Universal algorithm for transforming Hamiltonian eigenvalues

Tatsuki Odake,[1] Hlér Kristjánsson,[2, 3, 1] Philip Taranto,[1] and Mio Murao[1]

[1] *Department of Physics, Graduate School of Science, The University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-0033, Japan*
[2] *Perimeter Institute for Theoretical Physics, 31 Caroline Street North, Waterloo, Ontario, N2L 2Y5, Canada*
[3] *Institute for Quantum Computing, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, N2L 3G1, Canada*

Manipulating the Hamiltonians governing physical systems has found a broad range of applications, from quantum chemistry to semiconductor design. In this work, we provide a new way of manipulating Hamiltonians, by transforming their eigenvalues while keeping their eigenvectors fixed. If a classical description of the initial Hamiltonian is known, then one can—in principle—diagonalize it and compute the Hamiltonian transformation on a classical computer. However, this comes with a significant computational cost, and a classical description of the initial Hamiltonian is not always available, in particular for complex systems. In this work, we develop a universal algorithm that deterministically implements any desired (suitably differentiable) function on the eigenvalues of any unknown Hamiltonian, whose dynamics is given as a black box. Our algorithm makes use of correlated randomness to efficiently combine two subroutines—namely controlization and Fourier series simulation—using a general compilation procedure developed in this work. We show that the runtime of our algorithm is significantly reduced using our general compilation framework, compared to a naïve concatenation of the subroutines, and moreover outperforms similar methods based on the quantum singular value transformation.

I. INTRODUCTION

The physical properties of any given system are determined by its Hamiltonian. As such, realizing a Hamiltonian which exhibits properties that are desirable in a given context is an important task in fields ranging from condensed matter physics and materials science to quantum chemistry. For example, an important problem in materials science is the discovery of new materials for specific tasks [1, 2].

In many cases, the physical properties of interest depend in particular on the eigenvalues or eigenstates of a Hamiltonian. The eigenstates of a given Hamiltonian can be easily transformed through a unitary evolution, however, tuning the eigenvalues of a Hamiltonian while keeping the eigenstates fixed is a much more difficult problem. The ability to transform Hamiltonian eigenvalues in an efficient manner would open up a new way of manipulating the physical properties of a system, leading to increased flexibility in the simulation, control, and design of quantum systems.

In this work, we propose a quantum algorithm to transform the eigenvalues of a Hamiltonian by any given (suitably differentiable) function, while keeping the eigenstates fixed. Our algorithm is *universal* in the sense that it does not require knowledge of the Hamiltonian, whose dynamics can be given as a black box. In the standard setting of Hamiltonian simulation, a classical description of the Hamiltonian to be transformed is known in advance [3–9]; as such, the task is to construct a circuit composed of a minimal number of high-fidelity gates that implements the desired Hamiltonian. However, in many cases of interest, the Hamiltonian of the system is not available but rather only its dynamics is accessible, especially for many-body or complex systems. Our algorithm can be seen as an alternative Hamiltonian simulation method which is based on transformations of black-box dynamics [10]. Possible applications of our method include constructing oracles for Grover's algorithm [11], and we envisage that further developments in this direction could find applications more broadly in condensed matter physics, chemistry, and material science, for example in tuning energy gaps to enable precise transitions between energy levels [12] and in the materials design of semiconductors [13].

To achieve our universal Hamiltonian eigenvalue transformation, we employ the framework of higher-order quantum operations, a research area that has attracted significant attention in recent years [14–24]. The adjective "higher-order" here refers to the fact that such processes take *quantum dynamics* as input, returning a transformed quantum dynamics as output, e.g., by appropriately applying quantum operations before and after the dynamics (see Fig. 1). We construct a method to achieve this by first controlizing the black-box dynamics [16] and then applying a novel algorithm for Fourier series simulation on the controlled black-box dynamics, which implements the desired function on the eigenvalues.

Our methodology shows how component subroutines can be compiled to build complex and efficient algorithms from simpler ones. Concatenating subroutines—as we do with controlization and Fourier series simulation—provides an instance of quantum functional programming [21, 25, 26], which endows quantum algorithms with a modular flexibility similar to their classical counterparts. Yet, whenever two subroutines relying on random sampling are concatenated directly, the overall runtime scales poorly due to the independence of the two sampling distributions. To overcome this issue, we develop a general *compilation* procedure for the concatenation of randomised algorithms, which makes use of correlated randomness as a key resource to optimize the algorithm at a global level. As such, our compilation technique goes beyond the existing techniques of quantum functional programming,

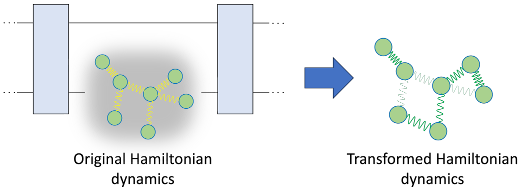

Original Hamiltonian dynamics
Transformed Hamiltonian dynamics

FIG. 1. *Higher-order quantum transformation.*—By appropriately applying fixed quantum operations before and after the unknown Hamiltonian dynamics, the physical properties of a system can be transformed as desired.

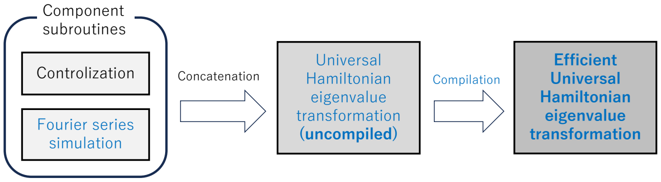

Component subroutines
Controlization
Fourier series simulation
Concatenation
Universal Hamiltonian eigenvalue transformation (uncompiled)
Compilation
Efficient Universal Hamiltonian eigenvalue transformation

FIG. 2. *Description of our work.*—Tasks written in blue denote our contributions. We first develop an "uncompiled" algorithm for UHET by concatenating two subroutines, controlization and Fourier series simulation. We then present a "compiled" version of the UHET algorithm, which improves upon the efficiency.

leading to more efficient compiled algorithms. By introducing the notion of compilation to functional programming in the context of quantum information processing, we align it with classical software design principles that allow code to be compiled and executed at various levels of abstraction, opening the possibilities for computations to be deployed across many different applications.

The reminder of the paper will be structured as follows. First, we present a summary of our main results in Sec. II. In Sec. III, we formalize the task of universal Hamiltonian eigenvalue transformation and the envisaged scenario. The subsequent three sections each pertain to our key results. In Sec. IV, we combine two subroutines, controlization and Fourier series simulation, to achieve said task. Then, in Sec. V, we use the intuition gained in order to construct a more efficient "compiled" algorithm. Finally, in Sec. VI, we compare our new methods with a protocol based upon the Quantum Singular Value Transformation (**QSVT**) [27, 28]. We finish with some concluding discussions in Sec. VII.

II. SUMMARY OF MAIN RESULTS

The main results of our work are schematically depicted in Fig. 2 and summarised as follows.

1. In Sec. IV, we develop an algorithm that performs a universal Hamiltonian eigenvalue transformation by concatenating two subroutines, namely controlization [16] and Fourier series simulation. The controlization subroutine adds control to an unknown Hamiltonian dynamics H; by approximating the Fourier series of a desired function f, one can then use this controlled dynamics to simulate the dynamics of the transformed Hamiltonian $f(H)$. Both subroutines are implemented efficiently using a randomized Hamiltonian simulation technique.

2. In Sec. V, we present a method, hereby dubbed "compilation", which provides a framework for constructing a more efficient circuit of any quantum algorithm constructed by concatenating subroutines that make use of randomized Hamiltonian simulation. We achieve this by employing the resource of temporally correlated randomness in order to optimize the overall task at hand rather than the individual modules as per the "uncompiled" algorithm. In our case of UHET, we show that this leads to a more efficient algorithm.

3. In Sec. VI, we compare the performance of our two new algorithms for transforming eigenvalues of an unknown Hamiltonian with a method based upon the Quantum Singular Value Transformation (QSVT). We show that our compiled algorithm has a better runtime than the QSVT-based algorithm, which in turn outperforms our uncompiled algorithm in this regard.

We begin by formalizing the task and our envisaged framework.

III. TASK: UNIVERSAL HAMILTONIAN EIGENVALUE TRANSFORMATION (UHET)

The goal of *universal Hamiltonian eigenvalue transformation* (**UHET**) is to simulate the dynamics of a desired function of an unknown input Hamiltonian H. Here, the term "simulate" refers to implementing the corresponding Hamiltonian dynamics for an arbitrary evolution time up to an acceptable approximation error. More precisely, given a function $f : [-1, 1] \to \mathbb{R}$, time $t > 0$, and precision $\epsilon > 0$, a universal Hamiltonian eigenvalue transformation uses an *unknown* input dynamics $e^{\pm iH\tau}$ ($\tau > 0$) to approximate a desired transformed dynamics $e^{-if(H_0)t}$ up to precision ϵ and for all t.

Here, H is associated to an n-qubit Hilbert space $\mathcal{H}$ and $H_0 := H - (\mathrm{tr}(H)/2^n)I$ is its traceless part, which we assume to be upper bounded $\|H_0\|_{\mathrm{op}} \leq 1$. Note that taking only the traceless part of the Hamiltonian leads to no loss of generality, since for $H_1 = H_2 + \alpha I$ where H_1, H_2 are Hamiltonians and $\alpha \in \mathbb{R}$, $e^{-iH_1 t} = e^{-i\alpha t}e^{-iH_2 t}$, thus the dynamics corresponding to H_1 and H_2 are equivalent (up to a global phase). Furthermore, as long as an upper bound Δ_H of the difference between the maximum and the minimum energy eigenvalues of the Hamiltonian is known, then one can always rescale the Hamiltonian as $H \to H/\Delta_H$ and change the definition of the function f as $f \to f_H$ such that $f_H(x) := f(\Delta_H x)$, justifying the assumption $\|H_0\|_{\mathrm{op}} \leq 1$. We note that $f(H_0)$ is defined in the usual way, i.e., by applying f to the spectrum of H_0. Lastly, we assume access to both the positive and negative time dynamics; if only the positive time dynamics is accessible, then the negative time one can be constructed with a small overhead using the algorithm in [10]. In fact, the negative time dynamics can be simulated using a special case of the algorithm in [10] which does not require the use of an auxiliary system.

As mentioned previously, transforming Hamiltonians has applications ranging from condensed matter physics to quantum information processing. In many situations considered, at least a classical description of the initial seed Hamiltonian is known, e.g., in algorithms based upon QSVT. However, from a physics perspective, this scenario is not always the case; for instance, one may have access to a time-evolving many-body system without knowing any description of the dynamics *a priori*. Our setting of *universal* Hamiltonian transformations allows such scenarios: the classical description of the input Hamiltonian H can be completely unknown, as long as its (positive and negative time) dynamics $e^{\pm iH\tau}$ is accessible.

Throughout the remainder of this article, we will present various methods to achieve UHET. At their core, all such methods make use of classical randomness to simulate the desired dynamics by sampling many rounds of evolution; being approximate, one must therefore take the iteration number N sufficiently large to ensure a desired accuracy $\epsilon > 0$. To quantify said error in any such scheme, we make use of the following distance measures. First, note that simulating any Hamiltonian dynamics for a specified amount of time (ideally) leads to a unitary operation. Thus, we consider the situation where one attempts to simulate such a unitary operation $\mathcal{U} : \mathcal{L}(\mathcal{H}) \to \mathcal{L}(\mathcal{H})$ by a random protocol $\sum_j p_j \mathcal{F}_j$ where the index j is chosen with a probability p_j and $\mathcal{F}_j : \mathcal{L}(\mathcal{H}) \to \mathcal{L}(\mathcal{H})$ is the corresponding quantum operation. If the input state to said dynamics is not fixed (i.e., remains arbitrary), then we use the following error measure for the quantum operation:

$$\sup_{\substack{\dim(\mathcal{H}') \\ |\psi\rangle \in \mathcal{H}\otimes\mathcal{H}' \\ \||\psi\rangle\|=1}} \|\mathcal{U}\otimes\mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|) - \sum_j p_j \mathcal{F}_j \otimes \mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|)\|_1. \tag{1}$$

Here, $\mathcal{I}$ represents the identity channel, $\mathcal{H}'$ is an auxiliary Hilbert space of arbitrary dimension and $\|\cdot\|_1$ denotes the 1-norm. If, on the other hand, the input state is specified to be some known $|\psi\rangle \in \mathcal{H}$, then the accuracy of any protocol can be determined by comparing the post-transformation state with the ideal case, and hence we use the following error measure for quantum states:

$$\|\mathcal{U}(|\psi\rangle\langle\psi|) - \sum_j p_j \mathcal{F}_j(|\psi\rangle\langle\psi|)\|_1. \tag{2}$$

When the average state $\sum_j p_j \mathcal{F}_j(|\psi\rangle\langle\psi|)$ approximates the target state $\mathcal{U}(|\psi\rangle\langle\psi|)$ [in terms of Eq. (2)] with an error that is upper bounded by a fixed constant ϵ for *any* input state $|\psi\rangle$, then the mean square of the approximation is upper bounded by 2ϵ for any input, i.e.,

$$\sum_j p_j \|\mathcal{U}(|\psi\rangle\langle\psi|) - \mathcal{F}_j(|\psi\rangle\langle\psi|)\|_1^2 \leq 2\epsilon \tag{3}$$

(see App. B 1 for technical details).

IV. UNCOMPILED UHET ALGORITHM

We now move to present our first algorithm that performs the task of UHET. This algorithm is based upon a random protocol for simulating Hamiltonian dynamics. Our method can be seen as an extension of a Hamiltonian simulation technique called **qDRIFT** [5] to the case where the Hamiltonian is unknown. Using this technique as a basis, we concatenate two subroutines, namely controlization and Fourier series simulation, to develop our overall UHET algorithm. Finally, we present a resource and runtime analysis of the protocol. Additional technical details are provided throughout App. C.

A. Efficient Hamiltonian Simulation (qDRIFT)

Suppose one has access to a set of Hamiltonian dynamics, i.e., the ability to perform $e^{-iH_j\tau}$ $(\tau > 0)$ for a set of Hamiltonians $\{H_j\}$. Assuming w.l.o.g. that the Hamiltonians are normalized $\|H_j\|_{\mathrm{op}} = 1$, any dynamics of the form $e^{-i(\sum_j h_j H_j)t}$ $(t > 0)$ for a set of positive coefficients $h_j > 0$ can be approximated by the following protocol (see Fig. 3):

1. Define $\lambda := \sum_k h_k$ and the probability distribution $p_j := h_j/\lambda$, from which an index j is randomly sampled.
2. Apply the dynamics $e^{-iH_j t\lambda/N}$.
3. Repeat steps (1)–(2) N times.

This method is based on the Hamiltonian simulation technique qDRIFT [5], which makes use of the Trotter-Suzuki decomposition [4] of $e^{-i(\sum_j h_j H_j)t}$ to approximate $e^{-i(\sum_j p_j H_j)\delta t}$ for a small time interval $\delta t := t\lambda/N$ up to the first order of δt via a probabilistic mixture of dynamics $e^{-iH_j\delta t}$ with probability p_j. In order to suppress the approximation error below ϵ, taking the iteration number

$$N(\lambda, t, \epsilon) := \mathrm{ceil}[\max(10\lambda^2 t^2/\epsilon, 5\lambda t/2)] \tag{4}$$

is sufficient (see App. C 1).

We now describe the two subroutines that leverage this qDRIFT primitive for *unknown* Hamiltonians, namely controlization [16] and Fourier series simulation, to perform UHET upon their concatenation. We refer to such a straightforward concatenation as *uncompiled*, in contrast to a later algorithm which we dub *compiled* that optimizes UHET globally (see Sec. V).

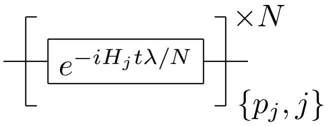


FIG. 3. *qDRIFT primitive.*—Circuit representation of the qDRIFT protocol for approximating $e^{-i(\sum_j h_j H_j)t}$ by randomly implementing $e^{-iH_j\tau}$ with probability p_j for each j for a number N times.

B. Controlization

Controlization is a method that adds control to an unknown Hamiltonian dynamics [16]. It takes a finite number of queries to the dynamics $e^{-iH\tau}$ as a resource and outputs a random unitary operator approximating $\mathtt{ctrl}(e^{-iH_0 t}) \in \mathcal{L}(\mathcal{H}_c \otimes \mathcal{H})$, where H_0 represents the traceless part of H, $\mathcal{H}_c$ is the Hilbert space associated to the control qubit, and $\mathtt{ctrl}(e^{-iH_0 t}) := |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes e^{-iH_0 t}$ represents controlled-$e^{-iH_0 t}$. Our following *uncompiled algorithm* for UHET first makes use of a variant of controlization which simulates $\mathtt{ctrl}_0(e^{-iH_0 t}) := |0\rangle\langle 0| \otimes e^{-iH_0 t} + |1\rangle\langle 1| \otimes I$ by reversing the roles of $|0\rangle$ and $|1\rangle$ of the control qubit, as presented in Subroutine 1 and the circuit in Fig. 4.

Subroutine 1 Controlization

Input:

- A finite number of queries to a black-box Hamiltonian dynamics $e^{-iH\tau}$ of a seed Hamiltonian H normalized as $\|H_0\|_{\mathrm{op}} \le 1$, where H_0 is the traceless part of H, i.e., $H_0 := H - (1/2^n)\mathrm{tr}(H)I$, with $\tau > 0$
- Allowed error $\epsilon > 0$
- Time $t > 0$

Output: A random unitary operator approximating

$$\begin{pmatrix} e^{-iH_0 t} & 0 \\ 0 & I \end{pmatrix} = \mathtt{ctrl}_0(e^{-iH_0 t}) \in \mathcal{L}(\mathcal{H}_c \otimes \mathcal{H}) \tag{5}$$

with an error according to Eq. (1) upper bounded by ϵ

Runtime: $\Theta(t^2 n/\epsilon)$
Used Resources:
System: n-qubit system $\mathcal{H}$ and one auxiliary qubit $\mathcal{H}_c$
Gates: $e^{-iH\tau}$ and Clifford gates on $\mathcal{H}_c \otimes \mathcal{H}$

Procedure:

```
Pre-processing:
1: Compute $N := N(1, t, \epsilon)$ using $N(\lambda, t, \epsilon)$ from Eq. (4)
Main Process:
2: Initialize $U_{\mathrm{current}} \leftarrow I$
3: for $m = 1, \ldots, N$ do
4:     Randomly choose $\vec{v} \in \{0,1,2,3\}^n$
5:     $U_{\mathrm{current}} \leftarrow \mathtt{ctrl}(\sigma_{\vec{v}})(I \otimes e^{-iHt/N})\mathtt{ctrl}(\sigma_{\vec{v}})$
6: end for
7: Return $U_{\mathrm{current}}$
```

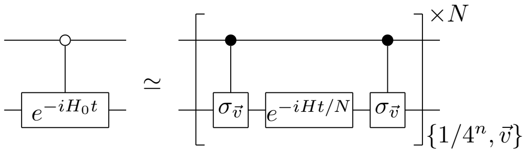


FIG. 4. *Controlization circuit.*—By randomly applying controlled Pauli gates $\mathtt{ctrl}(\sigma_{\vec{v}})$ before and after portions of unknown Hamiltonian dynamics $e^{-iHt/N}$ a sufficient number of times, said dynamics can be controllized, i.e., simulate $\mathtt{ctrl}_0(e^{-iH_0 t})$.

Intuitively, controlization makes use of a randomized algorithm to approximately implement a unitary dynamics (in a similar manner to the qDRIFT procedure) to simulate the Hamiltonian

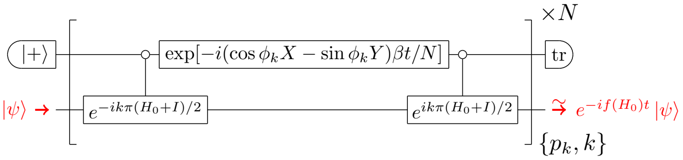
$$\sum_{\vec{v} \in \{0,1,2,3\}^n} \frac{1}{4^n} \begin{pmatrix} I & 0 \\ 0 & \sigma_{\vec{v}} \end{pmatrix} \begin{pmatrix} H & 0 \\ 0 & H \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & \sigma_{\vec{v}} \end{pmatrix} = \begin{pmatrix} H_0 & 0 \\ 0 & 0 \end{pmatrix} + \frac{\mathrm{tr}(H)}{2^n} I, \tag{6}$$

where $\vec{v} := (v_1, \ldots, v_n) \in \{0,1,2,3\}^n$ and $\sigma_{\vec{v}} := \sigma_{v_1} \otimes \cdots \otimes \sigma_{v_n}$. Equation (6) follows from the identity $\frac{1}{4^n}\sum_{\vec{v}} \sigma_{\vec{v}} H \sigma_{\vec{v}} = \frac{\mathrm{tr}(H)}{2^n} I$. Each term $\mathtt{ctrl}(\sigma_{\vec{v}})(I \otimes H)\mathtt{ctrl}(\sigma_{\vec{v}})$ in the sum on the l.h.s. of Eq. (6) can be implemented by applying the dynamics $e^{-iH\tau}$ in between the gate $\mathtt{ctrl}(\sigma_{\vec{v}})$, which follows from the identity $Ue^{-iHt}U^\dagger = e^{-i(UHU^\dagger)t}$ for a general Hamiltonian H and unitary U. When exponentiated, the r.h.s. of Eq. (6) yields the desired controlled Hamiltonian dynamics (up to a global phase contribution from the second term, which is irrelevant).

The runtime $\Theta(t^2 n/\epsilon)$ follows from the fact that $N := N(1, t, \epsilon)$ is $\Theta(t^2/\epsilon)$ and that implementing $\mathtt{ctrl}(\sigma_{\vec{v}})$ takes $\Theta(n)$ time.

C. Fourier Series Simulation

We now move to describe the second subroutine that comprises our uncompiled UHET algorithm, namely *Fourier series simulation.* This algorithm serves to transform eigenvalues of an unknown Hamiltonian using the theory of Fourier series at its core. A Fourier expansion of a function $f : [-1, 1] \to \mathbb{R}$ is given by $\sum_{k=-\infty}^{\infty} c_k e^{i\pi kx}$ for $c_k := (1/2)\int_{-1}^{1} \mathrm{d}x\, f(x) e^{-i\pi kx}$, which converges to $f(x)$ for a wide range of functions (see, e.g., [29]). The procedure



$|+\rangle$
$\exp[-i(\cos\phi_k X - \sin\phi_k Y)\beta t/N]$
tr
$\times N$
$|\psi\rangle \rightarrow$
$e^{-ik\pi(H_0+I)/2}$
$e^{ik\pi(H_0+I)/2}$
$\rightsquigarrow e^{-if(H_0)t}|\psi\rangle$
$\{p_k, k\}$

FIG. 5. *Fourier series simulation circuit.*—By randomly applying $\mathtt{ctrl}_0(e^{\pm ik\pi(H_0+I)/2})$ before and after the dynamics $e^{-i(\cos\phi_k X - \sin\phi_k Y)\beta t/N}$, the circuit simulates the Fourier series of the desired transformation function $\tilde{f}[(H_0 + I)/2] = f(H_0)$.

of Fourier series simulation is described in Subroutine 2, with a circuit representation presented in Fig. 5.

