbb{1} - Q) = e^{-i\pi\alpha/2}(c\mathbb{1} + isQ)$ .

*Proof.* The input state evolves as follows:

$$\begin{aligned} |0\rangle|\psi\rangle &\mapsto \frac{\sqrt{c}|0\rangle + \sqrt{s}|1\rangle}{\sqrt{c+s}} |\psi\rangle \\ &\mapsto \frac{1}{\sqrt{c+s}} (\sqrt{c}|0\rangle|\psi\rangle + \sqrt{s}|1\rangle Q|\psi\rangle) \end{aligned}$$

$$\begin{aligned}
 & \mapsto \frac{1}{c+s} [|0\rangle(c|\psi\rangle + isQ|\psi\rangle) + \sqrt{cs}|1\rangle(|\psi\rangle - iQ|\psi\rangle)] \\
 & = \sqrt{q_\alpha}(|0\rangle e^{i\pi\alpha/2} Q^\alpha |\psi\rangle + \sqrt{\sin(\pi\alpha)}|1\rangle e^{-i\pi/4} Q^{-1/2} |\psi\rangle) \\
 & = \sqrt{q_\alpha}|0\rangle e^{i\pi\alpha/2} Q^\alpha |\psi\rangle + \sqrt{1-q_\alpha}|1\rangle e^{-i\pi/4} Q^{-1/2} |\psi\rangle. \tag{A.8}
 \end{aligned}$$

Thus the output has the stated form.  $\square$

We can now collect these gadgets into a segment, which implements a fractional-query algorithm with constant query complexity with amplitude  $1/2$ .

**Lemma 4.4** (Segment Lemma). *Let  $V$  be a unitary implementable by a fractional-query algorithm with query complexity at most  $1/5$ , that is, there exists an  $m$  such that  $V = U_m Q^{\alpha_m} U_{m-1} \cdots U_1 Q^{\alpha_1} U_0$  with  $\alpha_i \geq 0$  for all  $i$  and  $\sum_{i=1}^m \alpha_i \leq 1/5$ . Let  $P$  and  $R_\alpha$  be as in Lemma 4.3. Then there exists a unitary  $\Upsilon$  on the additional ancilla such that the circuit in Figure 2 performs the map*

$$|0^{m+1}\rangle |\psi\rangle \mapsto \frac{1}{2} |0^{m+1}\rangle e^{i\vartheta} V |\psi\rangle + \frac{\sqrt{3}}{2} |\Phi^\perp\rangle \tag{4}$$

for some state  $|\Phi^\perp\rangle$  satisfying  $(|0^{m+1}\rangle \langle 0^{m+1}| \otimes \mathbb{1}) |\Phi^\perp\rangle = 0$  and some  $\vartheta \in [0, 2\pi)$ .

*Proof.* We first analyze the subcircuit in the dashed box in Figure 2, which is the entire circuit without the first qubit. The first qubit does not interact with the rest of the qubits and is only used at the end of the proof.

This subcircuit is built by composing several fractional-query gadgets (as in Figure 1) with a new control qubit for each gadget but with a common target. The  $m$  gadgets correspond to making the fractional queries  $Q^{\alpha_i}$ . The first register of a gadget indicates whether it has applied the fractional query successfully, in which case the register is  $|0\rangle$ , or not, in which case it is  $|1\rangle$ . For the  $i$ th gadget, the output state has amplitude  $q_{\alpha_i}$  on the state  $|0\rangle$  corresponding to the successful outcome, as shown in Lemma 4.3.

The state of the control qubits on the output is  $|0^m\rangle$  only when all the gadgets have successfully applied the fractional query. In this case, the target has been successfully transformed to  $V|\psi\rangle$ . Thus the dashed subcircuit in Figure 2 performs the map

$$|0^m\rangle |\psi\rangle \mapsto \sqrt{p} |0^m\rangle e^{i\vartheta} V |\psi\rangle + \sqrt{1-p} |\Phi^\perp\rangle \tag{A.9}$$

for some  $|\Phi^\perp\rangle$  satisfying  $(|0^m\rangle \langle 0^m| \otimes \mathbb{1}) |\Phi^\perp\rangle = 0$ , where  $p = \prod_{i=1}^m q_{\alpha_i}$  and  $\vartheta = -\sum_{i=1}^m \pi \alpha_i / 2 \pmod{2\pi}$ .

