

**Fixed-point adiabatic quantum search**

Alexander M. Dalzell, Theodore J. Yoder, and Isaac L. Chuang  
*Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, USA*  
 (Received 12 September 2016; published 11 January 2017)

Fixed-point quantum search algorithms succeed at finding one of  $M$  target items among  $N$  total items even when the run time of the algorithm is longer than necessary. While the famous Grover's algorithm can search quadratically faster than a classical computer, it lacks the fixed-point property—the fraction of target items must be known precisely to know when to terminate the algorithm. Recently, Yoder, Low, and Chuang [*Phys. Rev. Lett.* **113**, 210501 (2014)] gave an optimal gate-model search algorithm with the fixed-point property. Previously, it had been discovered by Roland and Cerf [*Phys. Rev. A* **65**, 042308 (2002)] that an *adiabatic* quantum algorithm, operating by continuously varying a Hamiltonian, can reproduce the quadratic speedup of gate-model Grover search. We ask, can an adiabatic algorithm also reproduce the fixed-point property? We show that the answer depends on what interpolation schedule is used, so as in the gate model, there are both fixed-point and non-fixed-point versions of adiabatic search, only some of which attain the quadratic quantum speedup. Guided by geometric intuition on the Bloch sphere, we rigorously justify our claims with an explicit upper bound on the error in the adiabatic approximation. We also show that the fixed-point adiabatic search algorithm can be simulated in the gate model with neither loss of the quadratic Grover speedup nor of the fixed-point property. Finally, we discuss natural uses of fixed-point algorithms such as preparation of a relatively prime state and oblivious amplitude amplification.

DOI: [10.1103/PhysRevA.95.012311](https://doi.org/10.1103/PhysRevA.95.012311)

**I. INTRODUCTION**

The remarkable discovery of Grover's algorithm [1], which solves the simplest of search problems quadratically faster than a classical computer, has helped fuel interest in quantum computing for the past two decades.

Grover's algorithm finds one of  $M$  target items among  $N$  total items in  $O(\sqrt{N/M})$  time with high probability by repeated application of an operation called the Grover iterate on an initial superposition of all of the items. The quantum state lies in an  $N$  dimensional Hilbert space, but the symmetry of the problem allows for a reduction to two dimensions. In this picture, the Grover iterate can be interpreted as a rotation which moves the quantum state closer to the direction of the target items. However, too many applications causes the state to overrotate; knowledge of  $\lambda = M/N$ , the fraction of target items, is needed to choose the right number of iterations.

How do we find a target item when  $\lambda$  is unknown, while maintaining the Grover-like quadratic speedup? One solution is to first estimate  $\lambda$  using quantum counting algorithms [2–4]; another involves choosing the number of iterations randomly from an exponentially increasing range of integers until a target state is found [5]. Although these methods have Grover-like scaling, they require measuring the system and repreparing the initial state. In contrast, Aaronson and Christiano [6] give a solution which prepares a state, albeit a mixed state, arbitrarily near to the target state subspace avoiding the need for measurement and requiring only knowledge of a (typically small) lower bound  $w$  for the fraction  $\lambda$ . Similarly, under the same assumption  $w \leq \lambda$ , Yoder, Low, and Chuang [7] provide an algorithm which coherently prepares the (pure) uniform superposition  $|E\rangle$  over the  $M$  target states with arbitrarily small error: it produces  $|E\rangle$  with fidelity at least  $\sqrt{1 - \delta^2}$  in only  $O(\log(1/\delta)/\sqrt{w})$  time, thus exhibiting the quadratic speedup. The algorithm is a *fixed-point* search algorithm [8,9] since the success probability remains high even when  $\lambda > w$  (i.e., when the algorithm was run longer than necessary).

Fixed-point search algorithms can be used for error-correcting schemes [10], oblivious amplitude amplification [11], or situations where a natural lower bound on  $\lambda$  exists, such as preparation of the relatively prime state (see Sec. VI). More generally, knowing the existence of even one target state provides a lower bound  $\lambda \geq 1/N$ . Additionally, Aaronson and Christiano [6] explain how a fixed-point algorithm might be a strategy for counterfeiters of quantum money to amplify imperfect copies.

The  $O(\sqrt{N/M})$  scaling of Grover's algorithm has been shown to be optimal [12], but this speedup is not restricted to the gate model; a similar speedup is found in continuous models. For example, Farhi and Gutmann introduced the Hamiltonian oracle model [13] and showed how an analog of search displays the quadratic speedup. Likewise, the speedup is reproduced [14] for searching in the model of adiabatic quantum computation (AQC) [15]. But these algorithms rely on knowledge of  $\lambda$ , the first to decide when to terminate the evolution and the second for defining the adiabatic interpolation schedule. Since Hamiltonians generate rotations, we might expect to be hindered by the same problem of overrotation that plagues Grover's algorithm in the absence of knowledge of  $\lambda$ . Indeed, search in the Hamiltonian oracle model faces this exact problem. However, for AQC, we will show how this issue is avoided by presenting a fixed-point version of the adiabatic search algorithm with run time which displays Grover-like scaling (meanwhile, other versions are shown not to share these properties).

In the context of AQC, fixed-point algorithms imply a robustness to systematic errors in the initial Hamiltonian. Moreover, a continuous algorithm might fit naturally into experiment, or perhaps provide a constant-factor speedup in implementation over a gate-model fixed-point algorithm. Indeed, implementing fixed-point search as a continuous algorithm, we gain continuous control over the lower bound  $w$ , which in the gate-model algorithm [7] is related to the number of Grover iterates and is therefore discrete.

Rigorously proving that a version of the adiabatic search algorithm is fixed point requires precise statements about the adiabatic theorem, which are often elusive. Many AQC works simply take a heuristic approach to the adiabatic theorem [15,16], and do not explicitly evaluate or bound the error in the adiabatic approximation, instead declaring the approximation valid if the Hamiltonian varies slowly in comparison to the square of its eigenvalue gap. Other works, including [14], use explicit bounds on the error which have since been shown to be incorrect in some cases [17,18]. More rigorous approaches have been taken. For example, in Refs. [19–23], the error is written as a series in powers of  $1/T$  where  $T$  is the total run time. Rezakhani *et al.* [23] do this explicitly for the adiabatic search algorithm, while some of the other treatments rely on assumptions or consider cases which are not applicable to the adiabatic search algorithm. This type of expansion is useful for understanding the asymptotic behavior of the algorithm, but less so for finding explicit bounds on the error when  $T$  and  $\lambda$  are finite. In Refs. [24–26], a bound on the error is proved in terms of the eigenvalue gaps and derivatives of the Hamiltonian, but these works are concerned mostly with establishing a polynomial relationship between  $T$ , the gap, and the adiabatic error. These bounds are not tight enough to imply Grover-like scaling for the search algorithm. It would be preferable to develop an intuitively motivated and explicit upper bound on the algorithm’s failure probability which is both tight enough to give reliable scaling estimates and easy to evaluate for finite run times of the adiabatic search algorithm.

We develop exactly such a method for analysis of the adiabatic search algorithm motivated by the geometric intuition that a two-level system evolving by a time-dependent Hamiltonian is equivalent to a spin precessing about a varying magnetic field. Using this framework we derive three main results.

First, we consider an arbitrary interpolation schedule and find an explicit expression (Theorem 1) which upper bounds the failure probability of the algorithm. We evaluate the upper bound for several different families of schedules, parametrized by  $w$ , a lower bound for  $\lambda$ , and  $\epsilon$ , a slowness parameter. We show examples of schedules which have the fixed-point property but lack Grover-like scaling, and vice versa. The “standard” schedule considered by Ref. [14] is shown to have both Grover-like scaling and the fixed-point property simultaneously (Theorem 3).

Second, we modify our argument to produce an exact expression for the failure probability for a particular schedule (Theorem 2), which reproduces the quantum speedup for search—in fact it is even faster than the standard schedule by a constant factor—but is not fixed point. Exact expressions such as this are a rare occurrence among results regarding the adiabatic theorem.

Third, we describe a gate-model algorithm in the same oracle framework of Ref. [7] which simulates the adiabatic search algorithm and we bound its failure probability (Theorem 4). This bound implies that the simulation of the adiabatic search algorithm with standard schedule retains the fixed-point property and Grover-like scaling, yielding an alternative gate-model fixed-point algorithm. We argue that the existence of such a gate-model simulation that does not compromise the quantum speedup is not obvious.

## II. ADIABATIC SEARCH ALGORITHM

In this section, we set up the adiabatic search problem and discuss the adiabatic theorem. Continuous-time search takes place in an  $N$ -dimensional Hilbert space, a superposition of the orthonormal states  $|0\rangle, |1\rangle, \dots, |N-1\rangle$ . Some number,  $M$ , of these states have been marked, dubbed *target states*, and our goal is to obtain one of them. To do so, we are given access to (scaled multiples of) the oracle Hamiltonian  $I - P$  where  $P$  is the projector onto the space spanned by the  $M$  target states, and  $I$  is the identity operator. We see that any target state is an eigenstate of  $I - P$  with eigenvalue 0, while any nontarget state is an eigenstate with eigenvalue 1. This is not the only oracle Hamiltonian which works; any Hamiltonian whose ground state is a solution to the search problem is sufficient.

The adiabatic solution to this search problem begins the system in the equal-superposition state

$$|B\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \quad (1)$$

at time  $t = 0$ , and further applies the Hamiltonian

$$H(t) = (1 - s(t))H_0 + s(t)H_1, \quad (2)$$

where

$$H_0 = I - |B\rangle\langle B|, \quad (3)$$

$$H_1 = I - P, \quad (4)$$

and  $s(t)$  is a continuous and monotonically nondecreasing function of time such that  $s(0) = 0$  and  $s(T) = 1$ . We see that the beginning state  $|B\rangle$  is the ground state of the initial Hamiltonian  $H_0$ . We are assuming, as is standard [13,15], that the ability to apply certain Hamiltonians individually also implies the ability to apply their sum. Experimentally this is quite justifiable (see, e.g., Refs. [27,28]).

With this physical setup, the system, represented by  $|\psi(t)\rangle$ , evolves based on the Schrödinger equation, which is given by

$$i \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle. \quad (5)$$

The adiabatic theorem suggests that, if the Hamiltonian is changed slowly enough, the system will remain approximately in the instantaneous ground state of the Hamiltonian. If this is true, at time  $t = T$ , the system will be approximately in the ground state of  $H_1$ , and when measured will yield a target state with high probability.

### A. Adiabatic theorem

The adiabatic theorem is used to approximate the evolution of a quantum state subject to a continuously varying Hamiltonian. Suppose a system represented by  $|\psi(t)\rangle$  is exposed to a time-dependent Hamiltonian  $H(t)$ , as is true for the algorithm we discuss in this paper. Further, suppose  $H(t)$  has spectral decomposition

$$H(t) |\phi_n(t)\rangle = E_n(t) |\phi_n(t)\rangle, \quad (6)$$

where  $E_n(t)$  and  $|\phi_n(t)\rangle$  are the instantaneous eigenenergies and eigenstates of the Hamiltonian, with  $E_0 < E_1 \dots < E_N$ . The adiabatic theorem states that if the state begins in the ground state  $|\psi(0)\rangle = |\phi_0(0)\rangle$ , then it will stay approximately in the ground state (up to a global phase) throughout the evolution  $|\psi(t)\rangle \approx |\phi_0(t)\rangle$ , provided that the Hamiltonian is changing sufficiently slowly. The criteria for slowness has been a notorious point of confusion for AQC. The general heuristic for adiabaticity used in early papers on AQC, such as Ref. [15], relates the rate of change of the Hamiltonian to the energy gap  $\Delta(t) \equiv E_1(t) - E_0(t)$  between the ground state and the first excited state

$$|\langle \phi_1(t) | \dot{H}(t) | \phi_0(t) \rangle| \ll \Delta(t)^2. \quad (7)$$

This makes the notion of slowness less vague, but still tells us nothing about how exactly the slowness of the algorithm affects the error in the approximation. Moreover, a closer examination reveals this criteria is not even completely correct [17,18]. However, more rigorous treatments of the adiabatic theorem have been done [19–24,26]. In this paper, we shall use the heuristic condition to inform our choice of interpolation schedule  $s(t)$ , but we do not rely on it mathematically. By analyzing the algorithm geometrically, we find a rigorous upper bound on the error of the approximation in terms of the schedule  $s(t)$ . The requirement that this bound on the error be small can be used as a more rigorous and less vague replacement for Eq. (7).

### B. Geometry of the algorithm

As in the gate-model analysis of Grover's algorithm, the symmetry of the adiabatic search algorithm allows us to reduce the state space to two dimensions, spanned by  $|E\rangle$  and  $|\bar{E}\rangle$  where  $|E\rangle$  is the equal superposition over the  $M$  target states (or “end” state), and  $|\bar{E}\rangle$  is the equal superposition over the  $N - M$  nontarget states. We let  $\lambda = M/N$  be the fraction of target states allowing us to write  $|B\rangle = \sqrt{1-\lambda}|\bar{E}\rangle + \sqrt{\lambda}|E\rangle$ . The projector  $P$  acts on this subspace like  $|E\rangle\langle E|$ . As the state evolves by Eq. (5) it will always remain in this subspace. We change basis by letting  $|0\rangle = \cos(\mu)|\bar{E}\rangle + \sin(\mu)|E\rangle$  and  $|1\rangle = -\sin(\mu)|\bar{E}\rangle + \cos(\mu)|E\rangle$ , with  $\cos(2\mu) = \sqrt{1-\lambda}$ . In this basis Eqs. (3) and (4) can be written as

$$H_0 = \frac{1}{2}I - \frac{1}{2}\hat{n}_0 \cdot \vec{\sigma}, \quad (8)$$

$$H_1 = \frac{1}{2}I - \frac{1}{2}\hat{n}_1 \cdot \vec{\sigma}, \quad (9)$$

where

$$\hat{n}_0 = \sqrt{\lambda} \hat{x} + \sqrt{1-\lambda} \hat{z}, \quad (10)$$

$$\hat{n}_1 = \sqrt{\lambda} \hat{x} - \sqrt{1-\lambda} \hat{z} \quad (11)$$

are normalized vectors and  $\vec{\sigma}$  is the vector of Pauli matrices ( $X, Y, Z$ ). The purpose of the basis change is to make  $\hat{n}_0$  and  $\hat{n}_1$  symmetric about the  $xy$  plane.

