PRX QUANTUM **6,** 020327 (2025)

Rapid Initial-State Preparation for the Quantum Simulation of Strongly Correlated Molecules

Dominic W. Berry,[1,*] Yu Tong,[2,3,4] Tanuj Khattar,[5] Alec White,[6] Tae In Kim,[7]
Guang Hao Low,[5] Sergio Boixo,[5] Zhiyan Ding,[8] Lin Lin,[8] Seunghoon Lee,[7,9]
Garnet Kin-Lic Chan,[9] Ryan Babbush,[5] and Nicholas C. Rubin[5,†]

[1] *School of Mathematical and Physical Sciences, Macquarie University, Sydney, New South Wales 2109, Australia*
[2] *Institute of Quantum Information and Matter, California Institute of Technology, Pasadena, California 91125, USA*
[3] *Department of Mathematics, Duke University, Durham, North Carolina 27708, USA*
[4] *Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina 27708, USA*
[5] *Google Quantum AI, Venice, California 90291, USA*
[6] *Quantum Simulation Technologies Inc., Boston, Massachusetts 02135, USA*
[7] *Department of Chemistry, Seoul National University, Seoul 08826, Republic of Korea*
[8] *Department of Mathematics, University of California, Berkeley, California 94720, USA*
[9] *Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, California 91125, USA*

(Received 28 November 2024; revised 21 March 2025; accepted 7 April 2025; published 8 May 2025)

Studies on quantum algorithms for ground-state energy estimation often assume perfect ground-state preparation; however, in reality the initial state will have imperfect overlap with the true ground state. Here, we address that problem in two ways: by faster preparation of matrix-product-state (MPS) approximations and by more efficient filtering of the prepared state to find the ground-state energy. We show how to achieve unitary synthesis with a Toffoli complexity about 7× lower than that in prior work and use that to derive a more efficient MPS-preparation method. For filtering, we present two different approaches: sampling and binary search. For both, we use the theory of window functions to avoid large phase errors and minimize the complexity. We find that the binary-search approach provides better scaling with the overlap at the cost of a larger constant factor, such that it will be preferred for overlaps less than about 0.003. Finally, we estimate the total resources to perform ground-state energy estimation of Fe-S cluster systems, including the FeMo cofactor by estimating the overlap of different MPS initial states with potential ground states of the FeMo cofactor using an extrapolation procedure. With a modest MPS bond dimension of 4000, our procedure produces an estimate of approximately 0.9 overlap squared with a candidate ground state of the FeMo cofactor, producing a total resource estimate of 7.3×10^{10} Toffoli gates; neglecting the search over candidates and assuming the accuracy of the extrapolation, this validates prior estimates that have used perfect ground-state overlap. This presents an example of a practical path to prepare states of high overlap in a challenging-to-compute chemical system.

DOI: 10.1103/PRXQuantum.6.020327

I. INTRODUCTION

The complexity of estimating ground-state energies of chemical and material systems using quantum phase estimation (QPE) is frequently analyzed in the ideal case in which the ground state has already been prepared accurately. In this restrictive setting, the main error in QPE originates from the eigenphase differing from the approximate output from QPE—a phenomenon known as “spectral leakage” [1] or “bit discretization error” [2]. In the case in which the ground state (or, more generally, an eigenstate) is not prepared exactly, the QPE protocol outputs an estimate of the ground-state energy with a probability proportional to the square of the overlap of the ground state and the input initial state. In order to estimate the total amount of quantum resources (logical qubits and gates) for the most important simulation problems and determine total run-times for high-confidence eigenenergies, both

---

[*] Contact author: dominic.berry@mq.edu.au

[†] Contact author: nickrubin@google.com

sources of error in QPE must be quantified and accounted for, along with the cost of performing phase estimation.

The problem of preparing an initial state with high overlap with the ground state has been of recent interest. This is in part because a generic state in the Hilbert space has exponentially small overlap with the ground state and a nontrivial state preparation might need to be performed for a reasonable success probability in QPE, which could add substantially to the total QPE cost. This has spurred a substantial body of work examining the cost of preparing various approximate wave functions and the associated success probability of QPE; e.g., analysis of product-state wave functions in the fermionic setting [3] along with informed orbital optimized improvements [4], state preparations analyzed in the context of embedding theories [5], truncated configuration interaction [6], and the use of matrix product states (MPSs) as input QPE states [7]. In terms of circuit compilations, direct synthesis costs for wave functions are known [8], along with a variety of MPS preparation techniques, including layers of two-qubit operations [9], low-depth circuits [10], and sequences of operations with an ancilla register [11]. Reference [11] in particular has been analyzed in detail for minimizing the Toffoli count [6]. A related consideration in the context of evaluating quantum advantage is that classical heuristic algorithms can also be viewed as having a state-preparation step, from which a classical estimate of the ground-state energy is then estimated and, in some cases, efficiently refined [12,13]. Hence it is relevant to ask, for specific systems, about the quantitative cost of preparing a good initial state for QPE.

Given that an appropriate initial state has been prepared, there remains the problem of how to measure the ground-state energy in a way that properly distinguishes it from excited states. If the overlap is not too small, then it may be expected that a simple sampling approach will suffice, and it will be necessary to sample enough times that there will be a high probability of sampling the ground-state energy at least once. The sample that corresponds to the ground-state energy can be identified by taking the minimum among all samples. There are two difficulties with this approach. First, the large number of samples may mean that there is exceptionally large underestimation error in at least one of the samples, resulting in an erroneously small estimate of the ground-state energy. Second, the number of samples will scale with the inverse of the squared overlap.

Amplitude amplification would suggest that the complexity should scale with the inverse overlap instead of the inverse overlap squared. Naively, in order to obtain the quadratic improvement, the range of energies for the ground state must be known. Binary search can be employed to go beyond this requirement and search for an unknown ground state. The binary-search approach has been used previously in Refs. [14–18] but these works, together with Refs. [19–21], typically aim to optimize for the circuit depth, while this work focuses on the number of non-Clifford gates needed for the whole algorithm, a more relevant metric for fault-tolerant quantum computers.

In this work, we provide improved results for both MPS preparation and filtering to determine the ground-state energy. For MPS preparation, we start by developing a method for synthesizing unitaries with low Toffoli count by decomposing the unitary into a sequence of diagonal-phasing operations together with low-cost operations. We then use that to construct a method for synthesizing only a fraction of the columns of the unitary, which we then apply to the method of Ref. [11] to provide a factor-of-7 improvement in Toffoli gates over prior work. For filtering the resulting state, we apply the theory of window functions in order to minimize the probability of estimates with large error. We describe the optimal performance provided by the Slepian prolate spheroidal window [22–24] and compare the costs to the Kaiser window previously reported in Ref. [25]. In the case of sampling, this suppresses the probability of exceptionally low energy estimates that would make the overall estimate low, overcoming the first problem. Moreover, we provide a tighter bound on the contribution to the error from excited states, by analyzing the interplay between the contribution to errors where the estimated energy is too low versus too high. We find that Kaiser windows can provide substantially improved performance.

We also provide improved scaling with the overlap by using a binary search together with amplitude estimation, similar to Ref. [26]. We successively reduce the possible range for the ground state by performing amplitude estimation at each step in order to eliminate a fraction of the range. In this way, we are able to achieve the speedup promised by amplitude amplification without any initial estimate of the ground-state energy. On the other hand, the overhead induced by this procedure means that it is preferable for small overlaps $p \lesssim 0.003$, and for large overlaps the sampling method is preferable. We apply the optimal windows for phase estimation to the amplitude-estimation procedure and thereby significantly reduce the resources compared to Ref. [26].

Armed with a detailed costing of the number of times QPE must be repeated, we estimate the full quantum resources necessary to refine the ground-state energy of several Fe-S clusters: [2Fe-2S], [4Fe-4S], and FeMo cofactor (FeMoco). In order to obtain overlaps of low-bond-dimension MPSs with the true ground state, we introduce an extrapolation protocol that uses two MPS wave functions to derive an empirical estimate of the overlap of a fixed-bond-dimension wave function and the infinite-(exact-) bond-dimension MPS. In FeMoco, at a finite bond dimension we can obtain low-bond-dimension MPSs that are candidates for different low-energy states of the cluster, although these initial MPSs do not give a reliable energy ordering. For each candidate MPS, we can estimate the overlap with the eventual eigenstate, which allows us to cost out energy estimation for FeMoco in the high-confidence regime (95% and 99% confidence level) to chemical accuracy. We also reanalyze the block-encoding costs for FeMoco and other Fe-S clusters using symmetry shifting, resulting in a reduction of the linear combination of unitaries (LCU) 1-norm by a factor of up to 2. We find that very few iterations of QPE are required (two or three iterations) due to the high overlap of low-bond-dimension MPSs. These additional iterations over single-shot QPE resource estimates provided for FeMoco in Ref. [27] along with symmetry shifting reductions in the LCU 1-norm result in 7.3×10^{10} Toffoli gates required for a full resource cost to refine an energy estimate for FeMoco. For a single candidate ground state, this amount of resources is only 2.3 times that in Ref. [27] (which uses the Hamiltonian defined in Ref. [28]) and can likely be reduced further through improved symmetry shifting. Once the accurate energies of different candidates are obtained, they can be ordered to determine the ground-state energy. Ultimately, the high extrapolated overlap achieved in this problem suggests that the combination of generating one (or more) candidate MPS initial states followed by QPE is a practical approach to refine the ground-state energy in a realistic challenging chemical system.

In the following we begin in Sec. I A by summarizing the results that will be presented. We then give the background for both MPS preparation and phase estimation in Sec. II. Section III describes our new method for unitary synthesis and preparing MPSs. Then Sec. IV summarizes the procedure to perform phase estimation optimally for confidence intervals. This is a procedure that is used for both approaches to searching for the ground-state energy. In Sec. V, we describe the sampling approach and in Sec. VI we describe the binary-search approach. These results are used to estimate resource requirements for real systems in Sec. VIII and we conclude in Sec. IX.

A. Results overview

Overall, quantum state preparation proceeds in three steps:

(1) classical calculation of an approximation of the quantum state (Sec. VIII)
(2) preparation of that classically approximated state on the quantum computer (Sec. III)
(3) filtering of that approximate state to give the ground state accurately (Secs. V–VII)

We analyze the energy estimate, as that is the result desired, but the filtering also produces the ground state accurately. Section III shows how the complexity depends on the MPS bond dimension, whereas the analysis of filtering in Secs. V–VII shows how the complexity scales with overlap with the ground state. There is a trade-off where an increased bond dimension makes the initial MPS preparation more costly but improves the overlap, making the filtering less costly. We analyze that trade-off in Sec. VIII, thereby providing the resource analysis for the complete procedure.

For MPS preparation, we first derive a new result for the synthesis of general unitary operations with reduced Toffoli count. This method reduces the synthesis to layers of phase shifts alternating with increment or decrement operations and Hadamard gates. The layers of phase shifts can be applied with reduced Toffoli count using the quantum read-only memory (QROM) [8,29,30]. As a result, there are at most $N_{\rm un}+1$ layers of QROM and each QROM has complexity $\mathcal{O}(\sqrt{N_{\rm un}})$, where $N_{\rm un}$ is the dimension of the unitary to be synthesized. The subscript "un" is used to distinguish this quantity from N used in phase estimation. For a summary of the notation used in this work, see Appendix A.

We then use this result to derive a procedure to synthesize columns of a unitary operation; i.e., a unitary operation where the input state is restricted to lie within a subspace. If only half the columns of the unitary need be synthesized, we reduce the problem to synthesis of an initial unitary of dimension $N_{\rm un}/2$, a controlled qubit rotation, and then controlled synthesis of a unitary of dimension $N_{\rm un}/2$. That controlled unitary has half the complexity of synthesizing a unitary operation of dimension $N_{\rm un}$, because only half the layers are required. Then, in order to synthesize $N_{\rm un}/d$ columns of a unitary, we can iterate this procedure another $d-2$ times.

The MPS preparation can then be achieved by a sequence of steps in which $N_{\rm un}/d$ columns of a unitary need to be synthesized. Moreover, the initial unitary in the above procedure can be merged with other operations, further reducing the complexity. As a result, the overall Toffoli complexity is significantly reduced over that obtained for methods in prior work, now making it a small complexity as compared to the complexity of phase estimation.

For energy estimation, there are two approaches that we consider: direct sampling and a binary search with amplitude estimation. The parameters of the problem are as follows:

(i) the initial squared overlap of the prepared state with the ground state, p
(ii) the block-encoding normalization factor, λ
(iii) the allowable error in the energy estimate, ϵ, and
(iv) the confidence level, $1-q$

For the direct-sampling approach, we provide expressions to determine exact costings in terms of special functions. We then derive asymptotic expressions to provide the expected scaling of the complexity in the above parameters. The total number of queries to the qubitized walk operator encoding the Hamiltonian is approximately

$$\frac{\lambda \ln(2/q)}{2p\epsilon} \ln\left[\frac{\ln(2/q)}{pq}\right], \tag{1}$$

corresponding to the leading term in Eq. (79). For the binary-search approach, under the same conditions, the query complexity is

$$\frac{7.77\lambda}{\sqrt{p}\epsilon} \ln\left(\frac{4}{\sqrt{p}}\right) \ln\left(\frac{\log_{\sqrt{2}}(\lambda/\epsilon)}{q}\right), \tag{2}$$

as in Eq. (118). This has improved scaling as $1/\sqrt{p}$, which is a square-root improvement, albeit with a constant factor about 16 times larger. This constant factor indicates that for best performance, the sampling approach should be used for $p \gtrsim 0.003$ and the binary-search approach only for smaller p.

These asymptotic results can be inaccurate for realistic values of parameters, so we provide improved approximations for phase measurements with window functions in Sec. V. We provide series expansions for the error and cost, both for the Kaiser window and for the prolate spheroidal window. In the process, we correct an error in the work of Slepian from 1965 [22], which gave incorrect terms. We show that the error is asymptotically lower for the prolate spheroidal window but the cost (the number of oracle calls to achieve a given error) is asymptotically the same for the two windows.

We show how to properly account for the excited states in sampling to obtain the ground state. These states contribute to two types of error:

Type I: The excited states contribute to the probability of a sample with large error, more than ϵ below the ground-state energy.

Type II: Samples corresponding to excited states contribute to estimates that are too high, because it is an accurate (or high) estimate of that energy, but the excited-state energy is higher than the ground-state energy.

When excited-state energies are close to the ground state, then they increase the probability of a type I error, but that also reduces the probability of a type II error. We provide a careful accounting of the contribution of these two errors to provide a more accurate estimate of the cost of the sampling approach. Surprisingly, we find that the Kaiser window can provide *better* results than the prolate spheroidal window when accounting for excited states. Taking all these considerations into account yields results close to those for the approximate asymptotic expression given above.

We provide numerical results for query complexities accounting both for the exact error with window functions and the effect of the excited states. The results for queries to block encoding of the Hamiltonian are provided in Fig. 15 and for the number of calls to the initial-state preparation in Fig. 23. This shows that it would take small values of the squared overlap p (below approximately 0.003 for the case of 95% confidence intervals) for the binary-search approach to be optimal, agreeing with the analytic result. For more typical values of p above 0.003, such as those we encounter in this study, the sampling approach is optimal.

For FeMoco resource estimates, we develop an extrapolation scheme that allows us to estimate the overlap of a fixed-bond-dimension MPS with an infinite-bond-dimension MPS—e.g., the true eigenstate. The extrapolation protocol is constructed from two empirically observed linear relationships,

$$\log\left(1 - \left|\langle\Phi(M')|\Phi(\infty)\rangle\right|^2\right) \text{ versus } \left(\log(M')\right)^2, \tag{3}$$

$$\log\left(\left|\langle\Phi(M')|\Phi(M'')\rangle\right|^2 - \left|\langle\Phi(M')|\Phi(\infty)\rangle\right|^2\right) \text{ versus } \left(\log(M'')\right)^2, \quad \text{where } M' \ll M'', \tag{4}$$

and verified on Fe_2S_2 and Fe_4S_4 systems where accurate estimates of the ground states can be computed and where the above extrapolation can be verified. We analyze three different MPS wave functions for FeMoco initialized through a procedure similar to Ref. [28,31] that is believed to generate candidates for competing low-energy eigenstates (corresponding to different spin couplings) of the $S = 3/2$ ground state. The high overlaps estimated with some eigenstate in the low-energy manifold, as produced by our protocol, suggest that the combination of initializing different candidate MPSs, followed by QPE, is a promising computational procedure to map out the lowest eigenenergies and subsequently refine the ground-state energy in this practically challenging chemical simulation problem.

II. BACKGROUND

A. Matrix-product-state preparation

MPSs provide a systematically improvable approximation of entangled states, thus providing a class of initial states with tunable overlap. The Toffoli complexity of MPS preparation has previously been analyzed in Ref. [6] using the approach from Ref. [11] together with the unitary-synthesis scheme of Low, Kliuchnikov, and Schaeffer (LKS) [8]. In this work, we provide a significantly more efficient unitary-synthesis scheme than that of Ref. [8], thereby enabling more efficient MPS preparation.

MPSs for $\mathcal{N}$ subsystems of dimension d are of the form

$$\sum_{\{n\}} \mathrm{Tr}\left[A_1^{(n_1)} A_2^{(n_2)} \ldots A_{\mathcal{N}}^{(n_{\mathcal{N}})}\right] |n_1, n_2, \ldots, n_{\mathcal{N}}\rangle\,. \tag{5}$$

The matrices $A_j^{(n_j)}$ are of dimension χ, called the bond dimension, and the indices n_j range over d values. The principle of the approach from Ref. [11] is to use an ancilla of dimension χ, together with a sequence of unitary operations on this ancilla together with each of the $\mathcal{N}$ subsystems of dimension d. Using the fact that one is free to represent the MPS of Eq. (5) in left-canonical form, the matrices of the MPS can be cast as unitaries such that

$$G[j]_{\alpha_j n_j,\alpha_{j-1}0} = (A_j^{(n_j)})_{\alpha_{j-1},\alpha_j}. \tag{6}$$

That is, the unitaries $G[j]$ are of dimension $d\chi$ but only χ columns are specified due to the input on the physical leg being zero. The notation $(A_j^{(n_j)})_{\alpha_{j-1},\alpha_j}$ indicates the matrix element α_{j-1},α_j of matrix $A_j^{(n_j)}$. There is a requirement for this technique that the specified columns are orthonormal, so they may correspond to columns of a unitary.

The sequence of unitary operations used to prepare the MPS is shown in Fig. 1. There are three distinct cases in which these unitary operations are performed:

1. The initial unitary $G[1]$ is on dimension $d\chi$ but is guaranteed to have input states that are zeroed. Therefore, it corresponds to simply preparing a state of dimension $d\chi$, which is simpler than synthesizing a general unitary operation on this dimension.
2. There are $\mathcal{N}-2$ unitaries $G[2]$ to $G[\mathcal{N}-1]$ on dimension $d\chi$ where one of the input registers is initialized to $|0\rangle$ but the other is the ancilla, which may be in a general state entangled with other registers. Therefore, only χ columns of the unitary need be synthesized.
3. The final unitary $G[\mathcal{N}]$ is required to reset the ancilla to $|0\rangle$.

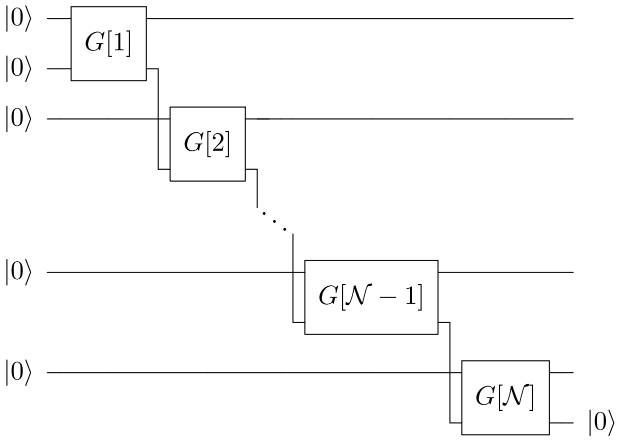

FIG. 1. The sequence of operations used to prepare an MPS. The input state $|0\rangle$ second from the top is of dimension χ, as is the final output state at the bottom $|0\rangle$.

For this last operation to be possible, it must be the case that the ancilla prior to the operation has support on dimension d. That is, the Schmidt decomposition describing the entanglement between the ancilla of dimension χ and the remaining system can have rank no more than d. The operation $G[\mathcal{N}]$ is unitary and so cannot affect the Schmidt rank; it can only transform one Schmidt basis to another Schmidt basis. Since this operation is transforming no more than d orthonormal states to another d orthonormal states, it may be described just as a unitary operation without introducing or discarding ancillas. In the cases in which we are interested, $d \leq \chi$, so $G[\mathcal{N}]$ may be described as just a unitary operation on dimension χ. Because $G[\mathcal{N}-1]$ already includes a unitary operation on this subsystem, the unitary operation transforming the Schmidt basis may be combined with $G[\mathcal{N}-1]$ to give a single unitary and $G[\mathcal{N}]$ does not contribute to the cost. This change just changes the entries in the χ columns of $G[\mathcal{N}-1]$ to be synthesized and does not affect its cost of synthesis.

B. Phase estimation

In phase estimation for ground-state energy estimation, a standard approach [32,33] is to construct a walk operator W from a block encoding of the Hamiltonian, which yields eigenvalues $e^{\pm i\phi_j}$, with

$$\phi_j = \arccos(\lambda_j/\lambda). \tag{7}$$

Here, λ is the constant in the block encoding of H; i.e., the block encoding gives H/λ. Note that there are two conventions in the definition of W depending on what factor of i is included (corresponding to the work in Refs. [32] and [33]). The other convention yields eigenvalues $\mp e^{\mp i \arcsin(\lambda_j/\lambda)}$ but use of either convention yields equivalent results.

From measuring $\pm\phi_j$, we can recover λ_j via $\lambda_j = \lambda\cos(\phi_j)$. This method has a number of useful features:

1. The cosine function is even and so eliminates the ± sign ambiguity.
2. The function $\lambda\cos(\phi_j)$ is monotonically decreasing in ϕ_j, from λ for $\phi_j = 0$ to $-\lambda$ for $\phi_j = \pi$. (We use the convention that arccos gives values in the range $[0,\pi]$.) Therefore λ_0, which is the smallest among all λ_j, corresponds to the largest among ϕ_j.
3. Because $|\cos(x) - \cos(y)| \leq |x-y|$ for all $x,y \in \mathbb{R}$, if we want to estimate λ_0 to within additive error ϵ, it suffices to estimate ϕ_0 to within ϵ/λ.

Each time we run the quantum phase-estimation circuit, we will obtain a phase estimate $\hat{\phi}$ that corresponds to ϕ_j for some j, up to a phase error that we will define later. Each phase estimate then gives us an estimate for the corresponding eigenvalue λ_j through

$$\hat{\lambda} = \lambda \cos(\hat{\phi}). \tag{8}$$

One of the problems that we will deal with is that we will not know with certainty which eigenvalue or eigenstate an estimate $\hat{\lambda}$ really corresponds to. For example, an estimate $\hat{\lambda}$ may correspond to an excited state but the phase error can make it smaller than the ground-state energy λ_0.

