Simulating the dynamics of time-dependent Hamiltonians with a truncated Dyson series

Mária Kieferová,[1,2] Artur Scherer,[1] and Dominic W. Berry[1]
[1]*Department of Physics and Astronomy, Macquarie University, Sydney, NSW 2109, Australia*
[2]*Department of Physics and Astronomy, University of Waterloo and Institute for Quantum Computing, Ontario N2L 3G1, Canada*

(Received 13 November 2018; published 5 April 2019)

We provide a general method for efficiently simulating time-dependent Hamiltonian dynamics on a circuit-model-based quantum computer. Our approach is based on approximating the truncated Dyson series of the evolution operator, extending the earlier proposal by Berry *et al.* [Phys. Rev. Lett. **114**, 090502 (2015)] to evolution generated by explicitly time-dependent Hamiltonians. Two alternative strategies are proposed to implement time ordering while exploiting the superposition principle for sampling the Hamiltonian at different times. The resource cost of our simulation algorithm retains the optimal logarithmic dependence on the inverse of the desired precision.

DOI: 10.1103/PhysRevA.99.042314

I. INTRODUCTION

Simulation of physical systems is envisioned to be one of the main applications for quantum computers [1]. Effective modeling of the dynamics and its generated time evolution is crucial to a deeper understanding of many-body systems [2], spin models, and quantum chemistry [3], and may thus have significant implications for many areas of chemistry and materials sciences. Simulation of the intrinsic Hamiltonian evolution of quantum systems was the first potential use of quantum computers suggested by Feynman [4] in 1982. Quantum simulation algorithms can model the evolution of a physical system with a complexity logarithmic in the dimension of the Hilbert space [5] (i.e., polynomial in the number of particles), unlike classical algorithms whose complexity is typically polynomial in the dimension, making simulations for practically interesting systems intractable for classical computers.

The first quantum simulation algorithm was proposed by Lloyd [6] in 1996. There have been numerous advances since then providing improved performance [7–16]. One advance was to provide complexity that scales logarithmically in the error, and is nearly optimal in all other parameters [14]. Further improvements were provided by quantum signal processing methodology [15,17] as well as qubitization [16], which achieve optimal query complexity.

An important case is that of time-dependent Hamiltonians. Efficient simulation of time-dependent Hamiltonians would allow us to devise better quantum control schemes [18] and describe transition states of chemical reactions [19]. Furthermore, simulation of dynamics generated by time-dependent Hamiltonians is a key component for implementing adiabatic algorithms [20] in a gate-based quantum circuit architecture.

The most recent advances in quantum simulation algorithms are for time-independent Hamiltonians. Techniques for simulating time-dependent Hamiltonians based on the Lie-Trotter-Suzuki decomposition were developed in Refs. [8,10], but the complexity scales polynomially with error. More recent advances providing complexity logarithmic in the error [12,14] mention that their techniques can be generalized to time-dependent scenarios but do not analyze this case. The most recent algorithms [15,16] are not directly applicable to the time-dependent case. Here we present an explicit algorithm for simulating time-dependent Hamiltonians with complexity logarithmic in the allowable error, matching the complexity of the algorithms for the time-independent case in Refs. [12,14], though not those in Refs. [15,16]. Similar results were independently achieved by Low and Wiebe [21].

Our approach is based on a truncated Dyson series, similar to Ref. [14]. Whereas Ref. [14] used a Taylor series, we here use a Dyson series to take account of time dependence of the Hamiltonians. Our approach is also conceptually similar to Ref. [10], which showed that classically choosing random times could eliminate dependence of quantum complexity on the derivatives of the Hamiltonian. Our technique uses a quantum superposition of times, which introduces a logarithmic dependence on the derivative, but improves upon Ref. [10] by providing logarithmic scaling in the error as well as linear scaling in time.

We consider two different models of the Hamiltonian. In the first model, the Hamiltonian is given by an oracle for the entries in a sparse matrix, similar to Ref. [12]. In that case we decompose the Hamiltonian into a linear combination of unitary terms. In the second model, we take the Hamiltonian to already be given as a linear combination of unitaries as in Ref. [14], and an oracle yields the values of the time-dependent coefficients in the decomposition. An important case for the latter scenario is the class of model Hamiltonians expressed as linear combinations of tensor products of Pauli matrices. For example, quantum chemistry Hamiltonians, which are an important application of quantum simulations, are naturally expressed in this framework.

This paper is organized as follows. In Sec. II we first state our main results. In Sec. III we introduce our framework of definitions and assumptions. Section IV then provides an overview of our simulation algorithm. The main technical difficulty is in ordering the time register, which is presented in Sec. V. We give two alternative methods for implementing time ordering using an additional control register. Our first approach given in Sec. V A uses a compression technique to encode ordered sequences of times with minimal overhead. Our second method is based on the quantum sort technique and presented in Sec. V B. In Sec. V C we show that oblivious amplitude amplification can be achieved with both approaches. We derive the overall query and gate complexity in Sec. VI. We conclude with a brief summary and discussion of our results in Sec. VII.

II. MAIN RESULTS

We consider a time-dependent Hamiltonian $H(t)$ to be given either by a time-dependent d-sparse matrix or by a time-dependent linear combination of unitary operators (e.g., tensor products of Pauli matrices). Access to oracles is assumed to compute the positions and values of the nonzero entries in the first case, or the values of the decomposition coefficients and the actions of the corresponding unitaries in the second case. For sparse matrices, the Hamiltonian is considered d-sparse on the time interval $[t_0, t_0+T]$ in the sense that in any row or column there are at most d entries that can be nonzero for any $t \in [t_0, t_0+T]$. Moreover, we take ϵ to be the maximum allowable error of the simulation and n to be the number of qubits for the quantum system on which $H(t)$ acts. We define

$$H_{\max} := \max_{t\in[t_0,t_0+T]} \|H(t)\|_{\max}, \tag{1}$$

$$\dot{H}_{\max} := \max_{t\in[t_0,t_0+T]} \|dH(t)/dt\|, \tag{2}$$

where $\|\cdot\|$ indicates the spectral norm.

Theorem 1. For a Hamiltonian $H(t)$ given by a time-dependent d-sparse matrix, the generated time evolution of a quantum system can be simulated for time T and within error ϵ using a number of oracle queries scaling as

$$O\left(d^2 H_{\max} T \frac{\log_2(dH_{\max}T/\epsilon)}{\log_2\log_2(dH_{\max}T/\epsilon)}\right), \tag{3}$$

and a number of additional elementary gates scaling as

$$O\left\{d^2 H_{\max} T \frac{\log_2(dH_{\max}T/\epsilon)}{\log_2\log_2(dH_{\max}T/\epsilon)} \times \left[\log_2\left(\frac{dH_{\max}T}{\epsilon}\right) + \log_2\left(\frac{\dot{H}_{\max}T}{\epsilon d H_{\max}}\right) + n\right]\right\}. \tag{4}$$

Theorem 2. For a Hamiltonian $H(t)$ given by a time-dependent linear combination of L unitary terms, the generated time evolution of a quantum system can be simulated for time T and within error ϵ using a number of oracle queries scaling as

$$O\left[L\alpha_{\max} T \frac{\log_2(L\alpha_{\max}T/\epsilon)}{\log_2\log_2(L\alpha_{\max}T/\epsilon)}\right], \tag{5}$$

and a number of additional elementary gates scaling as

$$O\left\{L\alpha_{\max} T \frac{\log_2(L\alpha_{\max}T/\epsilon)}{\log_2\log_2(L\alpha_{\max}T/\epsilon)} \times \left[\log_2\left(\frac{L\alpha_{\max}T}{\epsilon}\right) + \log_2\left(\frac{\dot{H}_{\max}T}{\epsilon L\alpha_{\max}}\right)\right]\right\}, \tag{6}$$

where $\alpha_{\max}$ denotes a known global upper bound on the modulus of each of the L time-dependent coefficients in the decomposition, implying $L\alpha_{\max} \geqslant H_{\max}$.

III. BACKGROUND AND FRAMEWORK

The unitary operator corresponding to time evolution under a time-dependent Hamiltonian $H(t)$ for a time period T is the time-ordered propagator

$$U(t_0, t_0+T) = \mathcal{T}\exp\left[-i\int_{t_0}^{t_0+T} d\tau\, H(\tau)\right], \tag{7}$$

where $\mathcal{T}$ is the time-ordering operator and we use natural units ($\hbar = 1$). Equation (7) can be understood as a limit

$$U(t_0, t_0+T) = \lim_{M\to\infty} \mathcal{T}\prod_{m=0}^{M-1} \exp\left\{\frac{-iT}{M} H\left(t_0 + \frac{mT}{M}\right)\right\}, \tag{8}$$

where $\mathcal{T}$ indicates the strict order of the terms in the product.

We distinguish between two different representations of $H(t)$. In the *SM scenario*, where SM stands for sparse matrix, Hamiltonians that are given by a d-sparse time-dependent Hermitian matrix. In the *QC scenario*, Hamiltonians that are time-dependent linear combinations of some time-independent unitaries. Here QC stands for quantum chemistry, as this would be the most prominent example. In quantum chemistry the model Hamiltonians are expressible as linear combinations of tensor products of Pauli matrices by using the Jordan-Wigner transformation. An explicit time dependence may arise when the motion of the nuclei is accounted for, or when a molecule is under the influence of an explicitly time-dependent electromagnetic driving field. Another important example that also fits into this framework is adiabatic algorithms which use time-dependent Hamiltonians of the form $H(t) = (1 - \frac{t}{T})H_0 + \frac{t}{T}H_1$ [20], where the two time-independent Hermitian operators H_0, H_1 can be further decomposed into unitaries.

In both scenarios our method relies on the ability to express the time-dependent Hamiltonian as a linear combination of efficiently implementable unitaries. For d-sparse matrices we need to decompose the Hamiltonian into a sum of unitaries, and it is more practical to take the unitaries to be time-dependent while making the coefficients in the decomposition constant. In contrast, for cases where the Hamiltonian is already given in the form of a sum of unitaries, we take the unitaries to be constant, and the time dependence is solely in the coefficients. This is natural for applications such as quantum chemistry or adiabatic algorithms. In either case we require access to oracles yielding information about the Hamiltonian.