From Eq. (2), we write the Hamiltonian in this 2D space as

$$H(s) = \frac{1}{2}I - \frac{\Delta_\lambda}{2} \hat{n} \cdot \vec{\sigma}, \quad (12)$$

where

$$\Delta_\lambda(s) = \sqrt{1 - 4s(1-s)(1-\lambda)}, \quad (13)$$

$$\hat{n}(s) = \frac{1}{\Delta_\lambda} (\sqrt{\lambda} \hat{x} + \sqrt{1-\lambda}(1-2s) \hat{z}) \quad (14)$$

$$= \sin(\theta(s)) \hat{x} + \cos(\theta(s)) \hat{z}, \quad (15)$$

and

$$\theta(s) = \arccos(\hat{n} \cdot \hat{z}) = \arccos\left(\frac{(1-2s)\sqrt{1-\lambda}}{\Delta_\lambda}\right) \quad (16)$$

is the angle the normalized vector  $\hat{n}$  makes with the  $z$  axis. Using Eqs. (13) and (16), it can be verified that

$$\frac{d\theta}{ds} = \frac{2\sqrt{\lambda(1-\lambda)}}{\Delta_\lambda^2}. \quad (17)$$

We will also find it useful to describe states by their position on the Bloch sphere. A normalized vector  $\hat{r}$  can be written as

$$\hat{r} = \sin(\xi) \cos(\phi) \hat{x} + \sin(\xi) \sin(\phi) \hat{y} + \cos(\xi) \hat{z}. \quad (18)$$

The state which is mapped to the point  $\hat{r}$  on the Bloch sphere is

$$|\hat{r}\rangle = \cos(\xi/2)|0\rangle + e^{i\phi} \sin(\xi/2)|1\rangle. \quad (19)$$

It is quick to show that

$$\hat{r} \cdot \vec{\sigma} |\hat{r}\rangle = |\hat{r}\rangle, \quad (20)$$

$$\hat{r} \cdot \vec{\sigma} |-\hat{r}\rangle = -|-\hat{r}\rangle. \quad (21)$$

Thus, referring to Eq. (12), the eigenvectors of  $H(s)$  are  $|\hat{n}(s)\rangle$  and  $|-\hat{n}(s)\rangle$ , and the gap between their eigenvalues is  $\Delta_\lambda$ .

By viewing the state of the system  $|\psi(t)\rangle$  by its position on the Bloch sphere and the Hamiltonian  $H(t) = I/2 - \vec{v}(t) \cdot \vec{\sigma}/2$  by the “Hamiltonian vector”  $\vec{v}(t)$ , it is apparent how we can think the system as a spin-1/2 particle exposed to a magnetic field of strength  $\Delta_\lambda$  pointing in the  $\hat{n}$  direction. The tip of the magnetic field vector  $\Delta_\lambda \hat{n}$  associated with the Hamiltonian traces out a vertical line in the  $xz$  plane as shown in Fig. 1. With this in mind, we can see intuitively how the algorithm works. The spin of the particle precesses on the Bloch sphere around the magnetic field at a rate proportional to the field strength. At first, the spin and field are aligned in the direction of  $|B\rangle$ . The field is rotated from the begin state  $|B\rangle$  to the end state  $|E\rangle$ . The precession of the spin around the field causes the field to drag the spin along with it as it changes, so at time  $T$  the spin is approximately in the same direction as the field, which corresponds to the desired final state  $|E\rangle$ . This is a specific example of the adiabatic theorem in two dimensions.

The gap  $\Delta_\lambda(s)$  can be interpreted as the length of the Hamiltonian vector as it changes linearly from  $|B\rangle$  to  $|E\rangle$  in Fig. 1. Notably,  $\Delta_\lambda(0) = \Delta_\lambda(1) = 1$  and  $\Delta_\lambda(1/2) = \sqrt{\lambda}$ .

The only piece left to the algorithm is what to choose for the function  $s(t)$ . Regardless of the form of  $s(t)$ , the Hamiltonian vector will follow the linear trajectory from the begin state to the end state; the function  $s(t)$  determines how fast it moves along this trajectory. We call  $s(t)$  the *schedule* of the algorithm.

Once the schedule has been specified, the final state  $|\psi(T)\rangle$  can be computed by evolution via the Schrödinger equation (5).

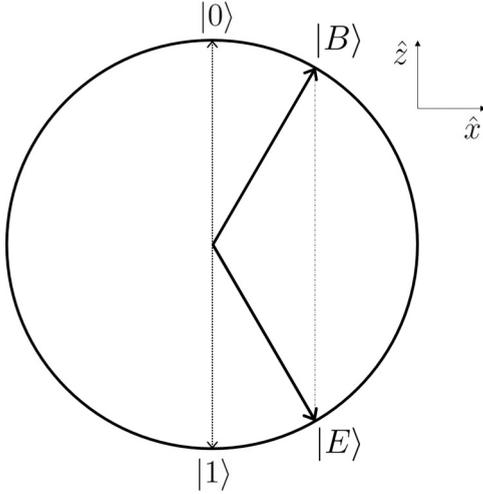


FIG. 1. Geometric picture of the algorithm. This is the  $xz$  plane of the Bloch sphere for  $\lambda = 1/4$ . The Hamiltonian vector  $\Delta_\lambda \hat{n}$  associated with  $H(s)$  follows the dotted line between the begin state and the end state.

For a given schedule  $s(t)$ , we can compute the probability of measuring a target state at the end of the algorithm as a function of  $\lambda$

$$P(\lambda) = |\langle E | \psi(T) \rangle|^2. \quad (22)$$

We also define the *error amplitude*  $\delta(\lambda)$  so that  $P(\lambda) = 1 - \delta(\lambda)^2$ . We note that when  $\lambda = 0$ , there are no target states, and when  $\lambda = 1$ , all the states are target states, so regardless of schedule,  $P(0) = 0$  and  $P(1) = 1$ .

In the next section, we discuss how the error amplitude  $\delta(\lambda)$  depends on the choice of schedule  $s(t)$  and whether the algorithm has the fixed-point property.

### III. SCHEDULING AND THE FIXED-POINT PROPERTY

#### A. Fixed-point property

For our analysis, we consider families of schedules  $s(t; \epsilon, w)$  parametrized by  $\epsilon$  and  $w$ . The  $\epsilon$  parameter is typically small and represents how fast the interpolation occurs. The  $w$  parameter represents a lower bound on the fraction of target states  $\lambda$ . Each schedule family does not depend on  $\lambda$  since we assume in general that we do not know  $\lambda$  precisely; all we know is the lower bound  $w \leq \lambda$ . This is a reasonable scenario: if we know that at least one item is marked, then  $\lambda \geq 1/N$ . Moreover, some problems admit tighter lower bounds, such as the problem of preparing the relatively prime state, which we discuss in Sec. VI.

Together,  $\epsilon$  and  $w$  determine the total run time  $T$  by requiring  $s(0; \epsilon, w) = 0$  and  $s(T; \epsilon, w) = 1$ . In Sec. III C, we give three examples of different families of schedules.

We denote the family of success probability functions which correspond to the family of schedules  $s(t; \epsilon, w)$  by  $P(\lambda; \epsilon, w)$  and, by extension, the family of error amplitudes by  $\delta(\lambda; \epsilon, w) = \sqrt{1 - P(\lambda; \epsilon, w)}$ . We now formally stipulate conditions on  $P(\lambda; \epsilon, w)$  that make the search algorithm under schedule  $s(t; \epsilon, w)$  a fixed-point algorithm.

*Fixed-point property.* The adiabatic search algorithm operating under schedule family  $s(t; \epsilon, w)$  has the fixed-point property if there exists a function  $f(\epsilon)$  independent of  $\lambda$  and  $w$  such that  $P(\lambda; \epsilon, w) \geq 1 - (f(\epsilon))^2$ , for all  $\lambda \geq w$  and  $f(\epsilon) \rightarrow 0$  as  $\epsilon \rightarrow 0$ .

If we have an adiabatic fixed-point search algorithm, and we wish to make  $P(\lambda) \geq 1 - \delta^2$  for all  $\lambda \geq w$ , we need only make  $\epsilon$  small enough so that  $f(\epsilon) \leq \delta$ . Thus we view  $\epsilon$  and  $w$  as independent input parameters which control two features of our success probability function  $P(\lambda)$ . The parameter  $\epsilon$  controls an upper bound on the failure probability  $f(\epsilon)^2$ , and  $w$  controls the range over which this success guarantee is valid. The price paid for a decrease in  $w$  or  $\epsilon$  is an increase in the run time  $T$ .

Although the notion of a schedule and the slowness parameter  $\epsilon$  might not make sense in the gate model, the adiabatic fixed-point definition is designed to mirror the definition in the gate model. A gate-model fixed-point algorithm, such as that presented in Ref. [7], is constructed given parameters  $\delta$  and  $w$  to succeed with probability  $P(\lambda) \geq 1 - \delta^2$  so long as  $\lambda \geq w$ .

We are also interested in the scaling of the algorithm. The algorithm has Grover-like scaling if the run time  $T = O(1/\sqrt{w})$ .

It is possible for a family of schedules to lack the fixed-point property if  $\delta(\lambda; \epsilon, w)$  cannot be bounded by a function only of  $\epsilon$ . It is also possible for a family of schedules to possess the fixed-point property but lack Grover-like scaling if the dependence of  $T$  on  $w$  is worse than  $O(1/\sqrt{w})$ . We present examples of these cases in Sec. III C, where we consider three different families of schedules. Determining if a certain schedule family has the fixed-point property requires analysis of the error in the adiabatic approximation. The tool we use for this purpose is Theorem 1, presented in the next section.

#### B. General bound on error probability

The heuristic condition for validity of the adiabatic theorem in Eq. (7) is inadequate for several reasons. First, we have not rigorously defended it, and it has actually been shown not to be a sufficient condition for adiabaticity in all cases [17, 18]. But also, it fails to provide information about the error amplitude  $\delta(\lambda)$ , besides that it is small. In particular, it does not allow us to understand how our choice of schedule  $s(t)$  affects the error amplitude  $\delta(\lambda)$ . Such information would be practically useful for actually running the adiabatic search algorithm and also allows us to understand the asymptotic behavior of  $\delta(\lambda)$  as  $T \rightarrow \infty$  and as  $\lambda \rightarrow 0$ .

We wish to provide this information by finding an upper bound for  $\delta(\lambda)$  in terms of the schedule  $s(t)$ . As a starting point, we can evaluate Eq. (7) using the geometric framework we set up in Sec. II B. From Eq. (12), we see that the ground and first-excited states of  $H(s)$  are  $|\hat{n}\rangle$  and  $|-\hat{n}\rangle$ , so the left-hand side of Eq. (7) is

$$\begin{aligned} | \langle -\hat{n} | \dot{H} | \hat{n} \rangle | &= \dot{s}(t) \left| \langle -\hat{n} | \frac{1}{2} (\hat{n}_0 - \hat{n}_1) \cdot \vec{\sigma} | \hat{n} \rangle \right| \\ &= \dot{s}(t) \sqrt{1 - \lambda} | \langle -\hat{n} | Z | \hat{n} \rangle | \\ &= \dot{s}(t) \frac{\sqrt{\lambda(1 - \lambda)}}{\Delta_\lambda}, \end{aligned} \quad (23)$$

where our computation relies on Eqs. (2), (8)–(11), and (19). We note that since the minimum of  $\Delta_\lambda$  is  $\sqrt{\lambda}$ , this quantity is at most  $\dot{s}(t)$ . Since the eigenvalue gap is  $\Delta_\lambda$ , Eq. (7) reads

$$\frac{ds}{dt} \ll \frac{\Delta_\lambda^3}{\sqrt{\lambda(1-\lambda)}}. \quad (24)$$

Now, in Theorem 1, we present an upper bound on the error amplitude  $\delta(\lambda)$  which can be computed given  $s(t)$ . Thus the requirement that  $\delta(\lambda)$  be small combined with Theorem 1 serves as an alternative adiabatic condition for  $s(t)$  which is more concrete (and more rigorous) than Eq. (24).

*Theorem 1.* If the adiabatic search algorithm is run according to schedule  $s(t)$  then  $\delta(\lambda) \leq d_0 + d_1$ , where

$$d_0 = 2\sqrt{\lambda(1-\lambda)}\dot{s}(0), \quad (25)$$

$$d_1 = \int_0^T dt \left| \frac{d}{dt} \left( \frac{\sqrt{\lambda(1-\lambda)} ds}{\Delta_\lambda^3} \right) \right|. \quad (26)$$

Theorem 1 is proved in Sec. IV A. The error is related to the derivatives of the schedule; we think of  $d_0$  as the initial error due to the choice of derivative at  $t = 0$  and  $d_1$  as the additional error accrued during the algorithm. If we require the bound in Theorem 1 on  $\delta(\lambda)$  to be small, then  $d_0$  and  $d_1$  must be small. Since  $d_0$  is small, the quantity  $\dot{s}\sqrt{\lambda(1-\lambda)}/\Delta_\lambda^3$  must be small for  $s = 0$  and since  $d_1$  is also small, that quantity must remain small for all  $s$ , implying Eq. (24) is satisfied.

However, the converse relationship is not true. Equation (24) implies that  $d_0$  is small but does not necessarily imply that  $d_1$  is small. It is possible that  $\dot{s}(t)$  is small throughout the algorithm while  $\ddot{s}(t)$  is not small. Thus the condition  $d_0 + d_1 \ll 1$  is strictly stronger than the heuristic adiabatic condition derived from Eq. (7).

### C. Schedules

In this section, we apply Theorem 1 to three families of schedules. Our first two families fail to have both Grover-like scaling and the fixed-point property, although we note how they can be made to have one of the two traits. The third family avoids the issues facing the first two and has both properties.