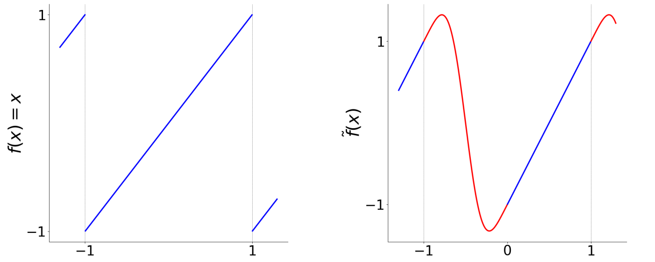
```
Subroutine 2 Fourier Series Simulation
  Input:
    • A finite number of queries to $\mathtt{ctrl}_0(e^{\pm iH_0\tau}) \in \mathcal{L}(\mathcal{H}_c \otimes \mathcal{H})$ $(\tau > 0)$ where $H_0 \in \mathcal{L}(\mathcal{H})$ is a traceless Hamiltonian normalized as $\|H_0\|_{\mathrm{op}} \leq 1$
    • A class $C^3$ (3 times continuously differentiable) function $f : [-1, 1] \to \mathbb{R}$, such that $f^{(4)}$ is piecewise $C^2$ (see App. A 1)
    • Input state $|\psi\rangle \in \mathcal{H}$
    • Allowed error $\epsilon > 0$
    • Time $t > 0$
  Output: A state approximating $e^{-if(H_0)t}|\psi\rangle$ $(t > 0)$ with an error according to Eq. (2) upper bounded by $\epsilon$, and mean squared error according to Eq. (3) upper bounded by $2\epsilon$

  Runtime: $\Theta(C_{2,f}t^2/\epsilon)$ for an $f$-dependent constant $C_{2,f}$ which is independent of $n$, $t$, and $\epsilon$
  Used Resources:
    System: $n$-qubit system $\mathcal{H}$ and one auxiliary qubit $\mathcal{H}_c$
    Gates: $\mathtt{ctrl}(e^{\pm iH_0\tau})$ and single-qubit gates on $\mathcal{H}_c$

  Procedure:
Pre-processing:
 1: Define modified version $\tilde{f}$ of $f$ as in Eq. (7)
 2: Compute Fourier coefficients $\tilde{c}_k := (1/2)\int_{-1}^{1} \mathrm{d}x e^{-ik\pi x}\tilde{f}(x)$ for $k \in \{-K, -K+1, \ldots, K\}$ where the cutoff number $K$ satisfies
      $\left|\tilde{f}(x) - \sum_{k=-K}^{K} \tilde{c}_k e^{ik\pi x}\right| < \frac{\epsilon}{4t} \quad \forall\, x \in [-1, 1]$
 3: Compute $N := N(\beta, t, \epsilon/2)$ using $N(\lambda, t, \epsilon)$ from Eq. (4), for $\beta := \sum_{k=-K}^{K} |\tilde{c}_k|$
Main Process:
 4: Initialize $|\mathrm{current}\rangle \leftarrow |+\rangle \otimes |\psi\rangle$
 5: for $m = 1, \ldots, N$ do
 6:     Randomly choose $k \in \{-K, \ldots, K\}$ with probability $p_k := |\tilde{c}_k|/\beta$
 7:     Define $Q := \mathtt{ctrl}_0(e^{-ik\pi H_0/2})$ and $R := \mathtt{ctrl}_0(e^{ik\pi H_0/2})$
 8:     $|\mathrm{current}\rangle \leftarrow (e^{ik\pi Z/4} \otimes I)R(e^{-i[\cos\phi_k X - \sin\phi_k Y]\beta t/N} \otimes I)Q(e^{-ik\pi Z/4} \otimes I)\,|\mathrm{current}\rangle$ for $\phi_k$ defined by $\tilde{c}_k = |\tilde{c}_k|e^{i\phi_k}$
 9: end for
10: Trace out $\mathcal{H}_c$ of $|\mathrm{current}\rangle$
11: Return $|\mathrm{current}\rangle$
```

Fourier series simulation makes use of a controlled dynamics $\mathtt{ctrl}_0(e^{\pm iH_0\tau})$ $(\tau > 0)$ (which can be constructed using Subroutine 1) to implement the target Hamiltonian dynamics $e^{-if(H_0)t}$ $(t > 0)$. First, the desired function f is deformed to a periodically smooth function $\tilde{f}$ as

$$\tilde{f}(x) := \begin{cases} g_f(x) & x \in [-1, 0] \\ f(2x - 1) & x \in [0, 1] \end{cases}, \tag{7}$$



(a) $f(x) = x$ (b) $\tilde{f}$ corresponding to f

FIG. 6. *Comparison between the function f and its corresponding $\tilde{f}$.*—The function $\tilde{f}$ is defined by merging f (blue) and an additional function g_f (red, see App. C 2). Defined appropriately, $\tilde{f}$ has a faster convergence of the absolute value of its Fourier coefficients $|\tilde{c}_k|$ than f.

with g_f defined in App. C 2. Due to its periodic smoothness, the absolute value of the Fourier coefficients of $\tilde{f}$ converges rapidly to 0, which avoids rapid growth of the cutoff number K in terms of the error ϵ.

We then compute the Fourier coefficients $\tilde{c}_k := (1/2)\int_{-1}^{1} \mathrm{d}x e^{-ik\pi x}\tilde{f}(x)$ for $k \in \{-K, -K+1, \ldots, K\}$, where the cutoff number K satisfies

$$\left|\tilde{f}(x) - \sum_{k=-K}^{K} \tilde{c}_k e^{ik\pi x}\right| < \frac{\epsilon}{4t} \quad \forall\, x \in [-1, 1], \tag{8}$$

and the iteration number $N := N(\beta, t, \epsilon/2)$ (using Eq. (4)) for $\beta := \sum_{k=-K}^{K} |\tilde{c}_k|$.

The remainder of Subroutine 2 simulates the Fourier-transformed Hamiltonian of H_0, which can be understood from the following analysis. The unitary applied in step 8, for a random k chosen with probability $p_k = |\tilde{c}_k|/\beta$, can be written as a Hamiltonian evolution given by

$$\begin{aligned}&(e^{ik\pi Z/4} \otimes I)\begin{pmatrix} e^{ik\pi H_0/2} & 0 \\ 0 & I \end{pmatrix}(e^{-i(\cos\phi_k X - \sin\phi_k Y)\beta t/N} \otimes I)\cdot\\ &\begin{pmatrix} e^{-ik\pi H_0/2} & 0 \\ 0 & I \end{pmatrix}(e^{-ik\pi Z/4} \otimes I)\\ &= \exp\left[-i\begin{pmatrix} 0 & e^{i\phi_k}e^{ik\pi(H_0+I)/2} \\ e^{-i\phi_k}e^{-ik\pi(H_0+I)/2} & 0 \end{pmatrix}\beta t/N\right].\end{aligned} \tag{9}$$

As such, steps 5 to 9 of Subroutine 2 correspond to simulating $e^{-iH't}$ for the following Hamiltonian H'

$$\begin{aligned}H' &:= \sum_{k=-K}^{K} |\tilde{c}_k| \begin{pmatrix} 0 & e^{i\phi_k}e^{ik\pi(H_0+I)/2} \\ e^{-i\phi_k}e^{-ik\pi(H_0+I)/2} & 0 \end{pmatrix}\\ &\simeq \sum_{k=-\infty}^{\infty} |\tilde{c}_k| \begin{pmatrix} 0 & e^{i\phi_k}e^{ik\pi(H_0+I)/2} \\ e^{-i\phi_k}e^{-ik\pi(H_0+I)/2} & 0 \end{pmatrix}\\ &= \begin{pmatrix} 0 & f(H_0) \\ f(H_0) & 0 \end{pmatrix} = X \otimes f(H_0),\end{aligned} \tag{10}$$

which follows from the fact that

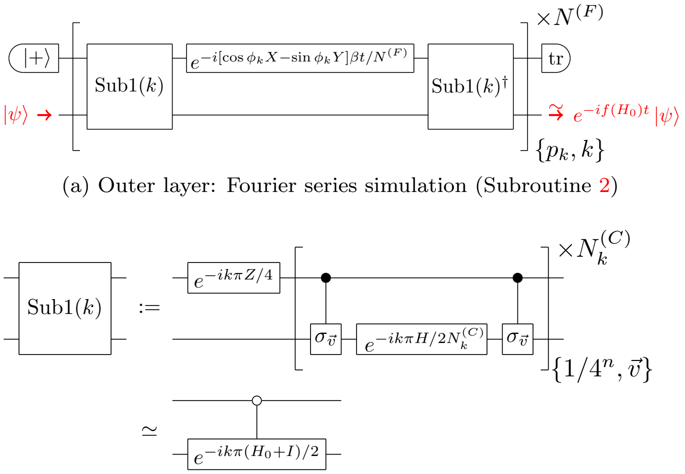
$$\begin{aligned}
&\sum_{k=-\infty}^{\infty} |\tilde{c}_k| e^{i\phi_k} e^{ik\pi(H_0+I)/2} \\
&= \sum_{k=-\infty}^{\infty} \sum_m \tilde{c}_k e^{ik\pi(E_m+1)/2} \left|E_m\right\rangle \left\langle E_m\right| \\
&= \sum_m \tilde{f}((E_m+1)/2) \left|E_m\right\rangle \left\langle E_m\right| = f(H_0)\,, \qquad (11)
\end{aligned}$$

where H_0 is diagonalized as $H_0 := \sum_m E_m \left|E_m\right\rangle \left\langle E_m\right|$. By taking the initial state as $|+\rangle \otimes |\psi\rangle$ in step 4 and tracing out the $\mathcal{H}_c$ subsystem in step 10, the dynamics $e^{-if(H_0)t}$ is applied to the input state $|\psi\rangle$.

We provide a full error analysis of Subroutine 2 in App. C 3. We evaluate its runtime as (Number of iterations N)×(Average runtime of each iteration). Given that N is $\Theta(\beta^2 t^2/\epsilon)$ and that each iteration on average takes time $\sum_k p_k \cdot \Theta(|k|)$, we have that the runtime is

$$\begin{aligned}
&\Theta\left(\frac{\beta^2 t^2}{\epsilon}\right) \times \sum_k p_k \cdot \Theta(|k|) = \\
&\Theta\left(\left(\sum_{k=-\infty}^{\infty} |\tilde{c}_k|\right)\left(\sum_{k=-\infty}^{\infty} |\tilde{c}_k||k|\right)\frac{t^2}{\epsilon}\right) =: \Theta\left(C_{2,f}\frac{t^2}{\epsilon}\right), \qquad (12)
\end{aligned}$$

where $C_{2,f}$ is a parameter that depends on f but is independent of n, t, ϵ. Note that the sum $\sum_{k=-\infty}^{\infty} |\tilde{c}_k||k|$ is guaranteed to converge due to the periodic smoothness of $\tilde{f}$ (see App. C 4 for details).



(a) Outer layer: Fourier series simulation (Subroutine 2)

(b) Inner layer: The procedure inside square brackets corresponds to controlization (Subroutine 1).

FIG. 7. *Circuit representation of Algorithm 3.*—The uncompiled UHET algorithm comprises an "outer" layer (a) that implements the Fourier series simulation Subroutine 2 upon the output of the "inner" layer (b), which itself controlizes the seed Hamiltonian dynamics via Subroutine 1.

D. Uncompiled UHET Algorithm

We now present the uncompiled UHET Algorithm 3, which results from concatenating the previous two subroutines and is depicted in Fig. 7. We first construct $\texttt{ctrl}_0(e^{\pm ik\pi(H_0+I)/2})$ from the input dynamics $e^{\pm iH\tau}$ via controlization and then perform the Fourier series simulation to simulate the desired dynamics $e^{-if(H_0)t}$. Thus, Algorithm 3 is a direct concatenation of Subroutines 1 and 2, and we therefore dub it *uncompiled*.

Algorithm 3 Universal Hamiltonian eigenvalue transformation (uncompiled)

Input:

- A finite number of queries to a black-box Hamiltonian dynamics $e^{\pm iH\tau}$ of a seed Hamiltonian H normalized as $\|H_0\|_{\mathrm{op}} \leq 1$ where H_0 is the traceless part of H, i.e., $H_0 := H - (1/2^n)\mathrm{tr}(H)I$, with $\tau > 0$
- A class C^3 (3 times continuously differentiable) function $f : [-1, 1] \to \mathbb{R}$, such that $f^{(4)}$ is piecewise C^2 (see App. A 1)
- Input state $|\psi\rangle \in \mathcal{H}$
- Allowed error $\epsilon > 0$
- Time $t > 0$

Output: A state approximating $e^{-if(H_0)t}|\psi\rangle$ $(t > 0)$ with an error according to Eq. (2) upper bounded by ϵ; additionally, the mean squared error according to Eq. (3) is upper bounded by 2ϵ)

Runtime:
Pre-processing (only once): T_3
Main Process: $\Theta(C_{3,f} t^4 n/\epsilon^3)$ for an f-dependent constant $C_{3,f}$ which is independent of n, t, and ϵ

Used Resources:
System: n-qubit system $\mathcal{H}$ and one auxiliary qubit $\mathcal{H}_c$
Gates: $e^{\pm iH\tau}$, single qubit gate on $\mathcal{H}_c$, and Clifford gates on $\mathcal{H}_c \otimes \mathcal{H}$

Procedure:

```
Pre-processing:
1: Run Pre-processing of Subroutine 2 for allowed error $\epsilon/2$
   to obtain iteration number $N^{(F)} := N(\beta, t, \epsilon/4)$, cutoff
   number $K$ and Fourier coefficients $\tilde{c}_k$
Main Process:
2: Run Main Process of Subroutine 2 with $N^{(F)}$ iterations
   using $K$ and $\tilde{c}_k$ obtained before, with step 7 modified to:
3:     Run Pre-processing of Subroutine 1 for allowed error $\epsilon/4N^{(F)}$ to obtain iteration number $N_k^{(C)} :=$
       $N(1, k\pi/2, \epsilon/4N^{(F)})$
4:     Run Main Process of Subroutine 1 with $N_k^{(C)}$ iterations
       and time $k\pi/2$ for $H$ to obtain unitary $Q'$
5:     Run Main Process of Subroutine 1 with $N_k^{(C)}$ iterations
       and time $k\pi/2$ for $-H$ to obtain unitary $R'$
6:     Define $Q := Q'$ and $R := R'$
```

As a consequence of its concatenated structure, two layers of iterations are used in Algorithm 3: $N_k^{(C)}$ for the controlization part and $N^{(F)}$ for the Fourier series simulation. We choose these numbers such that the total error of each subroutine is bounded from above by $\epsilon/2$ (so that overall error is upper bounded by ϵ). We begin by fixing the allowed error of the Fourier series simulation to be $\epsilon/2$. Then, fixing that for controlization to be $\epsilon/(4N^{(F)})$, it follows that iterating the outer layer of the procedure (see Fig. 7a) $N^{(F)}$ times implies that the error due to controlization is upper bounded by $[\epsilon/(4N^{(F)})] \cdot 2N^{(F)} = \epsilon/2$.

We now analyze the runtime of Algorithm 3. We split this cost into two parts. First, there is the *pre-processing step*, which refers to the processes that only need to be run once for a given set of inputs. In our case, this corresponds to the time required to compute the Fourier coefficients $\tilde{c}_k$ and cutoff number K until Eq. (8) is satisfied, plus the time for computing $N^{(F)}$. We define the sum of these two times as T_3. The *main process* takes $N^{(F)}\times$ (average runtime of each iteration), the latter of which in turn depends upon $N_k^{(C)}$.

The iteration number $N^{(F)} = N(\beta, t, \epsilon/4)$ scales as $\Theta[(\sum_{k=-\infty}^{\infty} |\tilde{c}_k|)^2 t^2/\epsilon]$ (note that $\sum_{k=-\infty}^{\infty} |\tilde{c}_k|$ is an upper bound of β which is independent of ϵ). Furthermore, the average runtime of each iteration is $\sum_{k=-K}^{K} p_k \times$(runtime of circuit inside Fig. 7a for each k), which scales as

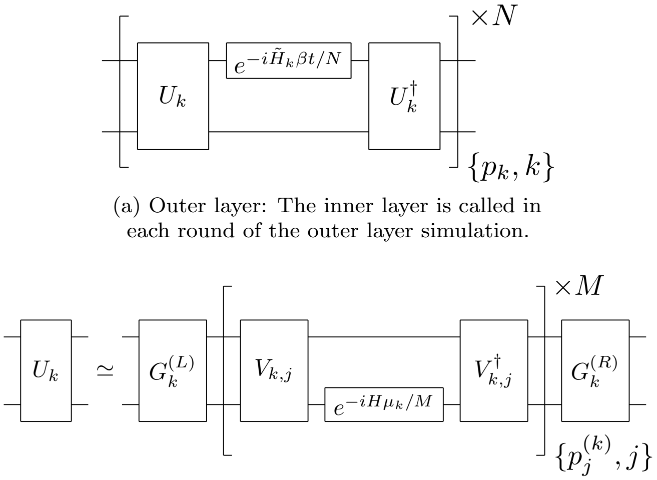
$$\sum_{k=-K}^{K} p_k \Theta(nN_k^{(C)}) = \sum_{k=-K}^{K} p_k \Theta\left(\frac{\beta^2 t^2 k^2 n}{\epsilon^2}\right) \\ \leq \Theta\left(\left(\sum_{k=-\infty}^{\infty} |\tilde{c}_k| k^2\right)\left(\sum_{k=-\infty}^{\infty} |\tilde{c}_k|\right)\frac{t^2 n}{\epsilon^2}\right).$$

The inequality is obtained by replacing K with ∞ in order to remove the ϵ dependence. Taking the product of the scaling of the two iteration layers, the overall runtime scaling of the main process of Algorithm 3 is bounded from above by

$$\Theta\left(\left(\sum_{k=-\infty}^{\infty} |\tilde{c}_k|\right)^3 \left(\sum_{k=-\infty}^{\infty} |\tilde{c}_k| k^2\right)\frac{t^4 n}{\epsilon^3}\right) =: \Theta\left(C_{3,f}\frac{t^4 n}{\epsilon^3}\right).$$

V. COMPILED UHET ALGORITHM

We now move to introduce a more efficient algorithm that implements UHET. At its core, this algorithm is inspired by the components that make up Algorithm 3, but rather than optimizing the subroutines independently, here we *compile* the algorithm at the level of the overall task by making use of *correlated classical randomness* to provide a more efficient implementation. We begin by describing a general framework of this novel notion of compilation, which can be used in many situations in which random Hamiltonian simulation subroutines are concatenated to perform a particular task. We subsequently apply this technique specifically to the task of UHET, compiling Algorithm 3 in order to construct the better Algorithm 4. Finally, we provide a resource analysis. Details are provided throughout App. D.



(a) Outer layer: The inner layer is called in each round of the outer layer simulation.

(b) Inner layer: This circuit simulates U_k defined in Eq. (14) by random Hamiltonian simulation.

FIG. 8. *Circuit representation of a general algorithm involving two layers of Hamiltonian simulation.*—Our method of compilation can be applied to any general two-layer protocol in which each subroutine uses the method of random Hamiltonian simulation.

A. General Method of Compilation

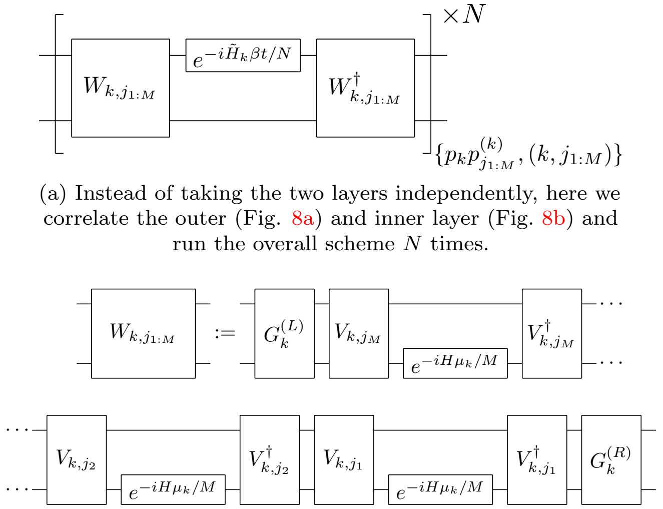
We begin by abstracting the key structural components of the uncompiled algorithm above. Notably, the algorithm consists of two independent layers, namely those depicted in Figs. 7a and 7b. These circuits are a special case of the more general form depicted in Figs. 8a and 8b respectively. The *outer layer* comprises portions of the input dynamics sandwiched between a unitary $U_k(H)$ that depends on the input Hamiltonian H and is iterated some $N \in \mathbb{Z}_{>0}$ times; for the sake of conciseness, we will simply write U_k instead of explicitly writing $U_k(H)$. Since

$$U_k^\dagger e^{-i\frac{\beta t}{N}\tilde{H}_k} U_k = e^{-i\frac{\beta t}{N} U_k^\dagger \tilde{H}_k U_k}, \tag{13}$$

this process corresponds to the simulation of the randomized dynamics $e^{-i(\sum_k p_k \beta U_k^\dagger \tilde{H}_k U_k)t}$. In this simulation, U_k is itself approximated using the *inner layer* procedure depicted in Fig. 8b. Here, j is a random index (depending on the choice of k) sampled from the probability distribution $\{p_j^{(k)}\}_j$, $\mu_k \in \mathbb{R}$, iteration number $M \in \mathbb{Z}_{>0}$ (which also depends on k, but we omit the subscript k for simplicity), and $V_{k,j}$, $G_k^{(L)}$, and $G_k^{(R)}$ are unitaries such that

$$U_k = G_k^{(R)} e^{-i(\sum_j p_j^{(k)} \mu_k V_{k,j}^\dagger H V_{k,j})} G_k^{(L)}. \tag{14}$$

Evidently, Algorithm 3 corresponds to the special case where $\tilde{H}_k = \cos\phi_k X - \sin\phi_k Y$, $\beta = \sum_{k=-K}^{K} |\tilde{c}_k|$, $p_k = |\tilde{c}_k|/\beta$, $U_k = \mathtt{ctrl}_0(e^{-ik\pi(H_0+I)/2})$, $M = N_k^{(C)}$,


$\times N$
$W_{k,j_{1:M}}$
$e^{-i\tilde{H}_k\beta t/N}$
$W^\dagger_{k,j_{1:M}}$
$\{p_k p^{(k)}_{j_{1:M}}, (k, j_{1:M})\}$

(a) Instead of taking the two layers independently, here we correlate the outer (Fig. 8a) and inner layer (Fig. 8b) and run the overall scheme N times.

(b) The subroutine constructs the unitary $W_{k,j_{1:M}}$ from M uses of the seed Hamiltonian dynamics.

FIG. 9. *Intermediate stage of compilation.*—By correlating the layers, an additional error is introduced due to the finite iteration of the previously inner layer. However, we next show how this error can be compensated for by choosing $W_{k,j_{1:M}}$ appropriately, therefore constructing an efficient compiled algorithm.

$j = \vec{v}$, $\mu_k = k\pi/2$, $p^{(k)}_{\vec{v}} = 1/4^n$, $V_{k,j} = \texttt{ctrl}(\sigma_{\vec{v}})$, $G_k^{(L)} = e^{-ik\pi Z/4} \otimes I$, and $G_k^{(R)} = I \otimes I$.

Having abstracted the key features of our previous algorithm, we are now in a position to introduce the general notion of *compilation*. The structure described above evidently consists of two *independent* layers of random Hamiltonian simulation; the method of compilation makes use of *correlated randomness* in order to correlate said layers in such a way that the error accumulation is reduced. In general, in order to reduce the approximation error of this overall procedure below ϵ, we must ensure that both errors introduced due to the simulations depicted in Figs. 8a and 8b are at most of $O(\epsilon)$. Subsequently, both iteration numbers N and M must have a $1/\epsilon$ dependence, which leads to an accumulation of $1/\epsilon$ dependence for each round of concatenated subroutines.

We do this by first modifying the circuit in Fig. 8a to an *intermediate circuit* in Fig. 9a, where $W_{k,j_{1:M}}(H) =: W_{k,j_{1:M}}$ is written with its dependence on H implicit and defined as in Fig. 9b; similarly, we compress the labels $j_{1:M} := \{j_1, \dots, j_M\}$ and $p^{(k)}_{j_{1:M}} := p^{(k)}_{j_1} \dots p^{(k)}_{j_M}$ for ease of notation. Finally, an additional modification—which depends upon the specific task at hand—can then be applied to said intermediate circuit in order to compensate for error accumulation, leading to the *compiled algorithm*, as we will show for the case of UHET in the coming section.

In the intermediate circuit, the random indices $j_{1:M}$ are correlated in such a way that the components before and after the dynamics $e^{-i\tilde{H}_k\beta t/N}$ are inverse to each other. Therefore, the circuit of Fig. 9 corresponds to the simulation of $e^{-iH_{\text{new}}t}$, where

$$H_{\text{new}} := \beta \sum_{k,j_{1:M}} p_k\, p^{(k)}_{j_{1:M}} W^\dagger_{k,j_{1:M}} \tilde{H}_k W_{k,j_{1:M}}. \tag{15}$$

It can be shown using analysis of qDRIFT [5] that

$$\sum_{j_{1:M}} p^{(k)}_{j_{1:M}} W^\dagger_{k,j_{1:M}} \tilde{H}_k W_{k,j_{1:M}} = U_k^\dagger \tilde{H}_k U_k + O\left(\frac{\mu_k^2}{M}\right), \tag{16}$$

since the error analysis for the qDRIFT procedure remains valid when replacing the simulation of a density operator with that of a Hamiltonian.