In the SM scenario, access to a d-sparse Hamiltonian is provided through the oracles

$$O_{\texttt{loc}}|j, s\rangle = |j, \nu(j, s)\rangle, \tag{9}$$

$$O_{\texttt{val}}|t, i, j, z\rangle = |t, i, j, z \oplus H_{ij}(t)\rangle, \tag{10}$$

where $\oplus$ represents a bitwise XOR. Here, $\nu(j, s)$ gives the position of the sth element in row j of $H(t)$ that may be nonzero at any time. Note that the oracle O_{loc} does not depend on time. Oracle O_{val} yields the values of non-zero elements at each instant of time. Furthermore, we say that a time-dependent Hamiltonian $H(t)$ is *d-sparse on the interval* if the number of entries in any row or column that may be nonzero at any time throughout the interval is at most d. This definition is distinct from the maximum sparseness of the instantaneous Hamiltonians $H(t)$, because some entries may go to zero while others become nonzero. (This definition of sparseness upper bounds the sparseness of instantaneous Hamiltonians.)

Our algorithm for d-sparse matrices builds on the decomposition of a Hermitian matrix into equal-sized 1-sparse self-inverse parts introduced in Lemma 4.3 in Ref. [12]. The Hamiltonian $H(t)$ can be decomposed using the technique of Ref. [12] for any individual time, giving

$$H(t) = \gamma \sum_{\ell=0}^{L-1} H_\ell(t), \tag{11}$$

where the $H_\ell(t)$ are 1-sparse, unitary, and Hermitian. The matrix entries in $H_\ell(t)$ can be determined using $O(1)$ queries according to Lemma 4.4 of Ref. [12]. The single coefficient γ is time-independent, and so is the sum of coefficients $\lambda := L\gamma$ in the decomposition. To give a contribution to the error that is $O(\epsilon)$, the value of γ should be chosen as (see Eq. (24) in Ref. [12] and the accompanying discussion)

$$\gamma \in \Theta[\epsilon/(d^3T)]. \tag{12}$$

The unitary decomposition Eq. (11) can be computed with a number of Hamiltonians scaling as

$$L \in O(d^2 H_{\max}/\gamma). \tag{13}$$

In the QC scenario, the Hamiltonian already has the form of a unitary decomposition with time-dependent coefficients:

$$H(t) = \sum_{\ell=0}^{L-1} \alpha_\ell(t) H_\ell. \tag{14}$$

Here the H_ℓ are all unitary and time-independent, while each $\alpha_\ell(t)$ is assumed to be a real-valued differentiable function with modulus upper bounded by a known constant $\alpha_{\max} \geqslant |\alpha_\ell(t)|$ for $t \in [0, T]$ and all $\ell \in \{0, \dots, L-1\}$. The condition $\alpha_\ell(t) \in \mathbb{R}$ can always be attained by decomposing any complex-valued coefficient into its real and imaginary parts and including the complex phase factor $\pm i$ in the associated unitary H_ℓ.

Similar to the SM scenario, access to the Hamiltonian is provided through oracles, as follows. We assume an efficient unitary procedure O_{unit} that applies a single unitary from the decomposition Eq. (14) to the system state according to

$$O_{\text{unit}} |\ell\rangle_{\mathtt{a}} |\Psi\rangle_{\mathtt{s}} = |\ell\rangle_{\mathtt{a}} H_\ell |\Psi\rangle_{\mathtt{s}}, \tag{15}$$

whereby the particular unitary H_ℓ is selected by the index value ℓ held in an ancilla register state $|\ell\rangle_{\mathtt{a}}$. An important example is when the H_ℓ are composed of tensor products of Pauli matrices, in which case each such oracle query can be implemented with $O[L(n + \log_2 L)]$ elementary gates [14]. However, in this paper we do not bind ourselves to a specific implementation and quantify the complexity only in terms of the total number of oracle calls. In addition, we also assume access to the time-dependent coefficients through a *coefficient oracle* defined as

$$O_{\mathtt{coeff}} |\ell, t, z\rangle = \big|\ell, t, z \oplus \alpha_\ell^{[\nu]}(t)\big\rangle. \tag{16}$$

That is, the oracle returns the target coefficients as $z \oplus \alpha_\ell^{[\nu]}(t)$, where z is a ν-bit integer encoded into an ν-qubit register, and $\alpha_\ell^{[\nu]}(t) := \lfloor 2^\nu \alpha_\ell(t)/\alpha_{\max} \rfloor$ is a ν-bit fixed-point approximation to the actual value of $\alpha_\ell(t)$ rescaled by $\alpha_{\max}$ [such that $\alpha_\ell(t)/\alpha_{\max} \leqslant 1$].

The sum of the coefficients, $\lambda(t) := \sum_{\ell=0}^{L-1} \alpha_\ell(t)$, in the decomposition Eq. (14) generally depends on time, which makes oblivious amplitude amplification [14] more difficult. For oblivious amplitude amplification to be successfully achieved for a time interval, the integral of $\lambda(t)$ over that interval should be equal to ln 2. To perform oblivious amplitude amplification on this decomposition directly would need a way to integrate $\lambda(t)$ and solve for an appropriate length of the time interval, which would increase the computational complexity. Instead, we alter the unitary decomposition Eq. (14) to a new decomposition such that the sum of the new coefficients is time-independent. This decomposition is

$$\begin{aligned} H(t) &= \sum_{\ell=0}^{L-1} \frac{\alpha_{\max} + \alpha_\ell(t)}{2} H_\ell + \sum_{\ell=0}^{L-1} \frac{\alpha_{\max} - \alpha_\ell(t)}{2} (-H_\ell) \\ &= \sum_{\ell=0}^{2L-1} \widetilde{\alpha}_\ell(t) \widetilde{H}_\ell, \end{aligned} \tag{17}$$

where

$$\widetilde{\alpha}_\ell(t) := \begin{cases} \frac{\alpha_{\max}+\alpha_\ell(t)}{2} & \text{for} \quad \ell = 0, \dots, L-1, \\ \frac{\alpha_{\max}-\alpha_{\ell-L}(t)}{2} & \text{for} \quad \ell = L, \dots, 2L-1, \end{cases} \tag{18}$$

$$\widetilde{H}_\ell := \begin{cases} H_\ell & \text{for} \quad \ell = 0, \dots, L-1, \\ -H_{\ell-L} & \text{for} \quad \ell = L, \dots, 2L-1. \end{cases} \tag{19}$$

This new decomposition has $2L$ terms and the sum of its coefficients is by construction time-independent and equal to $\lambda = L\alpha_{\max}$. This allows us to satisfy the condition that is sufficient for achieving oblivious amplitude amplification procedure globally for the entire time interval $[t_0, t_0 + T]$.

As is common, we will quantify the resource requirements of the algorithm in terms of two complexity measures: the "*query complexity*" and the "*gate complexity*." By the former measure we mean the number of queries to the oracles introduced above, i.e., the number of uses of the unitary procedures efficiently implementing those oracles, thereby dismissing all details of their implementation and treating them as black boxes. By the latter measure we mean the total number of additional elementary 1- and 2-qubit gate operations.

IV. ALGORITHM OVERVIEW

Suppose we wish to simulate the action of U in Eq. (7) within error ϵ. First, the total simulation time T is divided into r time segments of length T/r. Without loss of generality, we analyze the evolution induced within the first segment and set $t_0 = 0$. The simulation of the evolution within the following segments is accomplished in the same way. The overall complexity of the algorithm is then given by the number of segments r times the cost of simulation for each segment. To upper bound the overall simulation error by ϵ, we require the error of simulation for each segment to be at most ϵ/r.

We need a set of qubits to encode the time over the entire time interval $[0, T]$. It is convenient to use one set of qubits for the time that encodes the segment number, and another set of qubits that gives the time within the segment. The qubits encoding the segment number are only changed by incrementing by 1 between segments, which gives complexity $O(r \log_2 r)$ that is much smaller than other complexities that appear.

Second, we approximate the evolution within the first segment by a Dyson series up to the Kth order:

$$U(0, T/r) \approx \sum_{k=0}^{K} \frac{(-i)^k}{k!} \mathcal{T} \int_0^{T/r} d\mathbf{t}\, H(t_k) \dots H(t_1), \tag{20}$$

where, for term k in the Dyson series, $\mathcal{T} \int_0^{T/r} d\mathbf{t}\,(\cdot)$ represents integration over a k-tuple of time variables $(t_1, \dots, t_k)$, while keeping the times ordered: $t_1 \leqslant t_2 \leqslant \dots \leqslant t_k$. It is straightforward to show that the error of this approximation is $O[\frac{H_{\max}^{K+1}}{(K+1)!}(\frac{T}{r})^{K+1}]$.

Next, we discretize the integral over each time variable and approximate it by a sum with M terms, where we choose M to be a power of two so that we can use separate qubits to index the time within the segment and the segments. The time-dependent Hamiltonian is thereby approximated by its values at M uniformly separated times $\frac{jT}{rM}$ identified by an integer $j \in \{0, \dots, M-1\}$. Replacing all integrals by sums in Eq. (20) yields the following approximation of the time-evolution operator within the first segment:

$$\widetilde{U} := \sum_{k=0}^{K} \frac{(-iT/r)^k}{M^k k!} \sum_{j_1,\dots,j_k=0}^{M-1} \mathcal{T} H(t_{j_k}) \dots H(t_{j_1}). \tag{21}$$

Replacing the integrals by Riemann sums contributes an additional error upper bounded by $O[(T/r)^2 \dot{H}_{\max}/M]$ (see Sec. VI for more details). The overall error of the obtained approximation is thus,

$$\|\widetilde{U} - U(0, T/r)\| \in O\left[\frac{(H_{\max}T/r)^{K+1}}{(K+1)!} + \frac{(T/r)^2 \dot{H}_{\max}}{M}\right]. \tag{22}$$

Provided that $r \geqslant H_{\max}T$, the overall error can be bounded by ϵ/r if we choose

$$K \in \Theta\left[\frac{\log_2 (r/\epsilon)}{\log_2 \log_2 (r/\epsilon)}\right] \quad \text{and} \quad M \in \Theta\left(\frac{T^2 \dot{H}_{\max}}{\epsilon r}\right). \tag{23}$$

The main difference from time-independent Hamiltonians is that for time-dependent Hamiltonians we need to implement the evolution generated by $H(t)$ for different times in the correct order. We achieve this by introducing an additional multiqubit control register called "`time`." This ancilla register is prepared in a certain superposition state (depending on which approach we take). It is used to sample the Hamiltonian at different times in superposition in a way that respects time ordering.