#### 1. Constant-speed schedule

First, we consider the constant-speed schedule family  $s_c(t; \epsilon_c, w)$  defined by  $ds_c/dt = \epsilon_c$ . This definition does not depend on the  $w$  parameter. Boundary conditions imply  $T_c = 1/\epsilon_c$ , and

$$s_c(t; \epsilon_c, w) = \epsilon_c t = t/T_c. \quad (27)$$

We can evaluate the bound in Theorem 1 exactly, finding that  $d_0 = 2\sqrt{\lambda(1-\lambda)}\epsilon_c$  and  $d_1 = 2\sqrt{1-\lambda}(\lambda^{-1} - \sqrt{\lambda})\epsilon_c$ . So our bound reads  $\delta(\lambda; \epsilon_c, w) \leq 2\epsilon_c\sqrt{1-\lambda}/\lambda$ . Letting  $\delta$  be the maximum error amplitude for  $\lambda \geq w$ , our bound says that  $\delta = \max_{\lambda \geq w} \delta(\lambda; \epsilon_c, w) < 2\epsilon_c/w$ , which diverges as  $w$  approaches zero with  $\epsilon_c$  constant. Hence there is no function  $f(\epsilon_c)$  such that our bound guarantees  $\delta \leq f(\epsilon_c)$ ; this suggests the algorithm is not fixed point. Even equipped with a tighter upper bound than the one we present, it cannot be

possible to bound  $\delta \leq f(\epsilon_c)$  with  $\lim_{\epsilon_c \rightarrow 0} f(\epsilon_c) = 0$ : the error amplitude  $\delta(\lambda; \epsilon_c, w)$  has no  $w$  dependence, so  $\lim_{w \rightarrow 0} \delta \geq \lim_{\lambda \rightarrow 0} \delta(\lambda; \epsilon_c, w) = 1$ . The last equality comes about because all schedules fail to find a marked state in the limit of no states being marked. Thus the constant-speed schedule does not have the fixed-point property.

However, we succeed in getting a concrete upper bound on the error probability in terms of both  $w$  and  $\epsilon_c$ , if not  $\epsilon_c$  alone. Decreasing  $w$  with constant  $\epsilon_c$  causes the bound on the error probability to rise, but this effect can be canceled by a roughly proportional decrease in  $\epsilon_c$ . The fixed-point definition is not satisfied since both parameters must be adjusted to keep the error probability low, but we can use this fact to create a related family of schedules which is fixed point.

Consider the schedule  $s'_c(t; \epsilon'_c, w)$  defined by  $ds'_c/dt = \epsilon'_c w$ . The relationship between the primed and unprimed schedules is simply  $\epsilon_c = \epsilon'_c w$ , so our bound states that  $\delta = \max_{\lambda \geq w} \delta(\lambda) < 2\epsilon'_c$ . For this schedule,  $T'_c = 1/(\epsilon'_c w) = O(1/(\delta w))$ . It is fixed point but it lacks Grover-like scaling.

#### 2. Fast schedule

Our first attempt failed to achieve a quantum speedup. In our second attempt, we make the schedule faster when the gap is large, and slower when the gap is small. The idea of varying the speed according to the gap was first proposed in Ref. [14], but our schedule varies in proportion to the cube of the gap, while that in Ref. [14] varies in proportion to the square (and is presented as our third example). Additionally, previous works have not considered these schedules in the context of the fixed-point property and the lower bound  $w \leq \lambda$ .

The *fast* schedule family  $s_f(t; \epsilon_f, w)$  is defined by

$$\frac{ds_f}{dt} = \epsilon_f \frac{\Delta_w^3}{\sqrt{w(1-w)}}. \quad (28)$$

This schedule is designed to make the left and right sides of Eq. (24) proportional by the constant  $\epsilon_f$  for all  $s_f$  when  $\lambda = w$ . From this equation and the boundary conditions  $s_f(0) = 0$ ,  $s_f(T_f) = 1$ , it can be shown that the total time must be

$$T_f = \frac{\sqrt{1-w}}{\epsilon_f \sqrt{w}} \quad (29)$$

and the schedule is

$$s_f(t; \epsilon_f, w) = \frac{1}{2} - \frac{1}{2} \left( 1 - \frac{2t}{T_f} \right) \sqrt{\frac{w}{1 - \left( 1 - \frac{2t}{T_f} \right)^2 (1-w)}}. \quad (30)$$

Since  $T_f = O(1/\sqrt{w})$ , this schedule has Grover-like scaling, a quadratic speedup over the classical search complexity. However, the Grover-like scaling is only useful if the algorithm actually succeeds. We apply the result of Theorem 1 to get an upper bound on the failure probability. Supposing that  $\lambda = w$ , we calculate  $d_0 = 2\epsilon_f$  and  $d_1 = 0$ , meaning

$$\delta(w; \epsilon_f, w) \leq 2\epsilon_f. \quad (31)$$

When  $\lambda = w$ , the error probability is bounded by a function of  $\epsilon_f$ , and the run time  $T_f$  has Grover-like scaling in  $w$ . To show that the algorithm is fixed point, we would need this to also be true when  $\lambda > w$ .

In fact, for the fast schedule, we can modify the arguments for the bound in Theorem 1 to give an *exact* expression for the error amplitude when  $\lambda = w$  as follows.

*Theorem 2.* If the adiabatic search algorithm is run according to the fast schedule given by Eq. (30), then

$$\delta(w; \epsilon_f, w) = \frac{2\epsilon_f}{\sqrt{1 + 4\epsilon_f^2}} \left| \sin \left( \frac{\sqrt{1 + 4\epsilon_f^2}}{2\epsilon_f} \phi_w \right) \right|, \quad (32)$$

where

$$\phi_w = \arctan \left( \sqrt{\frac{1-w}{w}} \right). \quad (33)$$

Theorem 2 is proved in Sec. IV B. This result is significant. If the value of  $\lambda$  is known, we can run the adiabatic search algorithm with the fast schedule setting parameter  $w = \lambda$ , and we have an exact analytic expression for the error in the algorithm implying that it has Grover-like scaling in the fraction  $\lambda$ .

Varying the speed of the interpolation to be faster when the gap is larger allows us to recover the Grover speedup seen in the circuit model.

But what about when  $\lambda$  is unknown? Suppose all we know is that  $\lambda \geq w$ . The heuristic condition from Eq. (24) now reads

$$\epsilon_f \ll \frac{\sqrt{w(1-w)} \Delta_\lambda^3}{\sqrt{\lambda(1-\lambda)} \Delta_w^3}. \quad (34)$$

This is problematic when  $\lambda$  is large, and  $s_f$  is close to zero or 1. For example, fixing  $s_f = 0$  (so  $\Delta_\lambda = \Delta_w = 1$ ), and  $\lambda = 1/2$ , we have  $\epsilon_f \ll \sqrt{w}$ . Looking at Eq. (29), we see that if  $\epsilon_f = O(\sqrt{w})$  then  $T_f = O(1/w)$ , meaning the quadratic speedup is lost.

We confirm this notion with the more concrete bound in Theorem 1. We evaluate  $d_0 = 2\epsilon_f \sqrt{\lambda(1-\lambda)}/\sqrt{w(1-w)}$  and  $d_1 = 2\epsilon_f (1 - w^{3/2}\lambda^{-3/2})\sqrt{\lambda(1-\lambda)}/\sqrt{w(1-w)}$ . In this case,  $\delta(\lambda; \epsilon_f, w) \leq d_0 + d_1 \leq 4\epsilon_f \sqrt{\lambda(1-\lambda)}/\sqrt{w(1-w)}$  which increases like  $O(\epsilon_f/\sqrt{w})$  as  $w \rightarrow 0$ . We are unable to bound the error probability by a function of  $\epsilon_f$  alone, suggesting the algorithm is not fixed point. Other techniques could, in principle, yield a tighter bound on the error probability, but numerical simulations of the algorithm confirm that the algorithm lacks the fixed-point property, so any such attempt would suggest the same result.

As in the constant-speed case, the schedule can be modified to become fixed point. Defining  $s'_f(t; \epsilon'_f, w)$  by the relation  $ds'_f/dt = \epsilon'_f \Delta_w^3$  leads to the relationship  $T'_f = 1/(\epsilon'_f w)$  and the bound  $\delta(\lambda; \epsilon'_f, w) \leq 2\epsilon'_f$  whenever  $\lambda \geq w$ , meaning the primed algorithm is fixed point. In this case, the cost of the fixed-point property is the loss of the Grover-like scaling, since  $T'_f = O(1/w)$ .

### 3. Standard schedule

The third schedule family we present does not have this problem. The *standard* schedule family  $s_s(t; \epsilon_s, w)$ , is defined

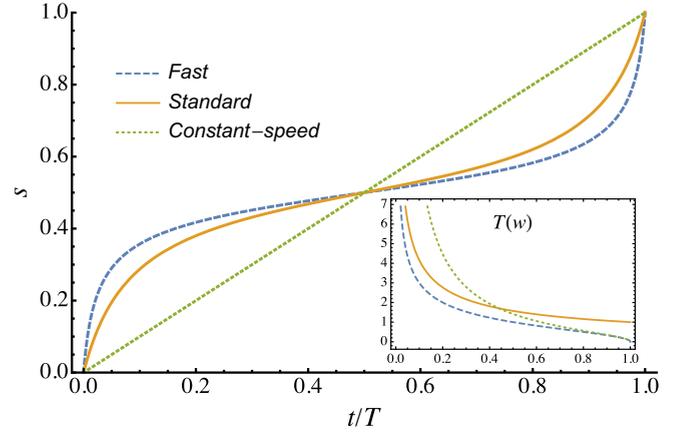


FIG. 2. Comparison of schedules  $s(t)$  at  $w = 1/20$  plotted with respect to normalized time  $t/T$ . Both the fast and standard schedules speed up when the gap is large and slow down when it is small, with the fast schedule exhibiting this behavior more strongly. As  $w \rightarrow 0$ , the fast and standard schedules approach the discontinuous function that is  $1/2$  for  $t/T \in (0, 1)$  and  $0$  ( $1$ ) for  $t/T = 0$  ( $1$ ). As  $w \rightarrow 1$  both schedules approach the constant-speed schedule. The inset shows the total time  $T$  taken (in units of  $1/\epsilon$ ) by these search algorithms as a function of  $w$ . The fast algorithm is the fastest.

by the relationship

$$\frac{ds_s}{dt} = \epsilon_s \Delta_w^2. \quad (35)$$

Using the boundary conditions  $s_s(0) = 0$ ,  $s_s(T_s) = 1$ , it can be shown that

$$T_s = \frac{\phi_w}{\epsilon_s \sqrt{w(1-w)}}, \quad (36)$$

where  $\phi_w$  is given by Eq. (33), and the schedule is

$$s_s(t; \epsilon_s, w) = \frac{1}{2} - \frac{1}{2} \sqrt{\frac{w}{1-w}} \tan \left( \left( 1 - \frac{2t}{T_s} \right) \phi_w \right). \quad (37)$$

By comparison with Eq. (29), we note that the fast schedule is a factor of  $\pi/2$  faster than the standard schedule for small  $w$  and  $\epsilon_s = \epsilon_f$ . The three schedules are plotted in Fig. 2. The standard schedule is so named because it is the schedule first introduced in Ref. [14]. It also appears as the “constant-norm interpolation” in Ref. [23].

We first examine the heuristic condition from Eq. (24), which reads  $\epsilon_s \ll \Delta_\lambda^3 / (\Delta_w^2 \sqrt{\lambda(1-\lambda)})$ , and is satisfied for all  $\lambda \geq w$  so long as  $\epsilon_s$  is small. This suggests that the error will be small for all  $\lambda \geq w$ , while the run time complexity is Grover-like  $O(1/\sqrt{w})$ .

To be more rigorous we apply the bound in Theorem 1. The computation is not as straightforward as for the other two schedules but is carried out in Appendix A and summarized by Theorem 3.

*Theorem 3.* If the adiabatic search algorithm is run with parameters  $w$  and  $\epsilon_s$  according to the standard schedule given by Eq. (37), then

$$\delta = \max_{w \leq \lambda \leq 1} \delta(\lambda; \epsilon_s, w) \leq 2\epsilon_s. \quad (38)$$

Theorem 3 is unique among previous rigorous treatments of the adiabatic theorem, as it relates to AQC. As mentioned, some previous papers [15] merely use Eq. (7) as a heuristic condition for adiabaticity or something similar. Some go further and prove asymptotic statements about the relationship between the error probability and the run time [19,24], and some even compute an asymptotic  $1/T$  expansion of the error probability [20–23], but often this series is truncated after a finite number of terms (which is valid only for large  $T$ ). Here we give an explicit upper bound on the error for general schedule in Theorem 1, and apply this bound to the standard schedule in Theorem 3, which confirms asymptotic notions, but is also valid for finite values of the run time  $T$  and parameter  $w$ .

Theorem 3 rigorously establishes that the standard schedule is a fixed-point algorithm with Grover-like scaling. Since  $\delta = O(\epsilon_s)$ , Eq. (36) implies that

$$T_s = \frac{\phi_w}{O(\delta)\sqrt{w(1-w)}} = O\left(\frac{1}{\delta\sqrt{w}}\right). \quad (39)$$

Comparing this adiabatic algorithm to the fixed-point algorithm in Ref. [7], which has scaling  $O(\log(2/\delta)/\sqrt{w})$ , we can see that it has similar scaling in  $w$ , but exponentially worse scaling in  $\delta$ .

However, if the goal is simply to find the target state, logarithmic scaling in  $\delta$  can be achieved by repetition of our algorithm. To see this, suppose we fix  $\delta^2 = 1/3$ , and repeat the algorithm  $\log(\delta^2)/\log(1/3)$  times. The algorithm fails only if each individual run fails, which occurs with probability  $\delta^2$ . The total time of this algorithm is logarithmic in the eventual error amplitude  $\delta'$ :  $T'_s = O(\log(1/\delta'))$ .

Additionally, other schedules are possible which might have better scaling in  $\delta$ . In particular, it is discussed in Ref. [23] how setting the first  $k$  time derivatives of  $s(t)$  to zero at the beginning and end of the algorithm makes the first  $k$  terms in the  $1/T$  expansion of the error amplitude vanish, suggesting that  $\delta = O(1/T^{k+1})$ . If all of the time derivatives vanish at  $s = 0$  and  $s = 1$ , we might even be able to reproduce the logarithmic dependence on  $\delta$  without resorting to repetition. These notions are explored in an asymptotic sense in Ref. [23], but no bound for finite  $T$  and  $w$  is given as we do in Theorem 1.

#### IV. GEOMETRIC INTUITION AND PROOFS

Theorems 1 and 2 are proved in this section, while Theorems 3 and 4 are proved in the Appendices. Throughout, when we refer to a Hamiltonian  $\mathcal{H} = I/2 - \vec{v} \cdot \vec{\sigma}/2$  in a geometric context, we mean the Hamiltonian vector  $\vec{v}$ .