1. The quantum phase-estimation circuit

A main building block of the quantum phase-estimation algorithm that we are going to use is the controlled walk operator,

$$|0\rangle\langle 0| \otimes W^{\dagger} + |1\rangle\langle 1| \otimes W. \tag{9}$$

Compared to the controlled-W, this operator needs the same number of gates to implement but doubles the resulting phase difference [29]. Using this version of the controlled walk operator, the controlled part in quantum phase estimation becomes

$$\sum_{k=0}^{N-1} |k\rangle\langle k| \otimes W^{2k-N}. \tag{10}$$

By using one more controlled-W (rather than controlling between W and $W^{\dagger}$) and relabeling, the controlled unitary becomes

$$\sum_{n=0}^{2N-1} |n\rangle\langle n| \otimes W^{n-N}. \tag{11}$$

Now suppose that the control register is initialized in the state $|\Gamma\rangle = \sum_{n=0}^{2N-1} \gamma_n |n\rangle$ and the system register is initialized in a superposition of eigenstates $|\Phi\rangle = \sum_j \Phi_j |\psi_j\rangle$. Then, the state after applying the controlled unitary is

$$\begin{aligned}\sum_{n=0}^{2N-1} \gamma_n |n\rangle \otimes W^{n-N} |\Phi\rangle \\ = \sum_j \Phi_j \sum_{n=0}^{2N-1} \gamma_n e^{i(n-N)\phi_j} |n\rangle \otimes |\psi_j\rangle .\end{aligned} \tag{12}$$

After applying the QFT on the control register, the quantum state becomes

$$\sum_j \Phi_j \sum_{l=0}^{2N-1} (-1)^l e^{-(\phi_j - \pi l/N)/2}\, \Gamma\left(\phi_j - \frac{\pi l}{N}\right) |l\rangle \otimes |\psi_j\rangle , \tag{13}$$

where $\Gamma(x)$ is a kernel function defined to be

$$\Gamma(x) = \frac{1}{\sqrt{2N}} \sum_{n=0}^{2N-1} e^{inx} \gamma_n. \tag{14}$$

The phase factor does not affect the probabilities and so can be ignored in the following discussion.

2. The phase error

From Eq. (13), the probability of obtaining a phase estimate $\pi l/N$ is

$$\Pr\left[\hat{\phi} = \frac{\pi l}{N}\right] = \sum_j |\Phi_j|^2 \left|\Gamma\left(\phi_j - \frac{\pi l}{N}\right)\right|^2 . \tag{15}$$

Here, $\hat{\phi}$ is the random variable corresponding to the phase estimate. Note that the above probability resembles a convolution between two probability measures. From this observation, we can write down a decomposition for $\hat{\phi}$ in the following way:

$$\hat{\phi} = \phi + \Delta\phi, \tag{16}$$

where ϕ is a random variable satisfying

$$\Pr[\phi = \phi_j] = |\Phi_j|^2. \tag{17}$$

This means that ϕ is the output of an exact phase estimation for W and that $\Delta\phi$ is described by the following conditional distribution:

$$\Pr\left[\Delta\phi = \frac{\pi l}{N} - \phi \,\middle|\, \phi\right] = \left|\Gamma\left(\phi - \frac{\pi l}{N}\right)\right|^2 . \tag{18}$$

We call $\Delta\phi$ the *phase error*.

In QPE, we want the phase error $\Delta\phi$ to be small. One way to characterize this is through the variance of $\Delta\phi$, which is minimized by using amplitudes proportional to a cosine function [34]. The use of this control state for this application has been considered in Ref. [29]. But oftentimes we want to make $\Delta\phi$ small with a probability that is arbitrarily close to 1, thus giving us reliable estimates in many runs. More precisely, we want

$$\Pr\left[|\Delta\phi| \geq \epsilon_\phi | \phi\right] \leq \delta, \tag{19}$$

where $|\Delta\phi|$ is calculated modulo 2π. The task of optimizing the performance is described by the theory of window functions, where the optimal performance is provided by the Slepian prolate spheroidal window [22,23]. That is difficult to calculate but can be approximated by the Kaiser window, which can be calculated by Bessel functions [25,35]. Either of these can be used to obtain scaling $\mathcal{O}(\epsilon_\phi^{-1} \ln(\delta^{-1}))$ with a small constant factor.

3. Window functions

In the theory of window functions, we would replace γ_n with a continuous function $w(z)$, so that $w(n-N+1/2)=\gamma_n$. Then the kernel function is approximately

$$\Gamma(x)=\frac{1}{\sqrt{2N}}\sum_{n=0}^{2N-1}e^{inx}\gamma_n\approx\frac{e^{i(N-1/2)x}}{\sqrt{2N}}\int_{-N}^{N}e^{izx}w(z)\,dz. \tag{20}$$

In the phase estimation, it is also trivial to adjust the measurement so that a continuous range of outcomes is obtained. That is achieved by imposing an extra (known) phase shift in addition to ϕ and correcting for it in the estimate. Then we can consider the error probability distribution as a continuous function of x given by $|\Gamma(x)|^2$. When $\Gamma(x)$ is approximated by the integral rather than the sum, then it is no longer periodic modulo π. The integral gives a nonperiodic function of x over the whole real line.

The discrete case with γ_n corresponds to sampling $w(z)$ at integer spacing at $2N$ points from $-(N-1/2)$ to $N-1/2$. That corresponds to multiplying $w(z)$ by a comb function, so $\Gamma(x)$ is the periodic function obtained by convolving the Fourier transform of $w(z)$ with a comb function. As described in Ref. [25], the tail probabilities for this continuous case correspond to the average over the tail probabilities for the discrete case, where the samples are shifted (so starting from $-(N-\nu)$ for $\nu\in[0,1]$). This means that some discrete case must give at least the performance (in terms of small tail probabilities) as the continuous case. A further advantage of using continuous window functions is that they can be scaled to a unit interval and analyzed independently of the specific value of N.

Examples of the probability distributions for the error for a range of window functions are given in Fig. 2. The traditional textbook version [37] of quantum phase estimation uses a flat distribution, which corresponds to a control register of unentangled qubits in $|+\rangle$ states. That window gives a narrow peak for the error that decays slowly, resulting in both large variance and large tails. The cosine window gives tails that decay more rapidly, to yield excellent performance for the variance. The Kaiser and Slepian windows give tails that are lower, resulting in smaller tail probabilities for confidence intervals. The Kaiser window [35] is proportional to the modified Bessel function of the first kind,

$$w_{\text{Kaiser}}(x)\propto I_0\left(\pi\alpha\sqrt{1-(x/N)^2}\right), \tag{21}$$

whereas the Slepian window [22,36] is proportional to the prolate spheroidal function,

$$w_{\text{Slepian}}(x)\propto \mathrm{PS}_{0,0}(c,x/N). \tag{22}$$

The Digital Library of Mathematical Functions [38] uses the notation $\mathrm{Ps}_0^0(x/N,\gamma^2)$ instead, with γ instead of c. These window functions do not decay as fast as the sine window, so the variance is larger. Note that for this example, the Kaiser and Slepian windows are almost indistinguishable.

III. MPS PREPARATION

In this section, we analyze the Toffoli complexity needed for preparation of MPSs. We first introduce a general unitary-synthesis method that improves on the approach of LKS and then we provide a method to generalize this approach to synthesizing columns of a unitary, as is appropriate for MPS preparation.

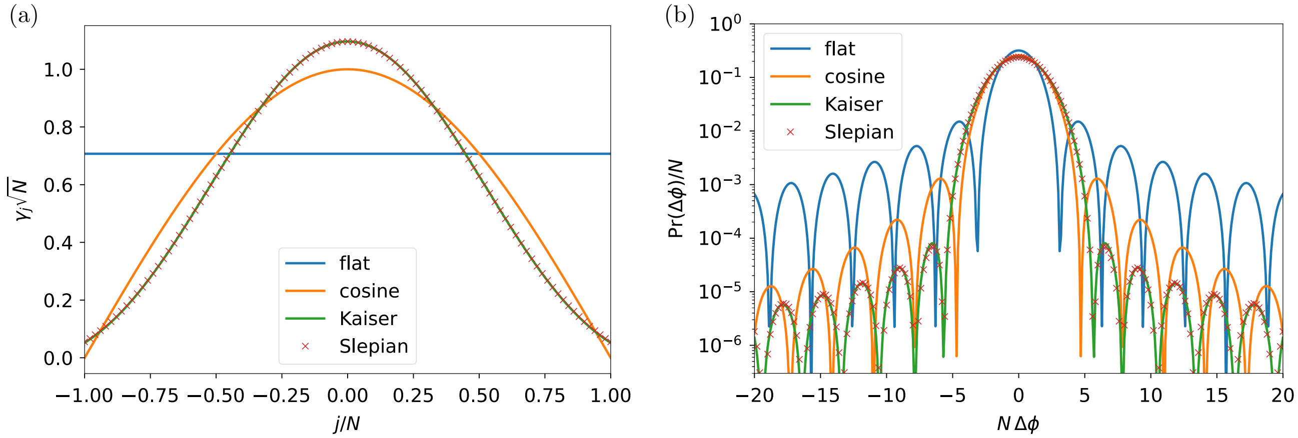

FIG. 2. The (a) windows and (b) error probability distribution for phase measurements. The cosine window is the one that provides the minimum phase variance. The Kaiser window [35] is used with $\alpha\approx 1.5$ [see Eq. (21)] and the Slepian window [22,36] is used with $c=5$ [see Eq. (22)].

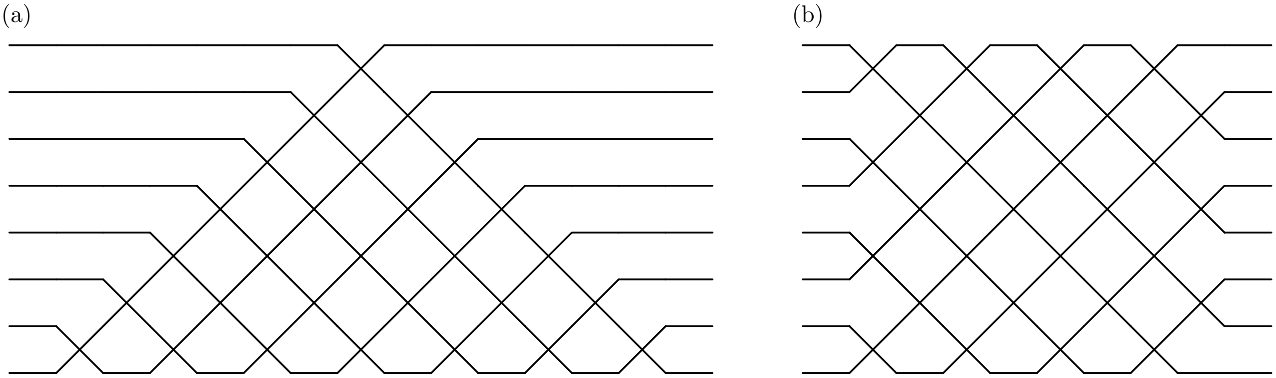

FIG. 3. The decomposition of a multiport interferometer for eight modes, equivalent to a transformation on three qubits: (a) the triangular Reck-Zeilinger [41] decomposition; (b) the rectangular decomposition from Ref. [39].

A. New unitary-synthesis method

We now provide an alternative method of unitary synthesis that significantly improves on the method of LKS. We consider the rectangular array of beam splitters for decomposing a multiport interferometer as in Ref. [39]. This has previously been considered for unitary synthesis in the optical context in Ref. [40] but we provide a significant improvement over that work. It has been shown in Ref. [39], that a general $N_{\rm un} \times N_{\rm un}$ interferometer can be decomposed into $N_{\rm un}$ layers of no more than $N_{\rm un}/2$ beam splitters. This is the problem of decomposing a unitary matrix (which describes an interferometer) into a product of Givens rotations (describing the beam splitters). The method used in Ref. [39] is to choose a sequence of Givens rotations such that multiplying the matrix by them zeros out entries of the matrix. We will show how to derive a quantum circuit given this decomposition; our method is not dependent on the technique used to construct the decomposition.

The form of the decomposition from Ref. [39] is shown in Fig. 3(b) for the example of $N_{\rm un} = 8$. An M-port interferometer is equivalent to an $N_{\rm un} \times N_{\rm un}$ unitary and each layer of beam splitters is equivalent to a block-diagonal matrix with 2×2 blocks. The first layer of beam splitters corresponds to a block-diagonal matrix where the first block is in rows and columns 1 and 2. The first layer of beam splitters in Fig. 3(b) would correspond to a unitary matrix with nonzero entries shown by the asterisks in

$$\begin{bmatrix} * & * & & & & & & \\ * & * & & & & & & \\ & & * & * & & & & \\ & & * & * & & & & \\ & & & & * & * & & \\ & & & & * & * & & \\ & & & & & & * & * \\ & & & & & & * & * \end{bmatrix}. \tag{23}$$

The next layer has blocks shifted by one and so would correspond to a matrix with nonzero entries:

$$\begin{bmatrix} * & & & & & & & \\ & * & * & & & & & \\ & * & * & & & & & \\ & & & * & * & & & \\ & & & * & * & & & \\ & & & & & * & * & \\ & & & & & * & * & \\ & & & & & & & * \end{bmatrix}. \tag{24}$$

The layers alternate between operators of these two forms. Because the second form in Eq. (24) is equivalent to that in Eq. (23) except shifted by 1, it can be transformed to the same form by increment and decrement operations.

In the optical interferometer, each beam splitter may be expressed as two 50:50 beam splitters with a phase shift in between, as in Fig. 4. The 50:50 beam splitter corresponds to a Hadamard matrix. That is equivalent to expressing a general qubit unitary as

$$\begin{bmatrix} e^{i\varphi_0} & 0 \\ 0 & e^{i\varphi_1} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} e^{i\theta} & 0 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} e^{i\phi} & 0 \\ 0 & 1 \end{bmatrix}. \tag{25}$$

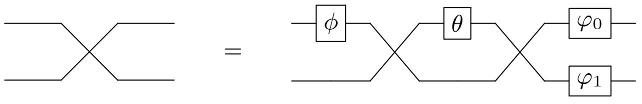

FIG. 4. The crossings depicted in Fig. 3 are general beam splitters with arbitrary phases and reflectivities (left). They can be implemented with two 50:50 beam splitters with phase shifts (right).

Since the layers correspond to 2×2 block-diagonal unitaries, this decomposition may be performed on the blocks individually; so, for example, Eq. (23) is of the form (with H the Hadamard)

$$\begin{bmatrix} e^{i\varphi_0} &&&&&&& \\ & e^{i\varphi_1} &&&&&& \\ && e^{i\varphi_2} &&&&& \\ &&& e^{i\varphi_3} &&&& \\ &&&& e^{i\varphi_4} &&& \\ &&&&& e^{i\varphi_5} && \\ &&&&&& e^{i\varphi_6} & \\ &&&&&&& e^{i\varphi_7} \end{bmatrix} \begin{bmatrix} H &&& \\ & H && \\ && H & \\ &&& H \end{bmatrix} \begin{bmatrix} e^{i\theta_0} &&&&&&& \\ & 1 &&&&&& \\ && e^{i\theta_1} &&&&& \\ &&& 1 &&&& \\ &&&& e^{i\theta_2} &&& \\ &&&&& 1 && \\ &&&&&& e^{i\theta_3} & \\ &&&&&&& 1 \end{bmatrix}$$

$$\times \begin{bmatrix} H &&& \\ & H && \\ && H & \\ &&& H \end{bmatrix} \begin{bmatrix} e^{i\phi_0} &&&&&&& \\ & 1 &&&&&& \\ && e^{i\phi_1} &&&&& \\ &&& 1 &&&& \\ &&&& e^{i\phi_2} &&& \\ &&&&& 1 && \\ &&&&&& e^{i\phi_3} & \\ &&&&&&& 1 \end{bmatrix}. \tag{26}$$

This can be further simplified, because the phases φ_j can be combined with the next layer. Then, the next layer can be similarly decomposed and its φ_j phases can be combined with the layer after, and so on. As a result, the phases on *all* basis states are only needed for the very last operation. As a result, we have $2N_{\mathrm{un}}$ layers of phase shifts with $\lfloor N_{\mathrm{un}}/2 \rfloor$ phases each and one final layer with N_{un} phases. Then the block-diagonal matrices have diagonals with all Hadamards, which can be achieved with just a Hadamard on a single qubit.

To illustrate how this procedure is performed, we show the steps in Fig. 5. In Fig. 5(a), each of the general beam splitters from Fig. 3(b) is expressed in terms of 50:50 beam splitters and phase shifts as per Fig. 4. Then, one can combine phases as in Fig. 5(b), where, e.g., φ_0^0 and φ_0^1 are combined. One can also commute some of these phase shifts through the 50:50 beam splitters. For example, φ_2^0 is moved from the left to the right of the 50:50 beam splitters on modes 2 and 3, so there are phases of φ_2^0 combined with φ_1^1 and φ_1^2 on the right. The fully simplified form, where this process of combining and commuting phase shifts has been applied all the way through, is then shown in Fig. 5(c). There are 16 layers of phase shifts with four or three phases and a final layer where phase shifts are applied on all modes (or computational basis states in the case of applying a unitary operation).

Note that the Toffoli cost of QROM is minimized if we are able to output more of the data together. It is possible to output the data for two layers of phase shifts at once, so there are $N_{\mathrm{un}} + 1$ uses of QROM with output size N_{un}. Considering the first layer of beam splitters, it is now decomposed into two layers of phase shifts that can be chosen to be only on odd (or even) modes, as well as two layers of 50:50 beam splitters. The equivalent quantum circuit corresponds to two layers of phase shifts that only depend on the first $n-1$ qubits and two layers of Hadamards on the last qubit, as shown in Fig. 6.

It is therefore possible to perform a QROM on the first $n-1$ qubits, perform the operations on qubit n, and then erase the QROM because it does not depend on the last qubit. The quantum circuit is shown in Fig. 7. A further simplification is possible because of the way in which QROM erasure is performed. The data qubits are measured in the X basis and sign corrections need to be performed on the control qubits. However, these sign corrections are phases that can be combined into φ_j. That is, we just need to modify the phases used in the remainder of the circuit to account for the sign fix-ups needed for the QROM erasure. The Toffoli complexity is therefore

$$\left\lceil \frac{N_{\mathrm{un}}}{2\Lambda} \right\rceil + (\Lambda - 1)2b - 1 \tag{27}$$

to output the $N_{\mathrm{un}}/2$ items of data of size $2b$ (for two rotations), with Λ a power of 2. In this expression, -1 accounts for the fact that the cost in the first term is for unary iteration [30] and the Toffoli cost of unary iteration is 1 less than the number of items [29]. This is a cost including a control, and this control will be needed in the overall scheme and so is allowed for here. The cost for the two rotations is $2(b-2)$, where the $b-2$ cost for a phase rotation is explained in Ref. [42]. In the following, we will bundle these two costs together.

A minor issue is the even layers in which the blocks are shifted by 1. This is easily accounted for in the quantum circuit by applying an increment in the computational basis

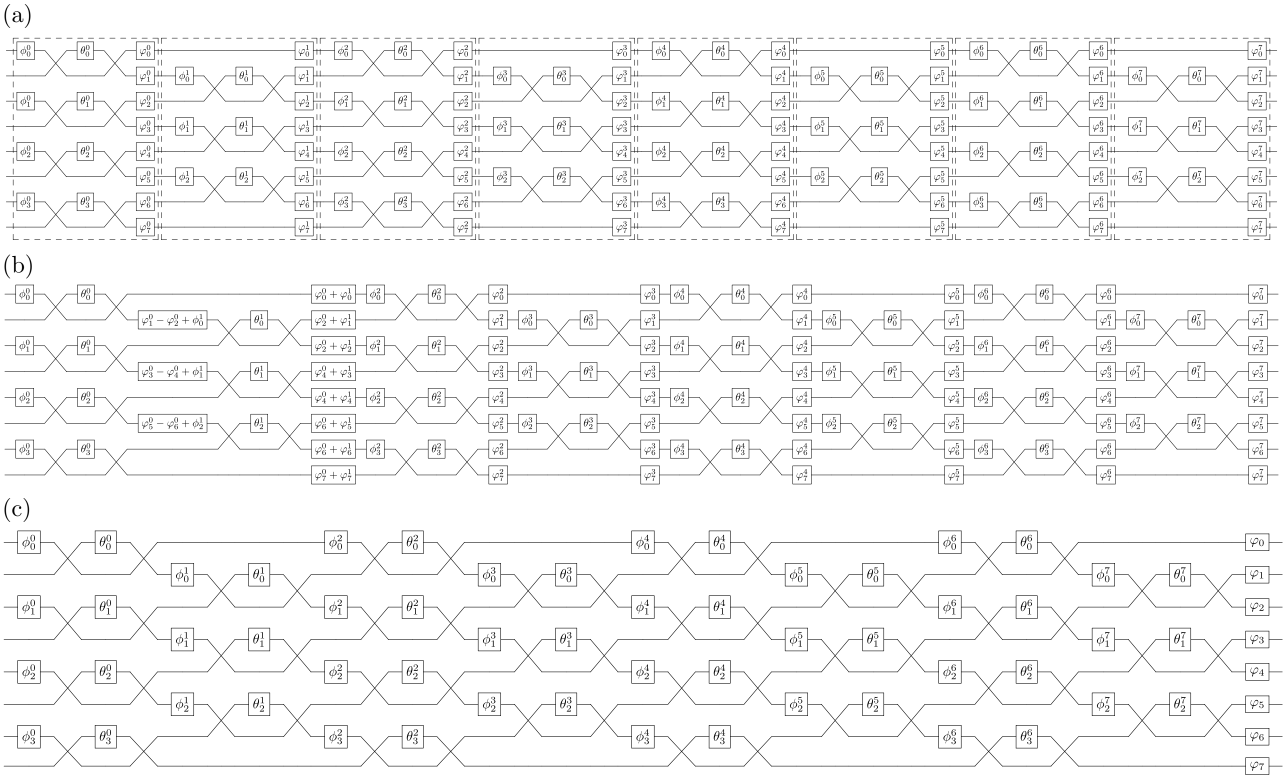

FIG. 5. The sequence of steps in simplifying the interferometer to one involving 50:50 beam splitters. (a) The interferometer from Fig. 3(b), with the general beam splitters implemented in terms of 50:50 beam splitters and phase shifts as in Fig. 4. The dashed boxes indicate the columns of general beam splitters. (b) The initial step of simplification in which the φ phases are combined with others. (c) The fully simplified form, in which the ϕ and φ phases are determined from combinations of those from (a) and (b).

to shift the blocks such that the layer may be implemented in the same way as for the first layer. Each increment (or decrement) can be performed with $n-2$ Toffolis and there are $N_{\mathrm{un}}-1$ needed, for a total complexity of $(n-2)(N_{\mathrm{un}}-1)$. (Recall that the cost of modular addition is $n-1$ as per Ref. [43], with a saving of one Toffoli when the number to be added is classical, rather than provided in a quantum register.)

The complete quantum circuit for the case of dimension $N_{\mathrm{un}}=8$ (three qubits) is shown in Fig. 8. There, it can be seen that $N_{\mathrm{un}}-1=7$ increment or decrement operations are needed. There are eight layers in which QROM is used to output ϕ_j and θ_j and these output registers are measured in the X basis to erase them. The appropriate sign

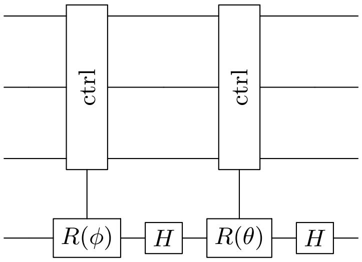

FIG. 6. Two layers of phases controlled by the first $n-1$ qubits and Hadamards on the last qubit. The boxes labeled $R(\phi)$ and $R(\theta)$ are phase rotations controlled by the first $n-1$ qubits.