In order to suppress the inner layer error $O(\mu_k^2/M)$ due to the random sampling of j in each of the N iterations of the outer layer below $O(\epsilon/N)$, the iteration number M must be chosen as $\Omega(\mu_k^2 N/\epsilon)$. Consequently, since N itself must have $1/\epsilon$ dependence (in order to suppress the overall error below ϵ, as explained previously) the overall runtime of the intermediate circuit $N \cdot M$ scales according to $1/\epsilon^3$. Next, by compiling the subroutines of the algorithm, one can reduce this resource scaling to $1/\epsilon$. This compilation procedure can be achieved by setting M as $O(\mu_k^2)$—removing the dependence on ϵ here comes at a cost of increasing the error. However, said $O(\mu_k^2/M) = O(1)$ inner layer error can be compensated for by introducing an additional modification to the procedure shown in Fig. 9, as we will discuss in the coming section. The total runtime will be proportional to $N \cdot M$ which scales in terms of ϵ in the same way as N. Therefore, the total runtime scales as $O(1/\epsilon)$ in terms of ϵ.

B. Compiled UHET Algorithm

We now move to apply this general method of compilation to the task of UHET, thereby providing a more efficient procedure than Algorithm 3. The compiled algorithm is presented in full in Algorithm 4, with details regarding the error and runtime provided throughout App. D.

With respect to the circuit depicted in Fig. 9a, here we choose each iteration of the circuit to correspond to a short time-evolution by the following Hamiltonian:

$$\left(\frac{1}{4}\right)^{nM}\sum_{\vec{v}_{1:M}} W^{\dagger}_{k,\vec{v}_{1:M}}([\cos\phi_k X - \sin\phi_k Y]\otimes I)W_{k,\vec{v}_{1:M}}$$
$$= A_{k,M}(e^{i\theta_{k,M}Z/2}\otimes I)\begin{pmatrix} 0 & e^{i\phi_k}e^{ik\pi(H_0+I)/2} \\ e^{-i\phi_k}e^{-ik\pi(H_0+I)/2} & 0 \end{pmatrix}(e^{-i\theta_{k,M}Z/2}\otimes I) \tag{17}$$

Algorithm 4 Efficient universal Hamiltonian eigenvalue transformation (Compiled)

Input:

- A finite number of queries to a black-box Hamiltonian dynamics $e^{\pm iH\tau}$ of a seed Hamiltonian H normalized as $\|H_0\|_{\mathrm{op}} \le 1$, where H_0 is the traceless part of H, i.e., $H_0 := H - (1/2^n)\mathrm{tr}(H)I$, with $\tau > 0$
- A class C^3 (3 times continuously differentiable) function $f : [-1,1] \to \mathbb{R}$, such that $f^{(4)}$ is piecewise C^2 (see App. A 1)
- Input state $|\psi\rangle \in \mathcal{H}$
- Allowed error $\epsilon > 0$
- Time $t > 0$

Output: A state approximating $e^{-if(H_0)t}|\psi\rangle$ $(t > 0)$ with an error in terms Eq. (2) upper-bounded by ϵ (also, the mean square of error in terms of Eq. (3) upper-bounded by 2ϵ)

Runtime:
Pre-processing (only once): $\Theta(\grave{K}^3 t^3 n/\epsilon^3) + T_4$, $\grave{K} = O((t/\epsilon)^{1/3})$
Main Process: $\Theta(C_{4,f}t^2 n/\epsilon)$ for an f-dependent constant $C_{4,f}$ which is independent of n, t, and ϵ

Used Resources:
System: n-qubit system $\mathcal{H}$ and one auxiliary qubit $\mathcal{H}_c$
Gates: $e^{\pm iH\tau}$, single qubit gate on $\mathcal{H}_c$, and Clifford gates on $\mathcal{H}_c \otimes \mathcal{H}$

Procedure:

```
Pre-processing:
Define $\tilde{f}$ as shown in Eq. (7) and compute Fourier coefficients $\tilde{c}_k := (1/2)\int_{-1}^{1} \mathrm{d}x e^{-ik\pi x}\tilde{f}(x)$ for $k \in \{-\grave{K}, -\grave{K}+1, \dots, \grave{K}\}$ for a $\grave{K} > 0$ satisfying Eq. (19)
for $k \in \{1, \dots, \grave{K}\}$ do
    Generate $(\hat{A}_k, \hat{\theta}_k)$ by Subroutine D5 of App. D 2 with allowed error set as $\sqrt{3}\epsilon/(12\pi(\sum_{k=-\infty}^{\infty}|\tilde{c}_k||k|))$
    $(\hat{A}_{-k}, \hat{\theta}_{-k}) \leftarrow (\hat{A}_k, -\hat{\theta}_k)$
end for
$(\hat{A}_0, \hat{\theta}_0) \leftarrow (1, 0)$
Compute $\grave{N} := N(\grave{\beta}, t, \epsilon/3)$ for $\grave{\beta} := \sum_{k=-\grave{K}}^{\grave{K}}(|\tilde{c}_k|/\hat{A}_k)$
Main Process:
Initialize $|\mathrm{current}\rangle \leftarrow |+\rangle \otimes |\psi\rangle$
for $m \in \{1, \dots, \grave{N}\}$ do
    Randomly choose $k$ with probability $\grave{p}_k := |\tilde{c}_k|/(\hat{A}_k\grave{\beta})$
    Randomly choose $j_k := \{\vec{v}_{1,10k^2}\} \in (\{0,1,2,3\}^n)^{10k^2}$
    $|\mathrm{current}\rangle \leftarrow \grave{W}^{\dagger}_{k,j_k}(e^{-i[\cos\phi_k X - \sin\phi_k Y]\grave{\beta}t/\grave{N}^{(F)}} \otimes I)\grave{W}_{k,j_k}|\mathrm{current}\rangle$
end for
Trace out $\mathcal{H}_c$ of $|\mathrm{current}\rangle$
Return $|\mathrm{current}\rangle$
```

where the iteration number M can depend upon k, $A_{k,M} = 1 - O(k^2/M) > 0$, $\theta_{k,M} = O(k^3/M^2) \in \mathbb{R}$ and

$$W^{(M)}_{k,\vec{v}_{1:M}} := \left[\prod_{l=1}^{M}\mathtt{ctrl}(\sigma_{\vec{v}_l})(I \otimes e^{-i(k\pi H)/(2M)})\mathtt{ctrl}(\sigma_{\vec{v}_l})\right] \times (e^{-ik\pi Z/4} \otimes I). \tag{18}$$

For simplicity, we sometimes suppress the superscript (M) and just write $W_{k,\vec{v}_{1:M}}$. We prove the validity of Eq. (17) in App. D 1.

As mentioned earlier, in order to improve the scaling behavior with respect to the uncompiled algorithm, we seek to modify the circuit to correct for the error introduced by the finite iteration number M of the inner layer. The factor $A_{k,M}$ can be compensated for by modifying the probability distribution p_k and the iteration number of the outer layer N; the rotation errors of the form $e^{\pm i\theta_{k,M}Z/2} \otimes I$ can be corrected via an inverse rotation. Thus, the general compilation procedure can be applied to UHET in order to reduce the runtime. In particular, we take M as $10k^2$ such that $A_{k,M} = 1 - O(k^2/M) > 1/2$ independently of k, which follows from a lower bound of $A_{k,M}$ [see Eq. (D3) in App. D 1].

If a classical description of the dynamics were given, then the compensation parameters $A_{k,M}$ and $\theta_{k,M}$ could be explicitly calculated via Eq. (17); however, since we only have access to the black-box dynamics $e^{\pm iH\tau}$, we must now construct a circuit that efficiently estimates them without relying on explicit knowledge of H. We present such a method that makes use of robust phase estimation [30] to obtain estimates $(\hat{A}_k, \hat{\theta}_k)$ of the error parameters $(A_{k,10k^2}, \theta_{k,10k^2})$ for $k \in \{-K, \dots, K\}$ (where K is a cutoff number) in App. D 2. This completes the pre-processing step of Algorithm 4.

The main part of Algorithm 4 then makes use of these estimates to compensate the error in simulating the desired transformed dynamics $e^{-if(H_0)t}$; we provide a complete error and runtime analysis in App. D 3. The circuit representation of Algorithm 4 is depicted in Fig. 10, in which we choose a new cutoff number $\grave{K}$ satisfying

$$\left|\tilde{f}(x) - \sum_{k=-\grave{K}}^{\grave{K}}\tilde{c}_k e^{ik\pi x}\right| < \frac{\epsilon}{6t} \quad \forall x \in [-1,1], \tag{19}$$

and set $j_k := (\vec{v}_{1:10k^2}) \in (\{0,1,2,3\}^n)^{10k^2}$, $\grave{\beta} := \sum_{k=-\grave{K}}^{\grave{K}}(|\tilde{c}_k|/\hat{A}_k)$, $\grave{N} := N(\grave{\beta}, t, \epsilon/3)$, $\grave{p}_k := |\tilde{c}_k|/(\hat{A}_k\grave{\beta})$,

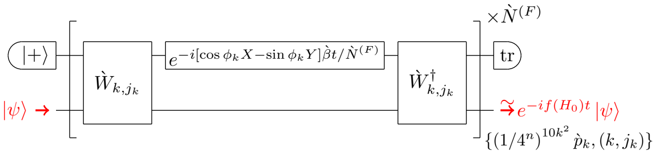

$\times \grave{N}^{(F)}$
$|+\rangle$
$\grave{W}_{k,j_k}$
$e^{-i[\cos\phi_k X - \sin\phi_k Y]\grave{\beta}t/\grave{N}^{(F)}}$
$\grave{W}^{\dagger}_{k,j_k}$
tr
$|\psi\rangle$
$e^{-if(H_0)t}\,|\psi\rangle$
$\{(1/4^n)^{10k^2}\ \grave{p}_k, (k, j_k)\}$

FIG. 10. *Compiled version of outer layer (Fig. 7a).*—By using modified values $\grave{p}_k$, $\grave{\beta}$, $\grave{N}^{(F)}$, and a modified operator $\grave{W}_{k,j_k}$ (see Algorithm 4 for details), the iteration number $N_k^{(C)} = \Theta(\beta^2 t^2 k^2/\epsilon^2)$ for controlization in Algorithm 3 is reduced to $10k^2 = \Theta(k^2)$ independent of f, n, t and ϵ, thereby leading to a more efficient algorithm for UHET.

and

$$\grave{W}_{k,j_k} := W_{k,j_k}^{(10k^2)}(e^{i\hat{\theta}_k Z/2} \otimes I). \tag{20}$$

The average number of times that the unknown dynamics $e^{-iH\tau}$ $(\tau > 0)$ is called in any one sampling of k is reduced via compilation from $\sum_k p_k \Theta(\beta^2 t^2 k^2/\epsilon^2)$ (Algorithm 3) to $\sum_k \grave{p}_k \Theta(k^2)$ (Algorithm 4) and the average depth of the overall circuit is subsequently reduced accordingly.

The runtime of Algorithm 4 is $\Theta(\grave{K}^3 t^3 n/\epsilon^3) + T_4$ for pre-processing, where $\grave{K} = O((t/\epsilon)^{1/3})$ is the cutoff number defined in step 1 and T_4 refers to the sum of the classical computation times for steps 1 (computation of the Fourier coefficients $\tilde{c}_k$ until Eq. (19) is satisfied) and step 7 (computation of $\grave{N}$). The main process has runtime $\Theta(C_{4,f} t^2 n/\epsilon)$ (see App. D 3 for the proof). The scaling of the runtime of the main process in the $t \to \infty$ and $\epsilon \to 0$ limits are reduced in Algorithm 4 compared to Algorithm 3.

VI. COMPARISON WITH A QSVT-BASED UHET ALGORITHM

Algorithms 3 and 4 above provide two novel methods to implement the task of UHET; we now move to compare these algorithms with another one that achieves the same task via a modified QSVT procedure. In principle, one can combine standard QSVT methods with the ability to simulate Hamiltonian dynamics to achieve the said task. Given a classical description of the Hamiltonian, this can be achieved by performing a block-encoding of H into a unitary, using QSVT to approximate $e^{-if(H_0)t}$ (up to a proportionality constant), and then amplifying the block via *robust oblivious amplitude amplification* [27, 28]. However, such a procedure requires a classical description of the Hamiltonian to be known *a priori*, in contradistinction to the UHET task we have so far considered. Here, instead of a classical description of H, one is only given access to a black-box Hamiltonian dynamics $e^{\pm iHt}$. In App. E, we provide a method that modifies the standard QSVT procedure by essentially applying control to a modified version of the unknown Hamiltonian and then applying appropriate gates to simulate the desired function.

This allows us to compare the runtimes of Algorithm 3 (uncompiled), Algorithm 4 (compiled) and the QSVT-based Algorithm 7 presented in full in App. E 1. We show that the scaling of the runtime of the main process (i.e., the part which must be run each time) of these algorithms behaves in the limit of $\epsilon \to 0$ according to:

$$\begin{aligned}&\text{Runtime of Algorithm 4 (compiled)}\\ \leq\ &\text{Runtime of QSVT-based Algorithm 7}\\ \leq\ &\text{Runtime of Algorithm 3 (uncompiled).}\end{aligned} \tag{21}$$

In other words, Algorithm 3, which is slower than the QSVT-based Algorithm 7 when uncompiled, becomes faster than it via compilation into Algorithm 4. In App. E 1, we first describe how QSVT techniques can be applied to the task of UHET and present an algorithm that leverages ideas from Hamiltonian singular value transformation [31] and the QSVT-based Hamiltonian simulation [27, 28]. In App. E 2, we subsequently calculate the ϵ dependence of the three studied algorithms, demonstrating Eq. (21), and we explain the technical reasons for the differences in scaling.

VII. CONCLUSION

In this work, we developed a universal quantum algorithm for transforming the eigenvalues of any Hamiltonian, by any (suitably differentiable) function, while keeping the eigenstates fixed. Our algorithm is universal in the sense that it does not rely on knowledge of the input Hamiltonian, whose dynamics can be given as a black box. The algorithm is initially constructed by concatenating two subroutines, namely controlization and Fourier series simulation. This is followed by a compilation step, based on a general framework we develop, using correlated randomness to perform the multiple layers of random sampling in a more efficient way. We show that the compilation step significantly reduces the runtime of the algorithm, making it even more efficient than simulation methods based on the quantum singular value transformation.

Our results have implications broadly across the realm of quantum information and beyond. First, the notion of compilation reconciles quantum computing practices with a key notion implemented in classical computing, where subroutines are compiled into larger functions to be implemented more efficiently. Extending this approach to different tasks could significantly improve our ability to develop complex and efficient quantum algorithms in a modular fashion, similar to that of classical software. Second, we expand the class of universal Hamiltonian transformations that can be efficiently performed to all suitably differential functions on the space of Hamiltonians, extending previously known results for linear functions of Hamiltonians [10].

In the future, developing a general theoretical framework for higher-order transformation of Hamiltonian dynamics will provide more insight into the possible manipulation of Hamiltonian dynamics for information processing tasks. In the longer term when fault-tolerant quantum computers become available, we envisage that our methods could be applied to practical use cases in the simulation and manipulation of quantum systems, such as in quantum chemistry or materials discovery.

ACKNOWLEDGMENTS

We would like to thank Toshinori Itoko, Antonio Mezzacapo, Kunal Sharma, and Satoshi Yoshida for helpful discussions. This work was supported by MEXT Quantum Leap Flagship Program (MEXT QLEAP) JPMXS0118069605, JPMXS0120351339, Japan Society for the Promotion of Science (JSPS) KAKENHI Grant No. 21H03394, and the IBM-UTokyo lab. Research at Perimeter Institute is supported in part by the Government of Canada through the Department of Innovation, Science and Economic Development and by the Province of Ontario through the Ministry of Colleges and Universities. P. T. acknowledges support from the Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellowship for Research in Japan.

[1] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, *et al.*, Commentary: The materials project: A materials genome approach to accelerating materials innovation, APL Materials **1** (2013).

[2] F. Tang, H. C. Po, A. Vishwanath, and X. Wan, Efficient topological materials discovery using symmetry indicators, Nature Physics **15**, 470 (2019).

[3] M. Suzuki, Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations, Physics Letters A **146**, 319 (1990).

[4] M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics, J. Math. Phys. **32**, 400 (1991).

[5] E. Campbell, Random compiler for fast Hamiltonian simulation, Phys. Rev. Lett. **123**, 070503 (2019), arXiv:1811.08017.

[6] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Simulating Hamiltonian dynamics with a truncated Taylor series, Phys. Rev. Lett. **114**, 090502 (2015), arXiv:1412.4687.

[7] G. H. Low and I. L. Chuang, Optimal Hamiltonian simulation by quantum signal processing, Phys. Rev. Lett. **118**, 010501 (2017), arXiv:1606.02685.

[8] G. H. Low and I. L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019), arXiv:1610.06546.

[9] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, Proc. Natl. Acad. Sci. U.S.A. **115**, 9456 (2018), arXiv:1711.10980.

[10] T. Odake, H. Kristjánsson, A. Soeda, and M. Murao, Higher-order quantum transformations of Hamiltonian dynamics (2023), arXiv:2303.09788.

[11] L. K. Grover, Quantum Mechanics Helps in Searching for a Needle in a Haystack, Phys. Rev. Lett. **79**, 325 (1997), arXiv:quant-ph/9706033.

[12] R. N. Zare, Laser control of chemical reactions, Science **279**, 1875 (1998).

[13] M. T. Greiner, M. G. Helander, W.-M. Tang, Z.-B. Wang, J. Qiu, and Z.-H. Lu, Universal energy-level alignment of molecules on metal oxides, Nat. Mater. **11**, 76 (2012).

[14] G. Chiribella, G. M. D'Ariano, and P. Perinotti, Quantum circuit architecture, Phys. Rev. Lett. **101**, 060401 (2008), arXiv:0712.1325.

[15] J. Miyazaki, A. Soeda, and M. Murao, Complex conjugation supermap of unitary quantum maps and its universal implementation protocol, Phys. Rev. Res. **1**, 013007 (2019), arXiv:1706.03481.

[16] Q. Dong, S. Nakayama, A. Soeda, and M. Murao, Controlled quantum operations and combs, and their applications to universal controllization of divisible unitary operations (2019), arXiv:1911.01645.

[17] M. T. Quintino, Q. Dong, A. Shimbo, A. Soeda, and M. Murao, Probabilistic exact universal quantum circuits for transforming unitary operations, Phys. Rev. A **100**, 062339 (2019), arXiv:1909.01366.

[18] M. T. Quintino, Q. Dong, A. Shimbo, A. Soeda, and M. Murao, Reversing unknown quantum transformations: Universal quantum circuit for inverting general unitary operations, Phys. Rev. Lett. **123**, 210502 (2019), arXiv:1810.06944.

[19] S. Yoshida, A. Soeda, and M. Murao, Reversing Unknown Qubit-Unitary Operation, Deterministically and Exactly, Phys. Rev. Lett. **131**, 120602 (2023), arXiv:2209.02907.

[20] G. Chiribella and H. Kristjánsson, Quantum Shannon theory with superpositions of trajectories, Proc. R. Soc. A **475**, 20180903 (2019), arXiv:1812.05292.

[21] G. Chiribella, G. M. D'Ariano, P. Perinotti, and B. Valiron, Quantum computations without definite causal structure, Phys. Rev. A **88**, 022318 (2013), arXiv:0912.0195.

[22] F. A. Pollock, C. Rodríguez-Rosario, T. Frauenheim, M. Paternostro, and K. Modi, Non-Markovian quantum processes: Complete framework and efficient characterization, Phys. Rev. A **97**, 012127 (2018), arXiv:1512.00589.

[23] O. Oreshkov, F. Costa, and Č. Brukner, Quantum correlations with no causal order, Nat. Commun. **3**, 1092 (2012), arXiv:1105.4464.

[24] G. Bai, Y.-D. Wu, Y. Zhu, M. Hayashi, and G. Chiribella, Efficient algorithms for causal order discovery in quantum networks (2020), arXiv:2012.01731.

[25] P. Selinger, Towards a Quantum Programming Language, Math. Struct. Comput. Sci. **14**, 527 (2004).

[26] P. Selinger and B. Valiron, A Lambda Calculus for Quantum Computation with Classical Control, in *Typed Lambda Calculi and Applications* (2005) pp. 354–368, arXiv:cs/0404056.

[27] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, in

Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (2019) pp. 193–204, arXiv:1806.01838.

[28] J. M. Martyn, Z. M. Rossi, A. K. Tan, and I. L. Chuang, Grand Unification of Quantum Algorithms, PRX Quantum **2**, 040203 (2021), arXiv:2105.02859.

[29] J. S. Walker, Fourier Series (Academic Press, 2003) pp. 167–183.

[30] S. Kimmel, G. H. Low, and T. J. Yoder, Robust calibration of a universal single-qubit gate set via robust phase estimation, Phys. Rev. A **92**, 062315 (2015), arXiv:1502.02677.

[31] S. Lloyd, B. T. Kiani, D. R. Arvidsson-Shukur, S. Bosch, G. De Palma, W. M. Kaminsky, Z.-W. Liu, and M. Marvian, Hamiltonian singular value transformation and inverse block encoding (2021), arXiv:2104.01410.

[32] J. Watrous, *The theory of quantum information* (Cambridge University Press, Cambridge, UK, 2018).

[33] M. Bocher, Introduction to the Theory of Fourier's Series, Ann. Math. **7**, 81 (1906).

[34] J. P. Boyd, *Chebyshev and Fourier spectral methods* (Dover Publications Inc., New York, NY, 2001).

[35] R. Chao, D. Ding, A. Gilyen, C. Huang, and M. Szegedy, Finding angles for quantum signal processing with machine precision (2020), arXiv:2003.02831.

APPENDICES

Appendix A: Preliminary Definitions

Here, we provide some basic definitions that will prove useful throughout our work.

1. J-Smoothness of Functions

Definition 1 (Piecewise C^J)**.** *Let $\mathcal{X} := \{x_0, \ldots, x_n\}$, where $-1 = x_0 < x_1 < \cdots < x_n = 1$. A function $g : [-1,1]\backslash\mathcal{X}' \to \mathbb{R}$, where $\mathcal{X}' \subseteq \mathcal{X}$, is **piecewise** $\mathbf{C^J}$ ($J \in \mathbb{N}_+$) $:\overset{\text{def}}{\Leftrightarrow}$ the derivatives $g^{(m)}$ for $m \in \{0,1,\ldots,J\}$ are well-defined and continuous everywhere in $[-1,1]\backslash\mathcal{X}$, and additionally, at the* exceptional points $x_1, \ldots, x_{n-1}$ *of g, we have that for all $j \in \{1,\ldots,n-1\}$, $\lim_{x\to x_j^+} g^{(m)}(x)$ and $\lim_{x\to x_j^-} g^{(m)}(x)$ exist for all $m \in \{0,\ldots,J\}$ (although the limits from above and below need not coincide), as well as for $x = \pm 1$, $\lim_{x\to -1^+} g^{(m)}(x)$ and $\lim_{x\to 1^-} g^{(m)}(x)$ exist. In particular, when $J = 1$, g is said to be **piecewise smooth***

Definition 2 (J-Smoothness)**.** *Here, we extend the notion of piecewise continuity to higher orders of smoothness.*

1. *A function $g : [-1,1] \to \mathbb{R}$ is* J-smooth $:\overset{\text{def}}{\Leftrightarrow}$ *g is piecewise C^{J-1} and $g^{(J)}$ is piecewise C^2.*

2. *A function $g : [-1,1] \to \mathbb{R}$ is* periodically J-smooth $:\overset{\text{def}}{\Leftrightarrow}$ *g is J-smooth and $g^{(m)}(+1) = g^{(m)}(-1)$ $\forall$ $m \in \{0,\ldots,J-1\}$.*

3. *A function $g : [-1,1] \to \mathbb{R}$ is* strictly J-smooth $:\overset{\text{def}}{\Leftrightarrow}$ *g is J-smooth and $g^{(J)}$ does not satisfy*

$$\lim_{x\to a^+} g^{(J)}(x) = \lim_{x\to a^-} g^{(J)}(x) \quad \forall\ a \in (-1,1). \tag{A1}$$

4. *A function $g : [-1,1] \to \mathbb{R}$ is* strictly periodically J-smooth $:\overset{\text{def}}{\Leftrightarrow}$ *g is periodically J-smooth and $g^{(J)}$ does not satisfy either or both of the following conditions:*

$$\begin{aligned} &1.\ \lim_{x\to a^+} g^{(J)}(x) = \lim_{x\to a^-} g^{(J)}(x) \quad \forall\ a \in (-1,1) \\ &2.\ \lim_{x\to 1^-} g^{(J)}(x) = \lim_{x\to -1^+} g^{(J)}(x). \end{aligned} \tag{A2}$$

2. Norms for Quantifying Errors

When quantifying the error of a simulated operation, we will often make use of the diamond norm, defined as

$$\|\Phi\|_\diamond := \max_{A\in\mathcal{L}(\mathcal{H}\otimes\mathcal{H});\|A\|_1=1} \|(\Phi\otimes\mathcal{I})(A)\|_1, \tag{A3}$$

where $\Phi : \mathcal{L}(\mathcal{H}) \to \mathcal{L}(\mathcal{H})$ is a quantum operation, $\mathcal{I}$ is an identity operation on $\mathcal{L}(\mathcal{H})$, and $\|\cdot\|_1$ denotes the 1-norm defined as $\|A\|_1 := \mathrm{tr}(\sqrt{A^\dagger A})$.