*Geometric intuition.* At its core, the method for arriving at Theorems 1 and 2 is very intuitive. We view the Hamiltonian and state vectors geometrically on the Bloch sphere. As the Hamiltonian is changed, the state vector strays away from the Hamiltonian vector. We would like to find how far away it moves, which we quantify by computing an upper bound on the angle between the state and Hamiltonian vectors at the end of the algorithm. We do this by looking at the state from the point of view of the Hamiltonian; we change coordinates so that the Hamiltonian always points in the  $\hat{z}$  direction, and the state is nearly in the  $\hat{z}$  direction. This coordinate transformation is time dependent, which gives rise

to another term in the effective Hamiltonian. For the fast schedule, the direction of the effective Hamiltonian remains fixed, although its magnitude changes. Since the state vector in these coordinates precesses about the effective Hamiltonian, the angle between the state vector and effective Hamiltonian is constant in time, allowing for exact calculation of the error amplitude  $\delta(w)$ , and leading to Theorem 2. For general schedule, the direction of the effective Hamiltonian is not constant in time, but it changes very little. As a result, the angle between the state vector and the Hamiltonian changes very little and can be bounded, yielding Theorem 1.

The vital mathematical concept is the time-dependent change of coordinates which rotates the Hamiltonian back to the  $\hat{z}$  direction, and is described by Lemma 1. The change of coordinates is defined by the state

$$|\phi(t)\rangle = \exp\left(i\frac{\theta}{2}Y\right)|\psi(t)\rangle, \quad (40)$$

where  $\theta$  is given by Eq. (16). At any instant of time, the relationship between  $|\psi\rangle$  and  $|\phi\rangle$  is simply a rotation of the Bloch sphere by angle  $-\theta$  about the  $\hat{y}$  axis.

*Lemma 1.*  $|\phi(t)\rangle$  evolves according to the Schrödinger equation

$$i\frac{d}{dt}|\phi(t)\rangle = H_\phi|\phi(t)\rangle, \quad (41)$$

where  $H_\phi$  is the effective Hamiltonian given by

$$H_\phi = \frac{1}{2}I - \frac{m}{2}\hat{n}_\phi \cdot \vec{\sigma}, \quad (42)$$

with

$$m = \sqrt{\Delta_\lambda^2 + (\dot{\theta})^2} \quad (43)$$

and

$$\hat{n}_\phi = \frac{1}{m}(\dot{\theta}\hat{y} + \Delta_\lambda\hat{z}). \quad (44)$$

*Proof of Lemma 1.* This lemma follows from application of the Schrödinger equation (5) on the definition of  $|\phi\rangle$  in Eq. (40):

$$i\frac{d}{dt}|\phi(t)\rangle = \left(-\frac{\dot{\theta}}{2}Y + \exp\left(i\frac{\theta}{2}Y\right)H \exp\left(-i\frac{\theta}{2}Y\right)\right)|\phi(t)\rangle.$$

Using the geometric expression (12) for  $H$ , we have

$$\begin{aligned} & \exp\left(i\frac{\theta}{2}Y\right)H \exp\left(-i\frac{\theta}{2}Y\right) \\ &= \frac{1}{2}I - \frac{\Delta_\lambda}{2} \exp\left(i\frac{\theta}{2}Y\right)(\hat{n} \cdot \sigma) \exp\left(-i\frac{\theta}{2}Y\right) \\ &= \frac{1}{2}I - \frac{\Delta_\lambda}{2}Z \end{aligned}$$

and thus

$$H_\phi = \frac{1}{2}I - \frac{\Delta_\lambda}{2}Z - \frac{\dot{\theta}}{2}Y$$

and the lemma is proved.  $\blacksquare$

The term  $\Delta_\lambda Z/2$  in  $H_\phi$  comes from rotation of the Hamiltonian back to the  $\hat{z}$  axis. The term  $\dot{\theta}Y/2$  comes from the time dependence of this rotation.

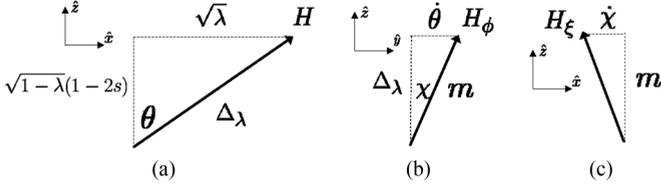


FIG. 3. Geometric guide to the proof of Theorem 1. We perform two time-dependent coordinate changes. (a) The state  $|\psi\rangle$  evolves by Hamiltonian  $H$ , consistent with Fig. 1. (b) The state  $|\phi\rangle$  is related to  $|\psi\rangle$  by Eq. (40) and evolves by Hamiltonian  $H_\phi$ . (c) The state  $|\xi\rangle$  is related to  $|\phi\rangle$  by Eq. (47) and evolves by Hamiltonian  $H_\xi$ . Each Hamiltonian  $\mathcal{H} = I/2 - \vec{v} \cdot \vec{\sigma}/2$  is represented by the vector  $\vec{v}$  in the figure.

### A. Proof of Theorem 1

To prove Theorem 1, we will make two coordinate changes and use Lemma 1 twice. The effective Hamiltonian vectors in each frame are depicted graphically in Fig. 3. Building off of Lemma 1, we define

$$\chi(t) = \arctan(\hat{\theta}(t)/\Delta_\lambda(t)) \quad (45)$$

to be the angle between  $\hat{n}_\phi$  and  $\hat{z}$ , allowing us to rewrite Eq. (44) as

$$\hat{n}_\phi = \sin(\chi)\hat{y} + \cos(\chi)\hat{z}, \quad (46)$$

noting the similarity to Eq. (15). If the interpolation is slow,  $\hat{\theta}$ , and hence  $\chi$  will be small, so  $\hat{n}_\phi$  will be close to  $\hat{z}$ .

Now, we repeat the process again, this time using  $\hat{n}_\phi$  in place of  $\hat{n}$  and rotating in the  $yz$  plane instead of the  $xz$  plane. We let

$$|\xi(t)\rangle = \exp\left(-i \frac{\chi}{2} X\right) |\phi(t)\rangle \quad (47)$$

and by invoking the result from Lemma 1 (except in the  $yz$  plane instead of  $xy$  plane), we see that  $|\xi(t)\rangle$  evolves according to

$$i \frac{d}{dt} |\xi(t)\rangle = H_\xi |\xi(t)\rangle, \quad (48)$$

with

$$H_\xi = \frac{1}{2}I - \frac{1}{2}\hat{n}_\xi \cdot \vec{\sigma} = \frac{1}{2}I - \frac{m}{2}Z + \frac{\dot{\chi}}{2}X. \quad (49)$$

We note that  $|\xi(0)\rangle = \cos(\chi(0)/2)|0\rangle - i \sin(\chi(0)/2)|1\rangle$ . The most general possible equation for  $|\xi(t)\rangle$  is

$$|\xi(t)\rangle = e^{iq(t)} \left( \cos\left(\frac{r(t)}{2}\right)|0\rangle + e^{ip(t)} \sin\left(\frac{r(t)}{2}\right)|1\rangle \right). \quad (50)$$

Note that  $r$  is the angle between  $\hat{z}$  and  $|\xi\rangle$  on the Bloch sphere, but since  $|\xi\rangle$  and  $|\phi\rangle$  are related by the rotation (47), which preserves angles,  $r$  is also the angle between the direction of the effective Hamiltonian  $H_\phi$  and the state  $|\phi\rangle$ .

By plugging Eq. (50) into the Schrödinger equation (48), we can prove the following.

*Lemma 2.*  $|\dot{r}| \leq |\dot{\chi}|$ .

*Proof of Lemma 2.* Applying the left-hand side of Eqs. (48) to (50) gives

$$e^{iq} \left( -\dot{q} \cos\left(\frac{r}{2}\right) - \frac{i}{2} \dot{r} \sin\left(\frac{r}{2}\right) \right) |0\rangle + e^{i(q+p)} \left( -\dot{q} \sin\left(\frac{r}{2}\right) + \frac{i}{2} \dot{r} \cos\left(\frac{r}{2}\right) - \dot{p} \sin\left(\frac{r}{2}\right) \right) |1\rangle$$

and applying the right-hand side gives

$$\frac{1}{2} e^{iq} \left( \cos\left(\frac{r}{2}\right) - m \cos\left(\frac{r}{2}\right) + e^{ip} \dot{\chi} \sin\left(\frac{r}{2}\right) \right) |0\rangle + \frac{1}{2} e^{iq} \left( e^{ip} \sin\left(\frac{r}{2}\right) + \dot{\chi} \cos\left(\frac{r}{2}\right) + e^{ip} m \sin\left(\frac{r}{2}\right) \right) |1\rangle.$$

These quantities must be equal. We can immediately cancel the  $e^{iq}$  factor. Now, the imaginary parts of the coefficient of  $|0\rangle$  for both the left- and right-hand sides must be equal, giving the equation

$$-\frac{1}{2} \dot{r} \sin\left(\frac{r}{2}\right) = \frac{1}{2} \dot{\chi} \sin(p) \sin\left(\frac{r}{2}\right), \\ \dot{r} = -\dot{\chi} \sin(p), \\ |\dot{r}| \leq |\dot{\chi}|, \quad (51)$$

which proves the lemma.  $\blacksquare$

Physically, Lemma 2 is telling us that the angle  $r$  between  $|\phi\rangle$  and the effective Hamiltonian  $H_\phi$  cannot change faster than the angle  $\chi$  between the effective Hamiltonian  $H_\phi$  and  $\hat{z}$ . This fact could have been proved using a purely geometric argument.

The success probability (22) can be expressed in our new coordinates using the relationships (40) and (47):

$$P(\lambda) = |\langle 0|\phi(T)\rangle|^2 = |\langle \xi(0)|\xi(T)\rangle|^2. \quad (52)$$

We define

$$A = \frac{r(0) + r(T)}{2}. \quad (53)$$

Using Eq. (50), we can see that

$$P(\lambda) = |\langle \xi(0)|\xi(T)\rangle|^2 \geq \cos^2(A) \geq 1 - A^2. \quad (54)$$

Since  $\int_0^T \dot{r} dt = r(T) - r(0)$ , we have

$$A = r(0) + \frac{1}{2} \int_0^T \dot{r} dt \\ \leq \chi(0) + \frac{1}{2} \int_0^T |\dot{\chi}| dt. \quad (55)$$

Plugging Eq. (17) into the definition of  $\chi$  in Eq. (45), we see that  $\chi(t) = \arctan(2\sqrt{\lambda}(1-\lambda)\dot{s}(t)/\Delta_\lambda(t)^3)$ . However, when  $u > 0$ ,  $\arctan(u) \leq u$  and  $|d(\arctan(u))/dt| = |\dot{u}/(1+u^2)| \leq |\dot{u}|$ . Thus  $\chi(0) \leq d_0$  and  $\int |\dot{\chi}|/2 \leq d_1$  where  $d_0$  and  $d_1$  are given by Eqs. (25) and (26). Since  $\delta(\lambda) = \sqrt{1-P(\lambda)} \leq A$ , this proves the theorem.

### B. Proof of Theorem 2

Theorem 2 gives an exact expression for the error probability where Theorem 1 only gives an upper bound. The structure of the proof is the same as that for Theorem 1.

We define  $\chi(t)$  as in Eq. (45) for the fast schedule with  $\lambda = w$ , but as can be verified by Eqs. (17) and (28),  $\chi = 2\epsilon_f$  is constant throughout time. So  $\hat{n}_\phi = \sin(\chi)\hat{y} + \cos(\chi)\hat{z}$  is fixed. In this frame, the effective Hamiltonian does not change direction, though its strength varies and is given by  $m(t) = (\Delta_w^2 + (\dot{\theta})^2)^{1/2} = \Delta_w(1 + 4\epsilon_f^2)^{1/2}$ . Thus the angle between the state vector and  $\hat{n}_\phi$  should remain constant throughout the evolution. Mathematically, this follows from Lemma 2, since  $\dot{\chi} = 0$  implies angle  $r$  is constant. It is this fact that allows for exact treatment of the error probability. This is not a coincidence. The proportionality between the gap  $\Delta_w$  and the angular velocity  $\dot{\theta}$  of  $H(t)$  is intimately related to the adiabatic condition in Eq. (7) and the definition of the fast schedule in Eq. (28). We define  $|\xi\rangle$  as in Eq. (47) but since  $\chi$  is constant the coordinate change is no longer time dependent. We see that  $H_\xi = I/2 - m(t)Z/2$ .

At  $t = 0$  we have  $|\xi(0)\rangle = \cos(\chi/2)|0\rangle - i \sin(\chi/2)|1\rangle$ . Evolution by  $H_\xi(t)$  simply applies a relative phase on the  $|1\rangle$  component of  $|\xi\rangle$ , so up to a global phase,  $|\xi(T_f)\rangle = \cos(\chi/2)|0\rangle - i \exp(-i \int_0^{T_f} m(t)dt) \sin(\chi/2)|1\rangle$ .

The phase given by the integral can be evaluated as

$$\begin{aligned} \int_0^{T_f} m(t)dt &= \frac{1}{\epsilon_f} \sqrt{1 + 4\epsilon_f^2} \sqrt{w(1-w)} \int_0^1 \frac{ds_f}{\Delta_w^2} \\ &= \sqrt{1 + 4\epsilon_f^2} \frac{\phi_w}{\epsilon_f}, \end{aligned} \tag{56}$$

where, incidentally, the integral is the same one used to arrive at Eq. (36), with  $\phi_w$  given by Eq. (33). Finally, we can evaluate the success probability  $P(w; \epsilon_f, w) = |\langle \xi(0) | \xi(T_f) \rangle|^2$ . We let the integral in Eq. (56) be denoted by  $p$  and evaluate

$$\begin{aligned} P(w; \epsilon_f, w) &= |\cos^2(\chi/2) + \exp(-ip) \sin^2(\chi/2)|^2 \\ &= 1 - \sin^2(\chi) \sin^2(p/2). \end{aligned} \tag{57}$$

Thus  $\delta(w; \epsilon_f, w) = \sin(\chi) |\sin(p/2)|$  where  $\sin(\chi) = 2\epsilon_f / (1 + 4\epsilon_f^2)^{1/2}$ , so the theorem has been proved.

**V. SIMULATION IN THE GATE MODEL**

What is the relationship between the standard-schedule adiabatic fixed-point algorithm and the gate-model fixed-point algorithms from Ref. [7]? They bear little resemblance beyond the fixed-point property. The evolution of the state in the adiabatic case is smooth and stays nearly in the  $xz$  plane of the Bloch sphere throughout the evolution, while the gate-model algorithm subjects the state to discrete rotations through large angles causing it to jump around all parts of the Bloch sphere. It seems unlikely that this latter sort of evolution could be exhibited by any adiabatic algorithm, but perhaps the former could be exhibited by a gate-model algorithm. Fortunately, simulating the adiabatic search algorithm in the gate model is simply done using a Trotter formula, as we explain in this section. We adapt our geometric analysis to show that the resulting gate-model algorithm can be made to have the fixed-point property, and thus act as an alternative to the algorithm in Ref. [7] with evolution more closely resembling that of the adiabatic search algorithm itself.