Substituting the unitary decomposition Eq. (11) or Eq. (17) into Eq. (21), the approximation of the time-evolution operator within the first segment takes the form $\widetilde{U} = \sum_{\boldsymbol{j} \in J} \beta_{\boldsymbol{j}} V_{\boldsymbol{j}}$, where $\boldsymbol{j}$ is a multi-index and the coefficients $\beta_{\boldsymbol{j}}$ comprise information about both the time discretization and the unitary decomposition weightings as well as the order k within the Taylor series. Explicitly, we define

$$\beta^{\mathrm{SM}}_{(k,\ell_1,\dots,\ell_k,j_1,\dots,j_k)} := \frac{(\gamma\, T/r)^k}{M^k k_1! k_2! \dots k_\varsigma!} \theta_k(j_1, \dots, j_k), \tag{24}$$

$$V^{\mathrm{SM}}_{(k,\ell_1,\dots,\ell_k,j_1,\dots,j_k)} := (-i)^k\, H_{\ell_k}(t_{j_k}) \dots H_{\ell_1}(t_{j_1}), \tag{25}$$

for the SM scenario with decomposition Eq. (11), and

$$\beta^{\mathrm{QC}}_{(k,\ell_1,\dots,\ell_k,j_1,\dots,j_k)} := \frac{(T/r)^k}{M^k k_1! k_2! \dots k_\varsigma!} \theta_k(j_1, \dots, j_k) \times \widetilde{\alpha}_{\ell_k}(t_{j_k}) \dots \widetilde{\alpha}_{\ell_1}(t_{j_1}), \tag{26}$$

$$V^{\mathrm{QC}}_{(k,\ell_1,\dots,\ell_k,j_1,\dots,j_k)} := (-i)^k\, H_{\ell_k} \dots H_{\ell_1}, \tag{27}$$

for the QC scenario with decomposition Eq. (14), where $\theta_k(j_1, \dots, j_k) = 1$ if $j_1 \leqslant j_2 \leqslant \dots \leqslant j_k$, and zero otherwise. The quantity ς is the number of distinct values of j, and $k_1, k_2, \dots, k_\varsigma$ are the number of repetitions for each distinct value of j. That is, we have the indices j for the times sorted in ascending order, and we have multiplied by a factor of $k!/(k_1!k_2! \dots k_\varsigma!)$ to take account of the number of unordered sets of indices which give the same ordered set of indices. The multi-index set J is defined for the SM and QC scenarios as

$$\begin{aligned} J^{\mathrm{SM}} := \{&(k, \ell_1, \dots, \ell_k, j_1, \dots, j_k) : k \in \{0, \dots, K\}, \\ &\ell_1, \dots, \ell_k \in \{0, \dots, L-1\}, \\ &j_1, \dots, j_k \in \{0, \dots, M-1\}\}, \end{aligned} \tag{28}$$

and

$$\begin{aligned} J^{\mathrm{QC}} := \{&(k, \ell_1, \dots, \ell_k, j_1, \dots, j_k) : k \in \{0, \dots, K\}, \\ &\ell_1, \dots, \ell_k \in \{0, \dots, 2L-1\}, \\ &j_1, \dots, j_k \in \{0, \dots, M-1\}\}, \end{aligned} \tag{29}$$

respectively. The only difference is the second has $2L$ rather than L, where we expanded the sum to ensure the λ is independent of time. For simplicity we take both L and M to be powers of two, so equal-weight superpositions can be produced with tensor products of Hadamard gates (denoted as HAD), for example, $\mathrm{HAD}^{\otimes \log_2 L}$ when preparing superposition states $\frac{1}{\sqrt{L}} \sum_{\ell=0}^{L-1} |\ell\rangle$. Equal superpositions over numbers of states that are not powers of two can also be obtained with logarithmic complexity by standard (but more complicated) techniques.

We use a standard technique to implement linear combinations of unitaries (referred to as *"LCU technique"*) involving the use of an ancillary register to encode the coefficients $\beta_{\boldsymbol{j}}$ [12,22]. In the next section, we present two approaches to implement the (multiqubit) ancilla state preparation,

$$B|0\rangle_{\mathtt{a}} = \frac{1}{\sqrt{\mathcal{N}}}\sum_{j\in J}\sqrt{\beta_j}|\boldsymbol{j}\rangle_{\mathtt{a}}, \tag{30}$$

as part of the LCU approach, where $\mathcal{N} := \sum_{j\in J}\beta_j$. For the LCU technique we introduce the operator $\textsc{Select}(V) := \sum_j |\boldsymbol{j}\rangle\langle \boldsymbol{j}|_{\mathtt{a}}\otimes V_j$ acting as

$$\textsc{Select}(V)|\boldsymbol{j}\rangle_{\mathtt{a}}|\Psi\rangle_{\mathtt{s}} = |\boldsymbol{j}\rangle_{\mathtt{a}}V_j|\Psi\rangle_{\mathtt{s}} \tag{31}$$

on the joint ancilla and system states. This operation implements a term from the decomposition of $\tilde{U}$*selected* by the ancilla state $|\boldsymbol{j}\rangle_a$ with weight β_j. Following the method in Ref. [14], we also define

$$R := \mathbb{I}_{\mathtt{as}} - 2(|0\rangle\langle 0|_{\mathtt{a}}\otimes\mathbb{I}_{\mathtt{s}}), \tag{32}$$

$$W := (B^\dagger\otimes\mathbb{I}_{\mathtt{s}})\textsc{Select}(V)(B\otimes\mathbb{I}_{\mathtt{s}}). \tag{33}$$

If $\widetilde{U}$ were unitary and $s \leqslant 2$, then a single step of oblivious amplitude amplification could be used to implement $\widetilde{U}$. When $\widetilde{U}$ is only approximately unitary, with $\|\widetilde{U} - U(0, T/r)\| \in O(\epsilon/r)$, a single step of oblivious amplitude amplification yields [14]

$$-WRW^\dagger RW|0\rangle_{\mathtt{a}}|\Psi\rangle_{\mathtt{s}} = |0\rangle_{\mathtt{a}}\tilde{U}|\Psi\rangle_{\mathtt{s}} + O(\epsilon/r), \tag{34}$$

which is the approximate transformation we aim to implement for each time segment.

The implementation of the unitary transformation W by a quantum circuit is illustrated in Fig. 1. In both scenarios, in addition to the system register holding the quantum state to be processed by Hamiltonian evolution, three auxiliary control registers are employed. The "K register" consisting of K qubits is used to hold the k value corresponding to the order to be taken within the truncated Dyson series. The time register consisting of K subregisters each of size $\log_2 M$ is prepared in a special superposition state "$|\text{clock}\rangle$" that also involves the K register and which is used to sample the Hamiltonian at different instances of time in superposition such that time-ordering is accounted for. Note that the same $|\text{clock}\rangle$ state and the two alternative approaches to its preparation presented in Sec. V A can be used for both considered scenarios.

Finally, the "L register" consisting of K subregisters each of size $\log_2 L$ is used to select which term out of the decomposition into unitaries is to be applied. The ancillary L register states need to be prepared with amplitudes corresponding to the square roots of the weightings of the unitaries in the decomposition. In decomposition Eq. (11) for the SM scenario, all unitaries have equal weight; thus only equal-weight superpositions need to be prepared in the L register in this case. However, for decompositions Eq. (14) for the QC scenario, we need to prepare superposition states that encode the amplitudes of the time-dependent weightings $\widetilde{\alpha}_\ell(t)$ according to the unitary transformation

$$\begin{aligned}&\text{controlled-}\textsc{Prep}(\alpha)|t\rangle_{\mathtt{time}_k}|0\rangle_{\mathtt{L}_k}\\ &\quad := (\mathbb{I}\otimes\textsc{Prep}(\alpha(t)))|t\rangle_{\mathtt{time}_k}|0\rangle_{\mathtt{L}_k}\\ &\quad := \frac{1}{\sqrt{L\alpha_{\max}}}\sum_{\ell=0}^{2L-1}\sqrt{\widetilde{\alpha}_\ell(t)}|t\rangle_{\mathtt{time}_k}|\ell\rangle_{\mathtt{L}_k}.\end{aligned} \tag{35}$$

This is a controlled state preparation involving the kth L-subregister as target and the kth time subregister as control, to be executed for each $k = 1, \dots, K$. The subscript k is used on the register labels to indicate the kth subregisters. Note that $\textsc{Prep}(\alpha)$ encodes the coefficients of the altered (but equivalent) decomposition Eq. (17), which has been introduced to ensure that oblivious amplitude amplification can be performed correctly.

The $\textsc{Select}(V)$ operation is implemented by a series of K controlled-$\textsc{Select}(H)$ transformations, whose action is given by

$$\begin{aligned}&\text{controlled-}\textsc{Select}(H)|b\rangle_{\mathtt{K}_k}|t\rangle_{\mathtt{time}_k}|\ell\rangle_{\mathtt{L}_k}|\Psi\rangle_{\mathtt{s}}\\ &\quad := |b\rangle_{\mathtt{K}_k}|t\rangle_{\mathtt{time}_k}|\ell\rangle_{\mathtt{L}_k}(-iH_\ell(t))^b|\Psi\rangle_{\mathtt{s}}\end{aligned} \tag{36}$$

for the SM scenario, and by

$$\begin{aligned}&\text{controlled-}\textsc{Select}(H)|b\rangle_{\mathtt{K}_k}|\ell\rangle_{\mathtt{L}_k}|\Psi\rangle_{\mathtt{s}}\\ &\quad := |b\rangle_{\mathtt{K}_k}|\ell\rangle_{\mathtt{L}_k}(-i\widetilde{H}_\ell)^b|\Psi\rangle_{\mathtt{s}},\end{aligned} \tag{37}$$

for the QC scenario. Note that the control on the $\mathtt{time}_k$ register is only necessary for the SM scenario, not the QC scenario, since in the latter case the individual unitaries H_ℓ in the decomposition are all time-independent. The phase factor $(-i)^b$ has been included in the definition of the operation, yet it is implemented simply by applying the phase gate $S^\dagger := |0\rangle\langle 0| + (-i)|1\rangle\langle 1|$ to each wire of the K register. If k is the value encoded in unary in all K wires of the K register, then this results in the overall factor $(-i)^k$ occurring in the kth order term of the Dyson series.