3. Scaling Notation

Throughout this article, we use the symbols $O(\cdot)$, $\Omega(\cdot)$, and $\Theta(\cdot)$ to denote the scaling behavior of algorithm runtimes, the definitions of which are provided in Table I. Furthermore, we consider various limits depending on the parameter of interest. In particular, we consider the limit $\to\infty$ for qubit number $n \in \mathbb{Z}_{>0}$, simulation time $t \in \mathbb{R}$, and the limit $\to 0$ for the allowed error $\epsilon > 0$. For instance $f(t,\epsilon) = O(t^2\epsilon^{-1})$ means that $\limsup_{t\to\infty}(f(t,\epsilon)/t^2) < \infty$ for all $\epsilon > 0$ and $\limsup_{\epsilon\to 0}(f(t,\epsilon)/\epsilon^{-1}) < \infty$ for all $t \in \mathbb{R}$. Intuitively speaking, $f(x) = O(g(x))$ if $g(x)$ grows at least as fast as $f(x)$ in $\lim x \to \infty$ (i.e., g asymptotically upper bounds f); $f(x) = \Omega(g(x))$ if $f(x)$ grows at least as fast as $g(x)$ in $\lim x \to \infty$ (i.e., g asymptotically lower bounds f); and $f(x) = \Theta(g(x))$ if g provides both an upper and lower bound of f asymptotically.

Notation	Definition
$f(x) = O(g(x))$	$\limsup_{x\to\infty}(f(x)/g(x)) < \infty$
$f(x) = \Omega(g(x))$	$\liminf_{x\to\infty}(f(x)/g(x)) > 0$
$f(x) = \Theta(g(x))$	$f(x) = O(g(x))$ and $f(x) = \Omega(g(x))$

TAB. I. *Scaling symbols.*—For any $g(x)$, we say that a function $f(x)$ is $O(g(x))$, $\Omega(g(x))$, or $\Theta(g(x))$, if the above are satisfied.

Appendix B: Universal Hamiltonian Eigenvalue Transformation (UHET)

1. Mean Squared Error Bound

Here we demonstrate the validity of Eq. (3), which bounds the mean squared error of an average operation in terms of some original error bound. More precisely, we prove:

Lemma 1 (Mean Squared Error Bound)**.** *Consider an arbitrary unitary operation defined by $\mathcal{U}(\rho) := U\rho U^\dagger$ with a unitary operator U and a density operator ρ on a Hilbert space $\mathcal{H}$. If the error (in terms of the 1-norm) of a set of deterministic quantum operations (completely-positive trace-preserving maps) $\mathcal{F}_j : \mathcal{L}(\mathcal{H}) \to \mathcal{L}(\mathcal{H})$ and a probability distribution $\{p_j\}$ satisfies*

$$\sup_{|\psi\rangle\in\mathcal{H}} \|\mathcal{U}(|\psi\rangle\langle\psi|) - \sum_j p_j \mathcal{F}_j(|\psi\rangle\langle\psi|)\|_1 \leq \Delta, \tag{B1}$$

for some $\Delta > 0$, then the mean squared error of the average operation $\sum_j p_j \mathcal{F}_j$ is upper bounded by

$$\sup_{|\psi\rangle;\||\psi\rangle\|=1} \sum_j p_j \|\mathcal{U}(|\psi\rangle\langle\psi|) - \mathcal{F}_j(|\psi\rangle\langle\psi|)\|_1^2 \leq 2\Delta, \tag{B2}$$

where $|\psi\rangle$ is any pure state on $\mathcal{H}$.

This lemma is proven in App. B of Ref. [10]. Furthermore, if a modified version of Eq. (B1) is applied to an extended Hilbert space, namely

$$\sup_{\substack{\dim(\mathcal{H}')\\ |\psi\rangle\in\mathcal{H}\otimes\mathcal{H}'\\ \||\psi\rangle\|=1}} \|\mathcal{U}\otimes\mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|) - \sum_j p_j \mathcal{F}_j\otimes\mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|)\|_1 \leq \Delta \tag{B3}$$

holds, then a stronger version of Eq. (B2), i.e.,

$$\sup_{\substack{\dim(\mathcal{H}')\\ |\psi\rangle\in\mathcal{H}\otimes\mathcal{H}'\\ \||\psi\rangle\|=1}} \sum_j p_j \|\mathcal{U}\otimes\mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|) - \mathcal{F}_j\otimes\mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|)\|_1^2 \leq 2\Delta, \tag{B4}$$

follows.

Appendix C: Uncompiled UHET Algorithm

Here we provide details relevant to the uncompiled UHET algorithm presented throughout Sec. IV of the main text.

1. Sufficient Number of qDRIFT Iterations

The qDRIFT procedure is a stochastic method for simulating Hamiltonian dynamics [5]. Here, we determine a sufficient number of iterations to ensure a sufficiently small error ϵ.

Lemma 2. *Suppose one is given access to the dynamics $e^{-iH_j\tau}$ ($\tau > 0$) corresponding to a set of Hamiltonians $\{H_j\}_j$ on $\mathcal{L}(\mathcal{H})$ which normalized as $\|H_j\|_{\mathrm{op}} = 1$. Then, the dynamics e^{-iHt} ($t > 0$) of a Hamiltonian represented as $H = \sum_j h_j H_j$ for a set of positive coefficients $\{h_j\}_j$ can be simulated using qDRIFT with an error of at most $(2\lambda^2 t^2/N)e^{2\lambda t/N}$ where $\lambda := \sum_j h_j$ and N refers to the number of iterations of the random sampling. Here, the error is quantified by*

$$\frac{1}{2}\|\mathcal{F}_{\mathrm{target}} - \mathcal{F}_{\mathrm{approx}}\|_{\diamond}, \tag{C1}$$

where $\mathcal{F}_{\mathrm{target}}(\rho) := e^{-iHt}\rho e^{iHt}$ and $\mathcal{F}_{\mathrm{approx}}$ is the average quantum operation simulated by the qDRIFT protocol.

The proof of this Lemma is provided in Ref. [5]. In particular, if N is chosen as $N(\lambda, t, 2\epsilon)$ (as defined in Eq. (4)), then the error in terms of Eq. (C1) is bounded from above by ϵ, i.e.,

$$\frac{2\lambda^2 t^2}{N(\lambda, t, 2\epsilon)} e^{2\lambda t/N(\lambda,t,2\epsilon)} \leq 2\lambda^2 t^2 \cdot \left(\frac{5\lambda^2 t^2}{\epsilon}\right)^{-1} e^{2\lambda t\cdot(5\lambda t/2)^{-1}} = \frac{2\epsilon}{5} e^{4/5} < \epsilon. \tag{C2}$$

Moreover, making use of Eqs. (B3) and (B4), it follows that the error in terms of Eq. (1), namely

$$\sup_{\substack{|\psi\rangle\in\mathcal{H}\otimes\mathcal{H}' \\ \||\psi\rangle\|=1 \\ \dim\mathcal{H}'}} \|\mathcal{F}_{\mathrm{target}} \otimes \mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|) - \mathcal{F}_{\mathrm{approx}} \otimes \mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|)\|_1 \tag{C3}$$

is upper bounded by twice that of Eq. (C1) and can therefore be upper bounded by ϵ by choosing the iteration number as $N(\lambda, t, \epsilon)$.

2. Definition of g_f in Fourier Series Simulation [Subroutine 2, Eq. (7)]

The first step in the Fourier Series Simulation part of our UHET algorithm is to modify the desired transformation function f to a suitable periodically smooth one $\tilde{f}$, which is in turn defined in terms of the function g below [see Eq. (7)]:

$$\begin{aligned} g_f(x) := {} & C_1\cos(\pi x) + S_1\sin(\pi x) + \frac{1}{2}S_2\sin(2\pi x) + C_2\cos(2\pi x) + C_3\cos(3\pi x) + \frac{1}{3}S_3\sin(3\pi x) \\ & + \frac{1}{4}S_4\sin(4\pi x) + C_4\cos(4\pi x), \end{aligned} \tag{C4}$$

where the coefficients $C_1,\ C_2,\ C_3,\ C_4,\ S_1,\ S_2,\ S_3,\ S_4$ are defined succinctly in terms of the coefficient matrix

$$\Phi := \begin{pmatrix} 9/16 & 1/16 & -9/16 & -1/16 \\ 2/3 & 1/24 & 2/3 & 1/24 \\ -1/16 & -1/16 & 1/16 & 1/16 \\ -1/6 & -1/24 & -1/6 & -1/24 \end{pmatrix} \tag{C5}$$

as

$$\begin{aligned} (C_1, C_2, C_3, C_4)^T &:= \Phi \cdot \left(f(-1), \frac{4f^{(2)}(-1)}{\pi^2}, f(1), \frac{4f^{(2)}(1)}{\pi}\right)^T, \\ (S_1, S_2, S_3, S_4)^T &:= \Phi \cdot \left(\frac{2f^{(1)}(-1)}{\pi}, \frac{8f^{(3)}(-1)}{\pi^3}, \frac{2f^{(1)}(1)}{\pi}, \frac{8f^{(3)}(1)}{\pi^3}\right)^T. \end{aligned} \tag{C6}$$

3. Error Analysis of Fourier Series Simulation (Subroutine 2)

Here, we rigorously analyze the performance of Subroutine 2. In particular, we prove the following Theorem:

Theorem 1. *Subroutine 2 outputs $e^{-if(H_0)t}|\psi\rangle$ with an error of at most ϵ. Here, the error is defined as:*

$$\sup_{\substack{\dim(\mathcal{H}') \\ |\psi\rangle\in\mathcal{H}\otimes\mathcal{H}' \\ \||\psi\rangle\|=1}} \|\mathcal{G}\otimes\mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|) - \sum_j p_j(\mathcal{G}_j\otimes\mathcal{I}_{\mathcal{H}'})(|\psi\rangle\langle\psi|)\|_1, \tag{C7}$$

where $\mathcal{G}(\rho) := e^{-if(H_0)t}\rho e^{if(H_0)t}$, j labels the tuple of all random indices k chosen in N iterations, p_j refers to the probability that j is chosen, which leads to the particular quantum operation $\mathcal{G}_j$ being simulated, and $\mathcal{H}'$ is an auxiliary system of arbitrary dimension.

We prove this theorem as follows. First, we decompose the error into two contributions: that of approximating the function $\tilde{f}$ via its (truncated) Fourier series and that of the qDRIFT protocol itself. We then upper bound each error to derive an upper bound for the total error of the subroutine.

Proof: First, we define three quantum operations $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_3 : \mathcal{L}(\mathcal{H}_c\otimes\mathcal{H}) \to \mathcal{L}(\mathcal{H}_c\otimes\mathcal{H})$:

$$\begin{aligned}\mathcal{G}_1(\rho) &:= e^{-i(X\otimes f(H_0))t}\rho e^{i(X\otimes f(H_0))t} \\ \mathcal{G}_2(\rho) &:= e^{-i(X\otimes \tilde{f}_K(H_0))t}\rho e^{i(X\otimes \tilde{f}_K(H_0))t} \\ \mathcal{G}_3 &:= \sum_j p_j\mathcal{G}'_j.\end{aligned} \tag{C8}$$

where $\tilde{f}_K(x) := \sum_{k=-K}^{K}\tilde{c}_k e^{ik\pi(x+1)/2}$ is the truncated Fourier representation of f (which is a real function due to $\tilde{c}_{-k} = \tilde{c}_k^*$) and $\mathcal{G}'_j : \mathcal{L}(\mathcal{H}_c\otimes\mathcal{H}) \to \mathcal{L}(\mathcal{H}_c\otimes\mathcal{H})$ is the quantum operation simulated between steps 5 and 9 of Subroutine 2. By applying $\mathcal{G}_1$ to the initial state $|+\rangle\otimes|\psi\rangle$ and finally tracing over the control Hilbert space $\mathcal{H}_c$, one yields the desired transformation, i.e., the ideal dynamics. The expression $\mathcal{G}_2$ corresponds to the simulated dynamics of a truncated Fourier series in the absence of any Trotterization error (i.e., a perfectly accurate simulation of the finite Fourier series). Lastly, $\mathcal{G}_3$ denotes the actual dynamics simulated in Subroutine 2, which includes errors due to both finite Fourier series cutoff and Trotterization.

Consider the norm E defined as

$$E(\mathcal{F}) := \sup_{\substack{|\psi\rangle\in\mathcal{H}_c\otimes\mathcal{H}\otimes\mathcal{H}' \\ \||\psi\rangle\|=1 \\ \dim\mathcal{H}'}} \|\mathcal{F}\otimes\mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|)\|_1, \tag{C9}$$

for any quantum operation $\mathcal{F} : \mathcal{L}(\mathcal{H}_c\otimes\mathcal{H}) \to \mathcal{L}(\mathcal{H}_c\otimes\mathcal{H})$. The difference $E(\mathcal{G}_3 - \mathcal{G}_1)$, therefore quantifies the error between the simulated dynamics and the ideal case.

Since $\mathcal{G}_3$ is a qDRIFT protocol approximating the ideal dynamics by simulating a Hamiltonian $X\otimes\tilde{f}_K(H_0)$ with finite cutoff number K as shown in Eq. (10), it follows from Lemma 2 that $E(\mathcal{G}_3 - \mathcal{G}_2) = \epsilon/2$ (since the precision is chosen as $\epsilon/2$). Furthermore, we can make use of the identity $\|\,|\beta\rangle\langle\beta| - |\gamma\rangle\langle\gamma|\,\|_1 = 2\sqrt{1 - |\langle\beta|\gamma\rangle|^2}$ (see, e.g., Eq. (1.186) of [32]) where $|\beta\rangle$, $|\gamma\rangle$ are unit vectors in the same Hilbert space, as well as the triangle inequality to yield

$$\begin{aligned}E(\mathcal{G}_3 - \mathcal{G}_1) &\leq E(\mathcal{G}_3 - \mathcal{G}_2) + E(\mathcal{G}_2 - \mathcal{G}_1) \\ &\leq \frac{\epsilon}{2} + \sup_{\substack{|\psi\rangle\in\mathcal{H}_c\otimes\mathcal{H}\otimes\mathcal{H}' \\ \||\psi\rangle\|=1 \\ \dim(\mathcal{H}')}} 2\left[1 - |\langle\psi| e^{-iX\otimes(f(H_0)-\tilde{f}_K(H_0))t}\otimes I|\psi\rangle|^2\right]^{1/2}.\end{aligned} \tag{C10}$$

We now lower bound the r.h.s. by decomposing $|\psi\rangle =: \sum_{s,m} a_{s,m}|s\rangle|E_m\rangle|\psi_{s,m}\rangle$ ($\sum_{s,m}|a_{s,m}|^2 = 1$), where E_m and $|E_m\rangle$ are the eigenvalues and eigenvectors of H_0 respectively, $|s\rangle \in \{|+\rangle, |-\rangle\}$ are eigenvectors of the operator X in

$\mathcal{H}_c$, and $|\psi_{s,m}\rangle$ is a unit vector in $\mathcal{H}'$. With this, it follows that $|\langle\psi| e^{-iX\otimes(f(H_0)-\tilde{f}_K(H_0))t}\otimes I|\psi\rangle|$ is lower bounded as

$$\begin{aligned}
|\langle\psi| e^{-iX\otimes(f(H_0)-\tilde{f}_K(H_0))t}\otimes I|\psi\rangle| &= \left|\sum_{s,m}|a_{s,m}|^2 e^{-i\langle s|X|s\rangle\cdot(f(E_m)-\tilde{f}_K(E_m))t}\right| \\
&\geq \left|\sum_{s,m}|a_{s,m}|^2 \mathrm{Re}(e^{-i\langle s|X|s\rangle\cdot(f(E_m)-\tilde{f}_K(E_m))t})\right| \\
&\geq \left|\sum_{s,m}|a_{s,m}|^2 \cos(\tfrac{1}{2}R(f-\tilde{f}_K)t)\right| \\
&= \cos(\tfrac{1}{2}R(f-\tilde{f}_K)t), \qquad \text{(C11)}
\end{aligned}$$

where $R(g) := 2\max_{x\in[-1,1]}|g(x)|$ for a function $g:[-1,1]\to\mathbb{R}$. Substituting Eq. (C11) into Eq. (C10) and invoking Eq. (8), we finally have

$$E(\mathcal{G}_3-\mathcal{G}_1)\leq\frac{\epsilon}{2}+2\sin[R(f-\tilde{f}_K)t/2]\leq\frac{\epsilon}{2}+R(f-\tilde{f}_K)t\leq\frac{\epsilon}{2}+\frac{\epsilon}{2}=\epsilon. \qquad \text{(C12)}$$

By substituting back to Eq. (C9), we have that

$$\begin{aligned}
\epsilon\geq E(\mathcal{G}_3-\mathcal{G}_1) &\geq \sup_{\substack{|\psi\rangle\in\mathcal{H}_c\otimes\mathcal{H}\otimes\mathcal{H}'\\ \||\psi\rangle\|=1\\ \dim\mathcal{H}'}}\|\mathrm{tr}_{\mathcal{H}_c}[(\mathcal{G}_3-\mathcal{G}_1)\otimes\mathcal{I}_{\mathcal{H}'}](|\psi\rangle\langle\psi|)\|_1 \\
&\geq \sup_{\substack{|\psi\rangle\in\mathcal{H}\otimes\mathcal{H}'\\ \||\psi\rangle\|=1\\ \dim\mathcal{H}'}}\|\mathrm{tr}_{\mathcal{H}_c}[(\mathcal{G}_3-\mathcal{G}_1)\otimes\mathcal{I}_{\mathcal{H}'}](|+\rangle\langle+|\otimes|\psi\rangle\langle\psi|)\|_1, \qquad \text{(C13)}
\end{aligned}$$

which is equal to the expression in Eq. (C7), therefore asserting our claim. □

4. Fourier Series Convergence

Lemma 3. *For arbitrary $f:[-1,1]\to\mathbb{R}$ which belongs to class C^3 with $f^{(4)}$ being piecewise C^2, the function $\tilde{f}$ defined in Eq. (7) is periodically 4-smooth.*

Proof: We will now demonstrate that $\tilde{f}$ belongs to class C^3, $\tilde{f}^{(j)}(1)=\tilde{f}^{(j)}(-1)$ ($j\in\{0,1,2,3\}$), and $\tilde{f}^{(4)}$ is piecewise C^2. Let $S\subset[-1,1]$ denote the finite set of points outside of which $f^{(4)}$ is defined and is continuously differentiable. It is straightforward to see that $\tilde{f}^{(4)}$ is piecewise C^2 with exceptional points $x\in\tilde{S}\cup\{0\}$ where $\tilde{S}:=\{(x+1)/2\mid x\in S\}$. Thus it suffices to show that $\tilde{f}$ belongs to class C^3 and $\tilde{f}^{(j)}(1)=\tilde{f}^{(j)}(-1)\ \forall\ j\in\{0,1,2,3\}$. This follows directly from the following equations, which in turn follow from the definition of g_f

$$\begin{aligned}
\frac{\mathrm{d}^j}{\mathrm{d}x^j}g_f(x)|_{x=0} &= \frac{\mathrm{d}^j}{\mathrm{d}x^j}f(2x-1)|_{x=0} \\
\frac{\mathrm{d}^j}{\mathrm{d}x^j}g_f(x)|_{x=-1} &= \frac{\mathrm{d}^j}{\mathrm{d}x^j}f(2x-1)|_{x=1}, \qquad \text{(C14)}
\end{aligned}$$

where $j\in\{0,1,2,3\}$. □

We now move on to analyze the convergence rate of the Fourier series being simulated, which leads to a relationship between the allowed error ϵ and the truncation number K. Whenever a function $g:[-1,1]\to\mathbb{R}$ is piecewise smooth, its Fourier series converges (except for at a finite number of discontinuities; see, e.g., Ref. [33]). In general, for a periodically J-smooth function, the rate of convergence can be determined via the following lemma.

Lemma 4. *For any periodically J-smooth function $g:[-1,1]\to\mathbb{R}$, its Fourier coefficients $c_k:=(1/2)\int_{-1}^{1}\mathrm{d}x\, g(x)e^{-i\pi kx}$ converge at the rate:*

$$\lim_{|k|\to\infty}|c_k||k|^J<\infty. \qquad \text{(C15)}$$

In particular, if g is strictly *periodically J-smooth, then*

$$\sum_{k=K}^{\infty} |c_k|^2 = \Omega(K^{-(2J+1)}) \quad \textit{and} \quad \sum_{k=\infty}^{-K} |c_k|^2 = \Omega(K^{-(2J+1)}) \tag{C16}$$

hold for $K > 0$.

Proof: If g is periodically J-smooth, then each coefficient c_k can be rewritten using the following Fourier asymptotic coefficient expansion method [34]:

$$\begin{aligned} c_k &= \frac{1}{2}\int_{-1}^{1} \mathrm{d}x\, g(x) e^{-i\pi k x} \\ &= \frac{1}{2}\frac{i}{\pi k}\left[(-1)^k \{g(1) - g(-1)\} - \int_{-1}^{1} \mathrm{d}x\, g^{(1)}(x) e^{-i\pi k x}\right] \\ &= \cdots = \frac{1}{2}\sum_{j=0}^{J-1} (-1)^{k+j} \left(\frac{i}{\pi k}\right)^{j+1} \left\{g^{(j)}(1) - g^{(j)}(-1)\right\} + \frac{(-1)^J}{2}\left(\frac{i}{\pi k}\right)^J \int_{-1}^{1} \mathrm{d}x\, g^{(J)}(x) e^{-i\pi k x}. \end{aligned} \tag{C17}$$

The final line follows by successively applying integration by parts on the relevant integral term; this technique is valid since the interval of integration $[-1, 1]$ can be decomposed into smaller intervals $[x_0, x_1], \ldots, [x_{n-1}, x_n]$ (with $x_0 = -1$ and $x_n = 1$) between the exceptional points, upon which all derivatives of g are C^2 by assumption. The rightmost part of Eq. (C17) is $O(1/k^J)$ because all of the terms in the summation vanish due to periodicity and the remaining integral in the second term is bounded. This proves the first part of our claim, i.e., the rate of convergence according to Eq. (C15).