We will do this simulation in two steps: first, we will discretize the continuous Hamiltonian  $H(t)$ , so it is constant

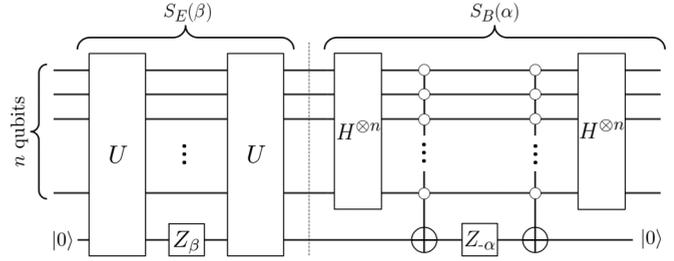


FIG. 4. Circuit which simulates the discretized Hamiltonian  $H_d$  over one time interval of width  $\delta t$ , performing the unitary operation  $U_t^{(j)}$ . The gate  $Z_\zeta$  applies the operation  $\exp(-i\zeta Z/2)$ . Also,  $H^{\otimes n}$  represents a Hadamard gate applied to each qubit, and the multiply-controlled NOT gate triggers upon the state  $|0\rangle^{\otimes n}$ .

over small time intervals  $\delta t$ ; then we will simulate the discrete Hamiltonian using a Trotter formula. Suppose we divide the total time  $T$  into  $l$  intervals of width  $\delta t = T/l$ . Then we let  $t_j = j \delta t$ , and  $s_j = s(t_j)$  for some schedule  $s(t)$ . The discretized Hamiltonian is

$$H_d(t) = H(\delta t \lfloor t/\delta t \rfloor). \tag{58}$$

Thus  $H_d$  agrees with  $H$  whenever  $t = t_j$  for some  $j$ , and  $H_d$  is constant between  $t_j$  and  $t_{j+1}$  for all  $j$ .

For a given interval  $j$ , the effect of applying the Hamiltonian  $H_d$  is the unitary operator

$$U_d^{(j)} = \exp(-i H(t_j) \delta t) \tag{59}$$

$$= \exp(-i (1 - s_j) H_0 \delta t - i s_j H_1 \delta t). \tag{60}$$

If  $\delta t$  is small, then we can use a Trotter formula to approximate this unitary as a product of two unitaries:

$$U_t^{(j)} = \exp(-i (1 - s_j) H_0 \delta t) \exp(-i s_j H_1 \delta t). \tag{61}$$

It can be shown that  $U_d^{(j)}(t) = U_t^{(j)}(t) + O(\delta t^2)$  [29]. The benefit of  $U_t$  is that it is a product of two unitaries which are easy to implement in a circuit. Exponentiating  $H_0$  is equivalent to a rotation about the begin state in the Bloch sphere framework we introduced above. Likewise, exponentiating  $H_1$  is a rotation about the end state. Specifically, we have

$$U_t^{(j)} = S_B(\alpha_j) S_E(\beta_j) \tag{62}$$

up to an overall phase, where

$$\begin{aligned} S_B(\alpha) &= I - (1 - e^{-i\alpha})|B\rangle\langle B| \\ &= \exp(-i\alpha) \exp(i\alpha H_0), \end{aligned} \tag{63}$$

$$\begin{aligned} S_E(\beta) &= I - (1 - e^{+i\beta})|E\rangle\langle E| \\ &= \exp(i\beta) \exp(-i\beta H_1), \end{aligned} \tag{64}$$

and  $\alpha_j = -(1 - s_j)\delta t$  and  $\beta_j = s_j\delta t$ . The circuit that implements  $U_t^{(j)}$  is the same one discussed in Ref. [7] and is shown in Fig. 4. It makes use of the discrete oracle  $U$ , which flips an ancilla bit when fed a target state:  $U|x\rangle|0\rangle = |x\rangle|1\rangle$  if  $|x\rangle$  is a target state, and otherwise acts as the identity. As an example, suppose we are using the standard schedule  $s_s(t; \epsilon_s, w)$  given by Eq. (37), then, recalling that we have split  $T_s$  into  $l$  equal

time steps of size  $\delta t$ , we can say that

$$\alpha_j = -\frac{\delta t}{2} - \frac{\delta t}{2} \sqrt{\frac{w}{1-w}} \tan\left(\left(1 - \frac{2j}{l}\right)\phi_w\right), \quad (65)$$

$$\beta_j = \frac{\delta t}{2} - \frac{\delta t}{2} \sqrt{\frac{w}{1-w}} \tan\left(\left(1 - \frac{2j}{l}\right)\phi_w\right). \quad (66)$$

Now we can formally define the gate-model algorithm as follows.

*Simulated Adiabatic Search Algorithm.*

*Inputs.* Schedule  $s(t)$  [where  $s(0) = 0$ ,  $s(T) = 1$ ],  $\delta t$ .

*Procedure.* Prepare the initial state  $|\psi_0\rangle$  to be  $|B\rangle$ . Apply circuit  $U_t^{(j)}$  to  $|\psi_j\rangle$  to yield  $|\psi_{j+1}\rangle$  for  $j = 0, 1, 2, \dots, l-1$ , where  $l = \lfloor T/\delta t \rfloor$ . Measure the system in the computational basis.

*Output.* The postmeasurement state is a target state with probability  $P(\lambda) \equiv |\langle \psi_l | E \rangle|^2$ , where  $\lambda$  is the fraction of target states.

We claim that the simulated adiabatic search algorithm, using the standard schedule  $s_s(t; \epsilon_s, w)$  given by Eq. (37), is also fixed point with Grover-like scaling. We prove this by bounding the error in the simulated adiabatic search algorithm for general schedule  $s(t)$  and then evaluating this bound for the standard schedule [recall  $\delta(\lambda) = \sqrt{1 - P(\lambda)}$ ].

*Theorem 4.* If the simulated adiabatic search algorithm is run according to schedule  $s(t)$  and step size  $\delta t$ , then

$$\delta(\lambda) < 3.1\sqrt{\delta t} + (d_0 + d_1)(1 + \delta t^2/25), \quad (67)$$

where  $d_0$  and  $d_1$  are functions of  $\lambda$  given by Eqs. (25) and (26) from Theorem 1.

Theorem 4 is proved in Appendix B. As  $\delta t$  approaches zero, this bound approaches the bound in Theorem 1. If  $s(t)$  is the standard schedule given by Eq. (37), then Theorem 3 says that  $d_0 + d_1 \leq 2\epsilon_s$ , so for the simulated adiabatic search algorithm with standard schedule,  $P(\lambda) \geq 1 - \delta^2$  with  $\delta = 3.1\sqrt{\delta t} + 2\epsilon_s(1 + \delta t^2/25)$ . The number of oracle queries associated with this sequence of length  $l$  is  $L = 2l + 1$  and  $l = \lfloor T/\delta t \rfloor$ . Thus we can express the query complexity in terms of the desired values for  $w$  and  $\delta$ :

$$L = 1 + 2\lfloor T_s/\delta t \rfloor = 1 + \frac{2\phi_w}{\epsilon \delta t \sqrt{w(1-w)}} = O\left(\frac{1}{\delta^3 \sqrt{w}}\right). \quad (68)$$

Thus Theorem 4 shows that the simulated adiabatic search algorithm with standard schedule is a fixed-point algorithm which shows Grover's quadratic speedup. This gives an alternative sequence of  $\alpha$  and  $\beta$  angles [see Eqs. (65) and (66)] that can be used in place of the fixed-point sequence given in Ref. [7]. It is known that adiabatic computation can be simulated with polynomial overhead [30], but this algorithm shows that the adiabatic search algorithm can be simulated while maintaining the same scaling in  $w$ .

We briefly mention an overview of how Theorem 4 is proved. One approach would be to bound the error introduced by discretization (the difference between applying the true Hamiltonian  $H$  and the discrete Hamiltonian  $H_d$ ), and then bound the error induced by Trotterization (the difference

between applying  $U_d^{(j)}$  and  $U_t^{(j)}$ ). We did not find this approach successful for the following reason: suppose the algorithm is run for time  $T$  with  $l$  time steps, so  $\delta t = T/l$ . The Trotterization error is  $O(\delta t^2)$  per time step, making the total error  $O(l\delta t^2) = O(T\delta t)$ . Since  $T = O(1/\sqrt{\lambda})$  and the error must be  $O(1)$ ,  $\delta t = O(\sqrt{\lambda})$ , and thus the number of time steps  $l = O(1/\lambda)$ , which does not have Grover-like scaling. The discretization error is not problematic in this way.

Instead, our proof finds an alternate continuous Hamiltonian  $H_t$  which, upon discretization, generates the exact evolution operations  $U_t^{(j)}$  applied in the simulated adiabatic search algorithm and given by Eq. (61). This Hamiltonian  $H_t$ , which is approximately equal to  $H$ , follows a slightly modified but still continuous path from the begin state  $|B\rangle$  to the end state  $|E\rangle$ . In particular, this path leaves the  $xz$  plane. We use the same strategy as in the proof from Theorem 1 to bound the error induced by the continuous algorithm defined by application of  $H_t$ . This error *does* have Grover-like scaling. Then, we discretize  $H_t$ , giving  $H_{td}$ , and bound the error induced by discretization.

While the first approach views Trotterization as an error-inducing step and tries to find an upper bound on this error, the second approach capitalizes on the fact that the Trotterization error at each time step is correlated in a way that can be interpreted as simply an alternate path from  $|B\rangle$  to  $|E\rangle$  on the Bloch sphere and hence should also be a valid adiabatic algorithm. As  $T$  gets longer with constant  $\delta t$ , the first approach finds that the error accrues while the second approach does not. The alternate path is a function only of  $\delta t$  and not of the schedule  $s(t)$  (which defines how fast the path is traversed) and hence a longer run time does not result in more error due to Trotterization under this approach.

## VI. APPLICATIONS OF THE FIXED-POINT PROPERTY

In this section, we elaborate on two direct applications of fixed-point algorithms: preparation of the relatively prime state and oblivious amplitude amplification. Other applications are mentioned briefly in the Introduction.

### A. Preparation of the relatively prime state

For a specific example of a problem which might benefit from a fixed-point algorithm, we look to number theory. Given an integer  $J$ , we wish to prepare the relatively prime state, the superposition over states which are relatively prime to  $J$ :

$$|\Phi\rangle = \frac{1}{\sqrt{\phi(J)}} \sum_{\gcd(x, J)=1} |x\rangle, \quad (69)$$

where  $\phi$  is the Euler totient function. That is,  $\phi(J)$  is the number of integers relatively prime to  $J$ . This can be done by beginning in the equal superposition  $|B\rangle$ , running the classical Euclidean algorithm (which is efficient) for calculation of the greatest common divisor (gcd), while in superposition, and copying the result to an ancilla register

$$|B\rangle|0\rangle \rightarrow \sum_{x=0}^{J-1} |x\rangle|\gcd(x, J)\rangle. \quad (70)$$

Then, measurement of the ancilla register yields outcome 1 and the state  $|\Phi\rangle$  with probability  $\phi(J)/J$ . The process is repeated until a 1 outcome is obtained. The algorithm is efficient because there is a lower bound [31] on  $\phi(J)/J$ ,

$$\frac{\phi(J)}{J} > \frac{1}{e^\gamma \log \log J + \frac{3}{\log \log J}}, \quad (71)$$

where  $\gamma \approx 0.577$  is the Euler-Mascheroni constant. so the state  $|\Phi\rangle$  will be produced with high probability after only  $O(\log \log J)$  repetitions of the Euclidean algorithm.

We can reformulate the algorithm as a search algorithm, with  $\lambda = \phi(J)/J$  and the Euclidean algorithm playing the role of the oracle. In the gate model, the oracle consists of doing the Euclidean algorithm and flipping an ancilla bit if the gcd is 1. In the adiabatic model, the oracle is a Hamiltonian  $H_J$  such that  $H_J|x\rangle = 0$  if  $\gcd(x, J) = 1$  and  $H_J|x\rangle = |x\rangle$  otherwise. Thus the ground state of  $H_J$  is  $|\Phi\rangle$ .

Running a fixed-point search algorithm with lower bound parameter  $w$  given by Eq. (71) will prepare the superposition of target states  $|\Phi\rangle$  with high fidelity using only  $O(\sqrt{\log \log J})$  queries to the Euclidean algorithm (or total run time in the adiabatic model). This is a quadratic speedup over the algorithm in Ref. [32] which does not make use of a quantum search algorithm. On an adiabatic computer, this would entail running the adiabatic search algorithm using the standard schedule. Moreover, the fixed-point algorithm allows the state  $|\Phi\rangle$  to be prepared to arbitrary fidelity without the need for measurement. Crucially, we only need to know the lower bound in Eq. (71) for  $\lambda = \phi(J)/J$  for the algorithm to work.

### B. Fixed-point oblivious amplitude amplification

Oblivious amplitude amplification [11] is a technique for implementing, on a state  $|\psi\rangle$ , a desired unitary  $V$  that we cannot construct directly with a larger unitary  $U$  that we can construct, at least when given some ancilla qubits. The ‘‘oblivious’’ in the name comes from the fact that amplitude amplification still works despite not having the full ability to reflect about the initial state  $|\psi\rangle$ . Thus this procedure is particularly useful when the initial state  $|\psi\rangle$  is not only unknown but also not renewable—we have only one copy and cannot or would prefer not to make another. For instance, this is the case in Hamiltonian simulation [11].

For convenience, we consider  $|\psi\rangle$  to be an  $n$ -qubit state and the extension to Hilbert space to be  $m$ -qubits. Then, we write the start state as [11]

$$U|0\rangle^{\otimes m}|\psi\rangle = \sqrt{\lambda}|0\rangle^{\otimes m}V|\psi\rangle + \sqrt{1-\lambda}|\Phi^\perp\rangle. \quad (72)$$

The target state is  $|\Phi\rangle = |0\rangle^{\otimes m}V|\psi\rangle$ . We note that for oblivious amplitude amplification to work, it is required of  $U$  that  $\lambda$  in Eq. (72) be independent of  $|\psi\rangle$ . Given this setup, the question is, how many uses of  $U$  do we need to make  $|\Phi\rangle$  with probability  $1 - \delta^2$ ?