V. PREPARATION OF AUXILIARY REGISTER STATES

The algorithm relies on efficient implementation of the unitary transformation B, which acts on the composite ancilla registers $\mathtt{K}\otimes\mathtt{time}\otimes\mathtt{L}$ and prepares a joint superposition state given in Eq. (30). This state preparation includes encoding the Dyson order k (in register K), the values $j_1 \leqslant \cdots \leqslant j_k$ specifying the ordered time instances (in register time), and the index values $\ell_1, \dots, \ell_k$ specifying the terms in the Hamiltonian decompositions (in register L), in superposition over all possible combinations of the values, and with amplitudes that are the square roots of the weightings $\beta^{\mathtt{SM}}_{(k,\ell_1,\dots,\ell_k,j_1,\dots,j_k)}$ or $\beta^{\mathtt{QC}}_{(k,\ell_1,\dots,\ell_k,j_1,\dots,j_k)}$. As noted above, the main difficulty of simulating time-dependent Hamiltonian dynamics is the implementation of time ordering. This is achieved by a weighted superposition named "clock" prepared as the joint state in the composite auxiliary control register $\mathtt{K}\otimes\mathtt{time}$, which has been introduced in addition to the ancilla register L used to implement the LCU technique in the time-independent case. Its key purpose and task is to control sampling the Hamiltonian for different instances of time in superposition in a way that respects the correct time order. We here present two alternative approaches to efficient preparation of such clock states in the composite $\mathtt{K}\otimes\mathtt{time}$ register.

The first approach is based upon generating the intervals between successive times according to an exponential distribution, in a similar way as in Ref. [23]. These intervals are then summed to obtain the times. Our second approach starts by creating a superposition over k and then a superposition over all possible times $t_1, \dots, t_k$ for each k. The times are then

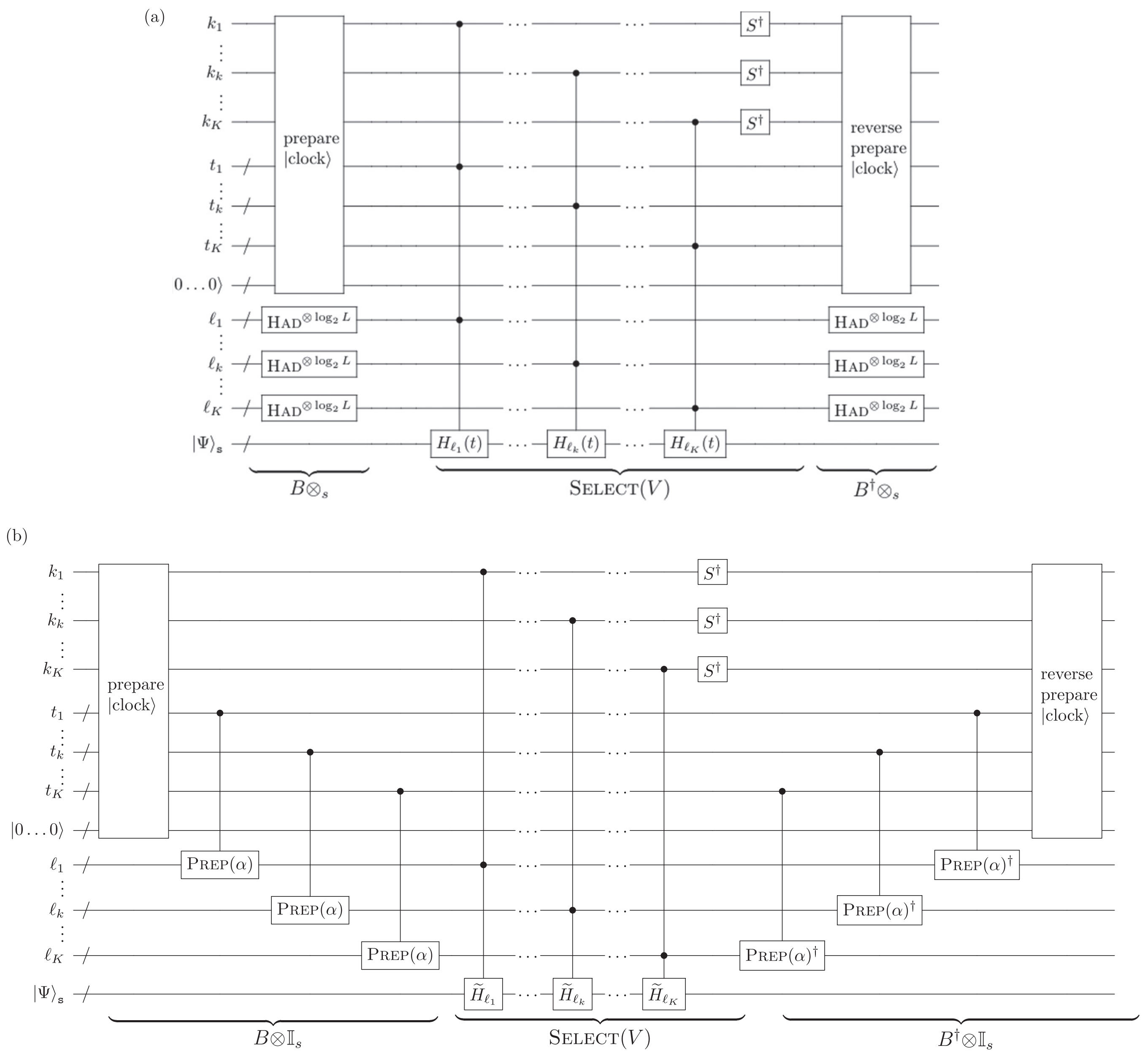

FIG. 1. Quantum circuit implementing the unitary transformation W [see definition in Eq. (33)] for (a) the SM scenario and (b) QC scenario. In the SM scenario the Hamiltonian is given as a time-dependent sparse matrix, and in the QC scenario the Hamiltonian is given as a time-dependent linear combination of time-independent unitaries. In both cases, W is composed of a SELECT(V) operation consisting of a series of K controlled-SELECT(H) transformations sandwiched between the operations B and $B^\dagger$ for the preparation and its reverse of the auxiliary register states required for implementing the LCU technique. The boxed subroutine "prepare $|\text{clock}\rangle$" acting on $\mathtt{K}\otimes\mathtt{time}$ is implemented either by the "compressed encoding" approach outlined in Sec. V A, or by using a quantum sorting network described in Sec. V B. Both approaches require additional ancillae indicated by $|0\dots0\rangle$. The phase gate $S^\dagger = |0\rangle\langle 0| + (-i)|1\rangle\langle 1|$ is used to implement the factor $(-i)^k$ as part of the kth order term of the Dyson series.

reversibly sorted using quantum sorting networks [24,25] and used to control ℓ as in the previous approach.

This section is organized as follows. We first present the two approaches to preparation of the superposition state $|\text{clock}\rangle_{\mathtt{K}\otimes\mathtt{time}}$ in Secs. V A and V B. In Sec. V C we then describe how the state preparation operation B is completed by transforming the auxiliary $\mathtt{L}$ register states, depending on the given Hamiltonian model.

A. Clock preparation using compressed rotation encoding

To explain the first approach, it is convenient to consider a conceptually simple but inefficient representation of the time register. An ordered sequence of times $t_1,\dots,t_k$ is encoded in a binary string $x\in\{0,1\}^M$ with Hamming weight $|x|=k$, where $0\leqslant k\leqslant K$. That is, if m_j is the jth value of m such that $x_m=1$, then $t_j=m_jT/(rM)$. This automatically forces the ordering $t_1<t_2<\cdots<t_k$, omitting the terms when two

or more Hamiltonians act at the same time. The binary string x then would be encoded into M qubits as $|x\rangle$.

Now consider these M qubits initialized to $|0\rangle^{\otimes M}$, then rotated by a small angle, to give

$$\begin{aligned}(\alpha|0\rangle+\beta|1\rangle)^{\otimes M} &= \sum_x \alpha^{M-|x|}\beta^{|x|}|x\rangle \\ &= \sum_{x,|x|\leqslant K}\alpha^{M-|x|}\beta^{|x|}|x\rangle+\sum_{x,|x|>K}\alpha^{M-|x|}\beta^{|x|}|x\rangle \\ &= \sqrt{1-\mu^2}|\Omega\rangle+\mu|\Omega^\perp\rangle, \quad (38)\end{aligned}$$

where $\alpha := \sqrt{\frac{rM}{\lambda T+rM}}$ and $\beta := \sqrt{\frac{\lambda T}{\lambda T+rM}}$. The state $|\Omega\rangle$ encodes the times in a way required by Dyson series up to the Kth order with the weighting as in Eqs. (24) and (26),

$$|\Omega\rangle = \frac{1}{\sqrt{S}}\sum_{|x|\leqslant K}\zeta^{|x|/2}|x\rangle, \quad (39)$$

with $S := \sum_{|x|\leqslant K}\zeta^{|x|}$, $\zeta := \lambda T/(rM)$. Recall that for the SM scenario $\lambda = L\gamma$, whereas for the QC scenario $\lambda = L\alpha_{\max}$. Also recall that the Hamming weight $|x|$ corresponds to the order $0\leqslant k\leqslant K$ within the truncated Dyson series. The states $|\Omega\rangle$ and $|\Omega^\perp\rangle$ are normalized and orthogonal. The state $|\Omega^\perp\rangle$ is analogous to the higher-order terms omitted in the Dyson series. The amplitude μ satisfies (as shown in Ref. [23]):

$$\mu^2 = O\left[\frac{(\lambda T/r)^{K+1}}{(K+1)!}\right]. \quad (40)$$

We choose K as in Eq. (23), and the choice $r>\lambda T$ implies $\mu^2 = O(\epsilon/r)$.