Furthermore, when g is *strictly* periodically J-smooth, then the integral term on the r.h.s. of Eq. (C17) can be rewritten as:

$$\begin{aligned} \int_{-1}^{1} \mathrm{d}x\, g^{(J)}(x) e^{-i\pi k x} &= \int_{-1}^{x_1} \mathrm{d}x\, g^{(J)}(x) e^{-i\pi k x} + \cdots + \int_{x_{n-1}}^{1} \mathrm{d}x\, g^{(J)}(x) e^{-i\pi k x} \\ &= \frac{i}{\pi k}\Big[e^{i\pi k}\{g^{(J)}(1^-) - g^{(J)}(-1^+)\} + e^{-i\pi k x_1}\{g^{(J)}(x_1^-) - g^{(J)}(x_1^+)\} + \\ &\quad \cdots + e^{-i\pi k x_{n-1}}\{g^{(J)}(x_{n-1}^-) - g^{(J)}(x_{n-1}^+)\}\Big] - \frac{i}{\pi k}\int_{-1}^{1} \mathrm{d}x g^{(J+1)}(x) e^{-i\pi k x}. \end{aligned} \tag{C18}$$

By rewriting the final integral similarly, we can show that this expression is of $O(1/k^2)$. Specifically, let us define g_m for $m \in \{0, \ldots n-1\}$ as $g_0 := g^{(J)}(1^-) - g^{(J)}(-1^+)$ and $g_m := g^{(J)}(x_m^-) - g^{(J)}(x_m^+)$ for $m > 1$; with this, Eq. (C18) can be expressed succinctly as $\frac{i}{\pi k}\sum_{m=0}^{n-1} g_m e^{-i\pi k x_m} + O(1/k^2)$. Substituting Eq. (C18) into Eq. (C17) then yields

$$c_k = \frac{(-1)^J}{2}\left(\frac{i}{\pi k}\right)^{J+1}\left(\sum_{m=0}^{n-1} g_m e^{-i\pi k x_m}\right) + O(k^{-(J+2)}). \tag{C19}$$

In order to reach the desired claim, we now seek to lower bound $|c_k|$. For any $k_0 \in \mathbb{Z}$ and $M > 0$, we have that

$$\begin{aligned} \sum_{k=k_0}^{k_0+M-1}\left|\sum_{m=0}^{n-1} g_m e^{-i\pi k x_m}\right|^2 &= \sum_{k=k_0}^{k_0+M-1}\left[\sum_{m=0}^{n-1}|g_m|^2 + \sum_{m_1\neq m_2} g_{m_1}^* g_{m_2} e^{-i\pi k(x_{m_2}-x_{m_1})}\right] \\ &\geq \sum_{k=k_0}^{k_0+M-1}\sum_{m=0}^{n-1}|g_m|^2 - \left|\sum_{m_1\neq m_2} g_{m_1}^* g_{m_2}\sum_{k=k_0}^{k_0+M-1} e^{-i\pi k(x_{m_2}-x_{m_1})}\right| \\ &\geq \sum_{k=k_0}^{k_0+M-1}\sum_{m=0}^{n-1}|g_m|^2 - \sum_{m_1\neq m_2}|g_{m_1}||g_{m_2}|\left|\sum_{k=k_0}^{k_0+M-1} e^{-i\pi k(x_{m_2}-x_{m_1})}\right| \\ &\geq M\sum_{m=0}^{n-1}|g_m|^2 - \frac{1}{\Delta}\left(\sum_{m=0}^{n-1}|g_m|\right)^2, \end{aligned} \tag{C20}$$

where $\Delta := \sin\left[\frac{\pi}{2}\left\{\min_{0\leq m_1<m_2\leq n-1}(x_{m_2}-x_{m_1}, 2-(x_{m_2}-x_{m_1}))\right\}\right] > 0$. The last inequality above follows by evaluating the geometric series and then bounding it appropriately, i.e., $|\sum_{k=k_0}^{k_0+M-1} e^{-i\pi kq}| = |(e^{-i\pi(k_0+M)q} - e^{-i\pi k_0 q})/(e^{-i\pi q} - 1)| \leq 2/|e^{-i\pi q}-1| = 1/\sin(\pi q/2)$ for $q>0$. Since $1/\sin(\pi q/2) = 1/\sin(\pi(2-q)/2)$ and $1/\sin(\pi q/2)$ is a decreasing function in $0<q\leq 1$, it follows that $\left|\sum_{k=k_0}^{k_0+M-1} e^{-i\pi k(x_{m_2}-x_{m_1})}\right|$ can be upper bounded by $1/\Delta$ for $m_1 \neq m_2$.

In particular, when M is chosen as $M' := \mathrm{ceil}\{[2(\sum_{m=0}^{n-1}|g_m|)^2]/[\Delta(\sum_{m=0}^{n-1}|g_m|^2)]\}$, the last line of Eq. (C20) yields

$$\begin{aligned} M'\sum_{m=0}^{n-1}|g_m|^2 - \frac{1}{\Delta}\left(\sum_{m=0}^{n-1}|g_m|\right)^2 &= M'\left[\sum_{m=0}^{n-1}|g_m|^2 - \frac{1}{M'\Delta}\left(\sum_{m=0}^{n-1}|g_m|\right)^2\right] \\ &\geq M'\left[\sum_{m=0}^{n-1}|g_m|^2 - \frac{1}{\Delta}\left(\sum_{m=0}^{n-1}|g_m|\right)^2 \cdot \frac{\Delta(\sum_{m=0}^{n-1}|g_m|^2)}{2(\sum_{m=0}^{n-1}|g_m|)^2}\right] \\ &= M'\cdot\frac{1}{2}\sum_{m=0}^{n-1}|g_m|^2. \end{aligned} \tag{C21}$$

Therefore, the average value of $\left|\sum_{m=0}^{n-1} g_m e^{-i\pi k x_m}\right|^2$ for $k\in\{k_0,\ldots,k_0+M'-1\}$ is lower bounded by $(\sum_{m=0}^{n-1}|g_m|^2)/2$. Since there exists at least one $k\in\{k_0,\ldots,k_0+M'-1\}$ such that $|\sum_{m=0}^{n-1} g_m e^{-i\pi k x_m}|^2 \geq (\sum_{m=0}^{n-1}|g_m|^2)/2$ for all $k_0>0$ and the $O(k^{-(J+2)})$ term converges to 0 quicker than $1/k^{J+1}$, there exists constants $C>0$ and $k_0'>0$ such that for all $k_0>k_0'$, there exists at least one $k\in\{k_0,\ldots,k_0+M'-1\}$ such that $|c_k| > C/k^{J+1} \geq C/(k_0+M'-1)^{J+1}$. Since $\sum_{k=K}^{\infty}|c_k|^2 = \sum_{l=1}^{\infty}\sum_{j=0}^{M'-1}|c_{K+(l-1)M'+j}|^2 \geq \sum_{l=1}^{\infty}|C/(K+lM'-1)^{J+1}|^2 = \Theta(K^{-(2J+1)})$ for sufficiently large K, it follows that $\sum_{k=K}^{\infty}|c_k|^2 = \Omega(K^{-(2J+1)})$ and $\sum_{k=\infty}^{-K}|c_k|^2 = \Omega(K^{-(2J+1)})$ for $K>0$, asserting the second part of our claim. □

Appendix D: Compiled UHET Algorithm

1. Proof of Eq. (17) for the Compiled Algorithm (Algorithm 4)

Here, we prove the validity of Eq. (17) of the main text. To be precise with notation, we denote the order of matrix multiplication as $\Pi_{j=1}^{n} M_j := M_1 \cdots M_n$. Formally, we show:

Lemma 5. *Let $k \in \mathbb{Z}$, $N \in \mathbb{Z}_{\geq 0}$, and $j := (\vec{v}_1, \dots, \vec{v}_N) \in (\{0,1,2,3\}^n)^N$. For any unitary $W_{k,j}^{(N)}$ defined as per Eq. (18), namely*

$$W_{k,j}^{(N)} := \left[\prod_{m=1}^{N} \mathtt{ctrl}(\sigma_{\vec{v}_m})(I \otimes e^{-i\frac{k\pi}{2N}H})\mathtt{ctrl}(\sigma_{\vec{v}_m})\right](e^{-i\frac{k\pi}{4}Z} \otimes I), \tag{D1}$$

there exist parameters $A_{k,N} > 0$ and $\theta_{k,N} \in [0, 2\pi)$ such that

$$\begin{aligned}&\left(\frac{1}{4}\right)^{nN} \sum_{\vec{v}_1,\dots,\vec{v}_N} (W_{k,j}^{(N)})^{\dagger}([\cos\phi_k X - \sin\phi_k Y] \otimes I)W_{k,j}^{(N)} \\ =&A_{k,N}(e^{i\frac{\theta_{k,N}}{2}Z} \otimes I)\begin{pmatrix} 0 & e^{i\phi_k}e^{i\frac{k\pi}{2}(H_0+I)} \\ e^{-i\phi_k}e^{-i\frac{k\pi}{2}(H_0+I)} & 0 \end{pmatrix}(e^{-i\frac{\theta_{k,N}}{2}Z} \otimes I),\end{aligned} \tag{D2}$$

where H_0 is the traceless part of H, i.e., $H_0 := H - (\mathrm{tr}[H]/2^n)I$. Furthermore, $A_{k,N}$ and $\theta_{k,N}$ satisfy

$$1 - \frac{\pi^2 k^2}{8N} \leq A_{k,N} \leq 1 \quad \text{and} \quad |\theta_{k,N}| \leq \frac{\pi^3 k^3}{32N^2} \quad (N \geq 0.625\pi k). \tag{D3}$$

Proof: For the special case $N = 0$ (in which case we employ the convention that for any matrices $\{X_m\}_m$, we have $\Pi_{m=1}^{N=0} X_m = I$), it is straightforward to verify that $(A_{k,N}, \theta_{k,N}) = (1, 0)$ satisfies Eq. (D2); thus, the remainder of the proof concerns $N > 0$.

For convenience, we define

$$\Upsilon_{k,N,j} := \prod_{m=1}^{N} \mathtt{ctrl}(\sigma_{\vec{v}_m})(I \otimes e^{-i\frac{k\pi}{2N}H})\mathtt{ctrl}(\sigma_{\vec{v}_m}) = \begin{pmatrix} e^{-i\frac{k\pi}{2}H} & 0 \\ 0 & \prod_{m=1}^{N} \sigma_{\vec{v}_m} e^{-i\frac{k\pi}{2N}H}\sigma_{\vec{v}_m} \end{pmatrix}. \tag{D4}$$

With this, Eq. (D2) can be rewritten as

$$\begin{aligned}&\left(\frac{1}{4}\right)^{nN} \sum_{\vec{v}_1,\dots,\vec{v}_N} \Upsilon_{k,N,j}^{\dagger}([\cos\phi_k X - \sin\phi_k Y] \otimes I)\Upsilon_{k,N,j} \\ &= \left(\frac{1}{4}\right)^{nN} \sum_{\vec{v}_1,\dots,\vec{v}_N} \begin{pmatrix} 0 & e^{i\phi_k}e^{i\frac{k\pi}{2}H}(\prod_{l=1}^{N} \sigma_{\vec{v}_l} e^{-i\frac{k\pi}{2N}H}\sigma_{\vec{v}_l}) \\ e^{-i\phi_k}(\prod_{l=1}^{N} \sigma_{\vec{v}_l} e^{-i\frac{k\pi}{2N}H}\sigma_{\vec{v}_l})^{\dagger}e^{-i\frac{k\pi}{2}H} & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & e^{i\phi_k}e^{i\frac{k\pi}{2}H}[\prod_{l=1}^{N}\{(\frac{1}{4})^n \sum_{\vec{v}_l} \sigma_{\vec{v}_l} e^{-i\frac{k\pi}{2N}H}\sigma_{\vec{v}_l}\}] \\ e^{-i\phi_k}[\prod_{l=1}^{N}\{(\frac{1}{4})^n \sum_{\vec{v}_l} \sigma_{\vec{v}_l} e^{-i\frac{k\pi}{2N}H}\sigma_{\vec{v}_l}\}]^{\dagger}e^{-i\frac{k\pi}{2}H} & 0 \end{pmatrix} \\ &= \left(\frac{1}{2}\right)^{nN} \begin{pmatrix} 0 & e^{i\phi_k}\{\mathrm{tr}[e^{-i\frac{k\pi}{2N}H}]\}^N e^{i\frac{k\pi}{2}H} \\ e^{-i\phi_k}\{\mathrm{tr}[e^{-i\frac{k\pi}{2N}H}]^*\}^N e^{-i\frac{k\pi}{2}H)} & 0 \end{pmatrix},\end{aligned} \tag{D5}$$

where we have made use of the matrix identity $\sum_{\vec{v}_1,\dots,\vec{v}_N}(\prod_{l=1}^{N} M_{\vec{v}_l}) = \prod_{l=1}^{N}(\sum_{\vec{v}_l} M_{\vec{v}_l})$. By now invoking $\{\mathrm{tr}(e^{-i\frac{\pi k}{2N}H})\}^N e^{i\frac{\pi k}{2}H} = \{\mathrm{tr}(e^{-i\frac{\pi k\alpha}{2N}}e^{-i\frac{\pi k}{2N}H_0})\}^N e^{i\frac{\pi k\alpha}{2}}e^{i\frac{\pi k}{2}H_0} = \{\mathrm{tr}(e^{-i\frac{\pi k}{2N}H_0})\}^N e^{i\frac{\pi k}{2}H_0}$, where $\alpha := \mathrm{tr}(H)/2^n$, the final line of Eq. (D5) can be rewritten as

$$\begin{aligned}&\left(\frac{1}{2}\right)^{nN} \begin{pmatrix} 0 & e^{i\phi_k}\{\mathrm{tr}(e^{-i\frac{\pi k}{2N}H})\}^N e^{i\frac{\pi k}{2}H} \\ e^{-i\phi_k}\{\mathrm{tr}(e^{-i\frac{\pi k}{2N}H})^*\}^N e^{-i\frac{\pi k}{2}H} & 0 \end{pmatrix} \\ =&\left(\frac{1}{2}\right)^{nN} \begin{pmatrix} 0 & e^{i\phi_k}\{\mathrm{tr}(e^{-i\frac{\pi k}{2N}H_0})\}^N e^{i\frac{\pi k}{2}H_0} \\ e^{-i\phi_k}\{\mathrm{tr}(e^{-i\frac{\pi k}{2N}H_0})^*\}^N e^{-i\frac{\pi k}{2}H_0} & 0 \end{pmatrix}.\end{aligned} \tag{D6}$$

Thus, by setting

$$A_{k,N}e^{i\theta_{k,N}} := \left[\frac{\mathrm{tr}(e^{-i\frac{\pi k}{2N}H_0})}{2^n}\right]^N, \tag{D7}$$

where $A_{k,N} \geq 0$ and $\theta_{k,N} \in \mathbb{R}$, we obtain

$$\left(\frac{1}{4}\right)^{nN} \sum_{\vec{v}_1,\ldots,\vec{v}_N} \Upsilon^\dagger_{k,N,j}([\cos\phi_k X - \sin\phi_k Y]\otimes I)\Upsilon_{k,N,j} = A_{k,N}\begin{pmatrix} 0 & e^{i(\phi_k+\theta_{k,N})}e^{i\frac{k\pi}{2}H_0} \\ e^{-i(\phi_k+\theta_{k,N})}e^{-i\frac{k\pi}{2}H_0} & 0 \end{pmatrix}, \tag{D8}$$

which is equivalent to Eq. (D2).

For the second part of the claim, note that when H_0 is diagonalized as $H_0 = \sum_{j=0}^{2^n-1} E_j |j\rangle\langle j|$, then the expression $\{\mathrm{tr}(e^{-i\frac{\pi k}{2N}H_0})/2^n\}^N$ reduces to

$$\{\mathrm{tr}(e^{-i\frac{\pi k}{2N}H_0})/2^n\}^N = \left(\frac{\sum_{j=0}^{2^n-1} e^{-i\frac{\pi k}{2N}E_j}}{2^n}\right)^N = \left(\frac{\sum_{j=0}^{2^n-1}\cos(\frac{\pi k}{2N}E_j) - i\sum_{j=0}^{2^n-1}\sin(\frac{\pi k}{2N}E_j)}{2^n}\right)^N, \tag{D9}$$

from which $A_{k,N} \leq 1$ follows. Further invoking $|E_j| \leq 1$ (which follows from our assumption that $\|H_0\|_{\mathrm{op}} = 1$) and the inequality $\cos(x) \geq 1 - x^2/2$ $(x \in \mathbb{R})$, it follows that $\cos(\frac{\pi k}{2N}E_j) \geq \cos\left(\frac{\pi k}{2N}\right) \geq 1 - \frac{1}{2}\left(\frac{\pi k}{2N}\right)^2$ and subsequently

$$A_{k,N} \geq \left|\frac{\sum_{j=0}^{2^n-1}\cos(\frac{\pi k}{2N}E_j)}{2^n}\right|^N \geq \left[1 - \frac{1}{2}\left(\frac{\pi k}{2N}\right)^2\right]^N \geq 1 - \frac{\pi^2 k^2}{8N}. \tag{D10}$$

Finally, due to the fact that $\sum_{j=0}^{2^n-1} E_j = 0$ (which follows from $\mathrm{tr}(H_0) = 0$) as well as the inequality $|\sin x - x| \leq (|x|^3/6)$ $(x \in \mathbb{R})$, we have

$$\left|\sum_{j=0}^{2^n-1}\sin\left(\frac{\pi k}{2N}E_j\right)\right| = \left|\sum_{j=0}^{2^n-1}\left[\sin\left(\frac{\pi k}{2N}E_j\right) - \frac{\pi k}{2N}E_j\right]\right| \leq \frac{1}{6}\sum_{j=0}^{2^n-1}\left|\frac{\pi k}{2N}E_j\right|^3 \leq \frac{2^n}{6}\left(\frac{\pi k}{2N}\right)^3. \tag{D11}$$

Then, since $\frac{1}{2^n}\sum_{j=0}^{2^n-1}\cos(\frac{\pi k}{2N}E_j) \geq \frac{1}{2^n}\sum_{j=0}^{2^n-1}\cos(\frac{\pi k}{2N}) = \cos(\frac{\pi k}{2N}) \geq 1 - \frac{1}{2}\left(\frac{\pi k}{2N}\right)^2$, it follows that $\theta_{k,N}$ is upper bounded by

$$|\theta_{k,N}| \leq N\tan^{-1}\left[\left\{\frac{1}{6}\left(\frac{\pi k}{2N}\right)^3\right\}\left\{1 - \frac{1}{2}\left(\frac{\pi k}{2N}\right)^2\right\}^{-1}\right] \leq \frac{\pi^3 k^3}{32N^2} \quad (N \geq 0.625\pi k). \tag{D12}$$

The final inequality follows from $\tan^{-1}[\frac{1}{6}x^3/(1-\frac{1}{2}x^2)] \leq \frac{1}{4}x^3$ $(0 \leq x \leq 0.8)$. □

2. Parameter Estimation for Compiled Algorithm (Algorithm 4)

The compiled UHET Algorithm 4 makes use of "correction" parameters $(\hat{A}_k, \hat{\theta}_k)$ to compensate the error of the main process and reduce the runtime from that of Algorithm 3. Here, we will develop two subroutines, namely Subroutine D5 and D6, that allow one to estimate such parameters without any knowledge of the seed Hamiltonian. Subroutine D6 is used in step 2 of Subroutine D5 to generate a state $|\phi(N', l, k\pi/N', 25k, \Phi)\rangle$ that is used for robust phase estimation, from which the parameters of interest can be estimated.

Subroutine D5 Generating parameters $(\hat{A}_l, \hat{\theta}_l)$

Input:

- A finite number of queries to a black-box Hamiltonian dynamics $e^{\pm iH\tau}$ of a seed Hamiltonian H with $\tau > 0$ on an n-qubit system $\mathcal{H}$
- Parameter $l \in \mathbb{Z}_{>0}$
- Allowed error $\epsilon > 0$
- Time $t > 0$

Output: Estimates $\hat{A}_l > \frac{1}{2}$ and $\hat{\theta}_l \in [0, 2\pi)$ of $A_{l,10l^2}$ and $\theta_{l,10l^2}$, respectively, with root mean square error of $|1 - (A_{l,10l^2} e^{i\theta_{l,10l^2}})/(\hat{A}_l e^{i\hat{\theta}_l})|$ that is upper bounded by ϵ/t.
Runtime: $\Theta(l^2 t^3 n/\epsilon^3)$

Procedure:

```
1: for $\Phi \in \{0, \frac{\pi}{2}\}$ do
2:     Perform robust phase estimation [30] with allowed root mean square of error set as $\epsilon/(\sqrt{2}t)$. Here, success of $|0\rangle -$measurements and $|+\rangle -$measurements are defined as obtaining outcomes for $|0\rangle \otimes I$ and $|+\rangle \otimes I$ when performing $Z-$ and $X-$basis measurements respectively on the first qubit of the state $|\phi(N', l, k\pi/N', 25k, \Phi)\rangle$, where $N' := N(1, k\pi, 1/(4\sqrt{2}))$, generated by Subroutine D6. This provides an estimate $\hat{v}_\Phi$.
3: end for
4: if $(1/2\pi)\sqrt{\hat{v}_0^2 + \hat{v}_{\pi/2}^2} \leq \frac{1}{2}$ then
5:     Return to step 1
6: end if
7: Compute $\hat{A}_l$, $\hat{\theta}_l$ by
       $$\hat{A}_l \cos(\hat{\theta}_l) = \frac{1}{2\pi}\hat{v}_0$$
       $$\hat{A}_l \sin(\hat{\theta}_l) = \frac{1}{2\pi}\hat{v}_{\pi/2}$$
8: Return $(\hat{A}_l, \hat{\theta}_l)$
```

Step 2 of this routine makes use of the following subroutine to generate an appropriate state.

Subroutine D6 Generating a state used in Subroutine D5

Input:

- A finite number of queries to a black box Hamiltonian dynamics $e^{\pm iH\tau}$ of a seed Hamiltonian H with $\tau > 0$ on an n-qubit system $\mathcal{H}$
- Parameters $N, l, M \in \mathbb{Z}_{>0}$
- $\gamma \in [0, 1]$
- $\Phi \in [0, 2\pi)$

Output: A random state $|\phi(N, l, \gamma, M, \Phi)\rangle$
Runtime: $\Theta(Nl^2 nM)$

Procedure:

```
1: Initialize:
     $|\mathrm{current}\rangle \leftarrow |0\rangle \otimes |0\rangle \in \mathcal{H}_c \otimes \mathcal{H}$
2: for $m \in \{1, \ldots, N\}$ do
3:     From $j' := \{\vec{u}_1, \ldots, \vec{u}_{10l^2}\}$, $j'' := \{\vec{w}_1, \ldots, \vec{w}_{l^2 M}\}$ where $\vec{u}_1, \ldots, \vec{u}_{10l^2}, \vec{w}_1, \ldots, \vec{w}_{l^2 M} \in \{0, 1, 2, 3\}^n$, and $s \in \{+1, -1\}$, uniformly randomly choose $j = (j', j'', s)$
4:     $|\mathrm{current}\rangle \leftarrow \widetilde{W}_{l,j,\Phi}(e^{-i\gamma Y} \otimes I)\widetilde{W}_{l,j,\Phi}^{\dagger} |\mathrm{current}\rangle$ for
       $$\widetilde{W}_{l,j,\Phi} := (e^{-i\frac{s\Phi}{2}Z} \otimes I)\left[\prod_{m''=1}^{l^2 M} \mathtt{ctrl}(\sigma_{\vec{w}_{m''}})(I \otimes e^{i\frac{s\pi}{2lM}H})\mathtt{ctrl}(\sigma_{\vec{w}_{m''}})\right]\left[\prod_{m'=1}^{10l^2} \mathtt{ctrl}(\sigma_{\vec{u}_{m'}})(I \otimes e^{-i\frac{s\pi}{20l}H})\mathtt{ctrl}(\sigma_{\vec{u}_{m'}})\right]$$
5: end for
6: Return $|\phi(N, l, \gamma, M, \Phi)\rangle := |\mathrm{current}\rangle$
```

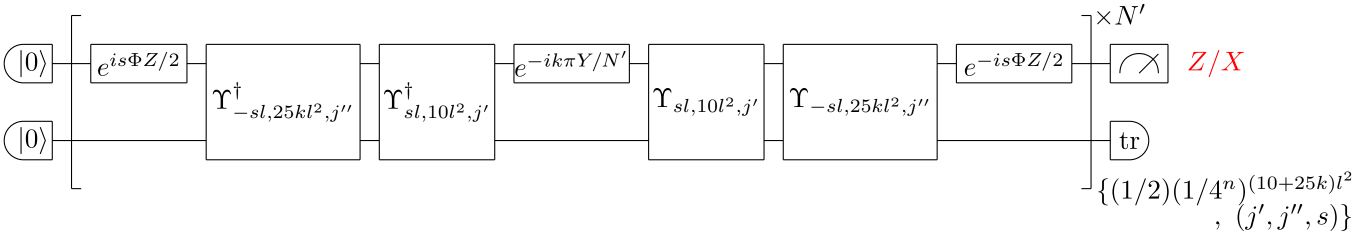


FIG. 11. *Compensation parameter estimation.*—The circuit used to obtain estimates (A_l, θ_l) that are subsequently used in Algorithm 4.