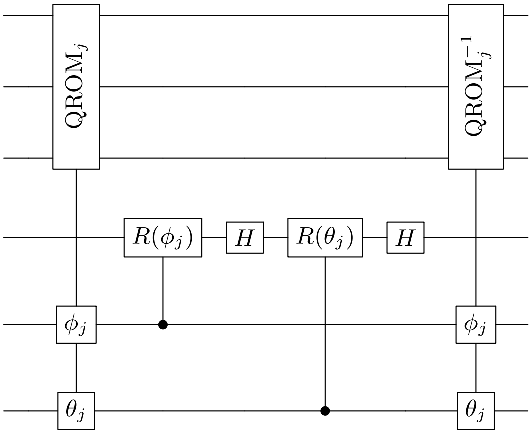

FIG. 7. Two layers of phases obtained by using a single QROM. The bottom two registers are temporary data registers used for the output of the QROM. The QROM on the left outputs both ϕ_j and θ_j and then the QROM on the right is an inverse QROM for erasure. In the middle, the rotations are controlled by the values in the data registers.

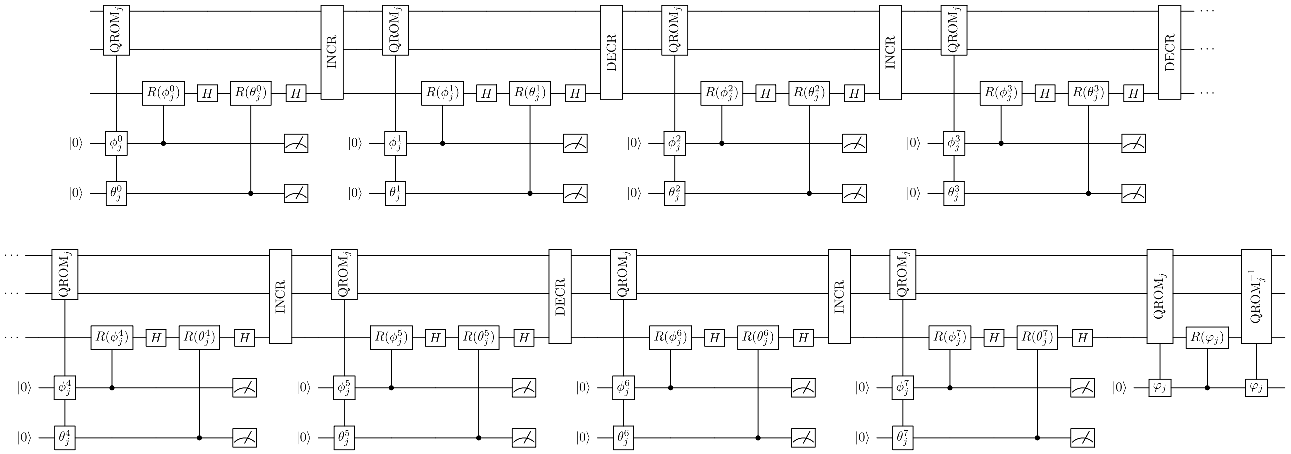

FIG. 8. The complete circuit diagram for synthesis of a unitary operation, shown in two parts due to the length of the circuit. We are applying the equivalent of the interferometer shown in Fig. 5. The pairs of phase shifts with ϕ and θ are applied with the circuit shown in Fig. 7. After each, there is an increment or decrement operation (INCR or DECR). The data-output registers are erased each time via a measurement in the X basis, indicated by the meter. At the end, a final layer of phases φ_j is applied using a QROM on all qubits.

corrections are applied using the phases in the next steps. At the end, QROM on all qubits is used to output φ_j and an explicit QROM erasure step is used since there are no further steps where sign corrections can be applied.

Bringing all these complexities together, we have the following:

(i) The complexity for the $2N_{\rm un}$ layers of phase shifts and Hadamards is

$$N_{\rm un}\left(\left\lceil\frac{N_{\rm un}}{2\Lambda}\right\rceil+2\Lambda b-5\right). \tag{28}$$

(ii) The increments and decrements have complexity $(n-2)(N_{\rm un}-1)$.

(iii) The final phase shifts have complexity

$$\left\lceil\frac{N_{\rm un}}{2\Lambda}\right\rceil+2\Lambda b+\left\lceil\frac{N_{\rm un}}{\Lambda'}\right\rceil+\Lambda'-6. \tag{29}$$

Note that the QROM cost appearing in Eq. (28) is for $N_{\rm un}/2$ items of data of size $2b$, whereas the QROM cost in Eq. (29) is for $N_{\rm un}$ items of data of size b. This expression for the cost would normally be given as $\lceil N_{\rm un}/\Lambda''\rceil+\Lambda'' b$ but, clearly, if Λ is optimal for minimizing the cost in Eq. (28), then $\Lambda''=2\Lambda$ will be optimal in Eq. (29). For this reason, we have simply used 2Λ in Eq. (29), rather than giving a third value of Λ to be optimized. In contrast to Eq. (28), it represents outputting 2Λ values of size b, rather than Λ values of size $2b$.

For Eq. (29), there is a term -6 that comes from -1 for the initial QROM, -2 for the addition into the phase gradient, and -3 for the sign fix-up. The sign fix-up is explained in Ref. [30, Fig. 6] and can be constructed from two unary iterations on subsets of the qubits. That is, Ref. [30, Fig. 6] has $|h\rangle$ indicating the most significant qubits of the register and $|l\rangle$ indicating the least significant qubits of the register. The X operation and S controlled by $|l\rangle$ in that work indicates the conversion of l to one-hot unary. This binary-to-unary conversion is not controlled and so has cost $\Lambda'-2$ Toffolis (where Λ' is the number of possible values of l). The T controlled by $|h\rangle$ is a unary iteration combined with a controlled phase on the one-hot unary representation of $|l\rangle$. We want the overall operation to be controlled and this unary iteration may be made controlled to give a cost of $\lceil N_{\rm un}/\Lambda'\rceil-1$. The -2 and -1 give an overall saving of -3 Toffolis for the sign fix-up. In numerical testing of this approach against the LKS approach, we find that the complexity is reduced by about a factor of 7 over a wide range of parameters. For details of the costing of the LKS approach, see Appendix B. There,

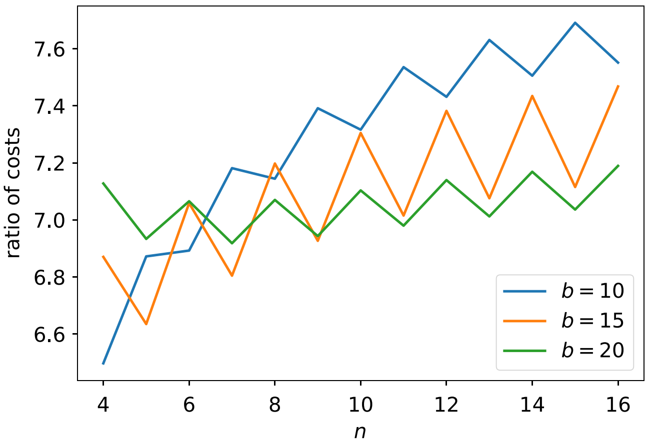

FIG. 9. The ratio of the Toffoli cost for synthesis via the LKS approach to our new approach as a function of $N_{\rm un}=2^n$ for a range of numbers of bits of precision in the rotations b.

we show analytically that the leading order of the factor for the improvement is $3+2\sqrt{2}\approx 5.8$, although the factor is larger when including all costs. The numerically determined factor of the improvement over a range of dimensions and numbers of bits is shown in Fig. 9.

B. Synthesis of columns of unitary

We can now use this approach to reduce the cost for the case in which it is only necessary to synthesize half the columns of the unitary. The key to this simplification is that the unitary can be simplified by an initial unitary on dimension $N_{\rm un}/2$ (ignoring the extra qubit) and then afterward using the extra qubit to control one of two unitaries on the remaining qubits. This then diagonalizes the two $N_{\rm un}/2\times N_{\rm un}/2$ blocks of the unitary that we need. That is, we have a decomposition of the unitary as

$$\begin{bmatrix} A & ? \\ B & ? \end{bmatrix} = \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}\begin{bmatrix} D_1 & ? \\ D_2 & ? \end{bmatrix}\begin{bmatrix} V & 0 \\ 0 & V \end{bmatrix}. \tag{30}$$

In this expression, the question marks indicate blocks that we do not need to specify, so A and B indicate half of the columns of the unitary matrix that need to be correctly produced. The blocks D_1 and D_2 are diagonal and U_1, U_2, and V are unitaries. That is, we require

$$A = U_1 D_1 V, \quad B = U_2 D_2 V. \tag{31}$$

The matrices U_1, D_1, and V are easily determined via a singular-value decomposition of A. Then, U_2 and D_2 can be determined from a QR decomposition of $BV^\dagger$. The QR decomposition guarantees that D_2 is upper triangular and the requirement that D_1 and D_2 are blocks of a unitary matrix ensures that D_2 is diagonal.

So, the procedure to apply the unitary that we need is as follows:

(1) Perform a unitary of dimension $N_{\rm un}/2\times N_{\rm un}/2$ on the first $n-1$ qubits.
(2) Use $n-1$ qubits to control rotation on the remaining qubit.
(3) Use that qubit to control unitaries on the $n-1$ qubits.

For the costing of this procedure, the cost of the first step is the same as in the formulas for unitary synthesis above, except replacing $N_{\rm un}$ with $N_{\rm un}/2$ and n with $n-1$.

However, in the MPS preparation, we have a unitary on these $n-1$ qubits first and this unitary can be combined with that one, so that it need not be performed and requires no additional complexity. This principle is illustrated in Fig. 10, where it can be seen that the dimension-$N_{\rm un}/2$ register where U_1 and U_2 are performed is the same as that where V' is applied for the next step.

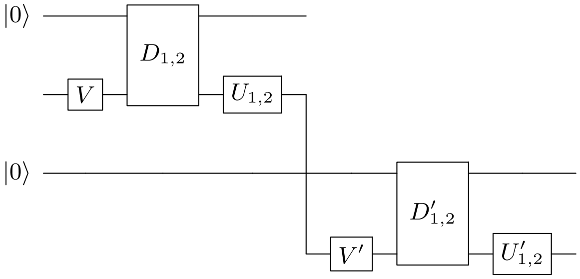

FIG. 10. Two consecutive steps in the MPS preparation, showing how the unitaries required for consecutive steps can be combined. The primes show the unitaries needed for the second step. The input registers with $|0\rangle$ are qubits and the top two outputs are qubits.

Step 2 is just a QROM and rotation, for complexity

$$\left\lceil \frac{N_{\rm un}}{2\Lambda}\right\rceil + \Lambda b - 2. \tag{32}$$

The -2 term is because the rotation is implemented by a controlled addition or subtraction on half the phase, which requires one more bit for the addition into the phase-gradient register. The QROM erasure does not introduce a Toffoli cost here, because the sign fix-up can be incorporated into the unitaries U_1 and U_2. A subtlety in this costing is that the blocks are distinguished by the *least* significant qubit, so we only need consider dimension $N_{\rm un}/2$ for the sub-blocks. For simplicity, we are assuming that $N_{\rm un}$ is even for this discussion.

For the controlled unitaries on dimension $N_{\rm un}/2$, in each layer we have the same size of the QROM as for the operations on the full dimension $N_{\rm un}$, with the only difference being that we now only need $N_{\rm un}/2$ layers rather than $N_{\rm un}$. Also, the increments and decrements are on $n-1$ qubits rather than n, slightly decreasing the complexity. The final phase shifts also have the same cost as before. The complete cost is therefore

$$\left\lceil \frac{N_{\rm un}}{2\Lambda}\right\rceil + \Lambda b - 2 \quad \text{(controlled qubit rotation)} \tag{33}$$

$$+\frac{N_{\rm un}}{2}\left(\left\lceil \frac{N_{\rm un}}{2\Lambda}\right\rceil + 2\Lambda b - 5\right) \quad \text{(phasing layers)} \tag{34}$$

$$+(n-3)(N_{\rm un}/2-1) \quad \text{(increments and decrements)} \tag{35}$$

$$+\left\lceil \frac{N_{\rm un}}{2\Lambda}\right\rceil + 2\Lambda b + \left\lceil \frac{N_{\rm un}}{\Lambda'}\right\rceil + \Lambda' - 6 \quad \text{(final phase shifts).} \tag{36}$$

This gives a similar factor-of-approximately-7 improvement over LKS as for synthesis of the complete unitary.

So far, we have discussed the case in which the dimension on each site is $d=2$ for the MPS. We can solve the case for $d > 2$ by iterating the procedure and the cost is multiplied by $d-1$. To explain the principle, we first explain how to obtain the result for $d = 3$. Then, we can construct a decomposition of the form

$$\begin{bmatrix} A & ? & ? \\ B & ? & ? \\ C & ? & ? \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & U_{11} & U_{12} \\ 0 & U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} A & ? & ? \\ R_{11} & R_{12} & ? \\ 0 & R_{22} & ? \end{bmatrix}. \quad (37)$$

In this expression, A, B, and C are the blocks of the unitary that we wish to construct, the U_{ij} are blocks of a unitary, and the R_{ij} are blocks of an upper-triangular matrix. The block R_{21} must be zero because of the upper-triangular form and so is just given as zero above. In this form, the U_{ij} and R_{ij} are obtained by a QR decomposition of the four blocks of the desired unitary operation including B and C and two unspecified blocks indicated by the question marks. It does not matter that the blocks indicated by question marks are unknown, because they can just be replaced with zeros for the purposes of the QR decomposition. We only need to determine R_{11}, U_{11}, and U_{21}, because the other blocks do not affect A, B, and C, and those blocks are obtained correctly by the QR decomposition with question marks replaced with zeros.

It is now possible to correctly apply the blocks U_{11} and U_{21} of the first unitary by the above procedure for synthesizing half of the columns of a unitary. Similarly, A and R_{11} have orthonormal columns and so can be regarded as half the columns of some unitary, which can also be synthesized by the above scheme. Moreover, the construction of the unitary with blocks U_{11} and U_{21} requires an initial unitary of dimension $N_{\text{un}}/3$. This unitary can be combined with R_{11} and so need not add to the cost of synthesizing U_{11} and U_{21}. Similarly, if this unitary synthesis is part of MPS preparation, the initial dimension-$N_{\text{un}}/3$ unitary for synthesis of A and R_{11} can be combined with other operations and does not add to the cost.

In the general case, we need to construct the correct first χ columns of a matrix of size $N_{\text{un}} = d\chi$. Consider the block consisting of the last $(d-1)\chi$ rows and the first $(d-1)\chi$ columns (equivalent to B and C and the question marks in the above $d = 3$ example). Apply a QR decomposition, to express it in the form of a unitary operation followed by an upper-triangular matrix. Similar to the above example, the question marks may be replaced with zeros in this decomposition, because they only affect blocks that we do not need to specify.

In exactly the same way as in the above example, we need only correctly synthesize the first χ columns of the first operation and the upper-triangular form guarantees that the only nonzero blocks are A and an upper-triangular $\chi \times \chi$ block (which is R_{11} in the above example). These blocks may be synthesized by the above procedure for $d = 2$, where we are synthesizing half the columns of a unitary.

The unitary that we have obtained by the QR decomposition is of size $(d-1)\chi \times (d-1)\chi$ and we only need to correctly reproduce the first χ columns. In the above example, these were U_{11} and U_{21}. Therefore, we have reduced the problem to synthesizing the first χ columns of a $(d-1)\chi \times (d-1)\chi$ unitary, which is the same as the initial problem, with d reduced by 1. We may therefore iterate this procedure to completely reduce the problem to that for $d = 2$. Therefore, we are able to reduce the problem to $d-1$ applications of the scheme for $d = 2$.

There is a very small increase in the cost for $d > 4$. Because the schemes for $d = 2$ are on a subspace, the QROMs need to be controlled, which is a cost accounted for above. However, for $d > 4$ we also need Toffolis to produce the qubit flagging the control for the QROMs. That is only performed once for each application of the $d = 2$ scheme and so is a negligible contribution to the overall cost.

We can also combine the $d = 2$ schemes in a slightly more efficient way. First, note that at each step, there is a controlled unitary performed on two dimension-χ subspaces but only one is used in the next $d = 2$ scheme. Instead of performing both, we can instead just perform one unitary of dimension $\chi \times \chi$ and then perform controlled unitaries on the d subspaces at the end. Then, the costs of $d-1$ controlled U_1, U_2 operations are replaced with $d-2$ unitaries of dimension $\chi \times \chi$, followed by selection between d unitaries of this size.

In either case, the increase in cost with d is linear in d, whereas for the LKS approach the factor is about $\sqrt{d/2}$, so LKS has better scaling with d. For $d = 4$, we find that the improvement over LKS is about a factor of 3.5 and the crossover where LKS is more efficient is for large values of d about 30.

It is also possible to prepare states more efficiently than in the LKS procedure. Instead of using QROM in sequence to output the rotations needed for each successive qubit, the principle is to output the data needed for multiple qubit rotations at once. In addition, it is possible to combine that with the QROM to output the phases for the basis states. This procedure is explained in detail in Appendix C and reduces the cost to about 40% of the original implementation.

Another method is to use interspersed layers of Hadamards and diagonal-phasing operators, and we find that three phasing layers are sufficient. That is, prepare an equal superposition state, apply phases in the computational basis, a layer of Hadamards, and then more phases. The initial two layers of phases and Hadamards produce the correct amplitudes. The phases may be found efficiently by a simple generalization of the Gerchberg-Saxton algorithm; we have tested this with dimension up to 2^{20}. These calculations may be performed quickly; using a laptop, they take only about 5 min to achieve machine precision for this dimension, or about 4 s for dimension 2^{15}. The computation time is approximately linear in the dimension.

The Gerchberg-Saxton algorithm is a method used for phase retrieval (e.g., for creating computer-generated holograms). That is, there are two distributions related by a Fourier transform and given only amplitude information, the goal is to retrieve the phase. To explain the method, let the original function be $f(x)$ and the Fourier transform be $F(k)$. The general principle is that one can start with $|F(k)|$ and take the inverse Fourier transform to yield some function of x. If it has the same amplitude as $f(x)$, then we are done, but otherwise we can use $|f(x)|$ multiplied by the phases from the inverse Fourier transform and take the Fourier transform. This will give a new function of k. Now take the phases from this function multiplied by $|F(k)|$ and take the inverse Fourier transform again. Iteration of this process then converges to phases such that the function and its Fourier transform match. In pseudocode, this algorithm is as follows, where FT and IFT represent the Fourier transform and the inverse Fourier transform:

```
$\varphi(x) = \arg(\mathrm{IFT}(|F(k)|))$
while error above threshold
    $\Phi(k) = \arg(\mathrm{FT}(|f(x)|e^{i\varphi(x)}))$
    $\varphi(x) = \arg(\mathrm{IFT}(|F(k)|e^{i\Phi(k)}))$
end while
```

In practice, this algorithm will be performed with discrete samples and a discrete Fourier transform. It is straightforward to generalize the algorithm for the case in which Hadamards are used instead of a Fourier transform and there are multiple layers. For the case in which there are three layers, let the phases used be φ_j^k for $k = 0, 1, 2$ and let the amplitudes of the desired state be ψ_j. The preparation starts with an equal superposition state $|+\rangle^{\otimes n}$. Then apply phases φ_j^0, Hadamards, phases φ_j^1, Hadamards, and then finally phases φ_j^2. In this generalization of Gerchberg-Saxton, we start with random initialization of φ_j^0 and φ_j^1, because otherwise the Hadamards would cancel.

In the following pseudocode, we will use $e^{i\varphi^k}$ to indicate operators applying $e^{i\varphi_j^k}$ to computational basis state j and we will use arg on quantum states to indicate the vector of phases of the amplitudes in the computational basis. The pseudocode for the three-layer generalization for state preparation is then as follows:

```
while error above threshold
    $\varphi^2 = \arg(|\psi\rangle) - \arg(H^{\otimes n} e^{i\varphi^1} H^{\otimes n} e^{i\varphi^0} |+\rangle^{\otimes n})$
    $\varphi^1 = \arg(H^{\otimes n} e^{-i\varphi^2} |\psi\rangle) - \arg(H^{\otimes n} e^{i\varphi^0} |+\rangle^{\otimes n})$
    $\varphi^0 = \arg(H^{\otimes n} e^{-i\varphi^1} H^{\otimes n} e^{-i\varphi^2} |\psi\rangle)$
    $\varphi^1 = \arg(H^{\otimes n} e^{-i\varphi^2} |\psi\rangle) - \arg(H^{\otimes n} e^{i\varphi^0} |+\rangle^{\otimes n})$
end while
```

That is, we loop back and forth through the layers, choosing the layers of phases so that they are consistent. The Toffoli cost of the preparation using three layers of phases is primarily that of the QROM to output the phases. The cost of each QROM is

$$\left\lceil \frac{N_{\mathrm{un}}}{\Lambda} \right\rceil + (\Lambda - 1)b, \tag{38}$$

the cost of each phase rotation is approximately b, and then the cost of each QROM erasure is

$$\left\lceil \frac{N_{\mathrm{un}}}{\Lambda'} \right\rceil + \Lambda'. \tag{39}$$

That gives a total cost for three layers of

$$3\left(\left\lceil \frac{N_{\mathrm{un}}}{\Lambda} \right\rceil + \Lambda b + \left\lceil \frac{N_{\mathrm{un}}}{\Lambda'} \right\rceil + \Lambda'\right). \tag{40}$$

In contrast, the cost of the procedure described in Appendix C is approximately $3/2$ times that of a single QROM, so its cost is about half in all cases. For this reason, one should always use the method in Appendix C for state preparation. On the other hand, this procedure is easily generalized by increasing the number of layers and applying the operations to an arbitrary input state. With enough layers of phase shifts and Hadamards, it is possible to generate a general unitary operation or correctly implement a limited number of columns of a unitary operation. Note that the state preparation can be considered to be the special case of just reproducing a single column of a unitary operation. The procedure for finding the phases is an obvious generalization of the method described above. One can simply run backward and forward through the layers of phases, choosing them for consistency. The problem is that the complexity of solving for the phases is now approximately scaling as the cube of the dimension, rather than linearly as in the state-preparation case. Although we have been able to solve for phases up to test cases of dimension about 128, the projected solution time for cases of interest in this work would be on the order of years. For this reason, we do not propose this as our main method for unitary synthesis, but if more efficient solution methods are developed, then it may be a viable method for unitary synthesis.

IV. PHASE ESTIMATION WITH WINDOW FUNCTIONS

First, we describe how to perform phase estimation that is optimal for confidence intervals. That is, for a given number of controlled applications of an operator and confidence level, it gives the smallest width of the confidence interval. This problem is related to that of window functions in classical signal-processing theory and optimal confidence intervals are given by the prolate spheroidal window. The analysis of the prolate spheroidal window has been given in Ref. [23] but it is difficult to calculate, so Ref. [25] gives an analysis of the Kaiser window, which gives near-optimal results. The use of the Kaiser window has also been mentioned in Ref. [42, p. 35]. The error for the Kaiser window is in terms of the sinc function and so can be calculated with standard mathematical software. The prolate spheroidal window can also be calculated using specialist mathematical software.

We will provide both asymptotic results for the cost of phase estimation using these window functions, as well as exact expressions using special functions and numerical results calculated from these functions. For the Kaiser window, we provide first-order approximations for the error and cost in Eqs. (45) and (48), respectively, and higher-order approximations for the error in Eq. (49) and the cost in Eq. (50). For the prolate spheroidal window, we provide a higher-order approximation of the error in Eq. (54) and for the cost in Eq. (55). We show that although the Slepian window provides asymptotically improved error, the cost is the same up to leading order.