Since $|\Omega\rangle$ includes only Hamming weights up to K, it includes strings that are highly compressible. In order to compress the string x, the lengths of the strings of zeros between successive ones are stored. That is, a string $x = 0^{s_1}10^{s_2}10^{s_3}\dots0^{s_k}10^{\sigma}$ can be represented by the integers $s_1s_2\dots s_k$. There is always a total of $K+1$ entries, regardless of the Hamming weight. In addition, we encode the Hamming weight k in an additional register. Thus, our encoding converts x into

$$\begin{aligned}\Xi|0^{s_1}10^{s_2}10^{s_3}\dots0^{s_k}10^{\sigma}\rangle &= |s_1+1\rangle|s_1+s_2+2\rangle\dots|s_1+s_2 \\ &\quad+\dots+k+1\rangle|\mathrm{junk}_x\rangle|k\rangle, \quad (41)\end{aligned}$$

where $|\mathrm{junk}_x\rangle$ is a remnant of the construction as well as a padding for strings with Hamming distance smaller than K. For example, for $K=4$, the state $|001000001000\rangle$ would be encoded as $|2,5\rangle|\mathrm{junk}_x\rangle|2\rangle$. Following Eq. (17) from Ref. [23], we define a more complex encoding B_M^K as

$$\begin{aligned}B_M^K|x\rangle &= |s_1,\dots,s_k\rangle \\ &\otimes\left(\sum_{j=0}^{q-\sigma-1}\alpha^j\beta|j+\sigma\rangle+\alpha^{q-\sigma}|q\rangle\right)|\phi_q\rangle^{\otimes(K-k)}, \end{aligned} \quad (42)$$

where $|\phi_q\rangle = \sum_{s=0}^{q-1}\beta\alpha^s|s\rangle+\alpha^q|q\rangle$ and we take $q=M$ for the encoding here. According to Theorem 3 in Ref. [23],

$$|\phi_q\rangle^{\otimes K+1} = \sum_{x,|x|\leqslant K}\alpha^{M-|x|}\beta^{|x|}B_M^K|x\rangle+\mu|\Omega'\rangle. \quad (43)$$

The idea is that one prepares the state $|\phi_q\rangle^{\otimes K+1}$ (see Eq. (13) of Ref. [23]), which gives an exponential distribution. Encoding $B_M^K|x\rangle$ includes spacing between consecutive ones and the expression in the second line of Eq. (42) represents $|\mathrm{junk}_x\rangle$. However, we require the absolute times, rather than the differences between the times, as well as an additional register encoding k. The state $B_M^K|x\rangle$ can be converted into $\Xi|x\rangle$ following the steps 1 and 2 of Section 4.4 in Ref. [23]. First, one computes the absolute position of 1's by computing the prefix sums. Then, it is possible to identify the register $k+1$ by finding the first register larger than m using the term in brackets in 42 as an indicator of overflow. The Hamming weight k is then recorded in an additional register. Unlike Ref. [23], there is no need to clean the $|\mathrm{junk}_x\rangle$ register or uncompute the Hamming weight.

B. Clock preparation using a quantum sort

In this section, we explain an alternative approach to implementing the preparation of the $|\mathrm{clock}\rangle_{\mathtt{K\otimes time}}$ state, which establishes time ordering via a reversible sorting algorithm. This approach first creates a superposition over all Dyson series orders $k\leqslant K$ with amplitudes proportional to $(\zeta^k/k!)^{1/2}$. It then generates a superposition over all possible k-tuples of times $t_1,\dots,t_k$ for each possible value of k. To achieve time-ordered combinations, sorting is applied to each of the tuples in the superposition using a quantum sorting network algorithm.

To be more concrete, in the first step we create a superposition over the allowed values of k in unary encoding by applying the following transformation to K qubits initialized to $|0\rangle$:

$$\textsc{Prep}(k)|0\rangle^{\otimes K} := \frac{1}{\sqrt{\mathcal{N}_s}}\sum_{k=0}^{K}\sqrt{\frac{\zeta^kM^k}{k!}}|1^k0^{K-k}\rangle, \quad (44)$$

where the constant $\mathcal{N}_s := \sum_{k=0}^{K}\frac{\zeta^kM^k}{k!}$ accounts for normalization. This transformation can be easily implemented (see Sec. 4 of Ref. [26] for details) using a series of $O(K)$ controlled rotations. In what follows, we denote the state $|1^k0^{K-k}\rangle$ simply as $|k\rangle$. We use it to determine which of the times t_j satisfy the condition $j\leqslant k$.

Next we wish to create an equal superposition over the time indices j. We take M to be a power of 2, so the preparation can be performed via the Hadamard transforms $\textsc{Had}^{\otimes\log_2 M}$ on all K subregisters of $\mathtt{time}$ (each of size $\log_2 M$) to generate the superposition state

$$\frac{1}{\sqrt{\mathcal{N}_s}}\sum_{k=0}^{K}\sqrt{\frac{\zeta^kM^k}{M^Kk!}}|k\rangle\sum_{j_1=0}^{M-1}\sum_{j_2=0}^{M-1}\cdots\sum_{j_K=0}^{M-1}|j_1,j_2,\dots,j_K\rangle \quad (45)$$

using $O(K\log_2 M)$ gates. At this stage, we have created all possible K-tuples of times without accounting for time-ordering. Note that this superposition is over *all* possible K-tuples of times, not only $k\leqslant K$ of them. We have a factor of M^k/M^K, but for any k in the superposition we ignore the times in subregisters $k+1$ to K. Hence, the amplitude for $|j_1,j_2,\dots,j_k\rangle$ is $\propto\sqrt{\zeta^k/k!}$.

The next step is to sort the values stored in the $\mathtt{time}$ subregisters. Common classical sorting algorithms are often inappropriate for quantum algorithms, because they involve actions on registers that depend on the values found in earlier steps of the algorithm. Sorting techniques that are suitable to be adapted to quantum algorithms are *sorting networks*, because they involve actions on registers in a sequence that is independent of the values stored in the registers. Methods to adapt sorting networks to quantum algorithms are discussed in detail in Refs. [24,25].

A sorting network involves a sequence of comparators, where each comparator compares the values in two registers and swaps them conditional on the result of the comparison. If used as part of a quantum algorithm, then a sorting network must be made reversible. This is achieved by recording the result of the comparison in an ancilla qubit. To be more specific, first the comparison operation COMPARE acts on two equally sized multiqubit registers storing the values q_1 and q_2 and an ancilla initialized in state $|0\rangle$ as follows:

$$\text{COMPARE}|q_1\rangle|q_2\rangle|0\rangle = |q_1\rangle|q_2\rangle|\theta(q_1 - q_2)\rangle, \tag{46}$$

where θ is the Heaviside step function (here using the convention $\theta(0) = 0$). In other words, COMPARE flips an ancilla if and only if $q_1 > q_2$. Such a comparison of two $\log_2 M$-sized registers can be implemented with $O(\log_2 M)$ elementary gates [27]. The overall action of a comparator module is completed by conditionally swapping the values of the two compared multiqubit registers. This is implemented by SWAP gates controlled by the ancilla.

For each $|k\rangle$ in the superposition Eq. (45), we only want to sort the values in the first k subregisters of `time`. The obvious approach would be to perform K sorts for each value of k. A much more efficient approach is to sort all the registers regardless of the value of k, and also perform the same controlled swaps on the registers encoding the value of k in unary. This means that the qubits encoding k still indicate whether the corresponding time register includes a time we wish to use. The controlled $H_\ell(t)$ operations are controlled on the individual qubits encoding k and will still be performed correctly.

There is a number of possible sorting networks. A simple example is the bitonic sort, and an example quantum circuit for that sort is shown in Fig. 2. Since we need to record the positions of the first k registers as well, we perform the same controlled-swap on the `K` register too. The bitonic sort requires $O(K \log_2^2 K)$ comparisons, but there are more advanced sorting networks that use $O(K \log_2 K)$ comparisons [28]. That brings the complexity of clock preparation to $O(K \log_2 K \log_2 M)$ elementary gates. Since each comparison requires a single ancilla, the space overhead is $O(K \log_2 K)$ ancillas.

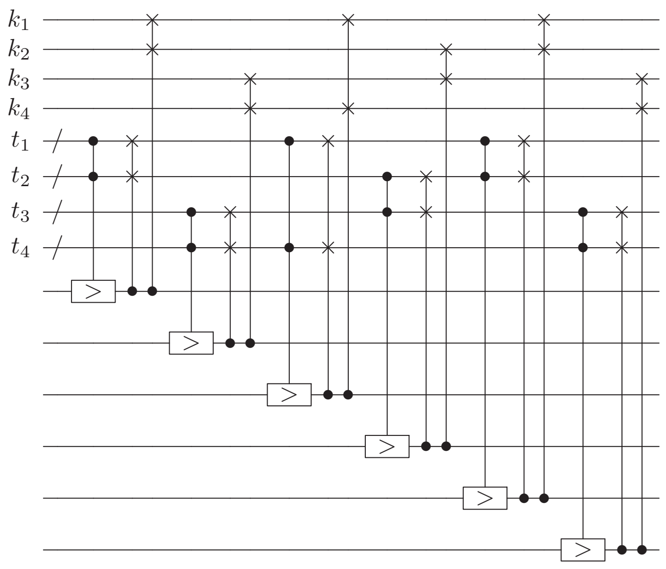

FIG. 2. An example of a bitonic sort for $K = 4$ and 6 ancillas as introduced in Ref. [27]. In each step, two registers encoding times are compared. If the compared values are in decreasing order, then an ancilla qubit is flipped. The states of the involved registers are then swapped conditioned on the value of the ancilla qubit. This results in sorting the values stored in a pair of registers. The same permutation is performed within the `K` register. Note that by recording the result of each comparator in an ancilla qubit the sorting network is made reversible.

C. Completing the state preparation

To complete the state preparation, we transform each of the K `L`-subregisters for encoding $\ell_1, \ldots \ell_K$ from $|0\rangle$ into specific superposition states, depending on the representation of the given Hamiltonian.

In the SM scenario, we only need to prepare equal-weight superposition states. If L is a power of two, then the transformation can be taken to be just Hadamards on the individual qubits. It is always possible to choose the value of γ so as to make L a power of two. The complexity is then $O(K \log_2 L)$ single-qubit operations. It is also possible to create an equal superposition state using $O(K \log_2 L)$ gates if L is not a power of two, though the circuit is more complicated.