The naive classical attempt would measure the first  $m$ -qubits, succeeding with probability  $\lambda$  in getting  $|\Phi\rangle$ . However, upon failure the state will generally be destroyed, and given only one copy of  $|\psi\rangle$ , we will not be able to retry except

perhaps in special cases when  $|\psi\rangle$  can be recovered from the remains after measurement.

In Ref. [11] Berry *et al.* give an exact quantum algorithm for oblivious amplitude amplification in the case of known  $\lambda$  using the expected number  $O(1/\sqrt{\lambda})$  of uses of  $U$ . They argue that, if  $\Pi = (|0\rangle\langle 0|)^{\otimes m} \otimes I$ , the operations  $Ue^{i\pi\Pi}U^\dagger$  and  $e^{i\pi\Pi}$  act as the reflection about the start and reflection about the target states, respectively. Thus Grover iterates constructed from these reflections act as expected on the two-dimensional span of  $|\Phi\rangle$  and  $|\Phi^\perp\rangle$ .

However, it is not immediately clear how to extend oblivious amplitude amplification to when  $\lambda$  is unknown. Standard Grover algorithms in the unknown- $\lambda$  setting use a set of sequences of Grover iterates that increase exponentially in length, with a measurement after each to determine whether the target state has been found [3]. Each such measurement (now performed on the  $m$ -qubit ancilla) will, like in the classical approach, destroy the state if it fails. Moreover, at least one measurement is likely to fail during the process of ramping up the sequence lengths.

A workable approach is to conduct a coherent version of this exponential step-up in sequence length. For the problem of standard Grover search, this is essentially the idea from the last section of Ref. [33], where amplitude estimation is used. With great enough precision (a lower bound  $w \leq \lambda$  is required here to ensure enough precision), the estimation will, with high probability, yield a state that has overlap  $1/2$  with the target state. Exact quantum search can from there complete the algorithm. Applying this approach to oblivious amplitude amplification, we get an algorithm that succeeds with probability  $1 - \delta^2$  using  $O(1/(\delta\sqrt{w}))$  applications of  $U$ . Yet, not only is the scaling with  $\delta$  poor, but there is also not a clear adiabatic version of this approach.

However, fixed-point search can also solve the problem of oblivious amplitude amplification with unknown  $\lambda$  (and a known lower bound  $w \leq \lambda$ ). If the optimal fixed-point algorithm in Ref. [7] is used this will take just  $O(\log(1/\delta)/\sqrt{w})$  uses of  $U$ . To see that this works is to realize that  $Ue^{-i\alpha\Pi}U^\dagger$  and  $e^{i\beta\Pi}$  implement partial reflections about the start and target states, respectively, following exactly the same reasoning as in Ref. [11]. Since fixed-point search in the gate model is just unitary application of these partial reflections, it will produce the same amplification profile as a function of  $\lambda$  as it does in the nonoblivious case. As an aside, a procedure using the digital-filter engineering that is also behind the fixed-point search [7] has recently improved the Hamiltonian simulation from Ref. [11] by bypassing oblivious amplitude amplification entirely [34].

An adiabatic oblivious search algorithm is also just as straightforward as the nonoblivious version, except with a modified Hamiltonian. One should construct  $H = (1 - s(t))H_B + s(t)H_E$ , where now  $H_E = \Pi$  and  $H_B = U\Pi U^\dagger$  (so that  $e^{-iH_B} = Ue^{-i\Pi}U^\dagger$ ). Assuming one can do this, the adiabatic search schedules  $s(t)$  we have presented can be applied to the oblivious case without change. The modified  $H_B$  may even be easier to implement than a full reflection about the initial state as you would find in a nonoblivious adiabatic search, as it can limit coupling between the  $n$ -qubit and  $m$ -qubit subsystems to just the bare essential coupling represented by  $U$ .

## VII. CONCLUSION

Inspired by the fixed-point algorithm in Ref. [7], we wondered whether the search algorithm in the adiabatic model could be made to have the fixed-point property. We found that the landscape of search algorithms in the adiabatic model is similar to that in the circuit model: some algorithms are fixed point but lack Grover-like scaling, some have Grover-like scaling but are not fixed point, and some have both properties. We have given examples for each of these cases. Moreover, we showed that simulation of a fixed-point adiabatic algorithm with Grover-like scaling can yield a fixed-point gate-model algorithm which retains the quantum speedup and acts as an alternative to the algorithms in Refs. [6,7].

Fixed-point search algorithms provide fault-tolerance to certain systematic errors and misestimations of  $\lambda$ , and also find applications in certain problems with natural lower bounds, like preparation of the relatively prime state, and even in counterfeiting quantum money [6].

That fixed-point Grover-like algorithms exist in the adiabatic model extends these benefits to AQC but also gives us continuous control over the parameters of the algorithm  $w$  and  $\epsilon$  which we did not have in the gate-model. This continuous control might be useful to create a scheme which estimates the value of  $\lambda$  by binary searching the interval  $[0, 1]$  using the adiabatic search algorithm with different parameters, or different schedule families altogether. Such a scheme would provide a novel way to perform quantum counting [2], but also, potentially, a practical method for calibrating adiabatic quantum computers.

Direct applications aside, the method by which we proved our results is an equally important takeaway from this work. The technique was motivated by the intuition that a two-dimensional system exposed to a Hamiltonian is equivalent to a spin immersed in a magnetic field and can be treated geometrically on the Bloch sphere. In this picture, the state vector precesses around the Hamiltonian vector, and by making a series of time-dependent coordinate changes we demonstrated an upper bound on the angle between the state and Hamiltonian vectors at the end of the algorithm. For the fast schedule when  $\lambda = w$ , this method actually allowed for the Schrödinger evolution to be solved exactly.

This approach contrasts with other rigorous methods of dealing with the adiabatic approximation, such as the asymptotic expansion of the error. Our bound is explicit and useful even in situations where the parameters are finite, giving us more than just asymptotic information about the algorithm. The ability to compute the error exactly (Theorem 2), albeit in a special case, is a unique result for AQC.

Further work might explore other families of schedules to achieve better dependence of  $T$  on the error amplitude  $\delta$ , modify our geometric method to also find a lower bound on the error in the adiabatic approximation, or extend our ideas to bound the error in the adiabatic approximation in more than two dimensions.

## ACKNOWLEDGMENTS

We thank Guang Hao Low for interesting discussions and perspective especially concerning Hamiltonian simulation.

A.M.D. acknowledges the Paul E. Gray UROP fund. T.J.Y. thanks the NDSEG fellowship program. The authors also acknowledge funding by the NSF CUA, and NSF RQCC Project No. 1111337.

## APPENDIX A: PROOF OF THEOREM 3

Theorem 3 follows from application of Theorem 1 to the standard schedule  $s_s(t; \epsilon_s, w)$  given by Eq. (37).

Since  $\dot{s}_s(t) = \epsilon_s \Delta_w^2$ , and  $\Delta_w(0) = 1$ , we evaluate  $d_0$  and  $d_1$  from Theorem 1 as

$$d_0 = 2\epsilon_s \sqrt{\lambda(1-\lambda)}, \quad (\text{A1})$$

$$d_1 = \int_0^{T_s} dt \left| \frac{d}{dt} \left( \frac{\epsilon_s \Delta_w^2 \sqrt{\lambda(1-\lambda)}}{\Delta_\lambda^3} \right) \right| \quad (\text{A2})$$

$$= \int_0^1 ds_s \left| \frac{d}{ds_s} \left( \frac{\epsilon_s \Delta_w^2 \sqrt{\lambda(1-\lambda)}}{\Delta_\lambda^3} \right) \right|. \quad (\text{A3})$$

In the integral for  $d_1$ , we note that if not for the absolute value, we would be integrating a derivative and the answer would simply be the net change in the value of  $c(s_s) := \epsilon_s \Delta_w^2 \sqrt{\lambda(1-\lambda)} / \Delta_\lambda^3$  from  $s_s = 0$  to  $s_s = 1$  [note that  $2c = \tan(\chi)$  with  $\chi$  from Eq. (45)]. The absolute value bars mean that we must count both positive and negative changes as positive, so we can evaluate the integral by observing intervals of  $s_s$  over which  $c(s_s)$  is monotonic, and summing the absolute value of the change of  $c(s_s)$  over each of these regions. Now we analyze over what regions this  $c(s_s)$  is monotonic.

First, let

$$c_0 = \epsilon_s \sqrt{\lambda(1-\lambda)}, \quad (\text{A4})$$

$$c_{1/2} = \frac{\epsilon_s w \sqrt{1-\lambda}}{\lambda}, \quad (\text{A5})$$

$$c_{\text{crit}} = \frac{2\epsilon_s \sqrt{\lambda}(1-w)^{3/2}}{3\sqrt{3}(1-\lambda)(\lambda-w)}, \quad (\text{A6})$$

$$s_{\text{crit}} = \frac{1}{2} - \frac{1}{2} \sqrt{\frac{2\lambda - 3w + \lambda w}{(1-\lambda)(1-w)}}. \quad (\text{A7})$$

Note that  $d_0 = 2c_0$ . There are three cases.

(i) *Case I.*  $\lambda \leq 3w/(2+w)$ . There is one local extremum:  $c(s_s)$  increases from  $c_0$  at  $s_s = 0$  to  $c_{1/2}$  at  $s_s = 1/2$  and back to  $c_0$  at  $s_s = 1$ ;  $d_0 + d_1 = 2c_{1/2}$ .

(ii) *Case II.*  $3w/(2+w) < \lambda < (1+2w)/3$ . There are three local extrema:  $c(s_s)$  increases from  $c_0$  at  $s_s = 0$  to  $c_{\text{crit}}$  at  $s_s = s_{\text{crit}}$ , decreases to  $c_{1/2}$  at  $s_s = 1/2$ , then increases back to  $c_{\text{crit}}$  at  $s_s = 1 - s_{\text{crit}}$  and finally decreases to  $c_0$  at  $s_s = 1$ ;  $d_0 + d_1 = 4c_{\text{crit}} - 2c_{1/2}$ .

(iii) *Case III.*  $\lambda \geq (1+2w)/3$ . There is one local extremum:  $c(s_s)$  decreases from  $c_0$  to  $c_{1/2}$  at  $s_s = 1/2$  then increases back to  $c_0$  at  $s_s = 1$ ;  $d_0 + d_1 = 4c_0 - 2c_{1/2}$ .

Thus we have explicitly calculated  $d_0 + d_1$  as a piecewise function of  $\lambda$  (supposing constant  $w$  and  $\epsilon_s$ ). We would like to show that in each of the three cases, we have  $d_0 + d_1 \leq 2\epsilon_s$ .

We can see this immediately in Case I, because  $\lambda \geq w$  implies  $c_{1/2} \leq \epsilon_s$  so  $d_0 + d_1 \leq 2\epsilon_s$ . Similarly for Case III, since  $c_0 \leq \epsilon_s/2$  for all  $\lambda$  we know that  $d_0 + d_1 \leq 4c_0 \leq 2\epsilon_s$ . Case II is difficult to analyze analytically, but the fact that

$d_0 + d_1 \leq 2\epsilon_s$  for all  $w$  and all  $\lambda \in [3w(2+w), (1+2w)/3]$  can be verified graphically. This proves the theorem. ■

## APPENDIX B: PROOF OF THEOREM 4

Since the maximum error amplitude  $\delta$  is at most 1, the theorem is trivially true when  $\delta t \geq 2\pi$ . Thus we assume  $\delta t < 2\pi$  and treat  $\delta t$  as a fixed input parameter. We define  $H_t(s)$  as the solution to the equation

$$\exp(-i(1-s)H_0\delta t)\exp(-isH_1\delta t) = \exp(-iH_t\delta t). \quad (\text{B1})$$

To solve this equation, first we assume that  $H_t$  has the form

$$H_t = \frac{1}{2}I - \frac{\gamma}{2}\hat{n}_t \cdot \vec{\sigma}, \quad (\text{B2})$$

where  $\hat{n}_t$  is a unit vector.

Then, we write  $H_0$  and  $H_1$  using Eqs. (8) and (9), expand both sides as a linear combination of  $\{I, X, Y, Z\}$  using  $\exp(i\theta\hat{u} \cdot \vec{\sigma}) = \cos(\theta)I + i\sin(\theta)\hat{u} \cdot \vec{\sigma}$  (where  $\hat{u}$  is a unit vector) and equate coefficients, arriving at

$$\gamma = \frac{2}{\delta t} \arccos(\cos(a-b) - 2\lambda \sin(a)\sin(b)), \quad (\text{B3})$$

$$\hat{n}_t \cdot \hat{x} = \sin\left(\frac{\gamma\delta t}{2}\right)^{-1} (\sqrt{\lambda} \sin(a+b)), \quad (\text{B4})$$

$$\hat{n}_t \cdot \hat{y} = \sin\left(\frac{\gamma\delta t}{2}\right)^{-1} (-2\sqrt{\lambda(1-\lambda)} \sin(a)\sin(b)), \quad (\text{B5})$$

$$\hat{n}_t \cdot \hat{z} = \sin\left(\frac{\gamma\delta t}{2}\right)^{-1} (\sqrt{1-\lambda} \sin(a-b)), \quad (\text{B6})$$

where  $a = (1-s)\delta t/2$  and  $b = s\delta t/2$ .

As expected, in the limit as  $\delta t$  approaches zero, we can see that  $\hat{n}_t$  approaches  $\hat{n}$  and  $\gamma$  approaches  $\Delta_\lambda$ . For finite  $\delta t$ ,  $\hat{n}$  and  $\hat{n}_t$  take approximately identical though slightly different paths. In particular,  $\hat{n}$  stays in the  $xz$  plane for all  $s$ , while  $\hat{n}_t$  has a small component in the  $\hat{y}$  direction.

We also discretize  $H_t$  to form

$$H_{td}(t) = H_t(\lfloor t/\delta t \rfloor \delta t), \quad (\text{B7})$$

and we can see that the unitary evolution operator corresponding to application of  $H_{td}$  between times  $t_j$  and  $t_{j+1}$  is precisely  $U_t^{(j)}$  from Eq. (61).

Furthermore, we let  $|\psi_t(t)\rangle$  and  $|\psi_{td}(t)\rangle$  be states which evolve by Hamiltonians  $H_t$  and  $H_{td}$ , respectively. So,  $|\psi_{td}(T)\rangle = |\psi_t\rangle$ , the final state of the simulated adiabatic search algorithm in the gate model, and hence  $P(\lambda) = |\langle\psi_{td}(T)|E\rangle|^2$ . We bound this probability by first showing that  $|\langle\psi_t(T)|E\rangle|^2$  is close to 1, and then bounding the distance between  $|\psi_t(T)\rangle$  and  $|\psi_{td}(T)\rangle$ .