This is similar to the desired statement, except that we want the amplitude in front of  $|0^m\rangle$  to be  $1/2$  instead of  $\sqrt{p}$ . We show that  $p > 1/4$  and then use the first qubit to decrease its value to exactly  $1/4$ .

Since  $\sum_{i=1}^m \alpha_i \leq 1/5$  by assumption, we can lower bound the value of  $p$  as follows. Since  $\alpha_i \geq 0$  for all  $i$ , using the inequalities  $\sin x \leq x$  (for  $x \geq 0$ ) and  $1/(1+x) \geq 1-x$  (for  $x \geq -1$ ) gives

$$\begin{aligned}
 p &= \prod_{i=1}^m q_{\alpha_i} = \prod_{i=1}^m \frac{1}{1 + \sin(\pi\alpha_i)} \geq \prod_{i=1}^m \frac{1}{1 + \pi\alpha_i} \geq \prod_{i=1}^m (1 - \pi\alpha_i) \geq 1 - \pi \sum_{i=1}^m \alpha_i \\
 &\geq 1 - \frac{\pi}{5} > \frac{1}{4},
 \end{aligned}
 \tag{A.10}$$

where the third inequality uses the fact that for  $x_i \in [0, 1]$ ,  $\prod_i (1 - x_i) \geq 1 - \sum_i x_i$ .

Thus we have  $\sqrt{p} > 1/2$ . Now let  $\mathcal{Y}$  be any unitary that maps  $|0\rangle$  to  $(1/2\sqrt{p})|0\rangle + (1-1/4p)^{1/2}|1\rangle$ . Since  $\sqrt{p} > 1/2$ , we have  $1/2\sqrt{p} < 1$ , so a unitary  $\mathcal{Y}$  exists. Then for the full circuit in Figure 2, the amplitude corresponding to the state  $|0^m\rangle$  is  $\sqrt{p} \cdot 1/2\sqrt{p} = 1/2$ .  $\square$

Finally, we show that the map in the previous lemma can be performed to error  $\epsilon$  using only  $O(\log(1/\epsilon)/\log \log(1/\epsilon))$  queries.

**Lemma 4.5** (Approximate Segment Lemma). *Let  $V$  be a unitary implementable by a fractional-query algorithm with query complexity at most  $1/5$ . Then for any  $\epsilon > 0$ , there exists a unitary quantum circuit that makes  $O(\log(1/\epsilon)/\log \log(1/\epsilon))$  discrete queries and, within error  $\epsilon$ , performs a unitary  $U$  acting as*

$$U|0^{m+1}\rangle|\psi\rangle = \frac{1}{2}|0^{m+1}\rangle e^{i\vartheta} V|\psi\rangle + \frac{\sqrt{3}}{2}|\Phi^\perp\rangle \tag{5}$$

for some state  $|\Phi^\perp\rangle$  satisfying  $\langle 0^{m+1} | \langle 0^{m+1} \otimes \mathbb{1} | \Phi^\perp \rangle = 0$  and some  $\vartheta \in [0, 2\pi)$ .

*Proof.* From Lemma 4.4 we know that the circuit in Figure 2 performs the claimed map with no error. However, the circuit makes  $m$  discrete queries, which can be arbitrarily large. We wish to construct a circuit with error at most  $\epsilon$  that makes only  $O(\log(1/\epsilon)/\log \log(1/\epsilon))$  queries, independent of  $m$ .