In the QC scenario, the target superposition states for the `L`-subregisters to be encoding $\ell_1, \ldots \ell_K$, respectively, are given by Eq. (35). We propose preparing the required target superposition states using a similar approach as in Ref. [29], namely, by transducing the computed values $\widetilde{\alpha}_\ell(t)/\alpha_{\max}$ (using the output of oracle $O_{\texttt{coeff}}$) into quantum amplitudes $\sqrt{\widetilde{\alpha}_\ell(t)/\alpha_{\max}}$ via inequality testing. According to that approach, the state preparation requires the use of several registers. For each $k = 1, \ldots, K$, in addition to the $\texttt{L}_k$ subregister for encoding the index value ℓ_k to identify the coefficients in the decomposition Eq. (17) and the $\texttt{time}_k$ subregister for encoding a particular time t_k as part of the $|\text{clock}\rangle$ state, we introduce two further registers named "$\texttt{coeff}_k$" and "$\texttt{ref}_k$," consisting of ν and $\nu + 1$ qubits, respectively. Here, ν is the number of bits used for the ν-bit fixed-point approximations $\alpha_\ell^{[\nu]}(t) = \lfloor 2^\nu \alpha_\ell(t)/\alpha_{\max} \rfloor$. Finally, an additional single-qubit ancilla called "`flag`" is required to distinguish between the two index ranges $\ell = 0, \ldots, L-1$ and $\ell = L, \ldots, 2L-1$, and thus indicate which of the corresponding two alternative amplitudes $\sqrt{[\alpha_{\max} \pm \alpha_\ell(t)]/2}$ has been generated.

The state preparation by inequality testing can be outlined as follows. For each $k = 1, \ldots, K$, we first create uniform superpositions in both the $\texttt{L}_k$ and $\texttt{ref}_k$ registers using Hadamards. We then query the oracle $O_{\texttt{coeff}}$ defined in Eq. (16) in order to give $\alpha_\ell^{[\nu]}(t)$ at time t specified by the $\texttt{time}_k$ subregister. These first two steps result in transforming each component $|0\rangle_{\mathtt{L}_k}|t\rangle_{\mathtt{time}_k}|0\rangle_{\mathtt{coeff}_k}|0\rangle_{\mathtt{ref}_k}|0\rangle_{\mathtt{flag}_k}$ as part of the superposition over all possible times and values k into

$$\frac{1}{\sqrt{2^{\nu+1}L}}\sum_{\ell=0}^{L-1}\sum_{x=0}^{2^{\nu+1}-1}|\ell\rangle_{\mathtt{L}_k}|t\rangle_{\mathtt{time}_k}\big|\alpha_\ell^{[\nu]}(t)\big\rangle_{\mathtt{coeff}_k}|x\rangle_{\mathtt{ref}_k}|0\rangle_{\mathtt{flag}_k}. \tag{47}$$

Creating the uniform superpositions in the first step requires only $\nu+\log_2 L$ elementary gates. The state preparation then proceeds by performing the inequality test $x \geqslant 2^\nu+\alpha_\ell^{[\nu]}(t)$ and placing the result in the flag qubit. This operation transforms the state Eq. (47) into

$$\frac{1}{\sqrt{L}}\sum_{\ell=0}^{L-1}|\ell\rangle_{\mathtt{L}_k}|t\rangle_{\mathtt{time}_k}\big|\alpha_\ell^{[\nu]}(t)\big\rangle_{\mathtt{coeff}_k} \otimes\left(\frac{1}{\sqrt{2^{\nu+1}}}\sum_{x=0}^{2^\nu+\alpha_\ell^{[\nu]}(t)-1}|x\rangle_{\mathtt{ref}_k}|0\rangle_{\mathtt{flag}_k} +\frac{1}{\sqrt{2^{\nu+1}}}\sum_{x=2^\nu+\alpha_\ell^{[\nu]}(t)}^{2^{\nu+1}-1}|x\rangle_{\mathtt{ref}_k}|1\rangle_{\mathtt{flag}_k}\right). \tag{48}$$

The state within the brackets of Eq. (48) is normalized with amplitudes equal to $\sqrt{\frac{\alpha_{\max}+\alpha_\ell(t)}{2\alpha_{\max}}}$ and $\sqrt{\frac{\alpha_{\max}-\alpha_\ell(t)}{2\alpha_{\max}}}$ on $|0\rangle_{\mathtt{flag}_k}$ and $|1\rangle_{\mathtt{flag}_k}$, respectively, up to error $O(2^{-\nu})$. This error must be bounded by $\epsilon/(Kr)$, implying $\nu=\Theta(\log_2\frac{Kr}{\epsilon})$. The complexity of the inequality test is $\mathcal{O}(\nu)$, and there are $\nu+\log_2 L+1$ Hadamards used to create the equal superposition. These operations are performed K times, giving a gate complexity of

$$O\{K[\log_2(Kr/\epsilon)+\log_2 L]\}. \tag{49}$$

There are also $O(K)$ oracle queries for $\alpha_\ell(t)$.

For the purpose of LCU implementation in our algorithm, it is not required to implement the full algorithm of Ref. [29]. Here it is not necessary to erase the ancillary registers immediately, and there is no need for amplitude amplification. It is allowable to leave the basis states $|\ell\rangle$ entangled with ancillae similar to state preparation for LCU given in Ref. [30]. The state given in Eq. (48) is already operationally equivalent [up to error $\epsilon/(Kr)$] to our target state Eq. (35), if we regard $\mathtt{flag}_k$ as part of an extended $\mathtt{L}_k$-register, while we ignore the entangled registers $\mathtt{coeff}_k$ and $\mathtt{ref}_k$. The entanglement with the latter registers does not affect the subsequent controlled-SELECT(H) operations. Hence, the uncomputation of registers $\mathtt{coeff}_k$ and $\mathtt{ref}_k$ may indeed be deferred to the stage when reversing the above state preparation as part of $B^\dagger$ after completing the SELECT(V) operation.

Next, let us consider the states prepared as a result of these procedures. In the first case, where we prepare the superposition of times by the compressed rotations, we first prepare the state $|\phi_q\rangle^{\otimes K+1}$, which contains the separations between successive times, then add these to obtain the times. The state is therefore

$$\sqrt{\frac{1-\mu^2}{S}}\sum_{k=0}^{K}\zeta^{k/2}|k\rangle\sum_{j_1<j_2\ldots<j_k}|j_1,\ldots,j_k\rangle+\mu|\Omega''\rangle. \tag{50}$$

The preparation of the registers encoding $\ell_1,\ldots\ell_K$ gives a factor of $1/L^{K/2}$ for both considered Hamiltonian models. For $k<K$ the registers past k are not used, and the effective amplitude factor is thus $1/L^{k/2}$.

In the SM scenario the amplitude of the states in the sum is therefore given by

$$\sqrt{\frac{1-\mu^2}{S}}\left(\frac{\zeta}{L}\right)^{k/2}=\sqrt{\frac{1-\mu^2}{S}}\left(\frac{\gamma T}{rM}\right)^{k/2}, \tag{51}$$

where we have used $\zeta=\lambda T/(rM)$ and $\lambda=\gamma L$. According to Eq. (30), the state preparation should give amplitudes that are the square roots of the weights in the linear combination of unitaries, with a normalization factor $\mathcal{N}$ that governs the complexity. The weights are given by Eq. (24) for the SM scenario. We can ignore the factorials because here we have the indices j_1 to j_k in sorted order without repetitions. Therefore, we have the desired amplitudes, with a normalization factor of $\mathcal{N}_{\mathrm{r}}=S/(1-\mu^2)$. There is imprecision of $O(\epsilon/r)$ due to the additional term weighted by μ, as well as imprecision due to the repetitions being omitted, which is bounded in Sec. VI below.

In the QC scenario we have the amplitudes obtained in Eq. (48) of $\sqrt{\widetilde{\alpha}_\ell(t)/\alpha_{\max}}$. These amplitude factors are obtained for each of the times. Using $\lambda=L\alpha_{\max}$, the amplitude of the states in the sum is then given by

$$\sqrt{\frac{1-\mu^2}{S}}\left(\frac{\zeta}{L\alpha_{\max}}\right)^{k/2}\sqrt{\widetilde{\alpha}_{\ell_k}(t_{j_k})\ldots\widetilde{\alpha}_{\ell_1}(t_{j_1})} =\sqrt{\frac{1-\mu^2}{S}}\left(\frac{T}{rM}\right)^{k/2}\sqrt{\widetilde{\alpha}_{\ell_k}(t_{j_k})\ldots\widetilde{\alpha}_{\ell_1}(t_{j_1})}. \tag{52}$$

In the QC scenario the weights are given by Eq. (26), and again we can ignore the factorials because there are no repetitions. Again, we see that we have amplitudes that are given by the square roots of the weights with a normalization factor of $\mathcal{N}_{\mathrm{r}}=S/(1-\mu^2)$. In the QC case there is additional imprecision due to approximating $\alpha_\ell(t)$ to finite precision.

In order for amplitude amplification to take one step, we require that the normalization factor is $\leqslant 2$. Note that it can be less than 2, and oblivious amplitude amplification can still be performed in a single step using an ancilla qubit [12]. To bound the value of S,

$$S=\sum_{|x|\leqslant K}\zeta^{|x|}=\sum_{k=0}^{K}\binom{M}{k}\zeta^k < \sum_{k=0}^{\infty}\frac{M^k\zeta^k}{k!}=e^{M\zeta}=e^{\lambda T/r}. \tag{53}$$

Therefore, by choosing $r\geqslant\lambda T/\ln[2(1-\mu^2)]$ we can ensure that $\mathcal{N}_{\mathrm{r}}\leqslant 2$, and a single step of amplitude amplification is sufficient. The value of μ^2 is $O(\epsilon/r)$, and therefore $r=\Theta(\lambda T)$.

In the second case, where we obtain the superposition over times via a sort, the state is as in Eq. (45) except sorted, with information about the permutation used for the sort in an ancilla. When there are repeated indices j, the number of initial sets of indices that yield the same sorted set of indices is $k!/(k_1!k_2!\ldots k_\varsigma!)$, using the same notation as in Eq. (24). That means for each sorted set of j there is a superposition over this many states in the ancilla with equal amplitude, resulting in a multiplying factor of $\sqrt{k!/(k_1!k_2!\dots k_\varsigma!)}$ for each sorted set of j.