A. The Kaiser window

First, we summarize the Kaiser window and its asymptotic scaling. The standard form of the Kaiser window is proportional to

$$w(x) = I_0\left(\pi\alpha\sqrt{1-(x/N)^2}\right), \tag{41}$$

for $|x| \le N$, and 0 otherwise. As discussed in Sec. II B 3, the control state used would correspond to samples of this continuous window at discrete points. The window function yields a probability distribution for the error θ proportional to

$$\frac{\sin^2\left(\sqrt{(N\theta)^2-(\pi\alpha)^2}\right)}{(N\theta)^2-(\pi\alpha)^2}; \tag{42}$$

i.e., the square of the Fourier transform of $w(x)$.

A simple approximation for the tail probabilities with the Kaiser window is given in Ref. [25]. The method used there is to first approximate the normalization by approximating the center of the distribution by a Gaussian. In that approximation, the integral over θ then gives

$$\int_{-\infty}^{\infty} \frac{\sinh^2(\pi\alpha)}{\pi^2\alpha^2} e^{-N^2(\pi\alpha\coth(\pi\alpha)-1)\theta^2/(\pi^2\alpha^2)}\, d\theta \approx \frac{e^{2\pi\alpha}}{4N\pi\alpha^{3/2}}. \tag{43}$$

The standard choice for the half-width of the confidence interval is $(\pi/N)\sqrt{1+\alpha^2}$. Later, we will adjust this width slightly to obtain improved performance, taking the half-width to be $(\pi/N)\sqrt{\Delta^2+\alpha^2}$ for an adjustable parameter Δ. With the standard choice of the width, the tail probabilities can be approximated by replacing $\sin^2$ with $1/2$ and integrating from the first zero at $\theta = (\pi/N)\sqrt{1+\alpha^2}$ to give

$$\int_{(\pi/N)\sqrt{1+\alpha^2}}^{\infty} \frac{1}{[N^2\theta^2-(\pi\alpha)^2]}\, d\theta \approx \frac{\ln(2\alpha)}{\pi N\alpha}. \tag{44}$$

Dividing by the approximation for the normalization in Eq. (43) then gives

$$\delta \approx 4\ln(2\alpha)\sqrt{\alpha}\, e^{-2\pi\alpha}. \tag{45}$$

Therefore, to obtain confidence level $1-\delta$ (so that the probability of error outside the range is δ), we take

$$\ln(1/\delta) \approx 2\pi\alpha - \ln[4\ln(2\alpha)\sqrt{\alpha}]. \tag{46}$$

Solving for α then gives the approximation

$$\alpha \approx \frac{1}{2\pi}\ln(1/\delta) + \frac{1}{4\pi}\ln(8\ln(4/\delta)/\pi) + \frac{1}{2\pi}\ln\left[\ln(\ln(4/\delta)/\pi)\right]. \tag{47}$$

The size of the confidence interval is $(\pi/N)\sqrt{1+\alpha^2}$, so if that needs to be ϵ, we should take

$$N = \frac{\pi}{\epsilon}\sqrt{1+\alpha^2} = \frac{1}{2\epsilon}\ln(1/\delta) + \mathcal{O}(\epsilon^{-1}\ln\ln(1/\delta)). \tag{48}$$

The order $\ln\ln$ term for α is larger than the correction term for approximating $\sqrt{1+\alpha^2}$ by α.

In Appendix D, we show how to use the properties of Bessel functions to derive higher-order terms, yielding

$$\delta = 4C_\alpha\sqrt{\alpha}e^{-2\pi\alpha}\left[1-\frac{5}{16\pi\alpha}+\mathcal{O}(\alpha^{-2})\right], \tag{49}$$

where $C_\alpha = \ln(2\alpha) + \mathrm{Ci}(2\pi)$. The expression for N is then

$$N = \frac{\pi}{\epsilon}\sqrt{1+\alpha^2} = \frac{\ln(1/\delta)}{2\epsilon} + \frac{\ln(8\ln(4/\delta)/\pi)}{4\epsilon} + \frac{1}{2\epsilon}\ln\left[\mathrm{Ci}(2\pi)+\ln(\ln(4/\delta)/\pi)\right] + \mathcal{O}\left(\frac{\ln\ln(1/\delta)}{\epsilon\ln(1/\delta)}\right). \tag{50}$$

To this order of approximation, $\sqrt{1+\alpha^2}\approx\alpha$ and the correction is in the order term above.

Alternatively, using $(\pi/N)\sqrt{\Delta^2+\alpha^2}$ for the half-width of the confidence interval gives

$$\delta = 4C_{\alpha,\Delta}\sqrt{\alpha}e^{-2\pi\alpha}\left[1-\frac{5}{16\pi\alpha}+\mathcal{O}(\alpha^{-2})\right], \tag{51}$$

where $C_{\alpha,\Delta} = \ln(2\alpha/\Delta) + \mathrm{Ci}(2\pi\Delta)$. This asymptotic approximation indicates that the asymptotically optimal choice of Δ is 1, so does not yield a distinct expansion for N (for further details, see Appendix D).

B. The prolate spheroidal window

For comparison with the optimal window, the error is given in Ref. [23, Eq. (13)] as

$$\delta = 4\sqrt{\pi c}\, e^{-2c}\left[1 - \frac{3}{32c} + \mathcal{O}(c^{-2})\right]. \tag{52}$$

We obtain a correction term of $7/16c$ rather than $3/32c$ and have verified it through numerical testing. The result in Ref. [23] has been derived from an expression given in Ref. [22] that appears to be incorrect. In the case $n = 0$, Ref. [22, Eq. (4.4)] gives

$$\delta = 4\sqrt{\pi c}\, e^{-2c}\left[1 - \frac{3}{32c} - \frac{389}{2^{11}c^2} + \mathcal{O}(c^{-3})\right]. \tag{53}$$

By repeating the derivation as given in Ref. [22, p. 138], together with some further analysis, we obtain

$$\delta = 4\sqrt{\pi c}\, e^{-2c}\left[1 - \frac{7}{16c} - \frac{91}{2^9c^2} - \frac{2657}{2^{13}c^3} + \mathcal{O}(c^{-4})\right] \tag{54}$$

(for the details of this derivation, see Appendix E).

In this expression, the factor of $\ln c$ that has been obtained for the Kaiser window error has been eliminated, so this error has asymptotically better scaling. According to the above analysis for the asymptotic expansion for N, we would obtain

$$N = \frac{c}{\epsilon} = \frac{\ln(1/\delta)}{2\epsilon} + \frac{\ln(8\pi\,\ln(4\sqrt{\pi}/\delta))}{4\epsilon} + \mathcal{O}\left(\frac{\ln\ln(1/\delta)}{\epsilon\,\ln(1/\delta)}\right). \tag{55}$$

That is, despite the optimal window giving asymptotically smaller error, the cost N is reduced by only removing the third term in Eq. (50). That term is triple logarithmic in $1/\delta$, meaning that the performance is only marginally improved.

The optimal window is the angular spheroidal function of the first kind $\mathrm{PS}_{0,0}(c,z)$ for $z \in [-1,1]$. Integrating then gives (for an explanation of why this integral is used, see Appendix E)

$$\begin{aligned}(1-\delta)\mathrm{PS}_{0,0}(c,0) &= \frac{c}{\pi}\int_{-1}^{1} \mathrm{sinc}(cz)\,\mathrm{PS}_{0,0}(c,z)\,dz \\ &= \frac{2c}{\pi}\,\mathrm{PS}_{0,0}(c,0)\,[S^1_{0,0}(c,1)]^2,\end{aligned} \tag{56}$$

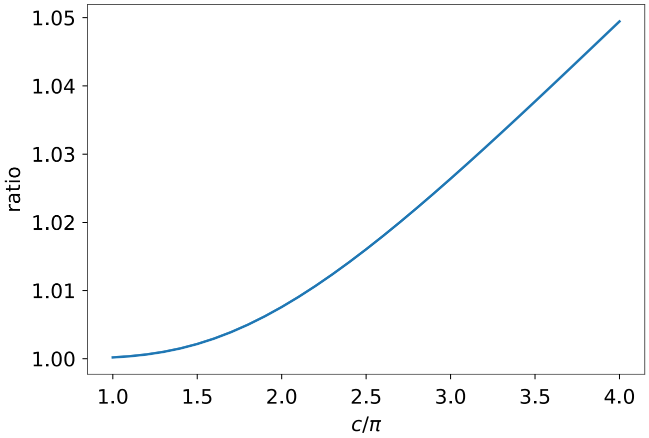

FIG. 11. The ratio of the error for the Kaiser window to that for the optimal Slepian window as a function of c. For the Kaiser window, the value of Δ is optimized to minimize the error.

so

$$\delta = 1 - \frac{2c}{\pi}\,[S^1_{0,0}(c,1)]^2, \tag{57}$$

where the function $S^1_{0,0}(c,1)$ is the radial spheroidal function of the first kind. Thus it is possible to compute the error in terms of special functions. Various methods are discussed in Appendix F.

In comparison, the value of N for root-mean-square (rms) error ϵ is [29]

$$N \approx \frac{\pi}{2\epsilon}. \tag{58}$$

That is, to leading order, the expression for N for the confidence interval replaces π with $\ln(1/\delta)$. For, e.g., a 95%

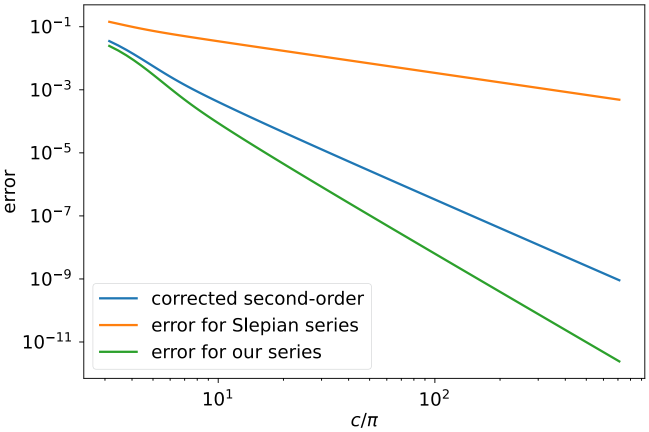

FIG. 12. The differences between the various asymptotic series and the exact error, divided by $4\sqrt{\pi c}\, e^{-2c}$. Results for the series of Slepian [see Eq. (53)] are shown in orange, our series in Eq. (54) is given in green, and results with just the corrected second-order term are given in blue.

confidence interval, $\ln(1/\delta)$ is less than π but calculation using the exact expression in Eq. (57) shows that the complexity is about 63% larger than for achieving rms error ϵ. For a 90% confidence interval, the complexity is only about 35% larger than for achieving rms error ϵ.

Next, we numerically compare the error outside the confidence interval for the Kaiser window versus that for the optimal window. According to the above asymptotic analysis for the error, the ratio of the errors should increase with c and we find that this occurs even optimizing for Δ. We show the ratio in Fig. 11 and it is very close to 1 for $c = \pi$, and increases to be about 5% larger for $c = 4\pi$.

We now test the series of Slepian compared to the one we have given in Eq. (54). In Fig. 12, we take the difference between the exact error and the asymptotic approximations divided by $4\sqrt{\pi c}\, e^{-2c}$. It can be seen that our expression is far more accurate. A significant improvement is provided by correcting the second-order term, with further accuracy provided by the third-order term.

V. THE SAMPLING METHOD

A. Asymptotic approximations

When we are performing phase estimation on a block-encoded Hamiltonian, the Hamiltonian is encoded as H/λ and the eigenvalues of the corresponding qubitized operator are $\pm e^{\pm i \arcsin(E_k/\lambda)}$ for eigenvalues E_k of H. This means that restricting the error to $\leq \epsilon/\lambda$ for the phase estimation means that the error in E_k is no more than ϵ. In the following, we denote the ground-state energy by E_0, so we aim to have an estimate in the region $[E_0 - \epsilon, E_0 + \epsilon]$. Then, the number of queries for phase estimation with either the Kaiser or Slepian window becomes, to leading order,

$$\frac{\lambda}{2\epsilon} \ln(1/\delta) + \mathcal{O}((\lambda/\epsilon) \ln\ln(1/\delta)). \tag{59}$$

If we take the amplitude for the ground state to be γ and $p = \gamma^2$, then the probability of failing to have the ground state once in sampling the energies n times is $(1-p)^n \approx e^{-pn}$. If we want the probability of failing to be less than q, we would need to take

$$n \approx (1/p)\ln(1/q). \tag{60}$$

If we are taking the minimum result for the eigenvalue, then the probability of the error in that measurement result being outside the ϵ interval is now bounded by $n\delta$. This suggests that we should divide δ by n in order to obtain an appropriately bounded error probability. However, the importance of the errors is asymmetric. If the measurement corresponds to an excited eigenstate but the measurement error yields an estimate of the energy that is exceptionally low, then that estimate could be taken to be the smallest out of all samples, yielding an inaccurate result. On the other hand, if the measurement error gives an estimate of the energy that is exceptionally high, then it is far less likely to be taken as the smallest estimate out of all samples. This means that the measurement errors in each direction need to be quantified differently.

To do this, let the confidence level for each individual estimate of the phase be $1 - \delta$, so that the probability of error on *one side* is $\delta/2$. That is because we are choosing a measurement technique where the error distribution is symmetric. Given squared overlap with the ground state p, then for each estimate there is probability $1-p$ of it corresponding to the wrong eigenstate, and $p\delta/2$ of it corresponding to the correct eigenstate but the true eigenvalue being below the confidence interval on the lower side (so that the estimate is too large). We will group these possibilities together as a "high" error, which has total probability $1 - p(1-\delta/2)$. The probability of there being high errors on all n samples is then $[1 - p(1-\delta/2)]^n$.

The probability of a "low" error outside the confidence interval is $\delta/2$ for any individual measurement, so the probability of any low error occurring in the n samples is $1 - (1-\delta/2)^n$. Note that it is possible for measurements corresponding to excited states to give a low error that is still not below the desired confidence interval for the measurement of the ground state. The above estimate does not take that into account and so is a fairly loose upper bound on the error. Our total upper bound on the error is then the left-hand side of

$$[1 - p(1-\delta/2)]^n + 1 - (1-\delta/2)^n = q. \tag{61}$$

Given an allowable error q (confidence level $1 - q$ in the final estimate), we can then solve for δ.

The overall complexity will then be approximately

$$\frac{n\lambda}{2\epsilon} \ln(1/\delta), \tag{62}$$

with δ as chosen by solving Eq. (61) and n chosen as at least $(1/p)\ln(1/q)$, resulting in a total complexity

$$\frac{\lambda}{2p\epsilon} \ln(1/q)\ln(1/\delta). \tag{63}$$

For any specific example, we can tweak the value of n in order to minimize the overall complexity. As an example, let us consider $\gamma = 0.1$ so $p = 0.01$ and require a 95% confidence interval so that $q = 0.05$. In this case, we have the factor $n\ln(1/\delta)/2$ as a function of n shown in Fig. 13. That is, this is the factor in the complexity that is multiplied by λ/ϵ to give the overall complexity. Here, it can be seen that the optimal value of n is 325, which is moderately above $(1/p)\ln(1/q) \approx 300$. Note that this choice of the optimal value of n is independent of λ and ϵ.

We can further develop asymptotic approximations for the solutions in the case of small p and q to estimate the

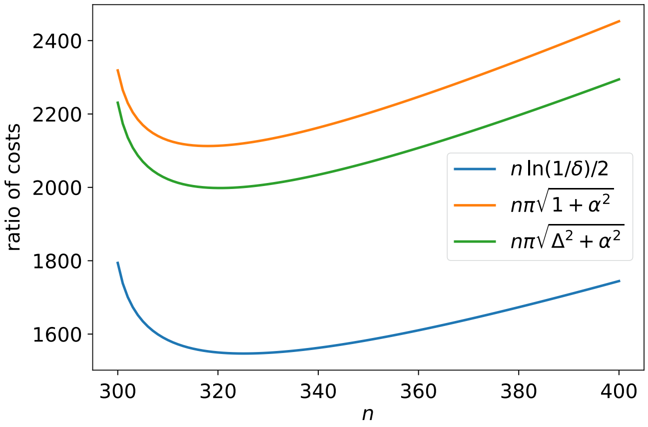

FIG. 13. The factor in the complexity as a function of n for $p = 0.01$ and $q = 0.05$, with δ obtained by solving Eq. (61). The blue curve is the approximation $n\ln(1/\delta)/2$, the orange curve is the factor $n\pi\sqrt{1+\alpha^2}$ for the Kaiser window, and the green curve is $n\pi\sqrt{\Delta^2+\alpha^2}$ with $\Delta = 0.3239$.

optimal values of n and δ. First, we linearize Eq. (61) to give

$$(1-p)^n + \frac{\delta n}{2}[1+(1-p)^{n-1}p] \approx q. \tag{64}$$

Solving for δ then gives

$$\delta \approx \frac{2}{n}\frac{q-(1-p)^n}{1+(1-p)^{n-1}p}. \tag{65}$$

If we expand δ to second order, then we obtain the next term in the expansion,

$$\delta \approx \frac{2}{n}\frac{q-(1-p)^n}{1+(1-p)^{n-1}p} + \frac{(n-1)[1-(1-p)^{n-2}]}{n^2[1+(1-p)^{n-1}p]^3}(q-(1-p)^n)^2. \tag{66}$$

Up to terms that are smaller by factors of q or p, we can give

$$\frac{1}{\delta} \approx \frac{n}{2[q-(1-p)^n]} = \frac{n}{2(q-e^{-n\rho})}, \tag{67}$$

where $\rho = -\ln(1-p) \approx p$.

Next, given the solution for δ, the task is to choose n to minimize $n\ln(1/\delta)$. Let us first consider this expression with only the leading order in the solution for δ, which is (ignoring the factor of 2 for the moment)

$$n\ln\left(\frac{n}{2(q-e^{-n\rho})}\right). \tag{68}$$

The higher-order terms in the solution for δ will result in corrections that are at least a factor of q smaller and so can be safely ignored in the following analysis where the terms in the expansions are significantly larger. Taking the derivative with respect to n gives

$$1 - \frac{e^{-n\rho}n\rho}{q-e^{-n\rho}} + \ln\left(\frac{n}{2(q-e^{-n\rho})}\right). \tag{69}$$

Setting this to zero and rearranging gives

$$h = g - \ln g, \tag{70}$$

where

$$g = \frac{e^{-n\rho}n\rho}{q-e^{-n\rho}}, \tag{71}$$

$$h = 1 + n\rho - \ln(2\rho). \tag{72}$$

We can then solve for g as a series in h as

$$g \approx h + \ln h + \frac{\log h}{h} + \frac{2\ln h - \ln^2 h}{2h^2} + \frac{6\ln h - 9\ln^2 h + 2\ln^3 h}{6h^3} + \mathcal{O}\left(\frac{\ln^4 h}{h^4}\right). \tag{73}$$

Now q can be given in terms of g as

$$q = e^{-n\rho} + \frac{e^{-np}n\rho}{g}. \tag{74}$$

We can then rewrite this as

$$\ln(2/q) = n\rho - \ln\left(\frac{1}{2} + \frac{n\rho}{2g}\right). \tag{75}$$

By substituting a series for g in terms of h and inverting to obtain a series for $n\rho$, we obtain

$$n\rho = \ln(2/q) - \frac{Q+1}{2\ln(2/q)}\left(1 - \frac{3Q-1}{4\ln(2/q)} - \frac{5+4Q-7Q^2}{12\ln^2(2/q)}\right) + \mathcal{O}\left(\frac{Q^4}{\ln^4(2/q)}\right), \tag{76}$$

$$Q := \ln(\ln(2/q)/2\rho). \tag{77}$$

The difficulty with using this expression is that the successive terms are not smaller if $Q \sim \ln(2/q)$. That will be the case if $q > p$. This expression does give accurate results if q is small compared to p.

A more accurate approximation may be given by

$$n\rho = \ln(2/q) - \frac{Q+1}{2\ln(2/q)}\left(1 + \frac{3Q-1}{4\ln(2/q)} + \frac{23-2Q-Q^2)}{48\ln^2(2/q)}\right)^{-1} + \mathcal{O}\left(\frac{Q^4}{\ln^4(2/q)}\right). \tag{78}$$

Using this expression with $p = 0.01$ and $q = 0.05$ gives $n \approx 325$, close to the true value determined numerically [as above, using $(n\lambda/2\epsilon)\ln(1/\delta)$ with δ found by solving Eq. (61); see Fig. 13]. This series then gives the leading-order terms for $n\ln(1/\delta)/2$ as

$$\begin{aligned}\frac{n}{2}\ln(1/\delta) &= \frac{1}{2\rho}\ln(2/q)\ln\left[\frac{\ln(2/q)}{\rho q}\right] - \frac{(1+Q)^2}{8\rho\ln(2/q)}\\ &\quad + \frac{(Q-1)(1+Q)^2}{16\rho\ln^2(2/q)}\\ &\quad + \frac{(1+Q)^2(5+10Q-7Q^2)}{192\rho\ln^3(2/q)}\\ &\quad + \mathcal{O}\left(\frac{Q^5}{\rho\ln^4(2/q)}\right). \qquad (79)\end{aligned}$$

For this example, the estimated value using the terms shown is 1525, within 1.5% of the true value of 1547. Using only the leading term gives 1634, still within 6% of the correct value. Note also that using only the leading-order term for δ in estimating the cost affects the result by very little, less than 0.06% for this example. This justifies omitting higher-order terms for δ in the above analysis. In contrast, using the approximation $\rho \approx p$ affects the result by about 0.6%, so it is useful to make that correction.

B. Real cost for window functions

Next, we consider the actual cost for the Kaiser and prolate spheroidal window functions, rather than just the asymptotic approximation. We can explicitly integrate the probability distribution for the error in the Kaiser window. That gives a factor of $n\pi\sqrt{1+\alpha^2}$ rather than $n\ln(1/\delta)/2$, with α solved to give error δ. The resulting factor is also shown in Fig. 13. The factor has been increased by about 37% above that for the asymptotic approximation given above, from about 1547 to 2113. It is also possible to adjust the cutoff used in the Kaiser window to $\sqrt{\Delta^2+\alpha^2}$ for $\Delta \neq 1$, which can give improved results. The result for $\Delta = 0.3239$ gives approximately the optimal result and is shown in Fig. 13. Now the constant factor is only increased by about 29%, to 1998. The accuracy of the asymptotic approximation is better for smaller q, so $q = 0.01$ results in the cost being about 25% above the asymptotic value, but it would take very small q for the approximation to be accurate.

The prolate spheroidal window further reduces the cost but only by a very small amount. The curve is indistinguishable from that for optimized Δ in Fig. 13 and so is not shown separately. In this case, we find the constant factor is reduced very slightly to 1997 with $n = 320$, a reduction of only 0.06%. This shows that it is possible to accurately approximate results for the optimal window using the Kaiser window and adjusting Δ.

C. Contribution to cost from excited states

We can give a tighter bound on the cost by more accurately accounting for the contribution to the error from excited states. In practice, there will be a small contribution to the probability of low estimates of the ground-state energy E_0 from excited states. When they are distant from E_0, there is very little probability of them yielding an estimate of the eigenvalue below the desired confidence interval, and if they are close to E_0, they will also increase the probability of having a result within the desired confidence interval.