We first analyze the subcircuit in the dotted box in Figure 2. The output of this subcircuit is  $|\zeta\rangle = \bigotimes_{i=1}^m R_{\alpha_i}|0\rangle = \bigotimes_{i=1}^m (1/\sqrt{c_i + s_i})(\sqrt{c_i}|0\rangle + \sqrt{s_i}|1\rangle)$ , where  $c_i := \cos(\pi\alpha_i/2)$  and  $s_i := \sin(\pi\alpha_i/2)$ . We also define  $q_i := q_{\alpha_i} = 1/(c_i + s_i)^2 = 1/(1 + \sin(\pi\alpha_i))$ . We can write  $|\zeta\rangle = \sum_{x \in \{0,1\}^m} w_x |x\rangle$  with  $\sum_x |w_x|^2 = 1$ .

Now consider the subnormalized state  $|\zeta_k\rangle := \sum_{|x| \leq k} w_x |x\rangle$ , where  $|x|$  denotes the Hamming weight of  $x$  and  $k \leq m$  is a positive integer. In the circuit, we approximate the state  $|\zeta\rangle$  with some  $|\zeta_k\rangle$ . Clearly  $|\zeta_m\rangle = |\zeta\rangle$ , and the approximation becomes worse as  $k$  decreases. To achieve a  $1 - \epsilon^2/2$  approximation, we claim it suffices to take  $k = \Omega(\log(1/\epsilon)/\log \log(1/\epsilon))$ . Since  $1 - \langle \zeta | \zeta_k \rangle = \sum_{|x| > k} |w_x|^2$ , we must upper bound  $\sum_{|x| > k} |w_x|^2$  in terms of  $k$ .

Consider  $m$  independent random variables  $X_i$  with  $\Pr(X_i = 0) = c_i/(c_i + s_i)$  and  $\Pr(X_i = 1) = s_i/(c_i + s_i)$ . The probability that  $\sum_i X_i > k$  is  $\sum_{|x|>k} |w_x|^2$ , since  $|w_x|^2$  is the probability of the event  $X_i = x_i$  for all  $i$ . For such events, the Chernoff bound (see for example [35, Theorem 4.1]) says that for any  $\delta > 0$ ,

$$\Pr\left(\sum_i X_i > (1 + \delta)\mu\right) < \frac{e^{\delta\mu}}{(1 + \delta)^{(1+\delta)\mu}}, \quad (\text{A.11})$$

where  $\mu := \sum_i \Pr(X_i = 1) = \sum_i (s_i/(c_i + s_i))$ . Since  $\alpha_i \geq 0$  and  $\sum_i \alpha_i \leq 1/5$ , we have  $\mu \geq 0$  and  $\mu = \sum_i (s_i/(c_i + s_i)) \leq \sum_i s_i = \sum_i \sin(\pi\alpha_i/2) \leq \sum_i \pi\alpha_i/2 \leq \pi/10 \leq 1$ , where we used the facts that  $\sin x \leq x$  for all  $x > 0$  and  $\sin \theta + \cos \theta \geq 1$  for all  $\theta \in [0, \pi/2]$ .

Setting  $k = (1 + \delta)\mu$ , we get  $\sum_{|x|>k} |w_x|^2 = \Pr(\sum_i X_i > k) < e^{k-\mu}/(1 + \delta)^k = e^{k-\mu} \mu^k/k^k < e^k/k^k$ . This is less than  $\epsilon^2/2$  for an appropriately chosen  $k = \Omega(\log(1/\epsilon)/\log \log(1/\epsilon))$ . For such a value of  $k$ , the state  $|\zeta_k\rangle$  has inner product at least  $1 - \epsilon^2/2$  with  $|\zeta\rangle$ . Let  $|\tilde{\zeta}\rangle$  denote the normalized  $|\zeta_k\rangle$  for some choice of  $k = \Omega(\log(1/\epsilon)/\log \log(1/\epsilon))$ . The state  $|\tilde{\zeta}\rangle$  also has inner product at least  $1 - \epsilon^2/2$  with  $|\zeta\rangle$ . We replace the dotted box in Figure 2 with  $|\tilde{\zeta}\rangle$ , a fixed state that requires no queries to create.

With this modification, the control qubits are in a superposition over states with Hamming weight at most  $k$ , suggesting that this circuit can be performed with at most  $k$  queries. We now show that this is possible.