As noted above, because the times t_{k+1} to t_K are ignored, the factors of M in the amplitude cancel. Similarly, when we perform the state preparation for the registers encoding $\ell_1,\dots\ell_K$, we obtain a factor of $1/L^{k/2}$. Including these factors in the amplitude from Eq. (45), as well as the factorials to account for repeated indices, we obtain amplitude

$$\frac{1}{\sqrt{\mathcal{N}_s k_1!k_2!\dots k_\varsigma!}}\left(\frac{\zeta}{L}\right)^{k/2}. \tag{54}$$

This result is the same as for the state preparation by compressed rotations, except the normalization factor is $\mathcal{N}_s$, and we have the factorial factors that account for repeated indices. These factorial factors agree with what is needed for the weightings in Eqs. (24) and (26). Following the same reasoning as for the state preparation by compressed rotations, we obtain the correct amplitudes and now the normalization factor is changed to $\mathcal{N}_s$.

Again we require the normalization factor $\leqslant 2$ for oblivious amplitude amplification to take one step. We can bound $\mathcal{N}_s$ via

$$\mathcal{N}_s = \sum_{k=0}^{K}\frac{M^k\zeta^k}{k!} < \sum_{k=0}^{\infty}\frac{M^k\zeta^k}{k!} = e^{M\zeta} = e^{\lambda T/r}. \tag{55}$$

Therefore, by choosing $r \geqslant \lambda T/\ln 2$ we can ensure that $\mathcal{N}_s \leqslant 2$, and a single step of amplitude amplification is sufficient. We take r to be a power of two, so the qubits encoding the time may be separated into a set encoding the segment number and another set encoding the time within the segment. Hence, in either case we choose $r = \Theta(\lambda T)$.

VI. COMPLEXITY REQUIREMENTS

We now summarize the resource requirements of all the components of the algorithm and provide the overall complexity. We start with elements that are necessary for both sparse matrices and quantum chemistry Hamiltonians and then discuss the complexities of their unique parts.

The full quantum circuit consists of r successively executed blocks, one for each of the r time segments. In order to perform the simulation over the entire time T, we need to perform all r segments. Since we took $r = \Theta(\lambda T)$ for oblivious amplitude amplification, the overall complexity is multiplied by a factor of λT.

Each segment requires one round of oblivious amplitude amplification, which includes two reflections R, two applications of W and one application of the inverse $W^\dagger$, as in Eq. (34). The cost of reflections is negligible compared to that of W. Hence, the overall cost of the algorithm amounts to $O(r)$ times the cost of transformation W, whose quantum circuit is depicted in Fig. 1. For the SM scenario, implementing W requires the procedure to prepare the $|\text{clock}\rangle$ state and its inverse, $2K$ applications of $\text{HAD}^{\otimes \log_2 L}$ and K controlled applications of unitaries $H_\ell(t)$. For the QM scenario, implementing W requires likewise preparing the $|\text{clock}\rangle$ state and the reverse of that procedure, $2K$ applications of the controlled-$\text{PREP}(\alpha)$ subroutine, and K controlled applications of H_ℓ.

Choosing K as in Eq. (23) yields error due to truncation for each segment scaling as $O(\epsilon/r)$, and therefore total error due to truncation scaling as $O(\epsilon)$. Taking $r = \Theta(\lambda T)$, K is chosen as

$$K = \Theta\left[\frac{\log_2(\lambda T/\epsilon)}{\log_2\log_2(\lambda T/\epsilon)}\right]. \tag{56}$$

Note that $\lambda \geqslant H_{\max}$ implies that the condition $r \geqslant H_{\max}T$ is satisfied, which was required for deriving Eq. (23). Recall that $\lambda = L\gamma$ in the SM scenario and $\lambda = L\alpha_{\max}$ for the QC scenario.

It was stated in Sec. IV that the error due to discretization of the integrals is $O[(T/r)^2\dot{H}_{\max}/M]$. That can be seen as follows. When approximating the integral over each time variable by a sum with M terms,

$$\begin{aligned}\int_0^{T/r} H(t)dt &= \sum_{j=0}^{M-1}\int_{t_j}^{t_{j+1}} H(t)dt \\ &= \sum_{j=0}^{M-1}\left(\frac{T}{rM}\right)H(t_j) \\ &\quad + \sum_{j=0}^{M-1}\int_{t_j}^{t_{j+1}}[H(t)-H(t_j)]dt \\ &\approx \sum_{j=0}^{M-1}\left(\frac{T}{rM}\right)H(t_j),\end{aligned} \tag{57}$$

where $t_j = jT/(rM)$, the incurred error can obviously be bounded by $\sum_{j=0}^{M-1}\int_{t_j}^{t_{j+1}}\|H(t)-H(t_j)\|dt$. In each of the time intervals $[t_j, t_{j+1}]$ of length $T/(rM)$ we have

$$\|H(t)-H(t_j)\| \leqslant \frac{T}{rM}\max_z\left\|\frac{dH(z)}{dz}\right\| \leqslant \frac{T\dot{H}_{\max}}{rM}, \tag{58}$$

where $\max_z$ indicates a maximum over that time interval. Hence, the overall error of approximation in the integral over time T/r is $O[(T/r)^2\dot{H}_{\max}/M]$. That is the error in the $k=1$ term for $\widetilde{U}$, and the error for terms with $k>1$ are higher order. The error for all r segments is then $O[T^2\dot{H}_{\max}/(rM)]$. We can ensure that the error due to the discretization of the integrals is $O(\epsilon)$, by choosing

$$M \in \Theta\left(\frac{T\dot{H}_{\max}}{\epsilon\lambda}\right), \tag{59}$$

where we have used $r \in \Theta(\lambda T)$. We remark that a slightly tighter bound in terms of a time-average of $\dot{H}(t)$ rather than in terms of $\dot{H}_{\max}$ is possible, as was achieved in Ref. [21].

In the case where the $|\text{clock}\rangle$ state is prepared using the compressed form of rotations, the operation that is performed is a little different than desired, because all repeated times are omitted. For each k, the proportion of cases with repeated times is an example of the birthday problem, and is approximately $k(k-1)/(2M)$. Therefore, denoting by $\widetilde{U}_{\text{unique}}$ the operation corresponding to $\widetilde{U}$ but with repeated times omitted, we have

$$\begin{aligned}\|\widetilde{U}-\widetilde{U}_{\text{unique}}\| &\lesssim \sum_{k=2}^{K}\frac{(T/r)^k}{k!}\frac{k(k-1)}{2M}H_{\max}^k\\ &= \sum_{k=0}^{K}\frac{(T/r)^{k+2}}{k!}\frac{1}{2M}H_{\max}^{k+2}\\ &< \frac{T^2H_{\max}^2}{2r^2M}e^{TH_{\max}/(rM)}.\end{aligned} \tag{60}$$

Therefore, over r segments the error due to omitting repeated times is upper bounded by

$$r\|\widetilde{U}-\widetilde{U}_{\text{unique}}\| \lesssim \frac{T^2H_{\max}^2}{2rM}e^{TH_{\max}/(rM)}. \tag{61}$$

Using $r\in\Theta(\lambda T)$ and $\lambda\geqslant H_{\max}$, we should therefore choose

$$M=\Omega\left(\frac{TH_{\max}^2}{\epsilon\lambda}\right). \tag{62}$$

In the case where the repeated times are omitted, we would therefore take

$$M=\Theta\left[\frac{T}{\epsilon\lambda}\left(H_{\max}^2+\dot{H}_{\max}\right)\right]. \tag{63}$$

Next we consider the gate complexity for the $|\text{clock}\rangle_{\mathtt{K\otimes time}}$ state preparation. In the case of the compressed rotation encoding, the complexity is $O(K\log_2 M)$, where we have used $r\in\Theta(\lambda T)$. In the case where $|\text{clock}\rangle_{\mathtt{K\otimes time}}$ is prepared with a quantum sort, the complexity is $O(K\log_2 M\log_2 K)$.

A. Complexity for SM scenario

Let us now consider the complexity of simulation of a Hamiltonian given as a sparse matrix. The preparation of the registers $\ell_1,\dots,\ell_k$ requires only creating an equal superposition. It is easiest if L is a power of two, in which case it is just Hadamards. It can also be achieved efficiently for more general L, and in either case the complexity is $O(K\log_2 L)$ elementary gates.

Next, one needs to implement the controlled unitaries H_ℓ. Each controlled H_ℓ can be implemented with $O(1)$ queries to the oracles that give the matrix entries of the Hamiltonian [12]. Since the unitaries are controlled by $O(\log_2 L)$ qubits and act on n qubits, each control-H_ℓ requires $O(\log_2 L+n)$ gates. Scaling with M does not appear here, because the qubits encoding the times are used just as input to the oracles, and no direct operations are performed. As there are K controlled operations in each segment, the complexity of this step for a segment is $O(K)$ oracle queries and $O[K(\log_2 L+n)]$ additional gates.

As the total cost for the entire simulation is obtained by multiplying the cost per segment by the number r of all segments, the overall oracle query complexity thus amounts to

$$O(\lambda TK). \tag{64}$$

Here $\lambda=L\gamma$ and L is chosen as in Eq. (13) as $\Theta(d^2H_{\max}/\gamma)$, so $\lambda\in O(d^2H_{\max})$. In addition, K is given by Eq. (56), giving the total query complexity

$$O\left[d^2H_{\max}T\frac{\log_2(dH_{\max}T/\epsilon)}{\log_2\log_2(dH_{\max}T/\epsilon)}\right]. \tag{65}$$

The overall complexity in terms of the additional gates is larger and depends on the scheme used to prepare the state $|\text{clock}\rangle_{\mathtt{K\otimes time}}$. In the case where the preparation is performed via the compressed encoding, we obtain the complexity

$$O[\lambda TK(\log_2 L+\log_2 M+n)]. \tag{66}$$

Regardless of the preparation approach, γ is chosen as in Eq. (12) as $\Theta[\epsilon/(d^3T)]$. Moreover, $L\in\Theta(d^5H_{\max}T/\epsilon)$, whereas the first term in the scaling for M in Eq. (63) is $\Theta[H_{\max}T/(\epsilon d^2)]$. This means that the first term in Eq. (63) can be ignored in the overall scaling due to the $\log_2 L$, and we obtain an overall scaling of the gate complexity as

$$O\left\{d^2H_{\max}T\frac{\log_2(dH_{\max}T/\epsilon)}{\log_2\log_2(dH_{\max}T/\epsilon)}\times\left[\log_2\left(\frac{dH_{\max}T}{\epsilon}\right)+\log_2\left(\frac{\dot{H}_{\max}T}{\epsilon dH_{\max}}\right)+n\right]\right\}. \tag{67}$$

There is also complexity of $O(r\log_2 r)$ for the increments of the register recording the segment number, but it is easily seen that this complexity is no larger than the other terms above.