To gauge the effect, let us consider just a single excited state with energy $\beta\epsilon$ above the ground state; i.e., when $\beta < 1$, the energy is actually within the desired confidence interval for estimating the ground state. It can be shown that the error is maximized for a single excited state, so our analysis for a single excited state is sufficient to bound the worst case for any spectrum of excited states (see Appendix G). We denote by δ_1 the probability for the estimate above the confidence interval for the excited state and by δ_2 that for the estimate below the desired confidence interval. It will be expected that $\delta < \delta_1 + \delta_2$, and in typical cases in which the eigenvalue is *much* higher than the ground value, there will be $\delta_1 \sim 1$ and $\delta_2 \sim 0$.

We can then replace $1-p$ for the probability of error due to the incorrect eigenstate being obtained with $(1-p)\delta_1$; i.e., the incorrect eigenstate is obtained *and* the estimate is too high. Then, the probability of all n samples being too high is

$$[(1-p)\delta_1 + p\delta/2)]^n. \qquad (80)$$

Then, the probability of a single sample being too low is $p\delta/2 + (1-p)\delta_2$, so the probability of *any* of the samples being too low is $1 - \{1 - [p\delta/2 + (1-p)\delta_2]\}^n$. Adding together these two probabilities of error then gives

$$\begin{aligned}P_{\rm err} &= [p\delta/2 + (1-p)\delta_1]^n\\ &\quad + 1 - \{1 - [p\delta/2 + (1-p)\delta_2]\}^n. \qquad (81)\end{aligned}$$

This expression can be expected to be smaller than that given before, because $p\delta/2 + (1-p)\delta_1$ will be smaller than $p\delta/2 + (1-p)$.

In numerical testing, we choose values of α, Δ, and n *without* knowing β (the excited-state energy), so the above error needs to be no greater than q for all choices of β. That is, for our choices of α, Δ, and n, we need

$$\max_{\beta\geq 0} P_{\rm err}(\alpha,\Delta,n,\beta) \leq q. \qquad (82)$$

We choose α, Δ, and n to minimize the cost $n\pi\sqrt{\Delta^2+\alpha^2}$ given this constraint. This will result in the total number of queries to the qubitized walk operator being

$$\frac{n\pi\lambda}{\epsilon}\sqrt{\Delta^2+\alpha^2}. \tag{83}$$

The value of P_{err} is determined from α, Δ, and n by calculating δ, δ_1, and δ_2 using

$$\delta = 2\int_{(\pi/N)\sqrt{\Delta^2+\alpha^2}}^{\infty} \mathrm{Pr}_{\mathrm{Kaiser}}(\theta)\,d\theta, \tag{84}$$

$$\delta_1 = \int_{(1-\beta)(\pi/N)\sqrt{\Delta^2+\alpha^2}}^{\infty} \mathrm{Pr}_{\mathrm{Kaiser}}(\theta)\,d\theta, \tag{85}$$

$$\delta_2 = \int_{-\infty}^{-(1+\beta)(\pi/N)\sqrt{\Delta^2+\alpha^2}} \mathrm{Pr}_{\mathrm{Kaiser}}(\theta)\,d\theta. \tag{86}$$

The probability distribution for the errors $\mathrm{Pr}_{\mathrm{Kaiser}}(\theta)$ is given as in Eq. (42). That is, δ_1 is the probability for an estimate above the confidence interval and the factor of $(1-\beta)$ reflects the fact that the center of the distribution is shifted for the excited state. Similarly, there is a shift in the bound of the integral for δ_2. In practice, we are interested in these results for the limit of large N, so the integrals are taken to infinity and θ is rescaled by N in the numerical calculation.

We find that in most cases, the value of P_{err} as a function of β has two peaks, one for $\beta=0$ and another for β a little above 2, as shown in Fig. 14. In such cases, we find that the best results are when both peaks satisfy $P_{\mathrm{err}}=q$. The numerical optimization can then proceed by an iterative process as follows.

```
choose $n$ and initial values of $\alpha$ and $\Delta$
while change in $\alpha$, $\Delta$,and $\beta_{\max}$ above convergence threshold
    $\beta_{\max} = \mathrm{argmax}_\beta P_{\mathrm{err}}(\alpha,\Delta,n,\beta)$
    solve $P_{\mathrm{err}}(\alpha,\Delta,n,\beta_{\max}) = q$, $P_{\mathrm{err}}(\alpha,\Delta,n,0) = q$ for $\alpha$ and $\Delta$
end while
```

That is, a choice of β is made that maximizes P_{err} and then α and Δ are found to solve the two simultaneous equations $P_{\mathrm{err}}=q$ at both $\beta=0$ and $\beta_{\max}$. Then $\beta_{\max}$ needs to be solved for again with the updated values of α, Δ and this process is repeated until the values converge. This iterative solution is performed for a range of values of n to find the one that yields the minimum value of $n\pi\sqrt{\Delta^2+\alpha^2}$.

We find that for $p=0.01$ and $q=0.05$, we can choose $\alpha=1.70116$, $\Delta=0.074476$, and $n=309$. These values minimize the constant factor while keeping the error below q (see Fig. 14). The constant factor $n\pi\sqrt{\Delta^2+\alpha^2}$ is approximately 1673, so it is significantly smaller than the results not taking this factor into account, and similar to the result with the very simple asymptotic approximation that we gave first. In testing with multiple excited-state eigenvalues, the results are no worse than with just one excited-state eigenvalue, as predicted.

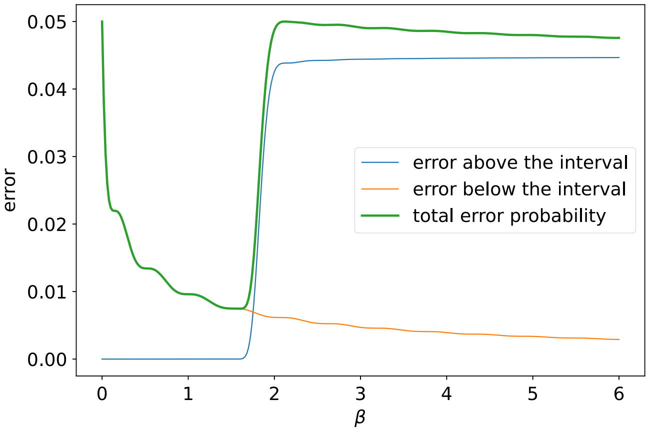

FIG. 14. The error according to Eq. (81) as a function of β with $p=0.01$, $\alpha=1.70116$, $\Delta=0.074476$, and $n=309$. The probability of an error above the interval is given by the first term in Eq. (81), $[p\delta/2+(1-p)\delta_1]^n$, and the probability of an error below the interval is given by the remaining expression, $1-\{1-[p\delta/2+(1-p)\delta_2]\}^n$. The total error probability P_{err} does not exceed $q=0.05$.

An interesting feature of the results is that the error is larger for larger β (around 2 and above) and then is smaller for small β, but with a spike near $\beta=0$. The reason for this is that as β is reduced below 2, there is greater overlap between the distribution of measurement results obtained for the excited state and the desired interval $[E_0-\epsilon, E_0+\epsilon]$ for estimates of E_0. That is, both of these are of half-width ϵ, so reducing the gap below $\beta\epsilon$ means that the two regions overlap. This means that measurements of energy for the excited state being within the desired region are reducing the overall error probability. It reduces quite quickly, because we are taking the minimum of the measurement results, and with many samples there is a high probability of a measurement result on the excited state being around ϵ below its true energy.

But then there is a separate spike for the error probability as the gap closes to zero. This is because the measurements of energy on the excited state now have a significant probability of being *below* the desired confidence interval. To bound the error by q, we need to ensure that the error for both $\beta=0$ and β around 2 is bound by q. We find that the best results (in terms of the lowest constant factor) tend to be those where the error reaches q for both. This is also illustrated in Fig. 14. It shows that the probability of errors more than ϵ above the true ground state rapidly approaches zero as β is reduced below 2, and the probability of errors that are too low gradually increases. This behavior is for small p, whereas for p close to 1, the error is less than q for $\beta=0$.

The method of solution for the prolate spheroidal wave function is a little simpler, because there is one fewer parameter to optimize, with α and Δ replaced by c. In this case, rather than the optimal solution having $P_{\rm err} = q$ for both $\beta = 0$ and a second value $\beta = \beta_{\max}$, there is a single maximum. Then, instead of solving α and Δ for two simultaneous equations as for the Kaiser window, we simply solve for c such that $P_{\rm err} = q$ at the single maximum; i.e., for each n, find c that solves

$$\max_{\beta} P_{\rm err}(c, n, \beta) = q \tag{87}$$

and then choose the value of n that gives the minimum of nc with this solution.

It turns out that for the smaller values of p, use of the prolate spheroidal wave function gives *worse* results than the Kaiser window. This is surprising, since the prolate spheroidal wave function is optimal for just a single phase measurement. For this example, we find that the performance is optimized for $n = 318$ and $\alpha = 1.71229$, where the constant factor is about 1711, or 2% *worse* than for the Kaiser window. To see why, consider the same parameters as for the Kaiser window, with $\beta = 2.12103$. We then find that most of the probabilities are the same but δ_2 is significantly lower for the Kaiser window. It is about 0.0000184942, versus $\delta_2 \approx 0.0000336569$ for the prolate spheroidal window, which is about 82% larger.

This is very significant in this case with small p, because it contributes to the chance of the excited state yielding an estimate of the energy that is too *low*. In this case, with the prolate spheroidal wave function, the probability of an estimate that is too low with the excited state is only about 1/5 that for the ground state. Since the excited state is about 100 times more common in this example, that is a significant contributor to the error. In contrast, the Kaiser window more strongly suppresses the tails, so estimates that are too low coming from the excited state are less of a problem. Again, the behavior is different for p close to 1, where the prolate spheroidal window provides better performance. That is because there are few repetitions, so the effect of the contribution to the error from δ_2 being amplified by repetitions is less.

VI. THE BINARY-SEARCH APPROACH

In the previous sections, we have focused on estimating the ground-state energy by directly reading off the energy samples from the quantum phase-estimation algorithm and taking the minimum among these samples. As shown in Eq. (60), the number of samples needed scales as $\widetilde{\mathcal{O}}(1/p)$. This is, however, not the optimal scaling. As shown in Ref. [26, Theorem 8], one can improve the dependence on p to $\widetilde{\mathcal{O}}(1/\sqrt{p})$. In this section, we will propose a method based on Ref. [26] and incorporating the window functions to reduce the resources needed in practical implementation. We also show numerically that this method is beneficial when $p < 10^{-3}$. Henceforth, we will refer to this method as *the binary-search approach*.

A. From the fuzzy-bisection problem to amplitude estimation

From the previous sections, we can see that in a single run of the QPE algorithm, in order to ensure that the phase error is below η with probability at least $1 - \delta$, the query complexity is given by a function $Q(\eta, \delta)$, which in the case of the Kaiser or Slepian window scales as

$$Q(\eta, \delta) \approx \frac{1}{2\eta} \ln(1/\delta) \tag{88}$$

for small η and δ, according to Eq. (50). In this notation, we use η for the phase-estimation error to distinguish it from the error ϵ for estimation of eigenvalues. We will use this expression in our computation of the asymptotic complexity of the binary-search approach but in numerics we will numerically compute the function $Q(\eta, \delta)$ in a more accurate manner.

In the binary-search process in Ref. [26], we gradually shrink an interval $[\lambda_L, \lambda_R]$ in which the ground-state energy is located. In particular, in the last search step, in order to estimate λ_0 to ϵ precision, we have λ_L and λ_R such that $\lambda_L \le \lambda_0 \le \lambda_R$, with $\lambda_R - \lambda_L = 3\epsilon$. We want to distinguish between two cases:

$$\lambda_0 > \frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R \quad \text{or} \quad \lambda_0 < \frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R. \tag{89}$$

If $\frac{2}{3}\lambda_L + \frac{1}{3}\lambda_R \le \lambda_0 \le \frac{1}{3}\lambda_L + \frac{2}{3}\lambda_R$, then we can output anything. Solving the problem of distinguishing the cases will give us an interval $[\lambda_L^{\rm new}, \lambda_R^{\rm new}] \ni \lambda_0$ of size 2ϵ. That interval corresponds to an estimate of λ_0 up to ϵ error. We call this problem the *fuzzy-bisection problem*.

Because arccos is monotonically decreasing in $[-1, 1]$, we only need to distinguish between

$$\begin{aligned} &\arccos\left(\frac{\lambda_0}{\lambda}\right) < \arccos\left(\frac{\lambda_L + 2\lambda_R}{3\lambda}\right) \quad \text{or} \\ &\arccos\left(\frac{\lambda_0}{\lambda}\right) > \arccos\left(\frac{2\lambda_L + \lambda_R}{3\lambda}\right). \end{aligned} \tag{90}$$

We then perform QPE with a Kaiser or Slepian window on W. We denote the phase output by $\hat{\phi} \in [-\pi, \pi]$. We want the phase error to be at most

$$\begin{aligned} \frac{\epsilon}{2\lambda} &= \frac{\lambda_R - \lambda_L}{6\lambda} \\ &\le \frac{1}{2}\left(\arccos\left(\frac{2\lambda_L + \lambda_R}{3\lambda}\right) - \arccos\left(\frac{\lambda_L + 2\lambda_R}{3\lambda}\right)\right), \end{aligned} \tag{91}$$

with probability at least $1 - \delta_1$.

If $\arccos(\lambda_0/\lambda) < \arccos((\lambda_L + 2\lambda_R)/3\lambda)$, then all eigenvalues $e^{\pm i \arccos(\lambda_k/\lambda)}$ of W satisfy $\arccos(\lambda_k/\lambda) < \arccos((\lambda_L + 2\lambda_R)/3\lambda)$. Therefore, if

$$|\hat{\phi}| > \bar{\phi} := \frac{1}{2}\left(\arccos\left(\frac{2\lambda_L + \lambda_R}{3\lambda}\right) + \arccos\left(\frac{\lambda_L + 2\lambda_R}{3\lambda}\right)\right), \tag{92}$$

then a phase error larger than

$$\bar{\phi} - \arccos\left(\frac{\lambda_L + 2\lambda_R}{3\lambda}\right) \geq \frac{\epsilon}{2\lambda} \tag{93}$$

must have occurred. This event has probability at most δ_1. Consequently,

$$\Pr[|\hat{\phi}| > \bar{\phi}] \leq \delta_1. \tag{94}$$

If $\arccos(\lambda_0/\lambda) > \arccos((2\lambda_L + \lambda_R)/3\lambda)$, suppose that $\hat{\phi}$ comes from eigenvalues $e^{\pm i \arccos(\lambda_0/\lambda)}$. Then, it will satisfy $|\hat{\phi}| > \bar{\phi}$ with probability at least $1 - \epsilon_1$. This is because

$$\arccos\left(\frac{2\lambda_L + \lambda_R}{3\lambda}\right) - \bar{\phi} \geq \frac{\epsilon}{2\lambda}. \tag{95}$$

The phase $\hat{\phi}$ comes from eigenvalues $e^{\pm i \arccos(\lambda_0/\lambda)}$ with probability at least p. Therefore,

$$\Pr[|\hat{\phi}| > \bar{\phi}] \geq p(1 - \delta_1). \tag{96}$$

We therefore only need to distinguish between two cases in Eqs. (94) and (96). This is an amplitude-estimation problem. We define

$$\gamma_1 = \sqrt{\delta_1}, \quad \gamma_2 = \sqrt{p(1 - \delta_1)}, \tag{97}$$

and aim to distinguish the cases in which the amplitude corresponding to $|\hat{\phi}| > \bar{\phi}$ is at most γ_1 or at least γ_2. We choose the parameters so that $\gamma_1 < \gamma_2$. To generate a single sample of $\hat{\phi}$, we need to run a coherent QPE circuit that involves

$$d_1 = Q(\eta, \delta_1) \tag{98}$$

queries to W, where $\eta = \epsilon/(2\lambda)$ is the allowed phase error.

B. Amplitude estimation with the Kaiser window

From the previous section, we can see that to solve the fuzzy-bisection problem, it suffices to estimate

$$A = \sqrt{\Pr[|\hat{\phi}| > \bar{\phi}]}. \tag{99}$$

We will do so using amplitude amplification. If this amplitude can be estimated with error at most $(\gamma_2 - \gamma_1)/2$, then we will be able to distinguish between $A > \gamma_2$ and $A < \gamma_1$. In amplitude estimation, we construct a walk operator $\mathcal{W}$ using the QPE circuit (two applications of it), such that

$$\mathcal{W}\,|\Phi^{\pm}\rangle = e^{\pm i 2 \arcsin(A)}\,|\Phi^{\pm}\rangle \tag{100}$$

and $|\Phi\rangle = \frac{1}{2}(|\Phi^+\rangle + |\Phi^-\rangle)$ can be prepared using the QPE circuit.

With this walk operator $\mathcal{W}$, we can then run QPE with the Kaiser or Slepian window to estimate $\arcsin(A)$. We only need to estimate A to precision $(\gamma_2 - \gamma_1)/2$, which means that it suffices to estimate the phase $2\arcsin(A)$ to precision $\gamma_2 - \gamma_1$. To do this with probability at least $1 - \delta_2$ requires running $\mathcal{W}$ d_2 times, where

$$d_2 = Q(\gamma_2 - \gamma_1, \delta_2). \tag{101}$$

For the last search step, we need to use W for a total of

$$d_1(2d_2 + 1) = Q\left(\frac{\epsilon}{2\lambda}, \delta_1\right)(2Q(\gamma_2 - \gamma_1, \delta_2) + 1) \tag{102}$$

times, which in the context of the Kaiser window, and using Eq. (59), is

$$d_1(2d_2 + 1) \approx 2d_1 d_2 \approx \frac{\lambda}{(\gamma_2 - \gamma_1)\epsilon}\ln(1/\delta_1)\ln(1/\delta_2) \tag{103}$$

times, up to the leading order. Note that in $2d_2 + 1$, the $+1$ comes from preparing the initial state $\mathcal{W}\,|0\rangle$ for amplitude estimation (see Ref. [44, Fig. 1]). The number of times we need to use U_{init} is $2d_2 + 1$, which in the context of the Kaiser window is

$$2d_2 + 1 \approx 2d_2 = \frac{1}{(\gamma_2 - \gamma_1)}\ln(1/\delta_2). \tag{104}$$

The value of δ_1 can be chosen to minimize the cost as follows. Using the expressions $\gamma_1 = \sqrt{\delta_1}$ and $\gamma_2 = \sqrt{p(1 - \delta_1)}$, we have a factor in the complexity

$$\frac{\ln(1/\delta_1)}{\sqrt{p(1 - \delta_1)} - \sqrt{\delta_1}}. \tag{105}$$

Approximating $\sqrt{1 - \delta_1} \approx 1$ and using $x = \sqrt{\delta_1}$, this factor is approximately proportional to

$$\frac{\ln(1/x)}{\sqrt{p} - x}. \tag{106}$$

Taking the derivative with respect to x then yields

$$\frac{-1/x}{\sqrt{p} - x} + \frac{\ln(1/x)}{(\sqrt{p} - x)^2}. \tag{107}$$

For this to be zero, we should have

$$x = \frac{\sqrt{p} - x}{\ln(1/x)}. \tag{108}$$

Starting with $x = \sqrt{p}/\ln(1/\gamma)$ and then iterating $x \mapsto (\sqrt{p} - x)/\ln(1/x)$ quickly gives the solution.

C. Query complexity of all search steps

Previously, we have focused on the cost of the last search step. Here, we account for the costs in all search steps: the binary search terminates in $L = \lceil \log_{3/2}(\lambda/\epsilon) \rceil$ steps. If we want a final success probability of q, we need $\delta_2 = q/L$. For the number of queries to W, we observe that each search step requires $2/3$ of the resolution of the next step and therefore the query complexity is also $2/3$ of that of the last step. Therefore, the total number of queries to W is

$$\begin{aligned} &Q\left(\frac{\epsilon}{2\lambda}, \delta_1\right)(2Q(\gamma_2 - \gamma_1, \delta_2) + 1)\left(1 + \frac{2}{3} + \left(\frac{2}{3}\right)^2 + \cdots\right) \\ &\quad \le 3Q\left(\frac{\epsilon}{2\lambda}, \delta_1\right)(2Q(\gamma_2 - \gamma_1, \delta_2) + 1). \end{aligned} \tag{109}$$

For the Kaiser window, the total number of queries needed is then

$$\frac{3\lambda}{(\gamma_2 - \gamma_1)\epsilon} \ln(1/\delta_1) \ln(1/\delta_2). \tag{110}$$

The total number of queries to U_{init} is the same for each search step. Therefore, it is

$$\frac{L}{\gamma_2 - \gamma_1} \ln(1/\delta_2). \tag{111}$$

We recall that

$$L \approx \log_{3/2}(\lambda/\epsilon), \quad \delta_2 = q/L. \tag{112}$$

At the end of the previous section, we have discussed how to choose δ_1 by solving an optimization problem, but here to get a concise expression we will choose a suboptimal δ_1, which does not have much effect on the final cost. Here, δ_1 is chosen to be

$$\delta_1 = p/16 \tag{113}$$

and then

$$\gamma_2 = \sqrt{p(1 - \delta_1)} = \sqrt{p(1 - p/16)}, \quad \gamma_1 = \sqrt{\delta_1} = \sqrt{p}/4. \tag{114}$$

Consequently,

$$\gamma_2 - \gamma_1 \approx (3/4)\sqrt{p}. \tag{115}$$

Substituting these values into Eqs. (110) and (111), we can see that in order to estimate the ground-state energy to precision ϵ with probability at least $1 - q$, with an initial squared overlap of p, we need to use W

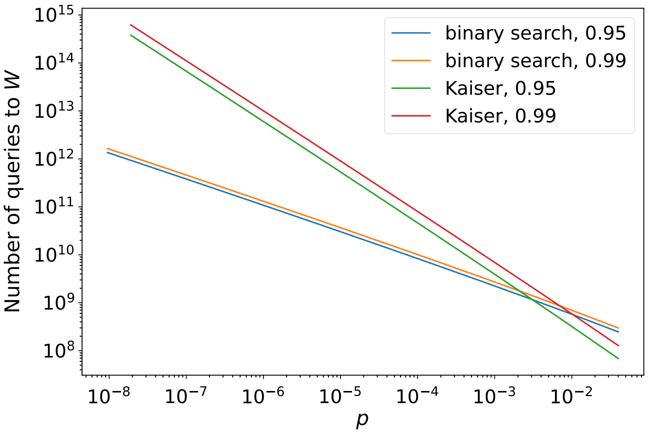

FIG. 15. The number of queries to W as a function of the squared overlap p for FeMoco with $\lambda = 306$ and $\epsilon = 0.0016$, using both the binary-search method discussed in this section and pure QPE+Kaiser window. At $p = 0.01$, the numbers of queries using the binary-search method are 571×10^6 and 691×10^6 for 95% confidence and 99% confidence, respectively, compared to 320×10^6 and 587×10^6 using pure QPE+Kaiser window. The binary-sampling cost is affected very little by whether the Kaiser or optimal window is used for phase estimation. For the 95% and 99% confidence intervals, the cost is only reduced by 0.04% and 0.06%, respectively. The value of ω is chosen to be $1/\sqrt{2}$.

$$\frac{8\lambda}{\sqrt{p}\epsilon} \ln\left(\frac{4}{\sqrt{p}}\right) \ln\left(\frac{\log_{3/2}(\lambda/\epsilon)}{q}\right) \tag{116}$$

times and we need to use U_{init}

$$\frac{4\log_{3/2}(\lambda/\epsilon)}{3\sqrt{p}} \ln\left(\frac{\log_{3/2}(\lambda/\epsilon)}{q}\right) \tag{117}$$

times. Adjusting the shrinking factor yields the slightly improved result (see Appendix H)

$$\frac{7.77\lambda}{\sqrt{p}\epsilon} \ln\left(\frac{4}{\sqrt{p}}\right) \ln\left(\frac{\log_{\sqrt{2}}(\lambda/\epsilon)}{q}\right). \tag{118}$$

In Fig. 15, we numerically compute the query complexities of the binary-search method and the direct-sampling approach, both based on the Kaiser window. The results suggest that a crossover of the query complexity takes place between $p = 10^{-3}$ and $p = 10^{-2}$ for 95% and 99% confidence levels. This agrees well with the estimated crossover of around $p \sim 0.003$ based on the constant factors in the scaling.