*Claim 1.*  $|\langle\psi_t(T)|E\rangle|^2 \geq 1 - (\sqrt{2.6\delta t} + (d_0 + d_1)(1 + \delta t^2/25))^2$  if  $\lambda \geq w$ .

*Claim 2.*  $|\langle\psi_{td}(T)|\psi_t(T)\rangle|^2 \geq 1 - 2\delta t$ .

*Lemma 3.* If  $|\langle Q|S\rangle|^2 \geq 1 - e_1^2$ , and  $|\langle R|S\rangle|^2 \geq 1 - e_2^2$ , then  $|\langle Q|R\rangle|^2 \geq 1 - (e_1 + e_2)^2$  for any normalized  $|Q\rangle, |R\rangle, |S\rangle$ .

We delay the proof of Lemma 3 until the end of the section. Since  $\sqrt{2} + \sqrt{2.6} < 3.1$ , Claims 1 and 2 combined with Lemma 3 are sufficient to prove Theorem 4. Now we explain why each claim is true.

*Proof of Claim 1.* Claim 1 is an upper bound on the error resulting from running the continuous search algorithm with Hamiltonian  $H_t$  instead of  $H$ . The claim highly resembles Theorem 1, and a very similar technique is used in the following proof.

As in Theorem 1, we make a time-dependent change of coordinates via a time-dependent unitary transformation so that the Hamiltonian vector is rotated back to the  $\hat{z}$  axis. In this case,  $\hat{n}_t$  has components in all three directions, so constructing this unitary is more complicated. We express this transformation as a product of three unitaries.

$$|\phi_t(t)\rangle = U_3(t)U_2(t)U_1(t)|\psi_t(t)\rangle, \quad (\text{B8})$$

with  $U_1 = \exp(i\frac{\eta_1}{2}\hat{n} \cdot \vec{\sigma})$ ,  $U_2 = \exp(-i\frac{\eta_2}{2}(\hat{n} \times \hat{y}) \cdot \vec{\sigma})$ , and  $U_3 = \exp(i\frac{\theta}{2}Y)$ .  $U_1$  is a rotation about  $\hat{n}$  by an angle  $\eta_1$ ;  $U_2$  is a rotation about the vector perpendicular to  $\hat{n}$  in the  $xz$  plane by an angle of  $\eta_2$ . We will choose  $\eta_1$  and  $\eta_2$  so these two operations alone rotate the vector  $\hat{n}_t$  onto the vector  $\hat{n}$ . Then,  $U_3$  is the same unitary we used to prove Theorem 1 that rotates  $\hat{n}$  back to  $\hat{z}$ . We find the rotation angles we need are

$$\eta_1 = \arctan\left(\frac{(\hat{n}_t \cdot \hat{x})\cos(\theta) - (\hat{n}_t \cdot \hat{z})\sin(\theta)}{\hat{n}_t \cdot \hat{y}}\right), \quad (\text{B9})$$

$$\eta_2 = \arccos((\hat{n}_t \cdot \hat{x})\sin(\theta) + (\hat{n}_t \cdot \hat{z})\cos(\theta)), \quad (\text{B10})$$

where the components of  $\hat{n}_t$  are given by Eqs. (B4)–(B6). Recall from Eq. (16) that  $\cos(\theta) = (1-2s)\sqrt{1-\lambda}/\Delta_\lambda = \hat{n} \cdot \hat{z}$  and  $\sin(\theta) = \sqrt{\lambda}/\Delta_\lambda = \hat{n} \cdot \hat{x}$ . Lemma 4 verifies that this transformation works.

*Lemma 4.*

$$(U_3U_2U_1)H_t(U_3U_2U_1)^\dagger = \frac{I}{2} - \frac{\gamma}{2}Z. \quad (\text{B11})$$

Moreover, when  $s = 1$  we have  $\gamma = 1$ ,  $|\hat{n}\rangle = |E\rangle$ , and  $\eta_2 = 0$ , so up to a global phase  $U_2U_1|E\rangle = |E\rangle$  and  $U_3U_2U_1|E\rangle = |0\rangle$ . Hence we can rewrite  $|\langle\psi_t(T)|E\rangle|^2 = |\langle 0|\phi_t(T)\rangle|^2$ . Now we calculate the effective Hamiltonian by which the state  $|\phi_t(t)\rangle$  evolves. Note that  $\|O\|$  denotes the maximum norm of  $O|\alpha\rangle$  over all normalized states  $|\alpha\rangle$ .

*Lemma 5.*  $|\phi_t(t)\rangle$  evolves according to  $i\frac{d}{dt}|\phi_t(t)\rangle = H_{\phi t}|\phi_t(t)\rangle$ , where

$$H_{\phi t} = \frac{I}{2} - \frac{\gamma}{2}Z - \frac{\dot{\theta}}{2}Y + E \quad (\text{B12})$$

and  $\int_0^T \|E\|dt \leq 1.3\delta t$ .

Since  $\gamma$  is approximately  $\Delta_\lambda$ ,  $H_{\phi t}$  is approximately  $H_\phi$  from Eq. (42) plus a bounded error term  $E$ . We write  $H_{\phi t} = H_{\phi t0} + E$ . If  $|\phi_{t0}\rangle$  evolves according to  $H_{\phi t0}$ , we wish to quantify how far away  $|\phi_{t0}\rangle$  is from  $|\phi_t\rangle$  at time  $T$ , which is facilitated by Lemma 6.

*Lemma 6.* If  $|\Psi_A(t)\rangle$  evolves by Hamiltonian  $H_A$ , while  $|\Psi_B(t)\rangle$  evolves by Hamiltonian  $H_B$ , and  $|\Psi_A(0)\rangle = |\Psi_B(0)\rangle$ , then  $|\langle\Psi_A(T)|\Psi_B(T)\rangle| \geq 1 - \int_0^T \|H_A - H_B\|dt$ .

Lemmas 5 and 6 tell us that

$$|\langle\phi_{t0}(T)|\phi_t(T)\rangle|^2 \geq 1 - 2.6\delta t. \quad (\text{B13})$$

*Lemma 7.*  $|\langle\phi_{t0}(T)|0\rangle|^2 \geq 1 - ((d_0 + d_1)(1 + \delta t^2/25))^2$  if  $\lambda \geq w$ , where  $d_0$  and  $d_1$  are given by Eqs. (25) and (26) in Theorem 1.

Compare this result to the conclusion of Theorem 1, that  $|\langle \phi(T)|0 \rangle|^2 \geq 1 - (d_0 + d_1)^2$ . We have now bounded in Eq. (B13) how close  $|\phi_{t_0}(T)\rangle$  and  $|\phi_t(T)\rangle$  are and, in Lemma 7, how close  $|\phi_{t_0}(T)\rangle$  and  $|0\rangle$  are. We can again use Lemma 3 to conclude that

$$|\langle \phi_t(T)|0 \rangle|^2 \geq 1 - (\sqrt{2.6\delta t} + (d_0 + d_1)(1 + \delta t^2/25))^2 \quad (\text{B14})$$

whenever  $\lambda \geq w$ . This proves Claim 1. We have shown that if the state is exposed to  $H_t$ , instead of  $H$ , we still have a search algorithm with bounded error probability that is related only to  $\delta t$  and the schedule  $s(t)$ . If we relax the fixed-point definition to allow algorithms whose error can be bounded by functions of both  $\epsilon$  and  $\delta t$  (but not  $w$ ), then applying  $H_t(s(t); \epsilon, w)$  would be a fixed-point algorithm for any schedule for which applying  $H(s(t); \epsilon, w)$  is a fixed-point algorithm. Claim 2 illustrates how the additional error incurred by using the discrete Hamiltonian  $H_{td}$  instead of  $H_t$  can be bounded as well. ■

*Proof of Claim 2.* First we examine the definition of  $H_t$  to bound  $\|dH_t/ds\|$ . We differentiate both sides of Eq. (B1) and cancel factors arriving at

$$\begin{aligned} \frac{dH_t}{ds} &= -H_0 + e^{-i(1-s)H_0\delta t} H_1 e^{i(1-s)H_0\delta t} \\ &= -\hat{n}_0 \cdot \vec{\sigma} / 2 + e^{-i(1-s)H_0\delta t} (\hat{n}_1 \cdot \vec{\sigma}) e^{i(1-s)H_0\delta t} / 2, \\ \left\| \frac{dH_t}{ds} \right\| &\leq \|\hat{n}_0 \cdot \vec{\sigma}\| / 2 + \|\hat{n}_1 \cdot \vec{\sigma}\| / 2 = 1. \end{aligned} \quad (\text{B15})$$

We define  $\delta H_t = H_t - H_{td}$ .  $\delta H_t(t_j) = 0$  for all  $j$ , and if  $t_j \leq t < t_{j+1}$ , then

$$\|\delta H_t(t)\| = (t - t_j) \frac{d}{dt} \|\delta H_t(c)\| \leq \delta t \frac{d}{dt} \|\delta H_t(c)\| \quad (\text{B16})$$

for some  $c$  with  $t_j < c < t$ . But

$$\frac{d}{dt} \|\delta H_t(c)\| = \frac{ds}{dt} \frac{d}{ds} \|\delta H_t(c)\| \leq \frac{ds}{dt} \left\| \frac{d}{ds} \delta H_t(c) \right\| \leq \frac{ds}{dt}. \quad (\text{B17})$$

So, now we have

$$\int_0^T \|\delta H_t\| dt \leq \int_0^T \delta t \frac{ds}{dt} dt = \delta t. \quad (\text{B18})$$

Again, invoking Lemma 6, we arrive immediately at the Claim. ■

*Proof of Lemma 3.* Since all states are qubits, this follows from the triangle inequality over a sphere. If the angle between states  $|Q\rangle$  and  $|R\rangle$  is  $2v_{qr}$ , between  $|Q\rangle$  and  $|S\rangle$  is  $2v_{qs}$ , and between  $|R\rangle$  and  $|S\rangle$  is  $2v_{rs}$ , then we can say that  $|\langle Q|R \rangle|^2 = \cos^2(v_{qr})$ ,  $|\langle Q|S \rangle|^2 = \cos^2(v_{qs})$ , and  $|\langle R|S \rangle|^2 = \cos^2(v_{rs})$ . So  $e_1 = \sin(v_{qs})$  and  $e_2 = \sin(v_{rs})$ . The triangle inequality says that  $v_{ab} \leq v_{ac} + v_{bc}$ , and therefore

$$\begin{aligned} \sin(v_{qr}) &\leq \sin(v_{qs} + v_{rs}) \\ &\leq \sin(v_{qs}) \cos(v_{rs}) + \sin(v_{rs}) \cos(v_{qs}) \\ &\leq \sin(v_{qs}) + \sin(v_{rs}) \\ &\leq e_1 + e_2, \end{aligned}$$

$$|\langle Q|R \rangle|^2 = 1 - \sin^2(v_{qr}) \geq 1 - (e_1 + e_2)^2$$

and the Lemma is proved. ■

*Proof of Lemma 4.* We will rely on the identity

$$\exp(-i\alpha \hat{r} \cdot \vec{\sigma})(\vec{v} \cdot \vec{\sigma}) \exp(i\alpha \hat{r} \cdot \vec{\sigma}) = R_{\hat{r}}(2\alpha) \vec{v} \cdot \sigma, \quad (\text{B19})$$

where  $R_{\hat{r}}(2\alpha)$  represents the operator which rotates vectors about  $\hat{r}$  through an angle of  $2\alpha$ .  $H_t$  undergoes three subsequent rotations. However, it will be helpful to rearrange the rotations as follows:

$$\begin{aligned} U_3 U_2 U_1 &= (U_3 U_2 U_3^\dagger)(U_3 U_1 U_3^\dagger) U_3, \\ U_3 U_2 U_3^\dagger &= \exp\left(-i \frac{\eta_2}{2} U_3((\hat{n} \times \hat{y}) \cdot \vec{\sigma}) U_3^\dagger\right) \\ &= \exp\left(-i \frac{\eta_2}{2} X\right), \\ U_3 U_1 U_3^\dagger &= \exp\left(-i \frac{\eta_1}{2} U_3(\hat{n} \cdot \vec{\sigma}) U_3^\dagger\right) \\ &= \exp\left(-i \frac{\eta_1}{2} Z\right), \\ \therefore U_3 U_2 U_1 &= \exp\left(-i \frac{\eta_2}{2} X\right) \exp\left(-i \frac{\eta_1}{2} Z\right) \exp\left(i \frac{\theta}{2} Y\right). \end{aligned}$$

So, expanded this way, the first rotation is about the  $y$  axis. If the Hamiltonian starts in direction  $\hat{n}_t$ , it will finish in direction

$$\begin{aligned} \hat{r}_t &= ((\hat{n}_t \cdot \hat{x}) \cos(\theta) - (\hat{n}_t \cdot \hat{z}) \sin(\theta)) \hat{x} + (\hat{n}_t \cdot \hat{y}) \hat{y} \\ &\quad + ((\hat{n}_t \cdot \hat{x}) \sin(\theta) + (\hat{n}_t \cdot \hat{z}) \cos(\theta)) \hat{z}. \end{aligned}$$

Now, if we rotate  $\hat{r}_t$  about the  $z$  axis by an angle of  $\eta_1 = \arctan((\hat{r}_t \cdot \hat{x})/(\hat{r}_t \cdot \hat{y}))$ , we will preserve the  $z$  component of  $\hat{r}_t$  while rotating the  $xy$  part onto the  $y$  axis. Then, we rotate about the  $x$  axis an angle  $\eta_2 = \arccos(\hat{r}_t \cdot \hat{z})$ , which will rotate it back to the  $z$  axis. Since the norm is preserved by all of these rotations, the final result is  $I/2 - \gamma Z/2$  as claimed. ■

*Proof of Lemma 5.* If we plug Eq. (B8) into the Schrödinger equation  $id|\psi_t\rangle/dt = H_t|\psi_t\rangle$ , we find that  $|\phi_t\rangle$  evolves according to Hamiltonian

$$\begin{aligned} H_{\phi t} &= i \frac{dU_3}{dt} U_3^\dagger + i U_3 \frac{dU_2}{dt} U_2^\dagger U_3^\dagger + i U_3 U_2 \frac{dU_1}{dt} U_1^\dagger U_2^\dagger U_3^\dagger \\ &\quad + U_3 U_2 U_1 H_t U_1^\dagger U_2^\dagger U_3^\dagger. \end{aligned}$$