In the case where $|\text{clock}\rangle_{\mathtt{K\otimes time}}$ is prepared using a sorting network, we obtain complexity

$$O[\lambda TK(\log_2 L+\log_2 M\log_2 K+n)], \tag{68}$$

with M given by Eq. (59). Technically this complexity is larger than that in Eq. (67), because of the multiplication by $\log_2 K$. Nevertheless, it is likely that in practice the preparation by a sorting network may turn out to be advantageous in some situations, because the value of M that is required for the first preparation scheme is larger. This is because the compressed encoding approach does not contain repeated times whereas the approach by sorting networks does.

B. Complexity for QC scenario

Let us now consider the cost of simulating the time evolution generated by quantum-chemistry type Hamiltonians. For each single time segment, we have $O(K)$ applications of the controlled-PREP(α) operation, which requires $O(K)$ queries to O_{coeff}. Furthermore, we have $O(K)$ controlled-SELECT(H) operations as part of SELECT(V) transformation, and thus require $O(K)$ queries to O_{unit}. The query complexities for $O_{\mathtt{coeff}}$ and O_{unit} are added to give the overall query complexity for the simulation over all r time segments $O(rK)$. By using $r\in\Theta(\lambda T)$ and $\lambda=L\alpha_{\max}$ as well as the choice for K given in Eq. (56), the oracle complexity is

$$O\left[L\alpha_{\max}T\frac{\log_2(L\alpha_{\max}T/\epsilon)}{\log_2\log_2(L\alpha_{\max}T/\epsilon)}\right]. \tag{69}$$

For each time segment, preparation of the auxiliary L-register states requires $O[K(\log_2 L+\log_2(rK/\epsilon)]$ additional elementary gates, as displayed earlier in Eq. (49). Furthermore, we need $O(K)$ gates to implement the minus signs given in Eq. (19) as well as the the $(-i)^k$ factors occurring in the Dyson series. Additional gate complexity comes from the preparation of the $|\text{clock}\rangle_{\mathtt{K\otimes time}}$ state, which is the same as for the SM scenario. Preparation via compressed encoding thus requires in total $O\{\lambda TK[\log_2 L+\log_2 M+\log_2(\lambda TK/\epsilon)]\}$ gates, where we have used $r\in\Theta(\lambda T)$. Using once more

$\lambda = L\alpha_{\max}$ and Eqs. (56) and (63) together with $H_{\max} \leqslant L\alpha_{\max}$, this finally yields

$$O\left\{\alpha_{\max}LT\frac{\log_2(\alpha_{\max}LT/\epsilon)}{\log_2\log_2(\alpha_{\max}LT/\epsilon)}\right.$$
$$\left.\times\left[\log_2\left(\frac{\alpha_{\max}LT}{\epsilon}\right)+\log_2\left(\frac{\dot{H}_{\max}T}{\epsilon\alpha_{\max}L}\right)+\log_2 K+\log_2 L\right]\right\} \tag{70}$$

as the overall gate complexity in this case, where we again ignored the first term in Eq. (63) in the overall scaling due to the occurrence of the larger $\log_2(\alpha_{\max}LT/\epsilon)$ term. The last two additive terms $\log_2 K$ and $\log_2 L$ in the square brackets of Eq. (70) are smaller than the first term, so are omitted in Theorem 2 in Sec. II. Similarly, we obtain the gate complexity $O\{\lambda TK[\log_2 L + \log_2 M \log_2 K + \log_2(\lambda TK/\epsilon)]\}$ for implementations based on quantum sorting networks, which is slightly larger due to the additional $\log_2 K$ factor in the second term.

Scaling with n (i.e., with the number of system qubits) does not appear in Eqs. (69) and (70), because the system qubits are just used as input to the oracle $O_{\texttt{unit}}$ [defined in Eq. (15)] as part of controlled-SELECT(H) operations. The scaling with n is therefore hidden in the gate cost of implementing that oracle.

VII. CONCLUSION

We have provided a quantum algorithm for simulating the evolution generated by a time-dependent Hamiltonian that is given either by a generic d-sparse matrix or by a time-dependent linear combination of some efficiently implementable unitary terms. For both scenarios, the complexity of the algorithm scales logarithmically in both the dimension of the Hilbert space and the inverse of the error of approximation ϵ. This is an exponential improvement in error scaling compared to techniques based on Lie-Trotter-Suzuki expansion. We utilize the truncated Dyson series, which is based on the truncated Taylor series approach by Berry *et al.* [14]. It achieves similar complexity, in that it has T times a term logarithmic in $1/\epsilon$. Interestingly, the complexity in terms of queries to the Hamiltonian is independent of the rate of change of the Hamiltonian, similarly to Ref. [10].

A lower bound on the complexity of Hamiltonian simulation is $O[T\|H\| + \frac{\log_2(1/\epsilon)}{\log_2\log_2(1/\epsilon)}]$. This bound was proven in the time-independent case [12], and also holds for time-dependent Hamiltonians because that is a more general case. Our complexity is optimal in that it is linear with respect to T and the norm of the Hamiltonian, and logarithmic in $1/\epsilon$. Our result does not completely match the lower bound, in that we have a product of the terms, rather than a sum of the terms as in the lower bound. In the time-independent case it is known that it is possible to match the lower bound [15], and it is an interesting open question whether it is possible to achieve similar complexity in the time-dependent case.

For the complexity in terms of additional gates, the complexity is somewhat larger, and it depends logarithmically on the rate of change of the Hamiltonian. This is in contrast with Ref. [10], which has complexity independent of the derivative. However, it should be noted that [10] has a classical gate complexity that does depend on the derivative. That is because the times are generated to finite precision by a classical random number generator. Those times need to be generated to a sufficient number of bits to capture the time variation of the Hamiltonian, in the same way as we need to generate time registers with a number of qubits depending logarithmically on the derivative of the Hamiltonian here.

The complexity also depends on the scheme that is used to prepare the state to represent the times. If one uses the scheme as in Ref. [23], which corresponds to a compressed form of a tensor product of small rotations, then there is additional error due to omission of repeated times. Alternatively one could prepare a superposition of all times and sort them, which eliminates that error but gives a multiplicative factor in the complexity. This tradeoff means that different approaches may be more efficient with different combinations of parameters.

Note added. An alternative approach has also been proposed in Ref. [21]. That work uses a simpler method of preparing the state representing the times by discarding times rather than ordering them. That results in a multiplicative cost.

ACKNOWLEDGMENTS

M.K. thanks Nathan Wiebe for insightful discussions. D.W.B. is funded by Australian Research Council Discovery Projects (Grants No. DP160102426 and No. DP190102633).

[1] J. Preskill, Quantum **2**, 79 (2018).
[2] D. S. Abrams and S. Lloyd, Phys. Rev. Lett. **79**, 2586 (1997).
[3] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Science **309**, 1704 (2005).
[4] R. P. Feynman, Int. J. Theor. Phys. **21**, 467 (1982).
[5] I. M. Georgescu, S. Ashhab, and F. Nori, Rev. Mod. Phys. **86**, 153 (2014).
[6] S. Lloyd, Science **273**, 1073 (1996).
[7] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, J. Phys. A: Math. Theoret. **43**, 065203 (2010).
[8] N. Wiebe, D. W. Berry, P. Høyer, and B. C. Sanders, J. Phys. A: Math. Theoret. **44**, 445308 (2011).
[9] A. M. Childs, Commun. Math. Phys. **294**, 581 (2010).
[10] D. Poulin, A. Qarry, R. Somma, and F. Verstraete, Phys. Rev. Lett. **106**, 170501 (2011).
[11] D. W. Berry and A. M. Childs, Quantum Inf. Comput. **12**, 0029 (2012).
[12] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC'14)* (ACM, New York, 2014), pp. 283–292.

[13] D. W. Berry, A. M. Childs, and R. Kothari, in *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS'15)* (IEEE, 2015), pp. 792–809.
[14] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Phys. Rev. Lett. **114**, 090502 (2015).
[15] G. H. Low and I. L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017).
[16] G. H. Low and I. L. Chuang, arXiv:1610.06546 (2016).
[17] G. H. Low, T. J. Yoder, and I. L. Chuang, Phys. Rev. X **6**, 041067 (2016).
[18] S. Pang and A. N. Jordan, Nat. Commun. **8**, 14695 (2017).
[19] L. J. Butler, Annu. Rev. Phys. Chem. **49**, 125 (1998).
[20] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, Science **292**, 472 (2001).
[21] G. H. Low and N. Wiebe, arXiv:1805.00675 (2018).
[22] A. M. Childs and N. Wiebe, Quantum Inf. Comput. **12**, 901 (2012).
[23] D. W. Berry, R. Cleve, and S. Gharibian, Quantum Inf. Comput. **14**, 0001 (2014).
[24] S.-T. Cheng and C.-Y. Wang, IEEE Trans. Circuits Syst. I: Reg. Papers **53**, 316 (2006).
[25] R. Beals, S. Brierley, O. Gray, A. W. Harrow, S. Kutin, N. Linden, D. Shepherd, and M. Stather, Proc. R. Soc. A **469** 20120686 (2013).
[26] R. Babbush, D. W. Berry, I. D. Kivlichan, A. Y. Wei, P. J. Love, and A. Aspuru-Guzik, New J. Phys. **18**, 033032 (2016).
[27] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney, and R. Babbush, npj Quant. Info. **4**, 22 (2018).
[28] M. T. Goodrich, in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC'14)* (ACM, New York, 2014), pp. 684–693.
[29] Y. R. Sanders, G.-H. Low, A. Scherer, and D. W. Berry, Phys. Rev. Lett. **122**, 020502 (2019).
[30] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Phys. Rev. X **8**, 041015 (2018).