The runtime of Subroutine D6 scales as (Number of iterations N)×(The runtime of $\widetilde{W}_{l,j,\Phi}$)$=N\times\Theta(l^2nM)=\Theta(Nl^2nM)$.

A circuit depiction of the combination of Subroutines D5 and D6 is provided in Fig. 11. In particular, the preparation of the state $|\phi(N', l, k\pi/N', 25k, \Phi)\rangle$ in Subroutine D6 corresponds to the part of Fig. 11 before the measurement, with M, γ set to $25k, k\pi/N'$ respectively. For many values of k, Subroutine D5 implements this circuit and performs appropriate measurements on the state in order to estimate $(\hat{A}_l, \hat{\theta}_l)$ via robust phase estimation [30]. We now formalize the validity of this algorithm and demonstrate its runtime.

Theorem 2. *Subroutine D5 outputs estimates $(\hat{A}_l, \hat{\theta}_l)$ of the parameters $(A_{l,10l^2}, \theta_{l,10l^2})$ defined in Lemma 5 for $l>0$ with a root mean square of $|1-(A_{l,10l^2}e^{i\theta_{l,10l^2}})/(\hat{A}_l e^{i\hat{\theta}_l})|$ upper bounded by ϵ/t with a runtime $\Theta(l^2\epsilon^3 n/t^3)$.*

In order to prove Theorem 2, we combine the results of Lemma 5 with Lemma 6 below, which concerns the robust phase estimation procedure and is proven in Ref. [30]. We first show that steps 2 to 5 of Subroutine D6 generates the state $|\phi(N, l, \gamma, M, \Phi)\rangle$ by simulating a particular Hamiltonian that is proportional to $\hat{v}_\Phi$ [which is, in turn, a function of $(A_{l,10^2}, \theta_{l,10l^2})$ and is defined below] via qDRIFT. We then show that the success probabilities of $|0\rangle$-measurements and $|+\rangle$-measurements for each k (respectively, denoted $(0,k)-$measurements and $(+,k)-$measurements in Lemma 6) differ from $\frac{1}{2}(1+\cos(k\hat{v}_\Phi))$ and $\frac{1}{2}(1+\sin(k\hat{v}_\Phi))$, respectively, at most by $\frac{1}{\sqrt{8}}$, demonstrating that $\hat{v}_\Phi$ can be well-estimated by robust phase estimation. Finally, we prove that $A_{l,10l^2}$ and $\theta_{l,10l^2}$ can be obtained with a root mean square of error smaller than or equal to ϵ/t.

Lemma 6 (Robust Phase Estimation [30])**.** *Let $k\in\mathbb{Z}_{>0}$. Suppose that one can perform two families of measurements, $(0,k)$-measurements and $(+,k)$-measurements, whose success probabilities for obtaining outcomes 0 and $+$ respectively are given in terms of $\theta\in(-\pi,\pi]$ as*

$$\begin{aligned}(0,k)\text{-measurement:}\quad p_{0,k}(\theta) &:= \frac{1+\cos(k\theta)}{2}+\delta_0(k)\\ (+,k)\text{-measurement:}\quad p_{+,k}(\theta) &:= \frac{1+\sin(k\theta)}{2}+\delta_+(k),\end{aligned}$$

where $\delta_0(k)$ and $\delta_+(k)$ satisfy

$$\sup_k\{|\delta_0(k)|, |\delta_+(k)|\} =: \delta_{\text{sup}} < \frac{1}{\sqrt{8}}.$$

Then for any allowed standard deviation $s>0$, an estimate $\hat{\theta}$ of θ can be obtained with a root mean squared error smaller than or equal to s by a classical computation with runtime $O(\text{poly}K)$. This computation is a function of the numbers of successful $(0,2^{j-1})$-measurements and $(+,2^{j-1})$-measurements ($j\in\{1,\ldots,K\}$), with both measurement choices being implemented M_j times. Here, M_j and K are defined as

$$\begin{aligned}K &:= \text{ceil}\left[\log_2\left(\frac{3\pi}{s}\right)\right]\\ M_j &:= F(\delta_{\text{sup}})(3(K-j)+1)\\ F(\delta_{\text{sup}}) &:= \text{ceil}\left[\frac{\log\left(\frac{1}{2}(1-\sqrt{8}\delta_{\text{sup}})\right)}{\log\left(1-\frac{1}{2}(1-\sqrt{8}\delta_{\text{sup}})^2\right)}\right].\end{aligned}\tag{D13}$$

This lemma is proven in Ref. [30] (see Theorem 1 therein). We are now in a position to prove Theorem 2.

Proof of Theorem 2: To begin, note that steps 2 to 5 of Subroutine D6 simulates the dynamics $e^{-iH'}$ using qDRIFT for the Hamiltonian H' defined as

$$H' = \frac{N\gamma}{2}\left(\frac{1}{4}\right)^{(10+M)l^2} \sum_{\substack{\vec{u}_1,\ldots,\vec{u}_{10l^2}\\ \vec{w}_1,\ldots,\vec{w}_{l^2M}\\ s\in\{1,-1\}}} \widetilde{W}_{l,j,\Phi}(Y\otimes I)\widetilde{W}^{\dagger}_{l,j,\Phi} =: t\sum_j h_j H_j, \tag{D14}$$

where $j := \{(\vec{u}_1,\ldots,\vec{u}_{10l^2}),(\vec{w}_1,\ldots,\vec{w}_{l^2M}),s\}$, $t := N\gamma$, $h_j := \frac{1}{2}(\frac{1}{4})^{(10+M)l^2}$, and $H_j := \widetilde{W}_{l,j,\Phi}(Y\otimes I)\widetilde{W}^{\dagger}_{l,j,\Phi}$. With respect to Lemma 2, we can evaluate $\lambda = \sum_j h_j = 1$ and thus the error of simulating this dynamics via qDRIFT in terms of Eq. (C1) is upper bounded by $(2t^2/N)e^{2t/N}$.

We can further simplify Eq. (D14) as follows. First, note that $\widetilde{W}_{l,j,\Phi}$ can be rewritten as

$$\widetilde{W}_{l,j,\Phi} = (e^{-i\frac{s\Phi}{2}Z}\otimes I)\Upsilon_{-sl,l^2M,j''}\Upsilon_{sl,10l^2,j'} \tag{D15}$$

by using $\Upsilon_{k,N,j}$ defined in Eq. (D4). Moreover, according to Eq. (D8), the equality

$$\left(\frac{1}{4}\right)^{nN}\sum_{\vec{v}_1,\ldots,\vec{v}_N}\Upsilon^{\dagger}_{k,N,j}\begin{pmatrix} 0 & e^{-i\phi}I\\ e^{i\phi}I & 0\end{pmatrix}\Upsilon_{k,N,j} = A_{k,N}\begin{pmatrix} 0 & e^{-i(\phi-\theta_{k,N})}e^{i\frac{k\pi}{2}H_0}\\ e^{i(\phi-\theta_{k,N})}e^{-i\frac{k\pi}{2}H_0} & 0\end{pmatrix} \tag{D16}$$

holds for $\phi\in[0,2\pi)$. Thus, Eq. (D14) can be rewritten as

$$\begin{aligned}
&\frac{N\gamma}{2}\left(\frac{1}{4}\right)^{(10+M)l^2}\sum_{\substack{\vec{u}_1,\ldots,\vec{u}_{10l^2}\\ \vec{w}_1,\ldots,\vec{w}_{l^2M}\\ s\in\{1,-1\}}}\widetilde{W}_{l,j,\Phi}(Y\otimes I)\widetilde{W}^{\dagger}_{l,j,\Phi}\\
&=\frac{t}{2}\left(\frac{1}{4}\right)^{(10+M)l^2}\sum_{\substack{\vec{u}_1,\ldots,\vec{u}_{10l^2}\\ \vec{w}_1,\ldots,\vec{w}_{l^2M}\\ s\in\{1,-1\}}}(e^{-i\frac{s\Phi}{2}Z}\otimes I)\Upsilon_{-sl,l^2M,j''}\Upsilon_{sl,10l^2,j'}\begin{pmatrix}0 & -iI\\ iI & 0\end{pmatrix}\Upsilon^{\dagger}_{sl,10l^2,j'}\Upsilon^{\dagger}_{-sl,l^2M,j''}(e^{i\frac{s\Phi}{2}Z}\otimes I)\\
&=\frac{t}{2}\left(\frac{1}{4}\right)^{(10+M)l^2}\sum_{\substack{\vec{u}_1,\ldots,\vec{u}_{10l^2}\\ \vec{w}_1,\ldots,\vec{w}_{l^2M}\\ s\in\{1,-1\}}}(e^{-i\frac{s\Phi}{2}Z}\otimes I)\Upsilon^{\dagger}_{sl,l^2M,j''}\Upsilon^{\dagger}_{-sl,10l^2,j'}\begin{pmatrix}0 & -iI\\ iI & 0\end{pmatrix}\Upsilon_{-sl,10l^2,j'}\Upsilon_{sl,l^2M,j''}(e^{i\frac{s\Phi}{2}Z}\otimes I)\\
&=\frac{t}{2}A_{-sl,10l^2}\left(\frac{1}{4}\right)^{Ml^2}\sum_{\substack{\vec{w}_1,\ldots,\vec{w}_{l^2M}\\ s\in\{1,-1\}}}(e^{-i\frac{s\Phi}{2}Z}\otimes I)\Upsilon^{\dagger}_{sl,l^2M,j''}\begin{pmatrix}0 & -ie^{i\theta_{-sl,10l^2}}I\\ ie^{-i\theta_{-sl,10l^2}}I & 0\end{pmatrix}\Upsilon_{sl,l^2M,j''}(e^{i\frac{s\Phi}{2}Z}\otimes I)\\
&=\frac{t}{2}A_{-sl,10l^2}A_{sl,l^2M}\sum_{s\in\{1,-1\}}(e^{-i\frac{s\Phi}{2}Z}\otimes I)\begin{pmatrix}0 & -ie^{i(\theta_{-sl,10l^2}+\theta_{sl,l^2M})}I\\ ie^{-i(\theta_{-sl,10l^2}+\theta_{sl,l^2M})}I & 0\end{pmatrix}(e^{i\frac{s\Phi}{2}Z}\otimes I),
\end{aligned}\tag{D17}$$

where in the third line we made use of the fact that $\Upsilon_{-k,N,j} = (\Upsilon_{k,N,j'})^{\dagger}$, where $j' := (\vec{v}_N,\ldots,\vec{v}_1)$. By substituting Eq. (D7) into Eq. (D17), we have

$$t\sum_j h_jH_j = \frac{t}{2}\sum_s(e^{-i\frac{s\Phi}{2}Z}\otimes I)\begin{pmatrix}0 & -ia'_{l,M,s}a'_{l,10,-s}I\\ ia'_{l,M,-s}a'_{l,10,s}I & 0\end{pmatrix}(e^{i\frac{s\Phi}{2}Z}\otimes I) = ta_{l,M,\Phi}Y\otimes I, \tag{D18}$$

where $a'_{l,m,s} := [\frac{1}{2^n}\mathrm{tr}\{e^{-is(\pi/(2ml))H_0}\}]^{ml^2}$ and $a_{l,M,\Phi} := \frac{1}{2}(e^{i\Phi}a'_{l,M,-s}a'_{l,10,s}+e^{-i\Phi}a'_{l,M,s}a'_{l,10,-s})$ (note that $a_{l,M,\Phi}\in\mathbb{R}$). In terms of the parameters $A_{k,N}$ and $\theta_{k,N}$, it is straightforward to show that $a_{l,M,\Phi}$ can be expressed as

$$a_{l,M,\Phi} = A_{l,Ml^2}A_{l,10l^2}\cos(\theta_{l,10l^2}-\theta_{l^2,Ml^2}+\Phi). \tag{D19}$$

In Subroutine D5, the input parameters of the state are specifically chosen as $(N,l,\gamma,M,\Phi) = (N',l,k\pi/N',25k,\Theta)$ where $N' = N(1,k\pi,1/(4\sqrt{2}))$, with respect to which, $ta_{l,M,\Phi}$ is expressed as

$$ta_{l,M,\Theta} = k\pi A_{l,25kl^2}A_{l,10l^2}\cos(\theta_{l,10l^2}-\theta_{l,25kl^2}+\Phi). \tag{D20}$$

Using the fact that $A_{l,10l^2} \leq 1$, we can upper bound the difference between the above equation and $k\pi A_{l,10l^2}\cos(\theta_{l,10l^2} + \Phi)$ by

$$\begin{aligned}|ta_{l,25k,\Phi} - k\pi A_{l,10l^2}\cos(\theta_{l,10l^2} + \Phi)| &= k\pi |A_{l,25kl^2} A_{l,10l^2}\cos(\theta_{l,10l^2} - \theta_{l,25kl^2} + \Phi) - A_{l,10l^2}\cos(\theta_{l,10l^2} + \Phi)| \\ &\leq k\pi |A_{l,25kl^2}\cos(\theta_{l,10l^2} - \theta_{l,25kl^2} + \Phi) - \cos(\theta_{l,10l^2} + \Phi)| \\ &\leq k\pi\{|A_{l,25kl^2}[\cos(\theta_{l,10l^2} - \theta_{l,25kl^2} + \Phi) - \cos(\theta_{l,10l^2} + \Phi)]| + |(1 - A_{l,25kl^2})\cos(\theta_{l,10l^2} + \Phi)|\}.\end{aligned} \tag{D21}$$

Invoking the inequality $|\cos(\theta + \theta') - \cos(\theta)| = |-\int_{\theta}^{\theta+\theta'} \mathrm{d}x \sin(x)| \leq \theta'$ and Eq. (D3), we then have that

$$\begin{aligned}&k\pi\{|A_{l,25kl^2}[\cos(\theta_{l,10l^2} - \theta_{l,25kl^2} + \Phi) - \cos(\theta_{l,10l^2} + \Phi)]| + |(1 - A_{l,25kl^2})\cos(\theta_{l,10l^2} + \Phi)|\} \\ &\qquad \leq k\pi(|\theta_{l,25kl^2}| + |1 - A_{l,25kl^2}|) \leq k\pi\left(\left|\frac{\pi^3}{20000k^2 l}\right| + \left|\frac{\pi^2}{200k}\right|\right) < 0.16 < \frac{1}{4\sqrt{2}}.\end{aligned} \tag{D22}$$

Now, let us define $\mathcal{F}_1,\ \mathcal{F}_2,\ \mathcal{F}_3 : \mathcal{L}(\mathcal{H}_c \otimes \mathcal{H}) \to \mathcal{L}(\mathcal{H}_c \otimes \mathcal{H})$ as

$$\begin{aligned}\mathcal{F}_1(\rho) &:= e^{-ik\pi A_{l,10l^2}\cos(\theta_{l,10l^2}+\Phi) Y\otimes I} \rho e^{ik\pi A_{l,10l^2}\cos(\theta_{l,10l^2}+\Phi) Y\otimes I} \\ \mathcal{F}_2(\rho) &:= e^{-ik\pi a_{l,25k,\Phi} Y\otimes I} \rho e^{ik\pi a_{l,25k,\Phi} Y\otimes I} \\ \mathcal{F}_3 &:= \text{quantum operation simulated by qDRIFT in steps 2 to 5 of Subroutine D6}.\end{aligned} \tag{D23}$$

With this, the success probability of $|0\rangle$-measurements and $|+\rangle$-measurements, which is given by $\langle 0|\,\mathrm{tr}_{\mathcal{H}}[\mathcal{F}_3(|0\rangle\langle 0| \otimes |0\rangle\langle 0|)]\,|0\rangle$ and $\langle +|\,\mathrm{tr}_{\mathcal{H}}[\mathcal{F}_3(|0\rangle\langle 0| \otimes |0\rangle\langle 0|)]\,|+\rangle$ respectively, satisfies

$$|S(\mathcal{F}_3, |\psi\rangle) - S(\mathcal{F}_1, |\psi\rangle)| \leq |S(\mathcal{F}_3, |\psi\rangle) - S(\mathcal{F}_2, |\psi\rangle)| + |S(\mathcal{F}_2, |\psi\rangle) - S(\mathcal{F}_1, |\psi\rangle)| < \frac{1}{4\sqrt{2}} + \frac{1}{4\sqrt{2}} = \frac{1}{\sqrt{8}}, \tag{D24}$$

where, for any quantum operation $\mathcal{F} : \mathcal{L}(\mathcal{H}_c \otimes \mathcal{H}) \to \mathcal{L}(\mathcal{H}_c \otimes \mathcal{H})$ and state $|\psi\rangle \in \mathcal{H}_c$, we define $S(\mathcal{F}, |\psi\rangle) := \langle\psi|\,\mathrm{tr}_{\mathcal{H}}[\mathcal{F}(|0\rangle\langle 0| \otimes |0\rangle\langle 0|)]\,|\psi\rangle$. Here, the upper bound of $|S(\mathcal{F}_3, |\psi\rangle) - S(\mathcal{F}_2, |\psi\rangle)|$ is obtained using Lemma 2, and that of $|S(\mathcal{F}_2, |\psi\rangle) - S(\mathcal{F}_1, |\psi\rangle)|$ is obtained by noting that $\mathrm{tr}_{\mathcal{H}}[\mathcal{F}_1(|0\rangle\langle 0| \otimes |0\rangle\langle 0|)]$ and $\mathrm{tr}_{\mathcal{H}}[\mathcal{F}_2(|0\rangle\langle 0| \otimes |0\rangle\langle 0|)]$ are pure states, and so we can use the fact that the operator norm of $|\beta\rangle\langle\beta| - |\gamma\rangle\langle\gamma|$ for unit vectors $|\beta\rangle$ and $|\gamma\rangle$ is $\sqrt{1 - |\langle\beta|\gamma\rangle|^2}$ (see Eq. (1.185) of Ref. [32]). In particular, $\langle\beta|\gamma\rangle$ in this case is evaluated as $\langle 0|\, e^{-ik\pi(A_{l,10l^2}\cos(\theta_{l,10l^2}+\Phi) - a_{l,25k,\Phi})Y}\,|0\rangle$ and $t = k\pi$. Invoking Eqs. (D21) and (D22), as well as a similar discussion to Eq. (C11), we have $|S(\mathcal{F}_2, |\psi\rangle) - S(\mathcal{F}_1, |\psi\rangle)| \leq 1/4\sqrt{2}$.

Setting the appropriate measurements as $|0\rangle$ and $|+\rangle$, we can express the success probabilities as

$$\begin{aligned}S(\mathcal{F}_1, |0\rangle) &= \frac{1 + \cos(2k\pi A_{l,10l^2}\cos(\theta_{l,10l^2} + \Phi))}{2} \\ S(\mathcal{F}_1, |+\rangle) &= \frac{1 + \sin(2k\pi A_{l,10l^2}\cos(\theta_{l,10l^2} + \Phi))}{2}.\end{aligned} \tag{D25}$$

Thus, an estimate $\hat{v}_\Phi$ of $2\pi A_{l,10l^2}\cos(\theta_{l,10l^2} + \Phi)$ can be successfully obtained by robust phase estimation. In particular, by setting K in Eq. (D13) as $K = \mathrm{ceil}[\log_2(3\sqrt{2}t/\epsilon)]$, one can estimate $A_{l,10l^2}\cos(\theta_{l,10l^2} + \Phi)$ with root mean square of error upper bounded by $\epsilon/(2\sqrt{2}t)$ with a runtime $\Theta(l^2 t^3 n/\epsilon^3)$. This follows from the fact that the runtime of generating $|\phi(N', l, k\pi/N', 25k, \Phi)\rangle$ is $\Theta(l^2 n k^3)$ and invoking

$$\sum_{j=1}^{K} (K - j) r^j = \frac{r^{K+1} - K r^2 + (K-1) r}{(r-1)^2} \qquad r > 1. \tag{D26}$$

Thus, the total runtime of the robust phase estimation procedure is given by $\sum_{j=1}^{K} M_j \Theta(l^2 n (2^{j-1})^3) = \Theta(l^2 n t^3/\epsilon^3)$. Finally, by setting $\Phi \in \{0, \frac{\pi}{2}\}$, one can obtain estimates $\hat{v}_0, \hat{v}_{\pi/2}$ of $2\pi A_{l,10l^2}\cos(\theta_{l,10l^2})$ and $2\pi A_{l,10l^2}\sin(\theta_{l,10l^2})$, respectively with the aforementioned error and runtime. When $A_{l,10l^2}\cos(\theta_{l,10l^2})$ and $A_{l,10l^2}\sin(\theta_{l,10l^2})$ are estimated with an error of δ_1 and δ_2, respectively, then the quantity $|1 - (A_{l,10l^2} e^{i\theta_{l,10l^2}})/(\hat{A}_l e^{i\hat{\theta}_l})|$ is upper bounded by

$$\begin{aligned}\left|1 - \frac{A_{l,10l^2} e^{i\theta_{l,10l^2}}}{\hat{A}_l e^{i\hat{\theta}_l}}\right| &= \frac{1}{|\hat{A}_l e^{i\hat{\theta}_l}|} |A_{l,10l^2} e^{i\theta_{l,10l^2}} - \hat{A}_l e^{i\hat{\theta}_l}| \\ &= \frac{1}{|\hat{A}_l e^{i\hat{\theta}_l}|} \left|\left(A_{l,10l^2}\cos(\theta_{l,10l^2}) - \frac{\hat{v}_0}{2\pi}\right) + i\left(A_{l,10l^2}\sin(\theta_{l,10l^2}) - \frac{\hat{v}_{\pi/2}}{2\pi}\right)\right| \\ &\leq 2\sqrt{\delta_1^2 + \delta_2^2},\end{aligned} \tag{D27}$$

where the inequality comes from the fact that Subroutine D5 returns values for the estimators only if $\hat{A}_l = \sqrt{\hat{v}_0^2 + \hat{v}_{\pi/2}^2} > 1/2$. Since the root mean square errors of δ_1 and δ_2 are upper bounded by $\epsilon/(2\sqrt{2}t)$, it follows that the root mean square error of $|1 - (A_{l,10l^2} e^{i\theta_{l,10l^2}})/(\hat{A}_l e^{i\hat{\theta}_l})|$ is upper bounded by ϵ/t, as claimed in the output of Subroutine D5. Finally, note that including the rejection condition of steps 4 and 5 of Subroutine D5 only increases the average runtime of generating $\hat{v}_0, \hat{v}_{\pi/2}$ by constant factor. This is because the root mean square of $|1-(A_{l,10l^2} e^{i\theta_{l,10l^2}})/(\hat{A}_l e^{i\hat{\theta}_l})|$ is upper bounded by ϵ/t and according to Eq. (D3), we have that $A_{l,10l^2} \geq 1-\pi^2/80 > 1/2$, and thus the probability that $\sqrt{\hat{v}_0^2 + \hat{v}_{\pi/2}^2} \leq 1/2$ is smaller than $1/2$ for sufficiently small ϵ/t. □

Above, we have demonstrated the ability to accurately obtain estimates $(\hat{A}_l, \hat{\theta}_l)$ for the case of $l > 0$. We now show how these estimates can be used to provide estimates for cases $l \leq 0$.

Lemma 7. *With respect to estimates $(\hat{A}_l, \hat{\theta}_l)$ with $l > 0$ obtained by Subroutine D5, define $(\hat{A}_l, \hat{\theta}_l)$ for $l \leq 0$ as*

$$(\hat{A}_l, \hat{\theta}_l) := \begin{cases} (\hat{A}_{-l}, -\hat{\theta}_{-l}) & l < 0 \\ (1, 0) & l = 0. \end{cases} \tag{D28}$$

Thus defined, these provide estimates of $(A_{l,10l^2}, \theta_{l,10l^2})$ in Lemma 5 with a root mean squared error of $|1 - (A_{l,10l^2} e^{i\theta_{l,10l^2}})/(\hat{A}_l e^{i\hat{\theta}_l})|$ upper bounded by ϵ/t.