The control qubits are in a superposition over states  $|b\rangle$  where  $b \in \{0, 1\}^m$ . The value of  $b_i$  decides whether the  $i$ th query occurs or not. The string  $b$  therefore completely determines the product of unitary matrices that is applied to  $|\psi\rangle$  when the control qubits are in the state  $|b\rangle$ . This product contains at most  $k$  query gates, and thus may be written as

$$W_{|b|}(b) Q W_{|b|-1}(b) \cdots Q W_1(b) Q W_0(b). \quad (\text{A.12})$$

Note that the  $W_i$  operators are functions of  $b$ . We may also write this unitary as

$$W_k(b) Q_k(b) W_{k-1}(b) \cdots Q_2(b) W_1(b) Q_1(b) W_0(b), \quad (\text{A.13})$$

where for  $i \leq |b|$  the  $W_i$  operators are as before and for  $i > |b|$ , we have  $W_i = \mathbb{1}$ . Here  $Q_i(b)$  is defined to be  $Q$  when  $i \leq |b|$  and  $\mathbb{1}$  when  $i > |b|$ . We can now construct a circuit that performs the unitary in (A.13) controlled on the value of  $b$ . This circuit has at most  $k$  query gates and performs the same unitary as the circuit in Figure 2 with  $|\zeta\rangle$  replaced with  $|\tilde{\zeta}\rangle$ .

Finally, we show that the actual operation performed, denoted  $\tilde{U}$ , is within error  $\epsilon$  of the ideal unitary  $U$ . The only difference between these operations is that  $\tilde{U}$

prepares  $|\tilde{\zeta}\rangle$  rather than  $|\zeta\rangle$  in the initial step. Therefore, the error between  $\tilde{U}$  and  $U$  is at most the error between an operation that prepares  $|\tilde{\zeta}\rangle$  and an operation that prepares  $|\zeta\rangle$ . If we required  $U$  to prepare  $|\zeta\rangle$  using  $\bigotimes_{i=1}^m R_{\alpha_i}$ , it would be difficult to design a nearby unitary that prepares  $|\tilde{\zeta}\rangle$ . However, the lemma does not specify the action of  $U$  on states not of the form  $|0^{m+1}\rangle|\psi\rangle$ , so we can make any convenient choice of the operation preparing  $|\zeta\rangle$  that is close to the operation preparing  $|\tilde{\zeta}\rangle$ .

Let  $R := \bigotimes_{i=1}^m R_{\alpha_i}$  and denote the unitary that prepares  $|\tilde{\zeta}\rangle$  by  $\tilde{R}$ . In the computational basis,  $R$  has first column  $\zeta$  and  $\tilde{R}$  has first column  $\tilde{\zeta}$ . We claim there is a unitary  $R'$  that is within  $\epsilon$  of  $\tilde{R}$  but that has the same first column as  $R$ .

To see this, let  $\theta$  satisfy  $\langle \tilde{\zeta} | \zeta \rangle = \cos \theta$ . Consider the 2-dimensional subspace spanned by  $|\zeta\rangle$  and  $|\tilde{\zeta}\rangle$ , and let  $E$  be the unitary that rotates by angle  $\theta$  in this subspace, but acts as the identity outside the subspace. In particular,  $E|\tilde{\zeta}\rangle = |\zeta\rangle$ . Taking  $R' := E\tilde{R}$ , we see that  $R'$  has the first column  $\zeta$  as required. The error is  $\|R' - \tilde{R}\| = \|E\tilde{R} - \tilde{R}\| = \|E - \mathbb{1}\| = \sqrt{2 - 2\cos\theta}$ .