VII. SINGLE-ANCILLA PHASE ESTIMATION

In the previous sections, we have explored phase-estimation algorithms based on QFT, including the sampling method and the binary-search approach. Recently, a class of “post-Kitaev” phase estimation algorithms [17, 19–21,45–47] has been proposed for early fault-tolerant quantum computers [48]. Similar to the original Kitaev algorithm [49], many of these algorithms utilize the output from the Hadamard test circuit. However, what distinguishes them from the Kitaev algorithm is the strategy for choosing the Hamiltonian simulation time and the classical signal-processing techniques to handle quantum noisy data. Unlike QFT-based algorithms, these new methods require only a single ancilla qubit while still achieving Heisenberg-limited scaling. Furthermore, when the initial state has a large overlap with the ground state (i.e., the initial overlap $p \approx 1$), the minimal circuit depth can be significantly reduced to $\mathcal{O}((1-p)/\epsilon)$ for a given precision ϵ, which vanishes in the limit of a perfect initial state ($p = 1$). In Appendix I, we demonstrate the core principles of these signal-processing-based algorithms, using the recent QMEGS algorithm [47] as an example.

Here, we numerically compare QMEGS [47] and QPE Kaiser. For simplicity, we only focus on the total (controlled-)Hamiltonian running time T_{total} of both methods, which is proportional to the number of queries to W or the number of non-Clifford gates. Also, instead of using traditional controlled-W in QPE and the Hadamard test circuit, we implement the quantum walk operator in Eq. (9), which doubles the resulting phase difference.

In our tests for QPE Kaiser, we choose optimal $\alpha \approx 0.765375$ and consider the version that incorporates a randomly perturbed angle, achieved by adding an extra (known) phase shift to the phase and correcting for it during estimation. This random perturbation allows us to obtain a continuous range of outcomes. Given a fixed maximum running time $T_{\max}$ and the number of measurements N_{QPE}, the total running time is given by $T_{\mathrm{total}} = N_{\mathrm{QPE}} \times T_{\max}$. When implementing the quantum walk operator, the maximal phase difference in QPE is $2T_{\max}\lambda$ for an eigenvalue λ. For QMEGS, we fix $\sigma = 1$ and vary T. Due to the implementation of the quantum walk operator, the maximum running time is $T_{\max} = \frac{1}{2}\max_n |t_n| \le T/2$, while the total running time is $T_{\mathrm{total}} = \sum_n |t_n|$.

We test both methods in estimating the ground-state energy of the transverse-field Ising model (TFIM) with eight qubits and the Hubbard model with four qubits, with $p_0 = 0.9$ and $p_1 = 1 - p_0 = 0.1$, where p_0 is the overlap with the ground state and p_1 is the overlap with the first excited states. For each test, we repeat both algorithms 1000 times with output energy $\{\widetilde{\lambda}_{0,n}\}_{n=1}^{1000}$. We then compute the half-width $\widetilde{\epsilon}_{95\%}$ of the 95% confidence interval for the error,

$$\widetilde{\epsilon}_{95\%} = \inf_E \left\{ \left| \left\{ |\widetilde{\lambda}_{0,n} - \lambda_0| \le E \right\} \right| \ge 950 \right\}. \qquad (119)$$

This serves as an approximation for the exact half-width of the 95% confidence interval for the error of the algorithm:

$$\epsilon_{95\%} = \inf_E \left\{ \mathbb{P}\left(|\widetilde{\lambda}_0 - \lambda_0| \le E \right) \ge 0.95 \right\}, \qquad (120)$$

where $\widetilde{\lambda}_0$ is the estimate output by the algorithm.

In the experiment, we choose $N_{\mathrm{QPE}} = 2, 4, 6$ and $N_{\mathrm{QMEGS}} = 40, 80$. Before showing the results, we first summarize two observations:

(i) For both algorithms, when the number of samples N is chosen optimally, the total cost T_{total} for QPE Kaiser is *50% lower* than that of QMEGS. For instance, when $p_0 = 0.9$, the error curve of QPE Kaiser with $N_{\mathrm{QPE}} = 4$ matches that of the optimal QMEGS with $N_{\mathrm{QMEGS}} = 40$, making QMEGS twice as costly as the optimal QPE Kaiser with $N_{\mathrm{QPE}} = 2$, as shown in Fig. 16.

(ii) When the optimal number of samples is not available, as is often the case in real applications in which p_0 is unknown, the error in QMEGS is *more stable* compared to QPE. For example, when $p_0 = 0.9$, the cost of QPE Kaiser increases by a factor of 1.5 if the number of samples N_{QPE} increases by a factor of 1.5 (from $N_{\mathrm{QPE}} = 4$ to $N_{\mathrm{QPE}} = 6$). In contrast, for QMEGS, the cost increases by only a factor of 1.4 when N_{QMEGS} doubles (from 40 to 80), making it more cost-effective than QPE Kaiser with $N_{\mathrm{QPE}} = 6$, as shown in Fig. 16.

We note that the optimal cost of QMEGS is roughly twice that of QPE, because two Hadamard tests are required to obtain a single signal estimation in Eq. (I1), doubling the T_{total} for QMEGS. It would be interesting to explore whether additional techniques could be employed to reduce the noise in the Hadamard test or to generate a signal estimation directly using a single shot of controlled-Hamiltonian evolution.

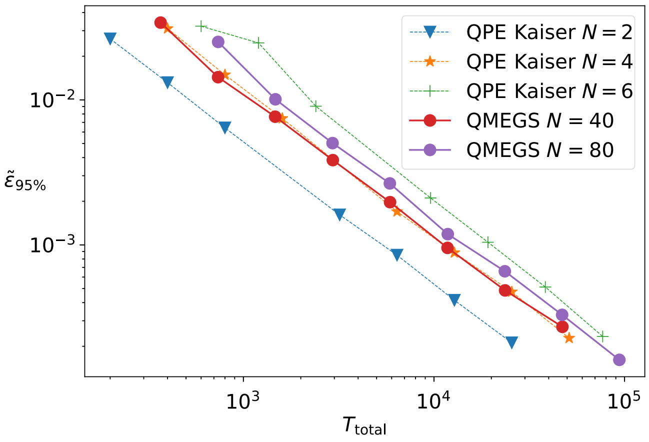

FIG. 16. QMEGS versus QPE Kaiser for the TFIM-8 model with $p_0 = 0.9$ and $p_1 = 0.1$. The half-width of the 95% confidence interval is denoted by $\widetilde{\epsilon}_{95\%}$ and defined in Eq. (119).

For the TFIM-8 model, we have rescaled the Hamiltonian so that $\lambda_0 = -1$ and $\lambda_1 \approx -0.8154$. The results are summarized in Fig. 16. In terms of T_{total}, QPE Kaiser with $N_{\text{QPE}} = 4$ performs similarly to QMEGS with $N_{\text{QMEGS}} = 40$, whereas QPE Kaiser with $N_{\text{QPE}} = 6$ performs worse than the suboptimal QMEGS with $N_{\text{QMEGS}} = 80$. For the Hubbard-4 model, we rescale the Hamiltonian so that $\lambda_0 = -1$ and $\lambda_1 \approx -0.9767$. The result is summarized in Fig. 16.

VIII. TOTAL RESOURCES FOR GROUND-STATE ENERGY ESTIMATION IN CHEMICAL SYSTEMS

As previously discussed, a key quantity in the cost of applying QPE to ground-state energy estimation is the overlap of the initial input state with the ground-state wave function. Prior work has investigated overlaps for simple small molecules [3] and found high overlaps with product-state wave functions substantially above the low-p settings discussed in this work. More recently, it has been shown that in realistic examples of more complicated molecules, product states can have a small overlap, while adiabatic state preparation from the lowest-energy mean-field state can be more costly than the phase estimation itself, motivating the development of more sophisticated state-preparation protocols [12]. However, determining an overlap, or a faithful estimate of it, in problems where classical algorithms have difficulty computing an accurate ground state (and thus which are of most interest for quantum applications) is by definition challenging.

In this section, we describe an extrapolation protocol that allows us to estimate the overlap with the ground state by extrapolating with respect to the bond dimensions of two MPS wave functions. Although this extrapolation does not provide any rigorous guarantees, we provide benchmark data to support the procedure (see Appendix K). In addition, in the most challenging systems, such as FeMoco, it is not possible to classically determine MPS wave functions with sufficiently large bond dimension to distinguish between the ground state and nearby excited states. In this case, under the assumption that the MPS we are preparing has good overlap with some low-energy eigenstate, we provide an estimation of the overlap with this eigenstate and QPE can then be used to resolve the energies of the different eigenstates associated with the different MPS initial states. Leveraging these estimates, we account for the full resources needed for resolving the energy landscape of competing spin structures in FeMoco.

A. Model generation

In this study, we have employed active-space models for Fe-S simulations suggested in previous studies for the Fe_2S_2, Fe_4S_4, and FeMoco iron-sulfur systems [31,50]. The active spaces for the Fe_2S_2 and Fe_4S_4 models consist of Fe-3d, S-3p, and σ-bonding orbitals between the Fe and thiolate ligands, defined as complete active spaces CAS(30e,20o) for 2Fe(III), CAS(54e,36o) for 2Fe(III)2Fe(II), and CAS(52e,36o) for 4Fe(III). In this notation, CAS(ne,mo) indicates the complete active space with n being the number of electrons and m the spatial orbitals in the active space (with “e” and “o” labeling electrons and spatial orbitals). The active-space model of FeMoco also includes additional Mo-4d and central C-2s and C-2p orbitals, resulting in a CAS(113e,76o) for 4Fe(III)3Fe(II)Mo(III).

The energy landscape of ansatz approximations to FeMoco ground-state wave functions with total spin $S = 3/2$ is characterized by numerous local electronic minima. Therefore, obtaining the correct ground state requires starting from a good initial guess. To achieve this, we have used a density matrix renormalization group (DMRG) initialization procedure as described in earlier studies [28,31] within the implementation in BLOCK2 [51,52]. For the initial guess in the spin-adapted DMRG calculations, we have first performed a spin-projected MPS calculation, initiated by a spin-projected broken-symmetry determinant. We have explored 35 different spin-projected determinants corresponding to the broken-symmetry configuration of $\{2\text{Fe(III)}\uparrow, 2\text{Fe(III)}\downarrow, 2\text{Fe(II)}\uparrow, \text{Fe(II)}\downarrow, \text{Mo(III)}\downarrow\}$, which have previously been studied using broken-symmetry density-functional theory in Ref. [53]. The resulting spin-projected MPS has then been optimized up to a bond dimension of 50.

These initial MPSs have subsequently been optimized using spin-adapted DMRG calculations, with the bond dimension increased to 2000. Out of these 35 MPSs, we have selected the three lowest-energy states at this bond dimension as example initial states, denoted MPS1, MPS2, and MPS3. The character of the spin couplings of these states, as represented by the spin-projected determinants used to initialize them, is shown in Fig. 17(a). The three MPSs have been further optimized, increasing the bond dimension to 7000 for MPS1 and 4000 for MPS2 and MPS3. In Fig. 17(b), we show the spin-correlation matrix $\langle S_A \cdot S_B \rangle$ between metal centers $\{A, B\}$ of these further optimized MPSs, which is defined by

$$S_A \cdot S_B = \sum_{\mu \in \{x,y,z\}} S_A^\mu S_B^\mu \quad \text{with} \tag{121}$$

$$S_A^\mu = \sum_{p \in A} s_p^\mu, \tag{122}$$

$$s_p^x = \frac{1}{2}\left(a_{p\uparrow}^\dagger a_{p\downarrow} + a_{p\downarrow}^\dagger a_{p\uparrow}\right), \tag{123}$$

$$s_p^y = \frac{1}{2i}\left(a_{p\uparrow}^\dagger a_{p\downarrow} - a_{p\downarrow}^\dagger a_{p\uparrow}\right), \tag{124}$$

$$s_p^z = \frac{1}{2}\left(a_{p\uparrow}^\dagger a_{p\uparrow} - a_{p\downarrow}^\dagger a_{p\downarrow}\right), \tag{125}$$

such that p indexes the orbitals local to metal center A.

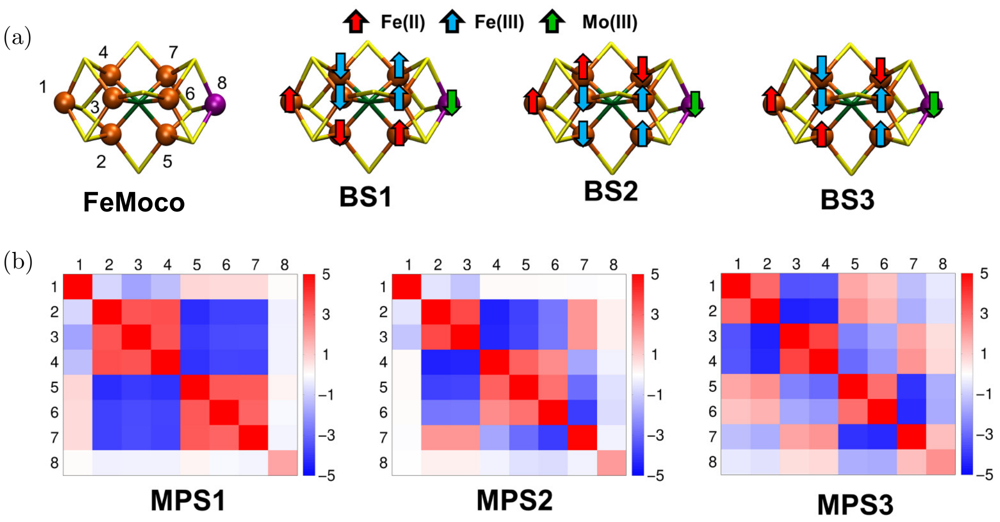

FIG. 17. (a) A schematic representation of the metal-site numbering in FeMoco, along with the spin-projected broken-symmetry determinants used in the spin-projected MPS calculation. Out of 35 broken-symmetry determinants, the three with the lowest DMRG energies at a bond dimension of 2000 are represented. (b) The spin-correlation matrices for the three chosen MPSs. MPS1, MPS2, and MPS3 started from the BS1, BS2, and BS3 guesses, respectively, and have converged to different states.

We have used a well-known energy-extrapolation scheme to estimate the energy errors of these MPSs [28,54–56]. In the zero-discarded-weight limit, which represents the exact MPS, the energies are expected to be 86, 109, and 92 millihartree lower than the DMRG energies of MPS1 ($M = 5500$), MPS2 ($M = 3500$), and MPS3 ($M = 3500$), respectively. We demonstrate linear extrapolation to these energies in Appendix K. Because these three states contain qualitatively different correlations, we consider these estimates to correspond to different low-energy eigenstates in the system. In this section, we use M for the bond dimension for consistency with the literature on DMRG, in contrast to the notation χ used for MPS preparation.

B. Overlap-extrapolation protocol

Here, we present an extrapolation scheme to predict the squared overlap between a spin-adapted DMRG state with a given bond dimension M and the exact wave function, $|\langle\Phi(M)|\Phi(\infty)\rangle|^2$. For the extrapolation, we have utilized the following two empirical linear relations:

$$\log\left(1 - \left|\langle\Phi(M')|\Phi(\infty)\rangle\right|^2\right) \text{ versus } \left(\log(M')\right)^2, \quad (126)$$

$$\log\left(\left|\langle\Phi(M')|\Phi(M'')\rangle\right|^2 - \left|\langle\Phi(M')|\Phi(\infty)\rangle\right|^2\right)$$
$$\text{versus } \left(\log(M'')\right)^2, \text{ where } M' \ll M''. \quad (127)$$

In Appendix K, we show that these linear relations are satisfied in the Fe_2S_2 system where the exact wave function for the complete active-space model of CAS(30e,20o) is accessible. Building on these empirical linear relations, here we demonstrate the extrapolation scheme in detail. Specifically, we show an example of estimating $|\langle\Phi(M = 1000)|\Phi(\infty)\rangle|^2$ for the 2Fe(II)2Fe(III) system. The main objective of this scheme is to accurately determine the squared overlap using data from MPSs with bond dimensions less than $M = 1000$. To achieve this, we have first generated MPSs with bond dimensions of 800, 600, 60, 40, and 20 using a reverse-sweep DMRG calculation. We have then estimated $|\langle\Phi(M' = 20)|\Phi(\infty)\rangle|^2$ using the values of $|\langle\Phi(M' = 20)|\Phi(M'')\rangle|^2$ for $M'' = 600, 800, 1000$ based on the second empirical linear relation of Eq. (127). We have performed a linear fit of

$$\log\left(\left|\langle\Phi(20)|\Phi(M'')\rangle\right|^2 - \left|\langle\Phi(20)|\Phi(\infty)\rangle\right|^2\right)$$
$$\text{versus } \left(\log(M'')\right)^2 \quad (128)$$

to determine the value of $|\langle\Phi(20)|\Phi(\infty)\rangle|^2$. In a similar fashion, we can estimate $|\langle\Phi(M')|\Phi(\infty)\rangle|^2$ for the other bond dimensions of $M' = 40$ and 60. The empty black triangles in Fig. 18(a) represent the infidelities for these estimated values using $M' = 20, 40, 60$. Each empty black triangle has been estimated from the blue, yellow, and green triangles directly below it. In Appendix K, we discuss the validity of this extrapolation in detail.

Finally, these estimated values have been used to obtain a linear fit represented by the dotted line in

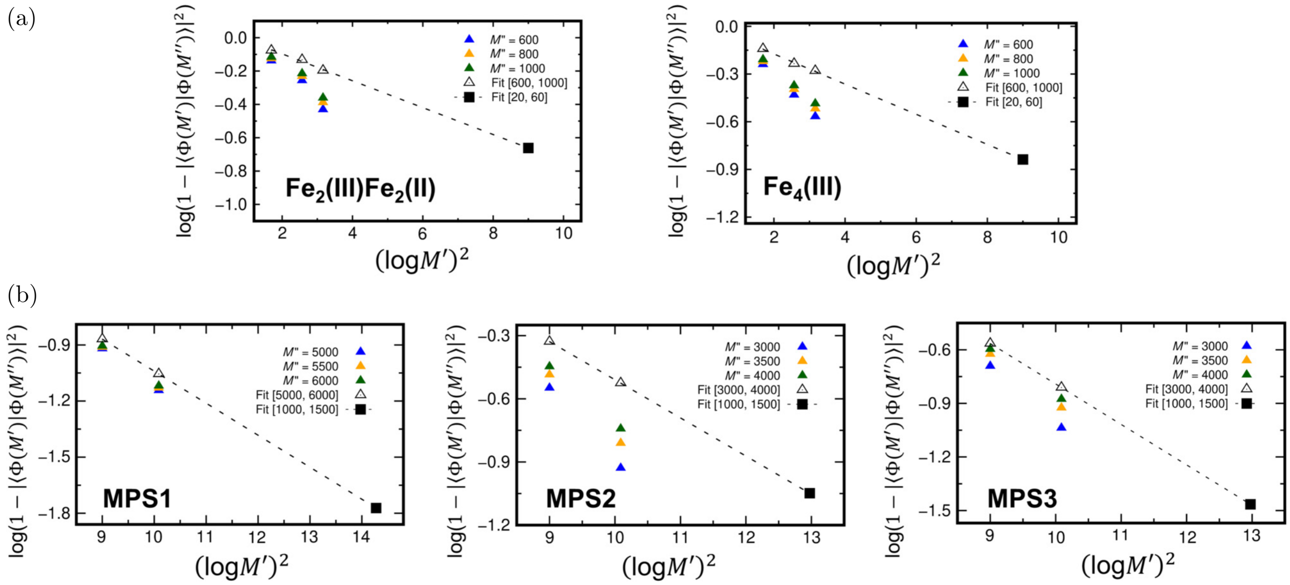

FIG. 18. The extrapolated overlap for (a) 2Fe(II)2Fe(III) and 4Fe(III), and (b) FeMoco.

Fig. 18(a), based on the first empirical linear relationship of Eq. (126). Using this line, we have estimated the value of $|\langle\Phi(1000)|\Phi(\infty)\rangle|^2$ at the black square. We have used the same extrapolation scheme to predict the overlap for the 4Fe(III) and FeMoco systems. For 4Fe(III), we have utilized MPSs with $M' = 20, 40, 60$ and $M'' = 600, 800, 1000$. For FeMoco, we have used MPSs with $M' = 1000, 1500$ and $M'' = 5000, 5500, 6000$ for MPS1 and $M'' = 3000, 3500, 4000$ for MPS2 and MPS3. It is noted that we have used MPS1 with $M = 6000$ for the overlap estimate although MPS1 with $M = 7000$ has been obtained from the DMRG calculation. This was because the site of the renormalized wave function with $M = 7000$ differed from the MPSs obtained through the reversed sweep, making it difficult to compute the overlap. The predicted overlap values are summarized in Table I along with the associated state-preparation costs using the unitary-synthesis techniques described earlier. Note that although the bond dimensions are on the order of thousands, they do not require many qubits in the preparation. The dimensions of 1000, 4000, and 6000 correspond to only 10, 12, or 13 qubits, respectively.

C. MPS preparation costs versus phase-estimation costs

With the costs of performing MPS preparation, extrapolated overlap values, and number of samples using phase estimation, we are in a position to perform resource estimation for the full task of ground-state energy estimation. The cost of performing block encodings for two [4Fe-4S] systems and FeMoco are determined using the tensor-hypercontraction (THC) factorization of the two-electron integral tensor [27] and double factorization [58] using number-operator symmetry shifting described in Ref. [59] on the one-body and two-body components of the Hamiltonian. For THC resource estimates, we apply symmetry shifting only to the one-body component. We provide symmetry-shifted and nonsymmetry-shifted block-encoding costs along with high-spin coupled-cluster correlation energies (obtained using PySCF [60,61]) for a variety of cutoff parameters in Appendix J. For resource estimates, we use the 1-millihartree threshold in correlation-energy differences (with respect to the nontruncated integrals) previously used

TABLE I. Extrapolated MPS overlaps as the output of the protocol described in Sec. III. The physical dimension of each subsystem for the MPS is taken to be $d = 4$, corresponding to $\{|\emptyset\rangle, |\uparrow\rangle, |\downarrow\rangle, |\uparrow\downarrow\rangle\}$. In the case of FeMoco, the overlap is with respect to a low-energy eigenstate, which may not be the ground state. The Toffoli and qubit counts are calculated using QUALTRAN [57].

System	Estimated $|\langle \mathrm{MPS}|\psi_0\rangle|$	Bond dimension	Spatial orbitals	MPS Toffolis	Qubits
Fe_2(III)Fe_2(II)	0.88	1000	36	42 200 000	359
Fe_4(III)	0.92	1000	36	42 200 000	359
FeMoco [50] MPS1	0.99	6000	76	1 360 000 000	833
FeMoco [50] MPS2	0.95	4000	76	733 000 000	682
FeMoco [50] MPS3	0.98	4000	76	733 000 000	682

in Ref. [27]. We find that for all iron-sulfur clusters studied, the 1-norm value is approximately halved by symmetry shifting.