Proof: For $l = 0$, it is shown in the proof of Lemma 5 that $(A_{l,10l^2}, \theta_{l,10l^2}) = (1, 0)$, thus Eq. (D28) provides an exact estimate of $(A_{l,10l^2}, \theta_{l,10l^2})$. For $l < 0$, note that by Eq. (D7), we have

$$A_{l,10l^2} e^{i\theta_{l,10l^2}} := \left[\frac{\mathrm{tr}(e^{-i\frac{\pi}{20l} H_0})}{2^n} \right]^{10l^2}, \tag{D29}$$

and so $(A_{l,10l^2}, \theta_{l,10l^2}) = (A_{-l,10(-l)^2}, -\theta_{-l,10(-l)^2})$ holds. We can then write

$$\left| 1 - \frac{A_{l,10l^2} e^{i\theta_{l,10l^2}}}{\hat{A}_l e^{i\hat{\theta}_l}} \right| = \left| 1 - \frac{A_{-l,10(-l)^2} e^{-i\theta_{-l,10(-l)^2}}}{\hat{A}_{-l} e^{-i\hat{\theta}_{-l}}} \right| = \left| \left(1 - \frac{A_{-l,10(-l)^2} e^{i\theta_{-l,10(-l)^2}}}{\hat{A}_{-l} e^{i\hat{\theta}_{-l}}} \right)^* \right| = \left| 1 - \frac{A_{-l,10(-l)^2} e^{i\theta_{-l,10(-l)^2}}}{\hat{A}_{-l} e^{i\hat{\theta}_{-l}}} \right|,$$

whose root mean square is upper bounded by ϵ/t, thus asserting our claim. □

In summary, we have shown that one can estimate the parameters $(\hat{A}_l, \hat{\theta}_l)$, which are necessary to construct appropriate corrections to the "intermediate circuit" in order to build the compiled Algorithm 4, as depicted in Fig. 12.

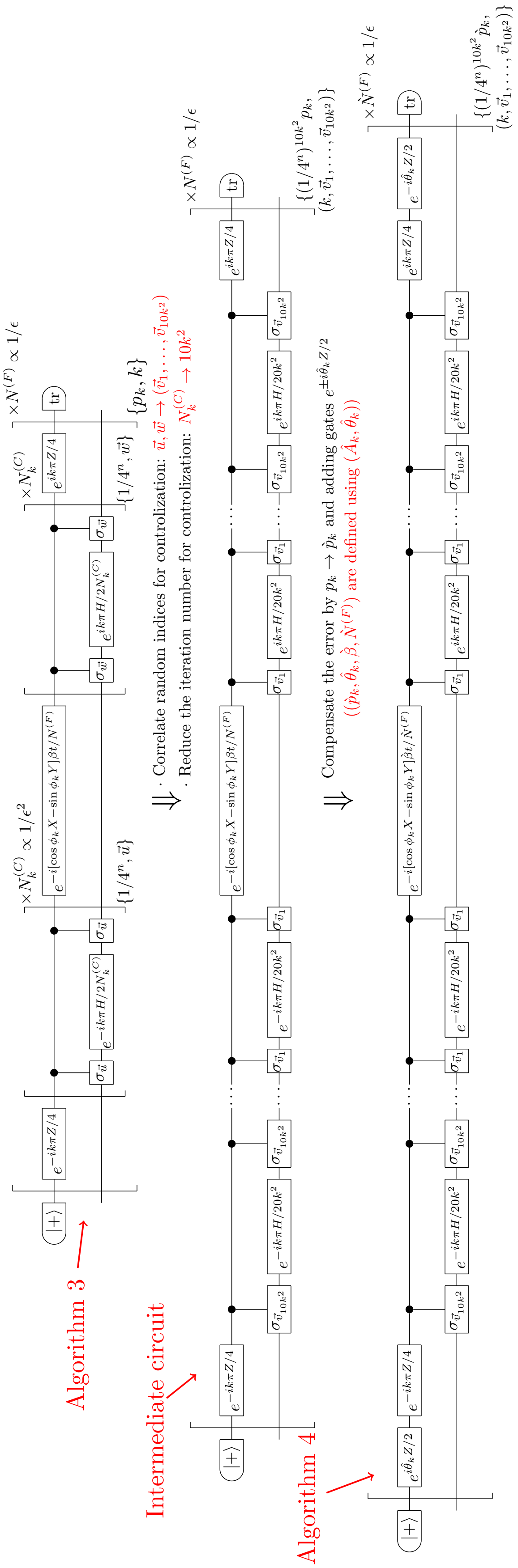

Algorithm 3
$|+\rangle$
$e^{-ik\pi Z/4}$
$\sigma_{\vec{u}}$
$e^{-ik\pi H/2N_k^{(C)}}$
$\sigma_{\vec{u}}$
$\times N_k^{(C)} \propto 1/\epsilon^2$
$\{1/4^n, \vec{u}\}$
$e^{-i[\cos\phi_k X - \sin\phi_k Y]\beta t/N^{(F)}}$
$\sigma_{\vec{w}}$
$e^{ik\pi H/2N_k^{(C)}}$
$\sigma_{\vec{w}}$
$\times N_k^{(C)}$
$\{1/4^n, \vec{w}\}$
$e^{ik\pi Z/4}$
tr
$\times N^{(F)} \propto 1/\epsilon$
$\{p_k, k\}$
$\Downarrow$ · Correlate random indices for controlization: $\vec{u}, \vec{w} \to (\vec{v}_1, \ldots, \vec{v}_{10k^2})$
· Reduce the iteration number for controlization: $N_k^{(C)} \to 10k^2$
Intermediate circuit
$|+\rangle$
$e^{-ik\pi Z/4}$
$\sigma_{\vec{v}_{10k^2}}$
$e^{-ik\pi H/20k^2}$
$\sigma_{\vec{v}_{10k^2}}$
$\sigma_{\vec{v}_1}$
$e^{-ik\pi H/20k^2}$
$\sigma_{\vec{v}_1}$
$e^{-i[\cos\phi_k X - \sin\phi_k Y]\beta t/N^{(F)}}$
$\sigma_{\vec{v}_1}$
$e^{ik\pi H/20k^2}$
$\sigma_{\vec{v}_1}$
$\sigma_{\vec{v}_{10k^2}}$
$e^{ik\pi H/20k^2}$
$\sigma_{\vec{v}_{10k^2}}$
$e^{ik\pi Z/4}$
tr
$\times N^{(F)} \propto 1/\epsilon$
$\{(1/4^n)^{10k^2} p_k, (k, \vec{v}_1, \ldots, \vec{v}_{10k^2})\}$
$\Downarrow$ Compensate the error by $p_k \to \grave{p}_k$ and adding gates $e^{\pm i\hat{\theta}_k Z/2}$
$((\grave{p}_k, \hat{\theta}_k, \grave{\beta}, \grave{N}^{(F)})$ are defined using $(\hat{A}_k, \hat{\theta}_k))$
Algorithm 4
$|+\rangle$
$e^{i\hat{\theta}_k Z/2}$
$e^{-ik\pi Z/4}$
$\sigma_{\vec{v}_{10k^2}}$
$e^{-ik\pi H/20k^2}$
$\sigma_{\vec{v}_{10k^2}}$
$\sigma_{\vec{v}_1}$
$e^{-ik\pi H/20k^2}$
$\sigma_{\vec{v}_1}$
$e^{-i[\cos\phi_k X - \sin\phi_k Y]\grave{\beta} t/\grave{N}^{(F)}}$
$\sigma_{\vec{v}_1}$
$e^{ik\pi H/20k^2}$
$\sigma_{\vec{v}_1}$
$\sigma_{\vec{v}_{10k^2}}$
$e^{ik\pi H/20k^2}$
$\sigma_{\vec{v}_{10k^2}}$
$e^{ik\pi Z/4}$
$e^{-i\hat{\theta}_k Z/2}$
tr
$\times \grave{N}^{(F)} \propto 1/\epsilon$
$\{(1/4^n)^{10k^2} \grave{p}_k, (k, \vec{v}_1, \ldots, \vec{v}_{10k^2})\}$

FIG. 12. *Summary of compilation of Algorithm 3.*—First, the random variables $\vec{u},\ \vec{w} \in \{0,1,2,3\}^n$ of the controlization become correlated. Since the error of this intermediate circuit for finite iteration number $10k^2$ of the controlization happens as in Eq. (18), this error is compensated in the final step by modifying $N^{(F)} \to \grave{N}^{(F)}$, and $p_k \to \grave{p}_k$, and introducing an additional gate $e^{\pm i\hat{\theta}_k Z/2}$ according to the parameters $(\hat{A}_k, \hat{\theta}_k)$ obtained by Subroutine D5 in Appendix D 2.

3. Error and Runtime Analysis of Compiled Algorithm (Algorithm 4)

Here, we will prove a theorem on the error, the mean square of the error, and the runtime of Algorithm 4.

Theorem 3.

1. *Algorithm 4 outputs $e^{-if(H_0)t}\left|\psi\right\rangle$ with an error (in terms of the 1-norm) upper bounded by ϵ, i.e.*

$$\sup_{\substack{\dim(\mathcal{H}') \\ |\psi\rangle\in\mathcal{H}\otimes\mathcal{H}' \\ \||\psi\rangle\|=1}} \|\mathcal{F}\otimes\mathcal{I}_{\mathcal{H}'}(\left|\psi\right\rangle\left\langle\psi\right|) - \sum_j p_j(\mathcal{F}_j\otimes\mathcal{I}_{\mathcal{H}'})(\left|\psi\right\rangle\left\langle\psi\right|)\|_1 \leq \epsilon, \tag{D30}$$

where $j \in [\bigcup_{k=-\grave{K}}^{\grave{K}}(\{k\}\times(\{0,1,2,3\}^n)^{10k^2})]^{\grave{N}^{(F)}} \times (\mathbb{R}\times[0,2\pi))^{\grave{K}}$ is chosen from the set of all random indices, namely (k, j_k) where $k \in \{-\grave{K},\dots,\grave{K}\}$ and $j_k = (\vec{v}_{1:10k^2}) \in (\{0,1,2,3\}^n)^{10k^2}$ for each of the $\grave{N}^{(F)}$ iterations and parameters $\Phi := [(\hat{A}_1,\hat{\theta}_1),\dots,(\hat{A}_{\grave{K}},\hat{\theta}_{\grave{K}})]$ where $\hat{A}_l \in \mathbb{R}$ and $\hat{\theta}_l \in [0,2\pi)$ (which are random due to dependence on measurement outcomes) with $l \in \{1,\dots,\grave{K}\}$, p_j is the probability that j is chosen, $\mathcal{F}(\rho) := e^{-if(H_0)t}\rho e^{if(H_0)t}$, $\mathcal{F}_j$ is the unitary performed when j is chosen, and $\mathcal{H}'$ is an auxiliary system of arbitrary dimension.

2. *Algorithm 4 outputs $e^{-if(H_0)t}\left|\psi\right\rangle$ with mean squared error upper-bounded by 2ϵ, i.e.*

$$\sup_{\substack{\dim(\mathcal{H}') \\ |\psi\rangle\in\mathcal{H}\otimes\mathcal{H}' \\ \||\psi\rangle\|=1}} \sum_j p_j\|(\mathcal{F}\otimes\mathcal{I}_{\mathcal{H}'})(\left|\psi\right\rangle\left\langle\psi\right|) - (\mathcal{F}_j\otimes\mathcal{I}_{\mathcal{H}'})(\left|\psi\right\rangle\left\langle\psi\right|)\|_1^2 \leq 2\epsilon. \tag{D31}$$

3. *The runtime of Algorithm 4 comprises of a pre-processing step and a main process. The pre-processing step has a runtime of $\Theta(\grave{K}^3t^3n/\epsilon^3) + T_4$, where T_4 is the sum of computation times (on a classical computer) for step 1 (calculation of Fourier coefficients $\tilde{c}_k$ until Eq. (19) is satisfied) and step 7 (computation of $\grave{N}^{(F)}$), and $\grave{K} = O[(t/\epsilon)^{1/3}]$. The main process has a runtime of $\Theta(C_{4,f}t^2n/\epsilon)$, where $C_{4,f}$ is a (function-dependant) constant.*

Part 1 of Theorem 3 implies in particular that for an arbitrary input state $\left|\psi\right\rangle \in \mathcal{H}$:

$$\|\mathcal{F}(\left|\psi\right\rangle\left\langle\psi\right|) - \sum_j p_j\mathcal{F}_j(\left|\psi\right\rangle\left\langle\psi\right|)\|_1 \leq \epsilon. \tag{D32}$$

In addition, part 2 of Theorem 3 implies in particular that the mean square of the difference between the ideal state and the single-shot output state is bounded above as:

$$\sum_j p_j\|\mathcal{F}(\left|\psi\right\rangle\left\langle\psi\right|) - \mathcal{F}_j(\left|\psi\right\rangle\left\langle\psi\right|)\|_1^2 \leq 2\epsilon. \tag{D33}$$

Therefore, we prove the error bounds stated in Algorithm 4.
Proof: We begin with the first statement.

1. Steps 2 to 13 of Algorithm 4 simulate the following quantum operation

$$\mathcal{F}_6 := \sum_\Phi p_\Phi\mathcal{F}_\Phi \tag{D34}$$

applied to the input state $\left|+\right\rangle\otimes\left|\psi\right\rangle$, where $\Phi := ((\hat{A}_1,\hat{\theta}_1),\dots,(\hat{A}_{\grave{K}},\hat{\theta}_{\grave{K}}))$, p_Φ is the probability Φ is obtained, and $\mathcal{F}_\Phi$ is the quantum operation simulated by qDRIFT in steps 9 to 13 of Algorithm 4 whenever Φ is chosen. The operation

$\mathcal{F}_\Phi$ approximates the dynamics $e^{-iH_\Phi t}$ for the Hamiltonian

$$\begin{aligned}
H_\Phi &:= \sum_{k=-\grave{K}}^{\grave{K}} \frac{|\tilde{c}_k|}{\hat{A}_k} \left(\frac{1}{4}\right)^{10nk^2} \sum_{j_k \in (\{0,1,2,3\}^n)^{10k^2}} \grave{W}_{k,j_k}^\dagger ([\cos(\phi_k)X - \sin(\phi_k)Y] \otimes I)\grave{W}_{k,j_k} \\
&= \sum_{k=-\grave{K}}^{\grave{K}} \frac{|\tilde{c}_k|}{\hat{A}_k} (e^{-i\frac{\hat{\theta}_k}{2}Z} \otimes I) \left[\left(\frac{1}{4}\right)^{10nk^2} \sum_{j_k \in (\{0,1,2,3\}^n)^{10k^2}} (W_{k,j_k}^{(10k^2)})^\dagger ([\cos(\phi_k)X - \sin(\phi_k)Y] \otimes I) W_{k,j_k}^{(10k^2)} \right] (e^{i\frac{\hat{\theta}_k}{2}Z} \otimes I) \\
&= \sum_{k=-\grave{K}}^{\grave{K}} \frac{|\tilde{c}_k|}{\hat{A}_k} A_{k,10k^2} (e^{i\frac{\theta_{k,10k^2} - \hat{\theta}_k}{2}Z} \otimes I) \begin{pmatrix} 0 & e^{i\phi_k} e^{i\frac{k\pi}{2}(H_0+I)} \\ e^{-i\phi_k} e^{-i\frac{k\pi}{2}(H_0+I)} & 0 \end{pmatrix} (e^{-i\frac{\theta_{k,10k^2} - \hat{\theta}_k}{2}Z} \otimes I). \qquad \text{(D35)}
\end{aligned}$$

By defining $\Delta_k := \frac{A_{k,10k^2} e^{i\theta_{k,10k^2}}}{\hat{A}_k e^{i\hat{\theta}_k}} - 1$, the last line of Eq. (D35) can be rewritten as

$$\sum_{k=-\grave{K}}^{\grave{K}} |\tilde{c}_k| \begin{pmatrix} 0 & (1+\Delta_k) e^{i\phi_k} e^{i\frac{k\pi}{2}(H_0+I)} \\ (1+\Delta_k^*) e^{-i\phi_k} e^{-i\frac{k\pi}{2}(H_0+I)} & 0 \end{pmatrix}. \qquad \text{(D36)}$$

This expression can be further simplified by defining a function $f_\Phi : [-1,1] \to \mathbb{R}$ as

$$f_\Phi(x) := \sum_{k=-\grave{K}}^{\grave{K}} \tilde{c}_k \Delta_k e^{ik\pi x}. \qquad \text{(D37)}$$

Note that the output is real since $\Delta_{-k} = \Delta_k^*$. With this, we have

$$H_\Phi = X \otimes \left[f_{\grave{K}}\left(\frac{H_0+I}{2}\right) + f_\Phi\left(\frac{H_0+I}{2}\right) \right] \qquad \text{(D38)}$$

where $f_{\grave{K}}(x) := \sum_{k=-\grave{K}}^{\grave{K}} \tilde{c}_k e^{ik\pi x}$.

Now, for any quantum operation $\mathcal{F} : \mathcal{L}(\mathcal{H}_c \otimes \mathcal{H}) \to \mathcal{L}(\mathcal{H}_c \otimes \mathcal{H})$ we define the following norm

$$E(\mathcal{F}) := \sup_{\substack{|\psi\rangle \in \mathcal{H}_c \otimes \mathcal{H} \otimes \mathcal{H}' \\ \||\psi\rangle\| = 1 \\ \dim \mathcal{H}'}} \|\mathcal{F} \otimes \mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|)\|_1, \qquad \text{(D39)}$$

where $\mathcal{I}_{\mathcal{H}'}$ is the identity operation in $\mathcal{L}(\mathcal{H}')$. Moreover, we define

$$\begin{aligned}
\mathcal{F}_4(\rho) &:= e^{-i(X \otimes f(H_0))t} \rho e^{i(X \otimes f(H_0))t} \\
\mathcal{F}_5(\rho) &:= \sum_\Phi p_\Phi e^{-iH_\Phi t} \rho e^{iH_\Phi t} =: \sum_\Phi p_\Phi \mathcal{G}_\Phi. \qquad \text{(D40)}
\end{aligned}$$

With these definitions at hand, we can upper bound the simulation error $E(\mathcal{F}_6 - \mathcal{F}_4)$ using Lemma 2 and similar arguments to those presented in Eqs. (C10) and (C11) as follows.

$$\begin{aligned}
&E(\mathcal{F}_6 - \mathcal{F}_4) \le E(\mathcal{F}_6 - \mathcal{F}_5) + E(\mathcal{F}_5 - \mathcal{F}_4) \\
&= \sup_{\substack{|\psi\rangle \in \mathcal{H}_c \otimes \mathcal{H} \otimes \mathcal{H}' \\ \||\psi\rangle\| = 1 \\ \dim \mathcal{H}'}} \left\| \sum_\Phi p_\Phi [\mathcal{F}_\Phi \otimes \mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|) - \mathcal{G}_\Phi \otimes \mathcal{I}_{\mathcal{H}'}(|\psi\rangle\langle\psi|)] \right\|_1 \\
&+ \sup_{\substack{|\psi\rangle \in \mathcal{H}_c \otimes \mathcal{H} \otimes \mathcal{H}' \\ \||\psi\rangle\| = 1 \\ \dim \mathcal{H}'}} \left\| \sum_\Phi p_\Phi [(e^{-iH_\Phi t} \otimes I)|\psi\rangle\langle\psi|(e^{iH_\Phi t} \otimes I) - (e^{-i(X \otimes f(H_0))t} \otimes I)|\psi\rangle\langle\psi|(e^{i(X \otimes f(H_0))t} \otimes I)] \right\|_1 \\
&\le \sum_\Phi p_\Phi \|\mathcal{F}_\Phi - \mathcal{G}_\Phi\|_\diamond + \sum_\Phi p_\Phi \sup_{\substack{|\psi\rangle \in \mathcal{H}_c \otimes \mathcal{H} \otimes \mathcal{H}' \\ \||\psi\rangle\| = 1 \\ \dim \mathcal{H}'}} 2[1 - |\langle\psi|(e^{-i\{X \otimes (\tilde{f}((H_0+I)/2) - f_{\grave{K}}((H_0+I)/2) - f_\Phi((H_0+I)/2))\}t} \otimes I)|\psi\rangle|^2]^{1/2} \\
&\le \frac{\epsilon}{3} + \sum_\Phi p_\Phi 2\sin(R(\tilde{f} - f_{\grave{K}} - f_\Phi)t/2) \le \frac{\epsilon}{3} + \sum_\Phi p_\Phi [R(\tilde{f} - f_{\grave{K}}) + R(f_\Phi)]t \le \frac{\epsilon}{3} + \frac{\epsilon}{3} + \sum_\Phi p_\Phi R(f_\Phi)t, \qquad \text{(D41)}
\end{aligned}$$

where for any function $g : [-1,1] \to \mathbb{R}$, we have $R(g) := 2 \max_{x\in[-1,1]} |g(x)|$.

The final term $\sum_{\Phi} p_{\Phi} R(f_{\Phi})t$ in the above equation can be upper bounded as follows. First note that, for f_{Φ} defined in Eq. (D37), we have

$$R(f_{\Phi}) \leq 2 \sum_{k=-\grave{K}}^{\grave{K}} |\tilde{c}_k||\Delta_k|. \tag{D42}$$

Now, because the allowed error of Subroutine D5 in step 3 of Algorithm 4 is set as $\sqrt{3}\epsilon/(12\pi(\sum_{k=-\infty}^{\infty}|\tilde{c}_k||k|))$, one can employ the Chebyshev inequality to show that the probability that $|\Delta_k| \leq |k|(\sqrt{3}\epsilon C)/(12\pi(\sum_{k=-\infty}^{\infty}|\tilde{c}_k||k|)t)$ for all k and a fixed positive value $C > 0$ is lower bounded by

$$\prod_{\substack{k\in\{-\grave{K},\dots,-1\\ 1,\dots,\grave{K}\}}} \left(1 - \frac{1}{k^2C^2}\right) \geq 1 - 2\left(\sum_{k=1}^{\infty}\frac{1}{k^2C^2}\right) = 1 - \frac{\pi^2}{3C^2}, \tag{D43}$$

where the final equality follows from the identity $\sum_{k=1}^{\infty}(1/k^2) = \pi^2/6$.

Independently, assuming that $|\Delta_k| \leq |k|(\sqrt{3}\epsilon C)/(12\pi(\sum_{k=-\infty}^{\infty}|\tilde{c}_k||k|)t)$ holds for all k, it follows that the r.h.s. of Eq. (D42) is upper bounded by $(\sqrt{3}C\epsilon)/(6\pi t)$.

Combining these two results, we have that

$$\Pr\left[R(f_{\Phi}) \geq \frac{\sqrt{3}C\epsilon}{6\pi t}\right] \leq \frac{\pi^2}{3C^2}. \tag{D44}$$

Making use of the identity

$$\sum_{\Phi} p_{\Phi} R(f_{\Phi}) = \int_0^{\infty} \mathrm{d}x \Pr[R(f_{\Phi}) \geq x], \tag{D45}$$

we can upper bound $\sum_{\Phi} p_{\Phi} R(f_{\Phi})$ via

$$\sum_{\Phi} p_{\Phi} R(f_{\Phi}) \leq \int_0^{\infty} \mathrm{d}x \min\left(1, \frac{\epsilon^2}{6^2t^2x^2}\right) = \frac{\epsilon}{6t} + \frac{\epsilon}{6t} = \frac{\epsilon}{3t}. \tag{D46}$$

Finally substituting Eq. (D46) into Eq. (D41), we obtain

$$E(\mathcal{F}_6 - \mathcal{F}_4) \leq \epsilon. \tag{D47}$$

Therefore,

$$\begin{aligned} \epsilon \geq E(\mathcal{F}_6 - \mathcal{F}_4) &\geq \sup_{\substack{|\psi\rangle\in\mathcal{H}_c\otimes\mathcal{H}\otimes\mathcal{H}'\\ \||\psi\rangle\|=1\\ \dim\mathcal{H}'}} \|\mathrm{tr}_{\mathcal{H}_c}[(\mathcal{F}_6 - \mathcal{F}_4)\otimes\mathcal{I}_{\mathcal{H}'}](|\psi\rangle\langle\psi|)\|_1 \\ &\geq \sup_{\substack{|\psi\rangle\in\mathcal{H}\otimes\mathcal{H}'\\ \||\psi\rangle\|=1\\ \dim\mathcal{H}'}} \|\mathrm{tr}_{\mathcal{H}_c}[(\mathcal{F}_6 - \mathcal{F}_4)\otimes\mathcal{I}_{\mathcal{H}'}](|+\rangle\langle+|\otimes|\psi\rangle\langle\psi|)\|_1 \\ &, \end{aligned} \tag{D48}$$

which is equal to the expression in D30, as required.

2. This statement follows directly by combining the above result with Lemma 1.

3. In order to prove this statement regarding the runtime, we use Lemma 3, which shows that $\tilde{f}$ defined for any input function f of Algorithm 4 is 4-smooth.