Since  $\langle \tilde{\zeta} | \zeta \rangle \geq 1 - \epsilon^2/2$ , we find  $\|R' - \tilde{R}\| \leq \epsilon$ . Because the remainder of the circuit is identical, the overall error between  $\tilde{U}$  and  $U$  is at most  $\epsilon$  as claimed.  $\square$

## References

- [1] D. Aharonov and A. Ta-Shma, ‘Adiabatic quantum state generation and statistical zero knowledge’, in *Proceedings of the 35th ACM Symposium on Theory of Computing* (ACM, New York, 2003), 20–29.
- [2] A. Ambainis, A. M. Childs, B. W. Reichardt, R. Špalek and S. Zhang, ‘Any AND-OR formula of size  $N$  can be evaluated in time  $N^{1/2+o(1)}$  on a quantum computer’, *SIAM J. Comput.* **39**(6) (2010), 2513–2530.
- [3] 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* (IEEE, Piscataway, NJ, 2011), 167–177.
- [4] R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love and A. Aspuru-Guzik, ‘Exponentially more precise quantum simulation of fermions II: Quantum chemistry in the CI matrix representation’, Preprint, 2015, [arXiv:1506.01029](https://arxiv.org/abs/1506.01029).
- [5] R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love and A. Aspuru-Guzik, ‘Exponentially more precise quantum simulation of fermions in second quantization’, *New J. Phys.* **18** (2016), 033032.
- [6] R. Beals, H. Buhrman, R. Cleve, M. Mosca and R. de Wolf, ‘Quantum lower bounds by polynomials’, *J. ACM* **48**(4) (2001), 778–797.
- [7] A. Belovs, ‘Variations on quantum adversary’, Preprint, 2015, [arXiv:1504.06943](https://arxiv.org/abs/1504.06943).
- [8] D. W. Berry, G. Ahokas, R. Cleve and B. C. Sanders, ‘Efficient quantum algorithms for simulating sparse Hamiltonians’, *Commun. Math. Phys.* **270**(2) (2007), 359–371.
- [9] D. W. Berry and A. M. Childs, ‘Black-box Hamiltonian simulation and unitary implementation’, *Quantum Inf. Comput.* **12**(1–2) (2012), 29–62.

- [10] 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 ACM Symposium on Theory of Computing* (ACM, New York, 2014), 283–292.
- [11] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari and R. D. Somma, 'Simulating Hamiltonian dynamics with a truncated Taylor series', *Phys. Rev. Lett.* **114**(9) (2015), 090502.
- [12] D. W. Berry, A. M. Childs and R. Kothari, 'Hamiltonian simulation with nearly optimal dependence on all parameters', in *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science* (IEEE, Piscataway, NJ, 2015), 792–809.
- [13] D. W. Berry, R. Cleve and S. Gharibian, 'Gate-efficient discrete simulations of continuous-time quantum query algorithms', *Quantum Inf. Comput.* **14**(1–2) (2014), 1–30.
- [14] A. M. Childs, 'Quantum information processing in continuous time', PhD Thesis, Massachusetts Institute of Technology, 2004.
- [15] A. M. Childs, 'On the relationship between continuous- and discrete-time quantum walk', *Comm. Math. Phys.* **294**(2) (2010), 581–603.
- [16] 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* (ACM, New York, 2003), 59–68.
- [17] A. M. Childs, R. Cleve, S. P. Jordan and D. Yonge-Mallo, 'Discrete-query quantum algorithm for NAND trees', *Theory Comput.* **5**(5) (2009), 119–123.
- [18] A. M. Childs, D. Gosset and Z. Webb, 'Universal computation by multi-particle quantum walk', *Science* **339**(6121) (2013), 791–794.
- [19] A. M. Childs and R. Kothari, 'Simulating sparse Hamiltonians with star decompositions', in *Theory of Quantum Computation, Communication, and Cryptography (TQC 2010)*, Lecture Notes in Computer Science, 6519 (Springer, Berlin, Heidelberg, 2011), 94–103.
- [20] A. M. Childs and N. Wiebe, 'Hamiltonian simulation using linear combinations of unitary operations', *Quantum Inf. Comput.* **12**(11–12) (2012), 901–924.
- [21] M. Christandl, N. Datta, A. Ekert and A. J. Landahl, 'Perfect state transfer in quantum spin networks', *Phys. Rev. Lett.* **92**(18) (2004), 187902.
- [22] B. D. Clader, B. C. Jacobs and C. R. Sprouse, 'Preconditioned quantum linear system algorithm', *Phys. Rev. Lett.* **110**(25) (2013), 250504.
- [23] R. Cleve, D. Gottesman, M. Mosca, R. D. Somma and D. Yonge-Mallo, 'Efficient discrete-time simulations of continuous-time quantum query algorithms', in *Proceedings of the 41st ACM Symposium on Theory of Computing* (ACM, New York, 2009), 409–416.
- [24] E. Farhi, J. Goldstone and S. Gutmann, 'A quantum algorithm for the Hamiltonian NAND tree', *Theory Comput.* **4**(8) (2008), 169–190.
- [25] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser, 'Limit on the speed of quantum computation in determining parity', *Phys. Rev. Lett.* **81**(24) (1998), 5442–5444.
- [26] E. Farhi and S. Gutmann, 'Analog analogue of a digital quantum computation', *Phys. Rev. A* **57**(4) (1998), 2403–2406.
- [27] R. P. Feynman, 'Simulating physics with computers', *Int. J. Theor. Phys.* **21**(6) (1982), 467–488.
- [28] A. W. Harrow, A. Hassidim and S. Lloyd, 'Quantum algorithm for linear systems of equations', *Phys. Rev. Lett.* **103**(15) (2009), 150502.
- [29] J. Huyghebaert and H. D. Raedt, 'Product formula methods for time-dependent Schrödinger problems', *J. Phys. A* **23**(24) (1990), 5777.
- [30] C. Jordan, 'Essai sur la géométrie à  $n$  dimensions', *Bull. Soc. Math. France* **3** (1875), 103–174.