For the cost of sampling to obtain the confidence intervals, one could use Eq. (1) but in this high-overlap regime that expression tends to overestimate the cost. For a more accurate estimate, one can perform the following procedure:

(1) For a given number of samples n, solve Eq. (61) for δ.
(2) Given that δ, solve Eq. (57) for c.
(3) Take the cost as $nc\lambda/\epsilon$.

This procedure is used for a number of values of n to find the one that gives the minimum cost. Here, we are using the prolate spheroidal window, which we find to give the best performance in the high-overlap regime. In the results below, we also use the method in Sec. V C to obtain a more accurate estimate accounting for the excited states. In this high-overlap regime, it only gives a small correction.

Given these sampling costs, we then use this overhead multiplied by the block-encoding cost. We then add n times the MPS-preparation cost to obtain the total complexity. For consistency with previous studies, we select $\epsilon = 1.0 \times 10^{-3}$ hartree, although here we are taking this to be the confidence interval half-width, rather than the rms error as in previous studies. As a result, the accuracy requirement is somewhat more demanding, as confidence intervals are significantly wider than the rms error.

We provide estimates for double factorization (DF) and denote the corresponding value of λ for the block encoding of double factorization by λ_{DF}. In our estimates, we have reduced the value of λ_{DF} with a symmetry shift of $(\alpha_2/2)\hat{N}\hat{N} + (\alpha_1 - \alpha_2/2)\hat{N}$, where $\hat{N}$ is the number operator and α_1 and α_2 are optimized to minimize λ_{DF}. With this shift, the 1-norm of the effective electron-repulsion-integral (ERI) tensor depends only on α_2, which is optimized first subject to the constraint that the ERI tensor remains positive semidefinite. After α_2 has been determined, α_1 is chosen to minimize the 1-norm of the effective one-electron part of the Hamiltonian, which depends on both α_1 and α_2. For THC, the LCU 1-norm λ is computed with a number-operator symmetry shift computed as the median of $\{f_i\}$, where the f_i are eigenvalues of the one-body operator being block encoded.

Using these results for λ values, we analyze the cost of Fe-S cluster ground-state energy estimation with 95% and 99% confidence levels. The total Toffoli and qubit requirements are listed in Table II. Due to the overlaps being so high, few QPE samples are needed, only two in the example of FeMoco. We should note that this corresponds to only one candidate MPS initial guess corresponding to a specific structure of spin coupling. Including all the BS-DFT–derived MPS initial guesses considered in this work would multiply this cost by 35. This number may be reduced by prefiltering some of the MPS initial guesses by the DMRG extrapolated energies.

For comparison with the costings given in Ref. [27], first note that we are using the Li Hamiltonian as from Ref. [50], whereas Ref. [27] has also considered the Reiher Hamiltonian from Ref. [62]. The Li Hamiltonian has higher cost to simulate and it is those costs in Ref. [27] with which we should compare. The relevant costs to compare with are therefore the right column in Ref. [27, Table III]. For THC, the Toffoli cost is 3.2×10^{10}, so the estimated cost here for a 95% interval is about 2.3 times higher. This factor comes from three main considerations:

(1) Here, we have used symmetry shifting to reduce λ to 781.8, which *reduces* the complexity by a factor of about 1.54 as compared to the λ in Ref. [27] of 1201.5. For the value of λ, see the line for $M = 450$ in Ref. [27, Table V].
(2) Requiring two samples doubles the complexity.
(3) Requiring the measurements to provide confidence intervals of half-width ϵ rather than rms error of ϵ is more demanding, and also increases the cost.

For DF, the estimated cost from Ref. [27] is 6.4×10^{10}, so the estimated cost for 95% confidence intervals here is about 1.7 times larger. The reduction in λ due to symmetry

TABLE II. The combined QPE and MPS preparation costs for confidence intervals of half-width $\epsilon = 1.0 \times 10^{-3}$ hartree at 95% and 99% confidence levels. The QPE costs for FeMoco are calculated using only the smallest 0.95 overlap for MPS2. The abbreviation "BE" stands for "block encoding".

System	Method	λ	BE Toffolis	Qubits	QPE cost 95%	QPE cost 99%
Fe_2(III)Fe_2(II)	THC	168.7143	9120	1149	1.33×10^{10}	2.45×10^{10}
	DF	154.7362	15 545	3111	2.08×10^{10}	3.82×10^{10}
Fe_4(III)	THC	164.1287	8573	1081	8.37×10^{9}	1.67×10^{10}
	DF	150.2923	15 602	3113	1.39×10^{10}	2.77×10^{10}
FeMoco [50]	THC	781.8172	16 923	2194	7.27×10^{10}	1.38×10^{11}
	DF	582.4211	35 006	6402	1.11×10^{11}	2.11×10^{11}

shifting for DF is more significant than for THC. The value of λ is given in the line for $L = 394$ in Ref. [27, Table XIV] as 1171.2, so the improvement in λ here is more than a factor of 2. The other considerations for the sampling cost for DF are the same as for THC.

IX. CONCLUSIONS

In this work, we have developed improved methods for ground-state energy estimation via both improved initial-state preparation and improved filtering of the initial state. For initial-state preparation, we have developed an improved method for preparing MPSs. This preparation has been based on our new technique for synthesizing general unitary operations, which improves the Toffoli count by about a factor of 7 over prior work. We have then used that to construct an iterative procedure to prepare MPSs with a substantially reduced complexity. The method for synthesizing general unitary operations is of independent interest, as this is a very common task in quantum computing. Moreover, we have found that it may be possible to further improve the complexity by a procedure interspersing phasing with Hadamard gates. The drawback to that approach is that we do not have an efficient procedure to determine the phases required, so the unitaries are restricted to lower dimension (no more than about eight qubits). In future work, it may be possible to develop more efficient procedures to solve for the phases, making that a more viable approach.

For improved filtering, we have proposed two approaches for ground-state energy estimation, both using quantum phase estimation, the efficiency of which is boosted through window functions. These window functions have been chosen to minimize the error in a confidence interval, as opposed to rms error, which is more commonly considered. This choice for the phase measurements is useful as we need to perform multiple phase measurements, all of which need to avoid large error. The two methods are direct sampling and a binary search using amplitude estimation. For both, we have provided both asymptotic expressions and numerical estimates of the complexity. The advantage of the binary search with amplitude estimation is that it provides a square-root speedup in the overlap, although it has a larger constant factor than the direct-sampling approach. This means that the direct-sampling approach is preferable for the case in which the initial guess has large overlap with the exact ground state, whereas the binary-search approach is more advantageous in the small-overlap situation. The asymptotic expressions suggest that the crossover is at $p \sim 0.003$, and that prediction is borne out by the numerics.

Building on the efficient MPS-preparation results along with the optimal window function analysis, we have analyzed the cost of refining energy estimates using QPE initialized with an MPS wave function. In order to determine total costs for energy refinement in the high-confidence regime, we have determined the overlap through an extrapolation. The extrapolation protocol uses two MPS wave functions to determine the infinite-bond-dimension overlap of a finite-bond-dimension wave function. The extrapolation is empirical but is supported in this set of systems by verifying against true overlaps computed in the smaller FeS cluster where exact ground states can be found through large-bond-dimension MPS calculations. In the case of FeMoco, where the ground-state energy manifold has many competing spin configurations, we have estimated the overlap for different MPS initial-state wave functions that are candidates for different low-energy eigenstates. The role of QPE in this setting is then to refine the energy ordering of the states, enabling the determination of the ground-state energy. Ultimately, due to the high extrapolated overlap, achieving 95% or 99% confidence intervals only requires two samples from QPE. Improvements to block encoding the LCU 1-norm through symmetry shifting result in total complexities that are only 2.3 times those of naive QPE, assuming perfect overlap (and with the less demanding requirement of rms error ϵ) [27].

The extrapolations and overlaps estimated here provide a concrete numerical example of a complex chemical problem in which classical precomputation can be used to prepare initial states of high overlap, enabling efficient QPE. As discussed previously [12,13], the degree of quantum advantage can then be evaluated from the relative cost of classical and quantum refinement from such an initial state. While the MPS wave functions considered here are attractive candidates for strongly correlated molecules up to a given finite size, other types of ansatz and techniques may also be used, particularly in the study of even larger strongly correlated molecular problems. In conclusion, our work has found that algorithms based on preparation of a classically computed state and QPE provide a practical approach in real-world examples of challenging molecular chemistry.

ACKNOWLEDGMENTS

D.W.B. worked on this project under a sponsored research agreement with Google Quantum AI. D.W.B. is also supported by Australian Research Council Discovery Projects No. DP210101367 and No. DP220101602. Y.T., L.L., and G.K.C. were supported by the U.S. Department of Energy, Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator. Work by S.L. was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (Grant No. RS-2025-00515475). Work by T.I.K. was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (Grant No. RS-2024-00415940).

APPENDIX A: SUMMARY OF NOTATION

N_{un}	The dimension of a unitary operation to be synthesized.
N	The dimension of the control register used for phase estimation.
$\mathcal{N}$	The number of subsystems in the MPS representation of a state.
$\hat{N}$	The number operator for fermions.
d	The dimension of the subsystems used in the MPS representation of a state.
χ	The bond dimension for the MPS.
M	This symbol is also used in some places for the MPS bond dimension, primarily in Sec. VIII.
p	The initial squared overlap of the prepared state with the ground state. It is also used in Sec. VIII as an index of summation, or in the text without italics to mean the name of an orbital, with the meaning clear by context.
λ	The block-encoding normalization factor.
ϵ	The allowable error in the energy estimate.
q	The confidence level of the desired confidence intervals is at least $1-q$.
δ	The confidence level for individual measurements (which are combined to give the overall ground-state energy estimate) is $1-\delta$.
n	Primarily used for the number of samples to distinguish the ground state. In some places, n is used for the number of qubits, or a variable of summation, with the meaning being clear by context.
$G[j]$	Unitaries used in the preparation of the MPS.
W	The walk operator constructed from the block encoding of the Hamiltonian.
ϕ_j	Used for eigenvalues of W, and elsewhere as rotation angles used in unitary synthesis, with the meaning being clear by context.
$\hat{\phi}$	An estimate of the eigenvalue of W.
λ_j	Eigenvalues of the Hamiltonian.
γ_n	Amplitudes used in a control state for phase estimation.
Γ	The kernel function describing the error in the phase estimation.
I_0	The modified Bessel function of the first kind.
$\mathrm{PS}_{0,0}$	The prolate spheroidal function.
$S^1_{0,0}$	The radial spheroidal function of the first kind.
Ci	The cosine integral function.
D_ν	The parabolic cylinder function.
$\phi_j, \theta_j, \varphi_j$	Angles used in rotations for unitary synthesis.
H	The Hadamard matrix.
Λ	A power of 2 that is chosen to reduce Toffoli costs for QROM.
b	The number of bits used in rotations.
A, B, C	Blocks of a unitary operation to be synthesized.
U_1, U_2, V	Unitaries generated in the decomposition of the unitary operation to be synthesized.
D_1, D_2	Diagonal matrices generated in the decomposition of the unitary operation to be synthesized.
α	The parameter used in the Kaiser window.
Δ	A parameter used to adjust the width of the confidence interval used with the Kaiser window.
c	The parameter used in the Slepian window.
ρ	Equal to $-\ln(1-p)$ and used in estimating sampling costs.
h	Equal to $1-n\rho-\ln(2\rho)$, and also used in estimating sampling costs.
δ_1, δ_2	In Sec. V, these are the single-sided errors in phase measurements on excited states.
β	The excited-state energy is $\beta\epsilon$ above the ground-state energy.
P_{err}	The probability of error in estimating the ground-state energy using a particular sampling scheme.
η	Used for the phase-estimation error (confidence interval half-width) in Sec. VI.
$Q(\eta, \delta)$	The query complexity (to steps of the walk W) for phase estimation with error η and confidence level $1-\delta$.
λ_L, λ_R	Lower and upper bounds on the estimation of an eigenvalue in the binary-search approach.
δ_1, δ_2	Particular choices of δ used in the binary-search approach in Sec. VI (distinct from its usage in Sec. V).
γ_1, γ_2	Choices of amplitudes in the binary-search approach.
$\mathcal{W}$	A walk operator that encodes the amplitude for phase estimation of W to be above a threshold.
d_1, d_2	Numbers of queries to W and $\mathcal{W}$ used in the binary search.
ω	A shrinking factor used in the binary search.
L	The number of steps in the binary search, and also a quantity used when referring to Ref. [27].
$a, a^\dagger$	Fermionic annihilation and creation operators.
$\lvert\Phi(M)\rangle$	An MPS approximation of the ground state with bond dimension M.

APPENDIX B: COSTING FOR LKS SCHEME

Here, we give a detailed analysis of the cost of unitary synthesis using the method of Low, Kliuchnikov, and Schaeffer [8]. For the unitary synthesis, the approach of Ref. [8] is to perform a sequence of K reflections $\mathbb{1} - 2\,|v_k\rangle\,\langle v_k|$ when K columns of the unitary need to be specified (here, $K = \chi$), as well as a diagonal operation (phases in the computational basis). The reflections are implemented by a sequence of two state preparations (one forward and one reverse). In the following description, we take the dimension to be $N_{\rm un}$, which need not be a power of 2, and the number of qubits to be n, so that $2^n \geq N_{\rm un}$. The qubits are ordered such that the most significant qubit is first. The state preparation is applied by the following procedure:

(1) Perform a rotation on the first qubit to prepare $\sqrt{p_0}\,|0\rangle + \sqrt{p_1}\,|1\rangle$. This has Toffoli complexity b for b bits of precision for the rotation, by using a phase-gradient state.

(2) Use the state of the first qubit to output b bits for the rotation on the second qubit. Then use that data to perform a controlled rotation on the second qubit, and then erase it. There is zero Toffoli complexity for the QROM on the single qubit and again there is complexity b for the rotation.

(3) For qubits $k = 3$ to $n-1$, use QROM on qubits 1 to $k-1$ to output a rotation angle. This has complexity

$$\left\lceil \frac{N_{\rm un}}{2^{n-k+1}\Lambda_k} \right\rceil + (\Lambda_k - 1)b, \tag{B1}$$

for the parameter Λ_k a power of 2 in the QROM. In Ref. [8], it is assumed that Λ is taken independent of k but it may be adjusted to minimize the complexity. Then there is complexity b for the rotation on qubit k and erasure of the QROM needs complexity

$$\left\lceil \frac{N_{\rm un}}{2^{n-k+1}\Lambda'_k} \right\rceil + \Lambda'_k, \tag{B2}$$

for total complexity

$$\left\lceil \frac{N_{\rm un}}{2^{n-k+1}\Lambda_k} \right\rceil + \Lambda_k b + \left\lceil \frac{N_{\rm un}}{2^{n-k+1}\Lambda'_k} \right\rceil + \Lambda'_k, \tag{B3}$$

where Λ_k and Λ'_k may be chosen independently.

(4) At the end, the phases are applied. This requires a QROM on all n qubits, then addition into a phase-gradient register, and erasure of the QROM, with complexity

$$\left\lceil \frac{N_{\rm un}}{\Lambda_n} \right\rceil + \Lambda_n b + \left\lceil \frac{N_{\rm un}}{\Lambda'_n} \right\rceil + \Lambda'_n. \tag{B4}$$

Steps 1–3 prepare a state with real amplitudes and then the final step applies phases. This final step applying phases would be applied twice between each of the K reflections. It is more efficient to combine these phases, so that there are K reflections by states with real coefficients, and $K+1$ diagonal-phasing operations. This procedure is illustrated for the case of preparation of a state on four qubits in Fig. 19.

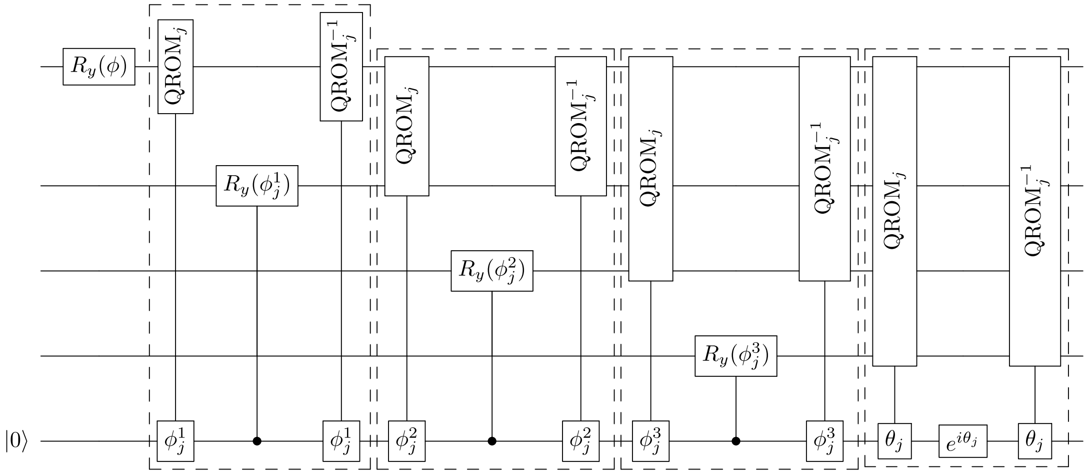

FIG. 19. An example of the LKS preparation on four qubits. The top four lines are these four qubits and the register at the bottom is for the data output of the QROMs (and is b qubits). The initial $R_y(\phi)$ is step 1 and then the first dashed rectangle is step 2. The second and third dashed rectangles correspond to step 3 for $k = 3, 4$. The last dashed rectangle is step 4, where $e^{i\theta_j}$ indicates a gate applying this phase factor using the data stored in this register.

The complete cost of this procedure, excluding the cost of the phases, can be given as

$$\sum_{k=3}^{n-1} 2\sqrt{\frac{N_{\mathrm{un}}b}{2^{n-k+1}}} \approx 2(1+\sqrt{2})\sqrt{N_{\mathrm{un}}b}. \tag{B5}$$

The cost including the cost of the phases is then approximately $2(2+\sqrt{2})\sqrt{N_{\mathrm{un}}b}$ and so about $1+\sqrt{2} \approx 2.4$ times the $2\sqrt{2N_{\mathrm{un}}b}$ cost for the improved method in Appendix C.

In the LKS approach for unitary synthesis, each step uses a forward and reverse preparation of the correct amplitudes, as well as applying the phases. That gives a cost of $2(3+2\sqrt{2})\sqrt{N_{\mathrm{un}}b}$ for each layer, as compared to $2\sqrt{N_{\mathrm{un}}b}$ for our improved unitary-synthesis method. This cost is a factor of $3+2\sqrt{2} \approx 5.8$ times larger than our improved method. In this approximation, the cost of the controlled rotations is omitted but this cost is significantly larger in LKS than in our improved scheme due to the need to perform rotations on all qubits. There is also a larger QROM erasure cost for the LKS approach. When including these contributions, our method provides an improvement by about a factor of 7, as shown in Fig. 9.

APPENDIX C: IMPROVED IMPLEMENTATION OF STATE PREPARATION

The Toffoli cost of the LKS approach can be significantly reduced by combining QROM with Grover-Rudolph in a new way. We choose a set of numbers $m < n$ and perform the preparation using the following sequence of steps:

1. Prepare the state with the correct amplitudes on qubits 1 to m by using standard QROM to output the correct rotation angle for each qubit in sequence.
2. Output the correct rotation angles for the remaining qubits, as well as the correct phases for the state, using QROM on qubits 1 to m.
3. Perform the sequence of rotations on the following qubits using data output from step 2 and controlled SWAPS.
4. Apply the correct phases for the state using the QROM output and controlled SWAPS.
5. Erase the QROM output by measuring in the X basis and applying sign corrections (the standard QROM erasure procedure).

Over the entire procedure, we have cost b for rotation of each qubit, for a total cost of nb for the qubit rotations to provide the correct amplitudes. In the following, we will quantify the QROM cost in the procedure. The costing for the first part can be determined by the following procedure:

(a) Rotate the first qubit.
(b) Do QROM on the first qubit to output the rotation angle for qubit 2. This has zero Toffoli cost since it can be done with controlled-NOTS (CNOTS). Then rotate qubit 2.
(c) Perform QROM on the first two qubits with Toffoli cost 2 to output the rotation angle for qubit 3 and then rotate qubit 3.
(d) For QROM on qubits up to j for rotation of qubit $j+1$, we have cost $2^j - 2$.

This gives a total cost of

$$\sum_{j=1}^{m-1}(2^j - 2) = 2^m - 2m. \tag{C1}$$

However, it is possible to further reduce the cost by outputting a one-hot unary representation of the binary value in each step. Then, the extra cost of the QROM on qubits 2 to j is 2^{j-1}, because it corresponds to a sequence of 2^{j-1} SWAPS controlled by qubit j between the 2^{j-1} qubits in the one-hot representation of qubits 1 to $j-1$ and an additional 2^{j-1} qubits. Using that approach, the total Toffoli cost is about half, at

$$2^{m-1} - 2. \tag{C2}$$

For an illustration of this procedure for the case $m=4$, see Fig. 20.

Now for step 2, the cost of QROM on qubits 1 to m is assuming that a one-hot representation is provided for qubits 1 to $m-1$, so the total QROM cost for steps 1 and 2 is $2^m - 2$. Then, for step 3, we have costs as follows:

(a) There is no Toffoli cost to select the register with the rotation angle for qubit $m+1$.
(b) There is a SWAP cost of b to select the correct output register for the rotation angle for qubit $m+2$.
(c) In general, there is a SWAP cost of $(2^{j-1}-1)b$ for the rotation angle for qubit $m+j$.

This gives a total cost for these SWAPS as

$$\sum_{j=1}^{n-m}(2^{j-1}-1)b = [2^{n-m} - (n-m+1)]b. \tag{C3}$$

The total cost so far for steps 1-3 is then

$$nb + 2^m - 2 + [2^{n-m} - (n-m+1)]b. \tag{C4}$$

For an illustration of the procedure for steps 2-4 for the case $n=7$, see Fig. 21.