Based on Lemmas 3 and 4, we have that $\sum_{k=K}^{\infty}|\tilde{c}_k| = O(1/K^3)$ and $\sum_{k=-\infty}^{-K}|\tilde{c}_k| = O(1/K^3)$ for $K > 1$. Thus, the truncation number $\grave{K}$ defined in Eq. (19) is shown to be $O((t/\epsilon)^{1/3})$, as claimed. Furthermore, the runtime can be split into one for pre-processing stage and one for the main process as follows.

Pre-processing: For each $k \in \{-\grave{K}, \ldots, \grave{K}\}$, the runtime of generating $(\hat{A}_k, \hat{\theta}_k)$ is shown in Lemma 2 to be $\Theta(k^2 t^3 n/\epsilon^3)$. Thus, by summing this over all k, the total runtime of steps 2 to 6 in Algorithm 4 is $\Theta(\grave{K}^3 t^3 n/\epsilon^3)$. Thus the total runtime of the pre-processing step is $\Theta(\grave{K}^3 t^3 n/\epsilon^3) + T_4$.

Main process: The average runtime of the main process in Algorithm 4 is evaluated as (number of iterations $\grave{N}^{(F)}$)×(average runtime of steps 10 to 12)= $N(\grave{\beta}, t, \epsilon/3) \times \Theta(\sum_{k=-\infty}^{\infty} \grave{p}_k n k^2) = \Theta((\sum_{k=-\infty}^{\infty} |\tilde{c}_k|)(\sum_{k=-\infty}^{\infty} |\tilde{c}_k| k^2) t^2 n/\epsilon) =: \Theta(C_{4,f} t^2 n/\epsilon)$ (note that $|\hat{A}_k| \geq 1/2$ due to steps 4 to 5 of Subroutine D5). □

Appendix E: QSVT-based UHET Algorithm

We will now present and analyze an alternative procedure to achieve UHET based upon a QSVT procedure. We will finally compare this method to Algorithms 3 and 4.

1. QSVT-based UHET Algorithm (Algorithm 7)

Formally, the QSVT-based algorithm is as follows.

Algorithm 7 QSVT-based algorithm

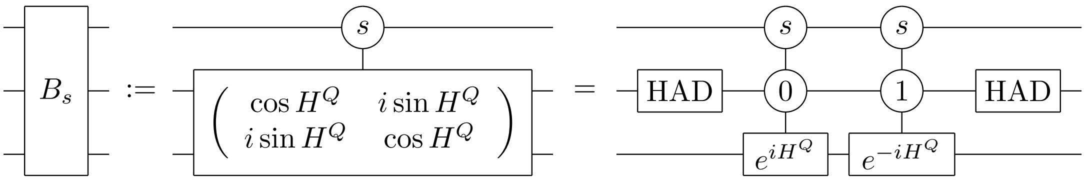
```
Input:
    • A finite number of queries to a black-box Hamiltonian dynamics $e^{\pm iH\tau}$ of a seed Hamiltonian $H$ normalized as $\|H_0\|_{\mathrm{op}} = 1$ where $H_0$ is the traceless part of $H$, i.e., $H_0 := H - (1/2^n)\mathrm{tr}(H)I$, with $\tau > 0$
    • A class $\mathrm{C}^3$ function $f : [-1,1] \to \mathbb{R}$ such that $f^{(4)}$ is piecewise $C^2$ (see Appendix A 1)
    • Input state $|\psi\rangle \in \mathcal{H}$
    • Allowed error $\epsilon > 0$
    • Time $t > 0$
Output: A state approximating $e^{-if(H_0)t}|\psi\rangle$ $(t > 0)$ with an error in terms Eq. (2) upper bounded by $\epsilon$

Runtime:
  Pre-processing (only once): $O((K^Q_{f,t,\epsilon})^3)$
  Main Process: $\Theta((K^Q_{f,t,\epsilon})^2/\epsilon)$ for $K^Q_{f,t,\epsilon}$ depending on $f$, $t$, and $\epsilon$
Used Resources:
  System: $\mathcal{H}$ and two auxiliary qubit $\mathcal{H}_b$, $\mathcal{H}_c$
  Gates: $e^{\pm iH\tau}$ $(\tau > 0)$ and controlled-Pauli gates on $\mathcal{L}(\mathcal{H}_b \otimes \mathcal{H}_c \otimes \mathcal{H})$

Procedure:
Pre-processing:
Define functions $f_0, f_1 : [-1,1] \mapsto \mathbb{R}$ as per Eq. (E2)
Compute $c_k^{(s)} := \int_{-1}^{1} \mathrm{d}x\, 2\sin(\frac{\pi}{10}) f_s[\cos(\pi x)]\cos(k\pi x)$ and $K^Q_s$ $(s \in \{0,1\})$ that satisfies

    $$\left| 2\sin(\tfrac{\pi}{10}) f_s[\cos(\pi x)] - \sum_{k=0}^{K^Q_s} c_k^{(s)} \cos(\pi k x) \right| < \Theta(\epsilon) \tag{E1}$$

    for all $x \in [0,1]$. Note that one can recast the above bound in terms of Chebyshev polynomials $T_k$ as $\left|2\sin(\pi/10) f_s(x') - \sum_{k=0}^{K^Q_s} c_k^{(s)} T_k(x')\right| < \Theta(\epsilon)$ for $x =: (1/\pi)\cos^{-1}(x')$ $(x' \in [-1,1])$, as we will employ in the coming steps.
Set $K^Q_{f,t,\epsilon} \leftarrow K^Q_0 + K^Q_1$
For $s \in \{0,1\}$, find gate sequence of QSVT for a $(1,2,0)$-block-encoding unitary $U_s$ of $\sum_{k=0}^{K^Q_s} c_k^{(s)} T_k[\cos(H^Q)]$ on $|s0\rangle\langle s0| \in \mathcal{L}(\mathcal{H}_b \otimes \mathcal{H}_c)$ using $B_s$ defined Eq. (E4) (see Fig. 13)
Main Process:
Initialize $U_{\mathrm{current}} \leftarrow I$
for $s \in \{0,1\}$ do
    Construct unitary $U'_s$ by replacing $B_s$ in the gate sequence for $U_s$ by the random unitary $B'_s$ which approximate $B_s$ up to an error of $\Theta(\epsilon/K^Q_{f,t,\epsilon})$ (see Fig. 14)
    $U_{\mathrm{current}} \leftarrow (S^{-s} \otimes I \otimes I)U'_s U_{\mathrm{current}}$        ▷ $S := |0\rangle\langle 0| + i|1\rangle\langle 1|$ is the phase gate
end for
Set $U_{\mathrm{current}} \leftarrow (\mathrm{HAD} \otimes I \otimes I)U_{\mathrm{current}}(\mathrm{HAD} \otimes I \otimes I)$
Perform robust oblivious amplitude amplification [27] with $n = 5$ on the block of $U_{\mathrm{current}}$ specified by $|00\rangle\langle 00| \in \mathcal{L}(\mathcal{H}_b \otimes \mathcal{H}_c)$
Return $U_{\mathrm{current}}(|0\rangle^{\otimes 2} \otimes |\psi\rangle)$
```



B_s
$:=$
s
$\begin{pmatrix} \cos H^Q & i\sin H^Q \\ i\sin H^Q & \cos H^Q \end{pmatrix}$
$=$
HAD
s
0
e^{iH^Q}
s
1
e^{-iH^Q}
HAD

FIG. 13. *Circuit representation of B_s.*—The circled number in the controlled qubits means that the unitary on the target qubit is applied if the control qubit is that state and otherwise identity is applied.

Begin by defining functions f_0 and f_1 as

$$
\begin{aligned}
f_0(x) &:= \begin{cases} f_{0,-}(x) & (0 \le x \le \frac{1}{2}) \\ \cos\left[f\left(\frac{12\cos^{-1}(x)}{\pi} - 3\right)t\right] & (\frac{1}{2} \le x \le \frac{\sqrt{3}}{2}) \\ f_{0,+}(x) & (\frac{\sqrt{3}}{2} \le x \le 1) \\ f_0(-x) & (-1 \le x \le 0) \end{cases} \\
f_1(x) &:= \begin{cases} f_{1,-}(x) & (0 \le x \le \frac{1}{2}) \\ \sin\left[f\left(\frac{12\cos^{-1}(x)}{\pi} - 3\right)t\right] & (\frac{1}{2} \le x \le \frac{\sqrt{3}}{2}) \\ f_{1,+}(x) & (\frac{\sqrt{3}}{2} \le x \le 1) \\ f_1(-x) & (-1 \le x \le 0) \end{cases},
\end{aligned}
\tag{E2}
$$

where $f_{0,-}, f_{0,+}, f_{1,-}, f_{1,+}$ are any functions that are infinitely differentiable on their domains and lead to f_0 and f_1 such that

1. $2\sin(\frac{\pi}{10})f_0(x)$ and $2\sin(\frac{\pi}{10})f_1(x)$ is bounded in $[-1,1]$ for all $x \in [-1,1]$. This ensures that they can be constructed using QSVT (in particular, using the technique presented in Theorem 10 of Ref. [28]).

2. $f_s^{(n)}(\frac{1}{2}^+) = f_s^{(n)}(\frac{1}{2}^-)$, $f_s^{(n)}(\frac{\sqrt{3}}{2}^+) = f_s^{(n)}(\frac{\sqrt{3}}{2}^-)$, and $f_s^{(n)}(0^+) = f_s^{(n)}(1^-) = 0$ for $s \in \{0,1\}$ and $n \in \{0,1,\dots,J-1\}$ for any integer $J \ge 4$.

The Hamiltonian H^Q in step 4 is defined as

$$
H^Q := \frac{\pi(H_0 + 3I)}{12}. \tag{E3}
$$

The functions f_0 and f_1 are defined in such a way that

$$
f_0(\cos H^Q) - i f_1(\cos H^Q) = e^{-if(H_0)t}.
$$

Furthermore, as we discuss in more detail below, f_0 and f_1 are implementable via QSVT. Since the runtime of QSVT scales polynomially on the cutoff number K_s^Q, the functions $f_0(x)$ and $f_1(x)$ are chosen to be class C^3 in $x \in [-1,1]$ so that the sum in Eq. (E1) converges rapidly to $2\sin(\frac{\pi}{10})f_s(\cos(\pi x))$ and K_s^Q scales slowly with ϵ.

With respect to the above Hamiltonian, the unitary B_s in step 4 is defined as

$$
B_s := |s\rangle\langle s| \otimes \begin{pmatrix} \cos(H^Q) & i\sin(H^Q) \\ i\sin(H^Q) & \cos(H^Q) \end{pmatrix} + |\bar{s}\rangle\langle\bar{s}| \otimes I \otimes I. \tag{E4}
$$

See Fig. 13 for its circuit representation.

By defining the following function for $s_b, s_c \in \{0,1\}$ and unitary U

$$
\texttt{ctrl}_2[U,(s_b,s_c)] := [I - (|s_b s_c\rangle\langle s_b s_c|)] \otimes I + |s_b s_c\rangle\langle s_b s_c| \otimes U, \tag{E5}
$$

the unitary B_s can be expressed as

$$
B_s = (I \otimes \mathrm{HAD} \otimes I)\texttt{ctrl}_2[e^{-iH^Q},(s,1)]\texttt{ctrl}_2[e^{iH^Q},(s,0)](I \otimes \mathrm{HAD} \otimes I). \tag{E6}
$$

The operator $\texttt{ctrl}_2[e^{\pm iH^Q t},(s_b,s_c)]$ can be constructed from $e^{\pm iHt}$ via double controlization [16], which makes use of qDRIFT. In this way, the unitary B_s can be approximated by the circuit shown in Fig. 14.

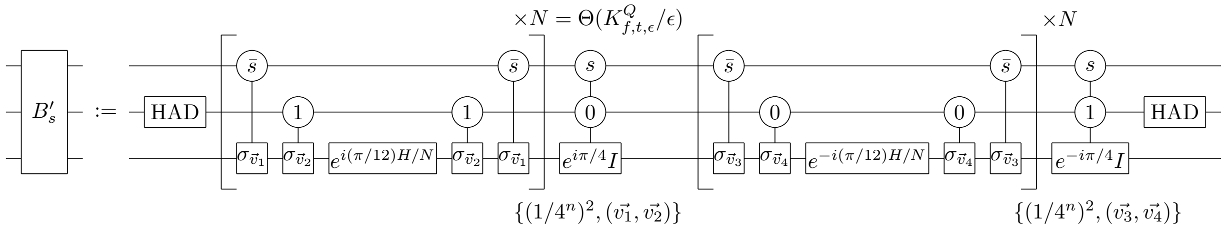


FIG. 14. *Definition of* B'_s.—$\texttt{ctrl}_2(e^{\pm iH^Q t}, (s_b, s_c))$ in Fig. 13 is replaced by its approximation by controlization.

In summary, the Algorithm 7 simulates the dynamics $e^{-if(H_0)t}$ by the following steps.

$$I \otimes I \otimes e^{\pm iHt} \xrightarrow{\text{double controlization}} \texttt{ctrl}_2[e^{\pm iH^Q t}, (s_b, s_c)] \xrightarrow{\text{Eq. (E4)}}$$

$$B_s = |s\rangle\langle s| \otimes \begin{pmatrix} \cos H^Q & i\sin H^Q \\ i\sin H^Q & \cos H^Q \end{pmatrix} + |\bar{s}\rangle\langle\bar{s}| \otimes I \otimes I \xrightarrow{\text{QSVT \& post-application of } (S^{-s\otimes I\otimes I})}$$

$$\left(\begin{array}{c|c} \begin{matrix} 2\sin(\frac{\pi}{10})f_0(\cos H^Q) & \cdot \\ \cdot & \cdot \end{matrix} & 0 \\ \hline 0 & \begin{matrix} -2i\sin(\frac{\pi}{10})f_1(\cos H^Q) & \cdot \\ \cdot & \cdot \end{matrix} \end{array}\right) \xrightarrow{\text{pre- and post-application of HAD}\otimes I\otimes I}$$

$$\begin{pmatrix} \sin(\frac{\pi}{10})e^{-if(H_0)t} & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{pmatrix} \xrightarrow[\text{amplitude amplification}]{\text{robust oblivious}} \begin{pmatrix} e^{-if(H_0)t} & 0 & 0 & 0 \\ 0 & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot \end{pmatrix}. \tag{E7}$$

The leftmost column and the top row of the final matrix in Eq. (E7) is filled with 0 except for the top-left block because both $e^{if(H_0)t}$ and the entire matrix are unitary.

The procedure of Algorithm 7 ensures that the approximation error of simulating $e^{-if(H_0)t}$ is $O(\epsilon)$. There are two sources of errors in the overall procedure: the error due to the approximation in Eq. (E1) and that due to the approximation of B_s by the circuit given in Fig. 14. The former error is $O(\epsilon)$ by definition and the latter is upper bounded by (approximation error of B'_s) × (number of queries to B'_s) $= \Theta(\epsilon/K^Q_{f,t,\epsilon}) \times K^Q_{f,t,\epsilon} = \Theta(\epsilon)$. Thus, the sum of these errors is $O(\epsilon)$.

The runtime of the main process is asymptotically proportional to (runtime of approximating B'_s) × (number of queries to B'_s) $= \Theta(K^Q_{f,t,\epsilon}/\epsilon) \times K^Q_{f,t,\epsilon} = \Theta((K^Q_{f,t,\epsilon})^2/\epsilon)$ (indeed, the total runtime of gates other than B'_s grows slower than $(K^Q_{f,t,\epsilon})^2/\epsilon$ and can thus be ignored). Moreover, the runtime for the pre-processing step of Algorithm 7 is $O((K^Q_{f,t,\epsilon})^3)$, which can be evaluated by noting that obtaining the gate sequence for QSVT implementing a function of polynomial of degree d takes $O(d^3)$ runtime [35].

2. Algorithm Comparison

We now compare the runtimes of the three algorithms: Algorithm 3 (uncompiled), Algorithm 4 (compiled), and Algorithm 7 (QSVT-based). The runtimes of the pre-processing and the main processes are summarized in Table II.

Pre-processing: Rigorous comparison of pre-processing runtimes is difficult in general due to difficulty in analyzing the time T_3 and T_4. Nevertheless, the runtime for Algorithm 4 increases more rapidly in terms of ϵ and n than that of Algorithm 7 because $K^Q_{f,t,\epsilon}$ does not depend on n and its ϵ dependence is given by $K^Q_{f,t,\epsilon} = O(1/\epsilon^{1/3})$ (this can be seen by considering that $f_s(x)$ defined in Eq. (E2) is 4-smooth and applying the result of Lemma 4, and subsequently noting that $K^Q_{f,t,\epsilon} = K^Q_0 + K^Q_1$).

	Pre-processing Step	Main Process
Algorithm 3	T_3	$\Theta(C_{3,f}t^4n/\epsilon^3)$
Algorithm 4	$\Theta(\grave{K}^3t^3n/\epsilon^3)+T_4$	$\Theta(C_{4,f}t^2n/\epsilon)$
Algorithm 7	$O((K^Q_{f,t,\epsilon})^3)$	$\Theta((K^Q_{f,t,\epsilon})^2/\epsilon)$

TAB. II. *Comparison of runtimes of Algorithm 3 (uncompiled), Algorithm 4 (compiled), and Algorithm 7 (QSVT-based).*—The times T_3 and T_4 refer to the total classical computation time for computing the Fourier coefficients and the values $N^{(F)}$ and $\grave{N}^{(F)}$ in Algorithms 3 and 4 respectively. $C_{3,f}$ and $C_{4,f}$ refer to function-dependent constants in Algorithms 3 and 4 respectively. The value $K^Q_{f,t,\epsilon}$ depends on the function f, the time t, and the allowed error ϵ.

Main Process: The main process scaling coefficients for Algorithms 3 and 4

$$C_{3,f} := \left(\sum_{k=-\infty}^{\infty}|\tilde{c}_k|\right)^3\left(\sum_{k=-\infty}^{\infty}|\tilde{c}_k|k^2\right) \quad \text{and} \quad C_{4,f} := \left(\sum_{k=-\infty}^{\infty}|\tilde{c}_k|\right)\left(\sum_{k=-\infty}^{\infty}|\tilde{c}_k|k^2\right) \tag{E8}$$

depend only on the function f and not on n, t, and ϵ. In terms of t and ϵ, the runtime of Algorithm 4 scales slower than that of Algorithm 3. Furthermore, both algorithms are linear in terms of n. Therefore, it follows that compilation reduces the runtime of the main process. For completeness, the explicit value of $C_{4,f}$ is calculated in the proof of Theorem 3. In contrast, the scaling coefficient for the main process of Algorithm 7, $K^Q_{f,t,\epsilon}$, depends on f, t, and ϵ, and its explicit expression is difficult to obtain in general. Nonetheless, below we show that for some class of functions, $K^Q_{f,t,\epsilon}$ depends on the allowed error ϵ as $\Omega(1/\epsilon^{2/9})$. Moreover, we show that a larger class of functions satisfies $K^Q_{f,t,\epsilon}=\Omega(1/\epsilon^q)$ for some $0<q\leq 2/9$. Also, since K^Q_s are the cutoff numbers to used to approximate functions $2\sin(\frac{\pi}{10})f_s[\cos(\pi x)]$ via its Fourier series and $f_s(x)$ oscillates with frequency proportional to t in the range $x\in[1/2,\sqrt{3}/2]$ as can be seen from Eq. (E2), it is expected that $K^Q_{f,t,\epsilon}=K^Q_0+K^Q_1$ increases as t grows. The scaling of $K^Q_{f,t,\epsilon}$ can be obtained as an instance of the following lemma.

Lemma 8. *For a periodically 4-smooth function $g:[-1,1]\to\mathbb{R}$, the following inequality is a necessary condition for $g_K(x):=\sum_{k=-K}^{K}c_ke^{-i\pi kx}$ (c_k are Fourier coefficients of g) to satisfy $|g(x)-g_K(x)|\leq\epsilon$ for all $x\in[-1,1]$:*

$$\sum_{k=-\infty}^{-K-1}|c_k|^2+\sum_{k=K+1}^{\infty}|c_k|^2\leq\epsilon^2. \tag{E9}$$

Proof: $\int_{-1}^{1}\mathrm{d}x\,|g(x)-g_K(x)|^2\leq 2\epsilon^2$ is a necessary condition of $|g(x)-g_K(x)|\leq\epsilon$ for all $x\in[-1,1]$. Due to Parseval's identity, $\int_{-1}^{1}\mathrm{d}x|g(x)-g_K(x)|^2=2[\sum_{k=-\infty}^{-K-1}|c_k|^2+\sum_{k=K+1}^{\infty}|c_k|^2]$, thus Eq. (E9) is a necessary condition of $|g(x)-g_K(x)|\leq\epsilon$ for all $x\in[-1,1]$. □

Assuming that f is strictly periodically J-smooth, f_0 and f_1 in Eq. (E2) can also be defined to be strictly periodically J-smooth, thus $K^Q_{f,t,\epsilon}$ satisfies $K^Q_{f,t,\epsilon}=\Omega(1/\epsilon^{2/(2J+1)})$ by Lemma 4. In some cases, e.g., when $f^{(4)}$ has some jump (i.e., non-removable) discontinuities, f becomes strictly periodically 4-smooth, and thus for some class of function f, $K^Q_{f,t,\epsilon}=\Omega(1/\epsilon^{2/9})$ holds.

We can now compare the scaling of runtimes of all three algorithms in terms of ϵ. For any class of functions f such that $K^Q_{f,t,\epsilon}$ scales as $\Omega(1/\epsilon^q)$ $(0<q\leq 2/9)$, the runtime scaling of Algorithms 3, 4, and 7 in terms of ϵ behaves as $\Theta(1/\epsilon^3)$, $\Theta(1/\epsilon)$, and $\Omega(1/\epsilon^{1+2q})$, respectively. By noting that the ϵ dependence of the runtime of Algorithm 7 is $O(1/\epsilon^{5/3})$ due to the relation $K^Q_{f,t,\epsilon}=O(1/\epsilon^{1/3})$, the hierarchy presented in Eq. (21) follows.

Finally, we describe two technical factors that make Algorithm 4 more efficient than Algorithm 7 in terms of ϵ dependence.

Efficient inputting of high-frequency terms: QSVT requires d queries in total to the block-encoding unitary and its inverse when implementing a polynomial function of degree d. Consequently, the runtime of the main process of Algorithm 7 increases proportionally to the total cutoff number $K^Q_{f,t,\epsilon}$, which gets larger as the precision increases. On the other hand, the runtime of the main process of Algorithm 4 has no explicit dependence on the cutoff number $\grave{K}$. This feature is enabled because the information of Fourier coefficients $\tilde{c}_k$ is input by random sampling according to the magnitude of $\tilde{c}_k$. As can be seen from the step 12 of Algorithm 4, applying unitaries corresponding to high frequency (i.e., large k) requires $\Theta(k^2)$ runtime and is an obstacle in reducing the total runtime in general. Fortunately, such high-frequency terms $\tilde{c}_k$ for larger k typically have smaller magnitude $|\tilde{c}_k|$ than their low-frequency counterparts, and therefore costly unitaries (i.e., for large k) are rarely chosen. Moreover, the average runtime of one iteration (steps 10 to 12 of Algorithm 4) given by $\Theta[(\sum_{k=-\grave{K}}^{\grave{K}} |\tilde{c}_k|k^2)/(\sum_{k=-\grave{K}}^{\grave{K}} |\tilde{c}_k|)]$ is tailored to converge in the limit $\grave{K} \to \infty$ by modifying input function f to a periodically smooth function $\tilde{f}$ with rapidly converging Fourier coefficients, and thus has no dependence on $\grave{K}$.

Iteration number in controlization procedure is independent of ϵ: In Algorithm 7, the allowed error of $\texttt{ctrl}(e^{-iHt})$ for making B'_s is proportional to $1/K^Q_{f,t,\epsilon}$ because B'_s and its inverse is called $K^Q_{f,t,\epsilon}$ times in total and the error accumulates with each query. Since the iteration number of controlization increases with increasing accuracy, the runtime of main process of Algorithm 7 gains an additional dependence on $K^Q_{f,t,\epsilon}$, which in turn depends upon ϵ. This same logic holds for Algorithm 3. On the other hand, the iteration number $10k^2$ of controlization in Algorithm 4 is independent of $\grave{N}^{(F)}$ and consequently on ϵ due to compilation, and thus the controlization does not introduce any additional ϵ dependence.