- [31] 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* (IEEE, Piscataway, NJ, 2011), 344–353.
- [32] S. Lloyd, ‘Universal quantum simulators’, *Science* **273**(5278) (1996), 1073–1078.
- [33] C. Marriott and J. Watrous, ‘Quantum Arthur–Merlin games’, *Comput. Complexity* **14**(2) (2005), 122–152.
- [34] C. Mochon, ‘Hamiltonian oracles’, *Phys. Rev. A* **75**(4) (2007), 042313.
- [35] R. Motwani and P. Raghavan, *Randomized Algorithms* (Cambridge University Press, 1995).
- [36] D. Nagaj, P. Wocjan and Y. Zhang, ‘Fast amplification of QMA’, *Quantum Inf. Comput.* **9**(11–12) (2009), 1053–1068.
- [37] A. Paetznic and K. M. Svore, ‘Repeat-until-success: Non-deterministic decomposition of single-qubit unitaries’, *Quantum Inf. Comput.* **14**(15–16) (2014), 1277–1301.
- [38] D. Poulin, A. Qarry, R. D. Somma and F. Verstraete, ‘Quantum simulation of time-dependent Hamiltonians and the convenient illusion of Hilbert space’, *Phys. Rev. Lett.* **106**(17) (2011), 170501.
- [39] A. Scherer, B. Valiron, S.-C. Mau, S. Alexander, E. van den Berg and T. E. Chapuran, ‘Concrete resource analysis of the quantum linear system algorithm used to compute the electromagnetic scattering cross section of a 2D target’, *Quantum Inf. Process.* **16**(3) (2017), 60.
- [40] M. Suzuki, ‘General theory of fractal path integrals with applications to many-body theories and statistical physics’, *J. Math. Phys.* **32**(2) (1991), 400–407.
- [41] J. Watrous, ‘Zero-knowledge against quantum attacks’, *SIAM J. Comput.* **39**(1) (2009), 25–38.
- [42] G. N. Watson, *A Treatise on the Theory of Bessel Functions* (Cambridge University Press, Cambridge, UK, 1922).
- [43] N. Wiebe, D. W. Berry, P. Høyer and B. C. Sanders, ‘Simulating quantum dynamics on a quantum computer’, *J. Phys. A* **44**(44) (2011), 445308.
- [44] N. Wiebe and M. Roetteler, ‘Quantum arithmetic and numerical analysis using Repeat-Until-Success circuits’, *Quantum Inf. Comput.* **16**(1–2) (2016), 134–178.