The  $Y$  term in Eq. (B12) comes immediately from the first term above:

$$i \frac{dU_3}{dt} U_3^\dagger = -\frac{\dot{\theta}}{2} Y.$$

The  $I$  and  $Z$  terms will come from the conjugation of  $H_t$  as shown in Lemma 4. So we have

$$\begin{aligned} E &= i U_3 \frac{dU_2}{dt} U_2^\dagger U_3^\dagger + i U_3 U_2 \frac{dU_1}{dt} U_1^\dagger U_2^\dagger U_3^\dagger \\ \|E\| &\leq \left\| \frac{dU_2}{dt} \right\| + \left\| \frac{dU_1}{dt} \right\|, \end{aligned}$$

and  $U_1 = \cos(\eta_1/2)I + i \sin(\eta_1/2)\hat{n} \cdot \vec{\sigma}$ , so  $dU_1/dt = -\dot{\eta}_1 \sin(\eta_1/2)I/2 + i(\dot{\eta}_1 \cos(\eta_1/2)\hat{n}/2 + \sin(\eta_1/2)\dot{\hat{n}}) \cdot \vec{\sigma}$

whose norm is given by

$$\begin{aligned} & \left\| \frac{dU_1}{dt} \right\| \\ &= \sqrt{\frac{\dot{\eta}_1^2}{4} \sin^2\left(\frac{\eta_1}{2}\right) + \frac{\dot{\eta}_1^2}{4} \cos^2\left(\frac{\eta_1}{2}\right) + |\dot{\hat{n}}|^2 \sin^2\left(\frac{\eta_1}{2}\right)} \\ &\leq \left| \frac{\dot{\eta}_1}{2} \right| + \left| \dot{\hat{n}} \sin\left(\frac{\eta_1}{2}\right) \right| \leq \frac{1}{2}(|\dot{\eta}_1| + |\eta_1 \dot{\hat{n}}|), \end{aligned}$$

where the second line follows since  $\hat{n}$  is orthogonal to  $\dot{\hat{n}}$ . Likewise,

$$\left\| \frac{dU_2}{dt} \right\| \leq \frac{1}{2}|\dot{\eta}_2| + \left| \frac{\eta_2}{2} \frac{d}{dt}(\hat{n} \times \hat{y}) \right| = \frac{1}{2}(|\dot{\eta}_2| + |\eta_2 \dot{\hat{n}}|), \quad (\text{B20})$$

with the final equality holding since  $\hat{n}$ ,  $\dot{\hat{n}}$ , and  $\hat{y}$  are mutually orthogonal.

Graphically, we can verify (can be justified analytically), that  $\eta_1$  is monotonically decreasing as a function of  $s$  with  $\eta_1(s=0) = \eta_{1,\max}$  and  $\eta_1(s) = -\eta_1(1-s)$  for all  $s$ . Meanwhile,  $\eta_2$  monotonically increases from  $\eta_2(s=0) = 0$  to its maximum  $\eta_2(s=1/2) = \eta_{2,\max}$ , and then returns to  $\eta_2(s=1) = 0$ , maintaining the symmetry  $\eta_2(s) = \eta_2(1-s)$  for all  $s$ .

$$\begin{aligned} & \int_0^T \|E\| dt \\ &\leq \frac{1}{2} \int_0^T (|\dot{\eta}_1| + |\dot{\eta}_2|) dt + \frac{1}{2} \int_0^T |\dot{\hat{n}}|(|\eta_1| + |\eta_2|) dt \\ &\leq \eta_{1,\max} + \eta_{2,\max} + \frac{\eta_{1,\max} + \eta_{2,\max}}{2} \int_0^T |\dot{\hat{n}}| dt \\ &\leq (1 + \pi/2)(\eta_{1,\max} + \eta_{2,\max}), \quad (\text{B21}) \end{aligned}$$

since  $\int_0^T |\dot{\hat{n}}| dt$  is the arclength of the path traversed by  $\hat{n}$ , which is always less than  $\pi$ . Now, we can verify that  $\eta_{1,\max} = \eta_1(s=0) \leq \delta t/4$  and  $\eta_{2,\max} = \eta_2(s=1/2) \leq \delta t/4$  as long as  $\delta t \leq 2\pi$  (analytically or graphically). Thus,  $\int_0^T \|E\| dt \leq (1 + \pi/2)(\delta t/2) \leq 1.3\delta t$ , which proves the lemma. ■

*Proof of Lemma 6.* We let  $C = \langle \Psi_0(T) | \Psi_1(T) \rangle$ , and we differentiate both sides,

$$\begin{aligned} \frac{dC}{dt} &= -i \langle \Psi_0(T) | H_1 | \Psi_1(T) \rangle + i \langle \Psi_0(T) | H_0 | \Psi_1(T) \rangle, \\ \left| \frac{dC}{dt} \right| &= |\langle \Psi_0(T) | H_1 - H_0 | \Psi_1(T) \rangle| \leq \|H_1 - H_0\|, \end{aligned}$$

and the lemma follows immediately. ■

*Proof of Lemma 7.* We go through the same process used to prove Theorem 1, except most of the work has already been

done for us. We define

$$\chi_t(t) = \arctan\left(\frac{\dot{\theta}(t)}{\gamma(t)}\right) \quad (\text{B22})$$

to mirror Eq. (45). And, as in Theorem 1, we can make the following bound:

$$\begin{aligned} |\langle 0 | \phi_{t0} \rangle|^2 &\geq \cos^2(A_t) \geq 1 - A_t^2, \\ A_t &= \chi_t(0) + \frac{1}{2} \int_0^T |\dot{\chi}_t| dt. \quad (\text{B23}) \end{aligned}$$

Also, since  $d(\arctan(u))/dt = \dot{u}/(1+u^2) \leq \dot{u}$ , we can say that

$$A_t \leq \frac{\dot{\theta}(0)}{\gamma(0)} + \frac{1}{2} \int_0^T \left| \frac{d}{dt} \left( \frac{\dot{\theta}(t)}{\gamma(t)} \right) \right| dt. \quad (\text{B24})$$

We note the following, where the subscript max denotes a quantities maximum value with  $0 \leq s \leq 1$ :

$$\begin{aligned} \left( \frac{\dot{\theta}}{\gamma} \right) &= \left( \frac{\dot{\theta}}{\Delta_\lambda} \right) \left( \frac{\Delta_\lambda}{\gamma} \right), \\ \partial_t \left( \frac{\dot{\theta}}{\gamma} \right) &= \partial_t \left( \frac{\dot{\theta}}{\Delta_\lambda} \right) \left( \frac{\Delta_\lambda}{\gamma} \right) + \left( \frac{\dot{\theta}}{\Delta_\lambda} \right) \partial_t \left( \frac{\Delta_\lambda}{\gamma} \right), \\ \left| \partial_t \left( \frac{\dot{\theta}}{\gamma} \right) \right| &\leq \left| \partial_t \left( \frac{\dot{\theta}}{\Delta_\lambda} \right) \right| \left( \frac{\Delta_\lambda}{\gamma} \right)_{\max} + \left( \frac{\dot{\theta}}{\Delta_\lambda} \right)_{\max} \left| \partial_t \left( \frac{\Delta_\lambda}{\gamma} \right) \right|, \end{aligned}$$

which allows us to rewrite

$$\begin{aligned} A_t &\leq \left( \frac{\dot{\theta}(0)}{\Delta_\lambda(0)} + \frac{1}{2} \int_0^T \left| \partial_t \left( \frac{\dot{\theta}(t)}{\Delta_\lambda(t)} \right) \right| dt \right) \left( \frac{\Delta_\lambda}{\gamma} \right)_{\max} \\ &\quad + \frac{1}{2} \left( \frac{\dot{\theta}(t)}{\Delta_\lambda(t)} \right)_{\max} \int_0^T \left| \frac{d}{dt} \left( \frac{\Delta_\lambda(t)}{\gamma(t)} \right) \right| dt. \quad (\text{B25}) \end{aligned}$$

If we look at  $\Delta_\lambda(s)/\gamma(s)$ , we can see that it is always at least 1, increasing to a maximum at  $s = 1/2$ , then decreasing back to 1. If  $(\Delta_\lambda(s)/\gamma(s))_{\max} = 1 + x$ . Graphically, we can see that  $x \leq \delta t^2/50$  whenever  $\delta t \leq 2\pi$  (can be justified analytically). Moreover, Theorem 1 tells us that the first term is at most  $d_0 + d_1$  given by Eqs. (25) and (26). Also, for any quantity  $u$ , we have  $u_{\max} \leq u(0) + \int |u|$ , so  $(\dot{\theta}(t)/\Delta_\lambda(t))_{\max} \leq d_0 + d_1$  as well. Therefore, we have

$$\begin{aligned} A_t &\leq (d_0 + d_1)(1 + x) + (d_0 + d_1)x, \\ |\langle \phi_{t0} | 0 \rangle|^2 &\geq 1 - ((d_0 + d_1)(1 + 2x))^2. \quad (\text{B26}) \end{aligned}$$

Using  $\delta t^2/50$  for  $x$  gives the Lemma. ■

- [1] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 1996), pp. 212–219.
- [2] G. Brassard, P. Høyer, and A. Tapp, Quantum counting, in *International Colloquium on Automata, Languages, and Programming* (Springer, Berlin, 1998), pp. 820–831.

- [3] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, *Contemp. Math.* **305**, 53 (2002).
- [4] I. Hen, How fast can quantum annealers count?, *J. Phys. A* **47**, 235304 (2014).
- [5] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Tight bounds on quantum searching, *Fortschr. Phys.* **46**, 493 (1998).

- [6] S. Aaronson and P. Christiano, Quantum money from hidden subspaces, in *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 2012), pp. 41–60.
- [7] T. J. Yoder, G. H. Low, and I. L. Chuang, Fixed-point Quantum Search With an Optimal Number of Queries, *Phys. Rev. Lett.* **113**, 210501 (2014).
- [8] L. K. Grover, Fixed-point Quantum Search, *Phys. Rev. Lett.* **95**, 150501 (2005).
- [9] T. Tulsı, L. K. Grover, and A. Patel, A new algorithm for fixed point quantum search, *Quantum Inf. Comput.* **6**, 483 (2006).
- [10] B. W. Reichardt and L. K. Grover, Quantum error correction of systematic errors using a quantum search framework, *Phys. Rev. A* **72**, 042326 (2005).
- [11] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Exponential improvement in precision for simulating sparse Hamiltonians, in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 2014), pp. 283–292.
- [12] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, Strengths and weaknesses of quantum computing, *SIAM J. Comput.* **26**, 1510 (1997).
- [13] E. Farhi and S. Gutmann, Analog analog of a digital quantum computation, *Phys. Rev. A* **57**, 2403 (1998).
- [14] J. Roland and N. J. Cerf, Quantum search by local adiabatic evolution, *Phys. Rev. A* **65**, 042308 (2002).
- [15] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106).
- [16] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem, *Science* **292**, 472 (2001).
- [17] K.-P. Marzlin and B. C. Sanders, Inconsistency in the Application of the Adiabatic Theorem, *Phys. Rev. Lett.* **93**, 160408 (2004).
- [18] D. M. Tong, K. Singh, L. C. Kwek, and C. H. Oh, Quantitative Conditions Do Not Guarantee the Validity of the Adiabatic Approximation, *Phys. Rev. Lett.* **95**, 110407 (2005).
- [19] J. Avron, R. Seiler, and L. Yaffe, Adiabatic theorems and applications to the quantum Hall effect, *Commun. Math. Phys.* **110**, 33 (1987).
- [20] G. A. Hagedorn and A. Joye, Elementary exponential error estimates for the adiabatic approximation, *J. Math. Anal. Appl.* **267**, 235 (2002).
- [21] A. Joye and C.-E. Pfister, Full asymptotic expansion of transition probabilities in the adiabatic limit, *J. Phys. A: Math. Gen.* **24**, 753 (1991).
- [22] D. A. Lidar, A. T. Rezakhani, and A. Hamma, Adiabatic approximation with exponential accuracy for many-body systems and quantum computation, *J. Math. Phys.* **50**, 102106 (2009).
- [23] A. T. Rezakhani, A. K. Pimachev, and D. A. Lidar, Accuracy versus run time in an adiabatic quantum search, *Phys. Rev. A* **82**, 052305 (2010).
- [24] B. W. Reichardt, The quantum adiabatic optimization algorithm and local minima, in *Proceedings of the Thirty-sixth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 2004), pp. 502–510; Erratum: <http://www-bcf.usc.edu/~breichar/Correction.txt>.
- [25] A. Ambainis and O. Regev, An elementary proof of the quantum adiabatic theorem, [arXiv:quant-ph/0411152](https://arxiv.org/abs/quant-ph/0411152).
- [26] S. Jansen, M.-B. Ruskai, and R. Seiler, Bounds for the adiabatic approximation with applications to quantum computation, *J. Math. Phys.* **48**, 102111 (2007).
- [27] M. Steffen, W. van Dam, T. Hogg, G. Breyta, and I. Chuang, Experimental Implementation of an Adiabatic Quantum Optimization Algorithm, *Phys. Rev. Lett.* **90**, 067903 (2003).
- [28] N. G. Dickson, M. Johnson, M. Amin, R. Harris, F. Altomare, A. Berkley, P. Bunyk, J. Cai, E. Chapple, P. Chavez *et al.*, Thermally assisted quantum annealing of a 16-qubit problem, *Nat. Commun.* **4**, 1903 (2013).
- [29] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2010).
- [30] W. Van Dam, M. Mosca, and U. Vazirani, How powerful is adiabatic quantum computation? in *42nd IEEE Symposium on Proceedings of the Foundations of Computer Science (2001)* (IEEE, New York, 2001), pp. 279–287.
- [31] J. Barkley Rosser and L. Schoenfeld, Approximate formulas for some functions of prime numbers, *Illinois J. Math.* **6**, 64 (1962).
- [32] P. J. Love, An efficient quantum algorithm for the moebius function, [arXiv:1412.3817](https://arxiv.org/abs/1412.3817).
- [33] M. Mosca, Counting by quantum eigenvalue estimation, *Theor. Comput. Sci.* **264**, 139 (2001).
- [34] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, [arXiv:1606.02685](https://arxiv.org/abs/1606.02685).