Next, we consider the complexity of applying the final phase for step 4. Rather than using a separate QROM at

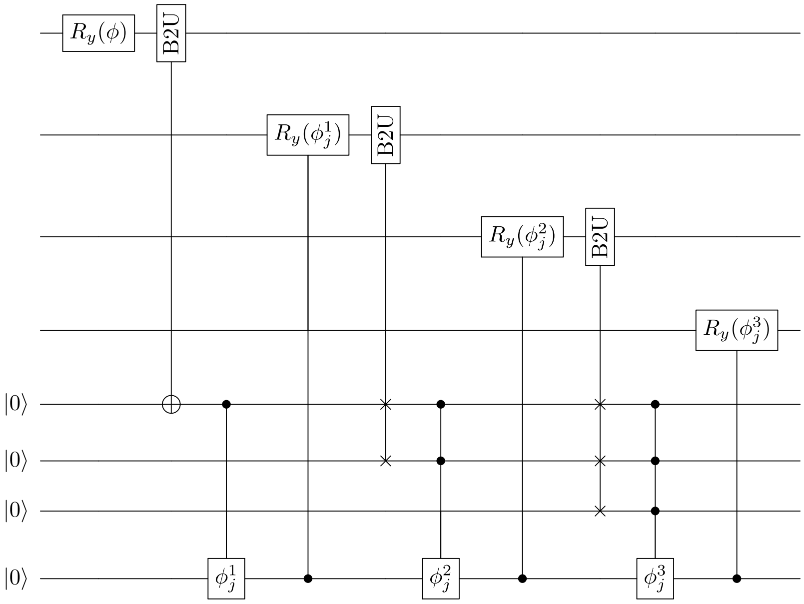

FIG. 20. An example of step 1 of the improved LKS-style preparation with $m = 4$ (the initial qubits for preparation). The top four lines are these four qubits, then below that are three registers for the one-hot unary representation, and then the register at the bottom is for the data output of the QROMs. "B2U" indicates the binary to (one-hot) unary representation. The first B2U can be performed with just controlled-NOTs (CNOTs) and two target qubits for the unary (which is why the target is shown as $\oplus$). The second B2U can be achieved with controlled SWAPs with a second register also containing two qubits (which is why it is shown as $\times$). The third B2U can again be achieved with controlled SWAPs, this time between the four qubits in the first two registers used for the unary and the third register of four qubits. The unary registers are used to output ϕ_j^1, ϕ_j^2, and ϕ_j^3 with CNOTs, and no erasure is used because it can be incorporated in the next output.

the end to output the phases for the various basis states, we can output these phases together with the final QROM on qubits $1 \ldots m$. Then, the cost of controlled SWAPs of the correct phase into an output register is

$$[2^{n-m} - 1]b. \tag{C5}$$

There is also a cost b for the rotation used for the phase. This gives a total cost of

$$2^m - 2 + [2^{n-m+1} + (m-1)]b. \tag{C6}$$

The optimal choice of m is then

$$m \approx (n + \log b + 1)/2. \tag{C7}$$

The leading term in the complexity is then $\sqrt{2^{n+3}b}$. This is only about 40% more than the complexity of a single phasing layer. For step 4, the erasure cost for the QROM is, to leading order, $2\sqrt{2^n}$, which is a smaller contribution to the complexity. Numerically, we find that it is possible to prepare a state of dimension 2^n using three phasing layers, but this cost is less than half the cost of three phasing layers, providing a significant improvement.

For the ancilla-qubit usage of this algorithm, there are 2^{m-1} qubits for the one-hot representation of qubits 1 to $m-1$, $2^{j-1}b$ qubits for the rotation angle(s) for qubit $m + j$ (for $j = 1$ to $n - m$), and then $2^{n-m}b$ qubits for the final phase rotations. That gives a total of

$$2^{m-1} + \sum_{j=1}^{n-m} 2^{j-1}b + 2^{n-m}b = 2^{n-m+1}b + 2^{m-1} - 1. \tag{C8}$$

Using the choice of m above, the total number of ancilla qubits is approximately $3\sqrt{2^{n-1}b}$. This ancilla-qubit usage is approximately $3/\sqrt{2} \approx 2.1$ times that for LKS or three phasing layers. In the standard form of the LKS approach, the maximum ancilla-qubit usage of approximately $\sqrt{2^n b}$ is in the final phasing step. That is also the ancilla-qubit usage for state preparation via three phasing layers. For our application, the number of ancilla qubits is less than used in the block encoding of the Hamiltonian and so does not increase the total qubit usage of the complete algorithm with state preparation and filtering.

As an example, consider the case in which $n = 20$ and $b = 16$ again. Then, the Toffoli cost via this approach is 14 510. In contrast, the cost of three diagonal-phasing operations is 30 642, which is a little more than double. Part of the reason for this is that it is necessary to erase the QROM multiple times. The cost of the original LKS approach for preparation would be 35 800 Toffolis. As another example, $n = 14$ and $b = 16$ gives a cost for this

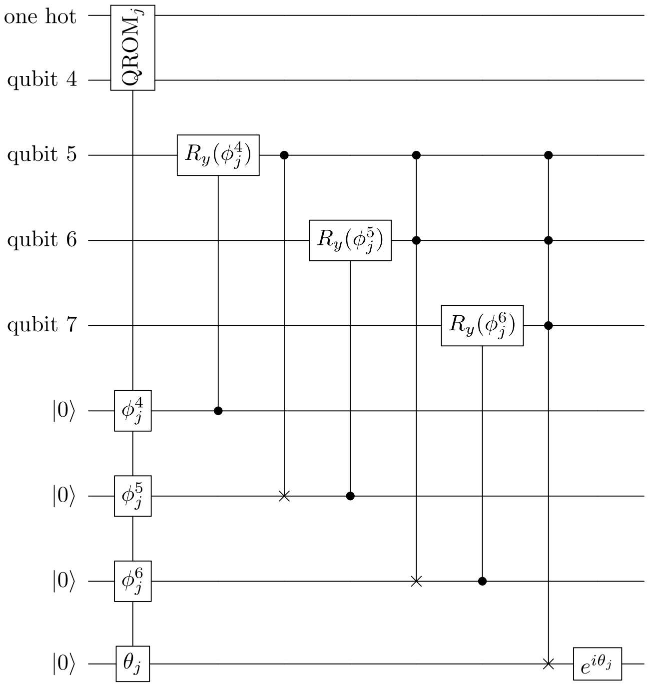

FIG. 21. An example of steps 2, 3, and 4 of the improved LKS-style preparation with $n = 7$. The top register is the one-hot register from Fig. 20 (shown there as three registers) storing the value of system qubits 1, 2, and 3. The next registers are system qubits 4 to $n = 7$ and then the registers below that are data-output registers. The initial QROM uses the value of that one-hot register together with the value in system qubit 4 to output the data needed for the remainder of the procedure. That is step 2. The angle ϕ_j^4 is then used to rotate qubit 5 and then the value in this qubit is used to SWAP the two possible values of ϕ_j^5, indicated by the ×. This register is shown as a single line for simplicity but has two subregisters of b qubits each containing the two possible rotations for qubit 6. We use a similar convention for the registers with ϕ_j^6 and θ_j, with these being composed of four and eight subregisters, respectively. Similarly for these registers, the × is used to indicate the controlled SWAPs to place the correct subregister in the output location. Step 3 is the selection of angles ϕ_j^k and rotations on qubits 5, 6, and 7. Then step 4 is the selection of θ_j, and the application of this phase indicated by $e^{i\theta_j}$.

method of 1918 Toffolis, versus 3780 for three layers of phases or 4446 for LKS.

In our application, the state preparation is used in the initial step of the MPS preparation, where the dimension of the state to be prepared is χd. The case $\chi = 4000$, $d = 4$ approximately corresponds to the example with $n = 14$ above, and for $b = 16$ the ancilla-qubit usage is less than 1100. Even increasing χ to 2^{13}, which is beyond that considered in this work, only about 1500 ancilla qubits are used. In contrast, the qubit usage for the block encoding of FeMoco in Table II is over 2000. Note that the qubit numbers provided in Table I are for three phasing layers. The Toffoli cost of this step in the MPS preparation is less than the rounding of the Toffoli costs reported in Table I.

These costings are assuming that the number of basis states is a power of 2. In the more general case, for QROM on qubits up to j, assuming a one-hot representation of qubits 1 to $j-1$ there should be

$$\left\lceil \frac{N_{\mathrm{un}}}{2^{n-j+1}} \right\rceil \tag{C9}$$

of these qubits. The number of possibilities to iterate through for the QROM on qubits up to j is

$$\left\lceil \frac{N_{\mathrm{un}}}{2^{n-j}} \right\rceil. \tag{C10}$$

If this is an even number, then it is exactly twice that in Eq. (C9). This means that for each one-hot qubit provided, a single Toffoli is used together with qubit j to distinguish the two cases. The Toffoli cost can alternatively be expressed as

$$\left\lceil \frac{N_{\mathrm{un}}}{2^{n-j}} \right\rceil - \left\lceil \frac{N_{\mathrm{un}}}{2^{n-j+1}} \right\rceil. \tag{C11}$$

Alternatively, if the value in Eq. (C10) is odd, then the maximum value encoded on qubits 1 to $j-1$ must correspond to the maximum value encoded on qubits 1 to j. That is, if the final one-hot qubit provided is equal to 1 (corresponding to the maximum value on qubits 1 to $j-1$), we know the value encoded on qubits 1 to j and no Toffoli is required. This means that in this case, the Toffoli cost is

$$\left\lceil \frac{N_{\mathrm{un}}}{2^{n-j+1}} \right\rceil - 1 = \left\lceil \frac{N_{\mathrm{un}}}{2^{n-j}} \right\rceil - \left\lceil \frac{N_{\mathrm{un}}}{2^{n-j+1}} \right\rceil. \tag{C12}$$

This means that the additional Toffoli cost for QROM on qubits 1 to j can be expressed as in Eq. (C11) in either case. As a result, the total Toffoli cost is equal to

$$\frac{N_{\mathrm{un}}}{2^{n-m}} - 2. \tag{C13}$$

That is, the Toffoli cost via this iterative process of using one-hot qubits to store the intermediate results of QROMs is just the same as the usual form of QROM. Moreover, the overall cost of the preparation should have leading order $\sqrt{8N_{\mathrm{un}}b}$. It is also possible to reduce the number of Toffolis slightly by an in-place binary-to-unary conversion, as in Ref. [42]. That requires significantly more Clifford gates and so would likely not be useful in practice.

APPENDIX D: HIGHER-ORDER APPROXIMATIONS FOR KAISER WINDOW

Here, we show how to use the properties of special functions to obtain the asymptotic results for the Kaiser window such as given in Eq. (49). First the integral can be given, using the Plancherel theorem, as

$$\int \frac{\sin^2\left(\sqrt{N^2\theta^2-(\pi\alpha)^2}\right)}{N^2\theta^2-(\pi\alpha)^2}\,d\theta = \frac{\pi}{2N}\int_{-1}^{1} I_0^2\left(\pi\alpha\sqrt{1-x^2}\right)\,dx. \tag{D1}$$

Ignoring the factor of $1/N$ for simplicity, this integral can be evaluated as

$$\frac{\pi}{2}\int_{-1}^{1} I_0^2\left(\pi\alpha\sqrt{1-x^2}\right)\,dx = \pi\int_0^1 I_0^2\left(\pi\alpha y\right)\frac{y\,dy}{\sqrt{1-y^2}} = \frac{\pi}{2}[I_0(2\pi\alpha)(2+\pi L_1(2\pi\alpha)) - \pi I_1(2\pi\alpha)L_0(2\pi\alpha)], \tag{D2}$$

where L_0 and L_1 are modified Struve functions. Expanding about $\alpha=\infty$ then gives

$$\frac{e^{2\pi\alpha}}{4\pi\alpha^{3/2}}\left(1+\frac{5}{2^4\pi\alpha}+\frac{129}{2^9\pi^2\alpha^2}+\frac{2655}{2^{13}\pi^3\alpha^3}+\frac{301\,035}{2^{19}\pi^4\alpha^4}+\frac{10\,896\,795}{2^{23}\pi^5\alpha^5}+\frac{961\,319\,205}{2^{29}\pi^6\alpha^6}+\mathcal{O}(\alpha^{-7})\right). \tag{D3}$$

For the integral over the tails, we can use

$$\begin{aligned}
2\int_{(\pi/N)\sqrt{1+\alpha^2}}^{\infty}\frac{\sin^2\sqrt{N^2\theta^2-\pi^2\alpha^2}}{N^2\theta^2-\pi^2\alpha^2}\,d\theta &= \frac{2}{N}\int_{\pi\sqrt{1+\alpha^2}}^{\infty}\frac{\sin^2\sqrt{\theta^2-\pi^2\alpha^2}}{\theta^2-\pi^2\alpha^2}\,d\theta\\
&= \frac{2}{N}\int_{\pi}^{\infty}\frac{\sin^2 x}{x\sqrt{x^2+\pi^2\alpha^2}}\,dx\\
&= \frac{1}{N}\int_{\pi}^{\infty}\frac{1-\cos 2x}{x\sqrt{x^2+\pi^2\alpha^2}}\,dx\\
&= \frac{\operatorname{arcsinh}(\alpha)}{\pi N\alpha} - \frac{1}{N}\int_{\pi}^{\infty}\frac{\cos 2x}{x\sqrt{x^2+\pi^2\alpha^2}}\,dx.
\end{aligned} \tag{D4}$$

For the remaining integral, we can use integration by parts to give

$$\begin{aligned}
&-\int_{\pi}^{\infty}\frac{\cos 2x}{x\sqrt{x^2+\pi^2\alpha^2}}\,dx\\
&\quad= -\left[\frac{\mathrm{Ci}(2x)}{\sqrt{x^2+\pi^2\alpha^2}}\right]_{\pi}^{\infty} - \int_{\pi}^{\infty}\frac{x\,\mathrm{Ci}(2x)}{(x^2+\pi^2\alpha^2)^{3/2}}\,dx\\
&\quad= \frac{\mathrm{Ci}(2\pi)}{\pi\sqrt{1+\alpha^2}} - \int_{\pi}^{\infty}\frac{x\,\mathrm{Ci}(2x)}{(x^2+\pi^2\alpha^2)^{3/2}}\,dx\\
&\quad= \frac{\mathrm{Ci}(2\pi)}{\pi\sqrt{1+\alpha^2}} + \left[\frac{\cos(2x)+2x\sin(2x)-4x^2\,\mathrm{Ci}(2x)}{8(x^2+\pi^2\alpha^2)^{3/2}}\right]_{\pi}^{\infty} + \int_{\pi}^{\infty}\frac{3x[\cos(2x)+2x\sin(2x)-4x^2\,\mathrm{Ci}(2x)]}{8(x^2+\pi^2\alpha^2)^{5/2}}\,dx\\
&\quad= \frac{\mathrm{Ci}(2\pi)}{\pi\sqrt{1+\alpha^2}} + \frac{4\pi^2\,\mathrm{Ci}(2\pi)-1}{8\pi^3(1+\alpha^2)^{3/2}} + \int_{\pi}^{\infty}\frac{3x[\cos(2x)+2x\sin(2x)-4x^2\,\mathrm{Ci}(2x)]}{8(x^2+\pi^2\alpha^2)^{5/2}}\,dx\\
&\quad= \frac{\mathrm{Ci}(2\pi)}{\pi\sqrt{1+\alpha^2}} + \frac{4\pi^2\,\mathrm{Ci}(2\pi)-1}{8\pi^3(1+\alpha^2)^{3/2}} - \left[\frac{3[(2x^2-3)\cos(2x)-2x(2x^2+3)\sin(2x)+8x^4\mathrm{Ci}(2x)]}{64(x^2+\pi^2\alpha^2)^{5/2}}\right]_{\pi}^{\infty}\\
&\qquad - \int_{\pi}^{\infty}\frac{15x[(2x^2-3)\cos(2x)-2x(2x^2+3)\sin(2x)+8x^4\mathrm{Ci}(2x)]}{64(x^2+\pi^2\alpha^2)^{9/2}}\,dx\\
&\quad= \frac{\mathrm{Ci}(2\pi)}{\pi\sqrt{1+\alpha^2}} + \frac{4\pi^2\,\mathrm{Ci}(2\pi)-1}{8\pi^3(1+\alpha^2)^{3/2}} + \frac{3[(2\pi^2-3)+8\pi^4\mathrm{Ci}(2\pi)]}{64\pi^5(1+\alpha^2)^{5/2}}\\
&\qquad - \int_{\pi}^{\infty}\frac{15x[(2x^2-3)\cos(2x)-2x(2x^2+3)\sin(2x)+8x^4\mathrm{Ci}(2x)]}{64(x^2+\pi^2\alpha^2)^{7/2}}\,dx,
\end{aligned} \tag{D5}$$

where Ci is the cosine integral. Note that the final integral here is of order α^{-7}, and repeating the integration by parts makes the remaining integral higher and higher order in α. Dividing by the above asymptotic expression for the normalization factor in Eq. (D3) gives, with $C_\alpha = \ln(2\alpha) + \mathrm{Ci}(2\pi)$,

$$\begin{aligned}\delta = 4\sqrt{\alpha}e^{-2\pi\alpha}\Bigg[&C_\alpha - \frac{5C_\alpha}{16\pi\alpha} - \frac{(64 + 79C_\alpha - 128\pi^2)}{2^9\pi^2\alpha^2} + \frac{5(64 - 323C_\alpha - 128\pi^2)}{2^{13}\pi^3\alpha^3}\\ &- \frac{195\,253C_\alpha + 128(497 - 994\pi^2 + 384\pi^4)}{2^{19}\pi^4\alpha^4} - \frac{7\,579\,195C_\alpha - 640(899 - 1798\pi^2 + 384\pi^4)}{2^{23}\pi^5\alpha^5}\\ &- \frac{2\,131\,315\,809C_\alpha + 64(6\,514\,017 - 13\,028\,034\pi^2 + 4\,733\,184\pi^4 - 655\,360\pi^6)}{2^{28}3\pi^6\alpha^6} + \mathcal{O}(\alpha^{-7})\Bigg]. \qquad \text{(D6)}\end{aligned}$$

Retaining the leading-order terms gives the result in Eq. (49).

Given error δ, expanding in a series solution for α yields

$$\alpha = \frac{\ln(1/\delta)}{2\pi} + \frac{\ln(8\ln(4/\delta)/\pi)}{4\pi} + \frac{1}{2\pi}\ln\left[\mathrm{Ci}(2\pi) + \ln(\ln(4/\delta)/\pi)\right] + \mathcal{O}\left(\frac{\ln\ln(1/\delta)}{\ln(1/\delta)}\right). \qquad \text{(D7)}$$

The leading-order terms here only differ from Eq. (47) in that there is an extra $\mathrm{Ci}(2\pi)$ in the third term, so the previous approximation given can be expected to be accurate. The expression for N is then that given in Eq. (50).

The performance of the Kaiser window can be improved by adjusting the width from $\pi\sqrt{1+\alpha^2}$ to $\pi\sqrt{\Delta^2+\alpha^2}$ for some general $\Delta \neq 1$. Then, the integral over the tails can be adjusted to

$$\begin{aligned}2\int_{(\pi/N)\sqrt{\Delta^2+\alpha^2}}^{\infty} \frac{\sin^2\sqrt{N^2\phi^2 - \pi^2\alpha^2}}{(N^2\phi^2 - \pi^2\alpha^2)}\,d\phi &= \frac{2}{N}\int_{\pi\sqrt{\Delta^2+\alpha^2}}^{\infty} \frac{\sin^2\sqrt{\phi^2 - \pi^2\alpha^2}}{(\phi^2 - \pi^2\alpha^2)}\,d\phi\\ &= \frac{2}{N}\int_{\pi\Delta}^{\infty} \frac{\sin^2 x}{x\sqrt{x^2+\pi^2\alpha^2}}\,dx\\ &= \frac{1}{N}\int_{\pi\Delta}^{\infty} \frac{1-\cos 2x}{x\sqrt{x^2+\pi^2\alpha^2}}\,dx\\ &= \frac{\operatorname{arcsinh}(\alpha/\Delta)}{\pi N\alpha} - \frac{1}{N}\int_{\pi\Delta}^{\infty} \frac{\cos 2x}{x\sqrt{x^2+\pi^2\alpha^2}}\,dx. \qquad \text{(D8)}\end{aligned}$$

Integration by parts gives

$$\begin{aligned}&-\int_{\pi\Delta}^{\infty} \frac{\cos 2x}{x\sqrt{x^2+\pi^2\alpha^2}}\,dx\\ &= -\left[\frac{\mathrm{Ci}(2x)}{\sqrt{x^2+\pi^2\alpha^2}}\right]_{\pi\Delta}^{\infty} - \int_{\pi\Delta}^{\infty} \frac{x\,\mathrm{Ci}(2x)}{(x^2+\pi^2\alpha^2)^{3/2}}\,dx\\ &= \frac{\mathrm{Ci}(2\pi\Delta)}{\pi\sqrt{\Delta^2+\alpha^2}} - \int_{\pi\Delta}^{\infty} \frac{x\,\mathrm{Ci}(2x)}{(x^2+\pi^2\alpha^2)^{3/2}}\,dx\\ &= \frac{\mathrm{Ci}(2\pi\sqrt{\Delta})}{\pi\sqrt{\Delta^2+\alpha^2}} + \left[\frac{\cos(2x) + 2x\sin(2x) - 4x^2\,\mathrm{Ci}(2x)}{8(x^2+\pi^2\alpha^2)^{3/2}}\right]_{\pi\Delta}^{\infty} + \int_{\pi\Delta}^{\infty} \frac{3x[\cos(2x) + 2x\sin(2x) - 4x^2\,\mathrm{Ci}(2x)]}{8(x^2+\pi^2\alpha^2)^{5/2}}\,dx\\ &= \frac{\mathrm{Ci}(2\pi\Delta)}{\pi\sqrt{\Delta^2+\alpha^2}} + \frac{4\pi^2\Delta^2\,\mathrm{Ci}(2\pi) - \cos(2\pi\Delta) - 2\pi\Delta\sin(2\pi\Delta)}{8\pi^3(\Delta^2+\alpha^2)^{3/2}}\\ &\quad + \int_{\pi\Delta}^{\infty} \frac{3x[\cos(2x) + 2x\sin(2x) - 4x^2\,\mathrm{Ci}(2x)]}{8(x^2+\pi^2\alpha^2)^{5/2}}\,dx\end{aligned}$$

$$
\begin{aligned}
&= \frac{\mathrm{Ci}(2\pi\Delta)}{\pi\sqrt{\Delta^2+\alpha^2}} + \frac{4\pi^2\Delta^2\,\mathrm{Ci}(2\pi) - \cos(2\pi\Delta) - 2\pi\Delta\sin(2\pi\Delta)}{8\pi^3(\Delta^2+\alpha^2)^{3/2}} \\
&\quad - \left[\frac{3[(2x^2-3)\cos(2x) - 2x(2x^2+3)\sin(2x) + 8x^4\mathrm{Ci}(2x)]}{64(x^2+\pi^2\alpha^2)^{5/2}}\right]_{\pi\Delta}^{\infty} \\
&\quad - \int_{\pi\Delta}^{\infty} \frac{15x[(2x^2-3)\cos(2x) - 2x(2x^2+3)\sin(2x) + 8x^4\mathrm{Ci}(2x)]}{64(x^2+\pi^2\alpha^2)^{9/2}}\,dx \\
&= \frac{\mathrm{Ci}(2\pi\Delta)}{\pi\sqrt{\Delta^2+\alpha^2}} + \frac{4\pi^2\Delta^2\,\mathrm{Ci}(2\pi) - \cos(2\pi\Delta) - 2\pi\Delta\sin(2\pi\Delta)}{8\pi^3(\Delta^2+\alpha^2)^{3/2}} \\
&\quad + \frac{3[(2\pi^2\Delta^2-3)\cos(2\pi\Delta) - 2\pi\Delta(2\pi^2\Delta^2+3)\sin(2\pi\Delta) + 8\pi^4\Delta^4\mathrm{Ci}(2\pi\Delta)]}{64\pi^5(\Delta^2+\pi^2\alpha^2)^{5/2}} \\
&\quad - \int_{\pi\Delta}^{\infty} \frac{15x[(2x^2-3)\cos(2x) - 2x(2x^2+3)\sin(2x) + 8x^4\mathrm{Ci}(2x)]}{64(x^2+\pi^2\alpha^2)^{7/2}}\,dx.
\end{aligned}
\tag{D9}
$$

Dividing by the normalization gives the error as

$$
\begin{aligned}
\delta = 4\sqrt{\alpha}e^{-2\pi\alpha}\Bigg[&C_{\alpha,\Delta} - \frac{5C_{\alpha,\Delta}}{16\pi\alpha} - \frac{(64\cos(2\pi\Delta) + 128\pi\Delta\sin(2\pi\Delta) + 79C_{\alpha,\Delta} - 128\pi^2\Delta^2)}{2^9\pi^2\alpha^2} \\
&+ \frac{5(64\cos(2\pi\Delta) + 128\pi\Delta\sin(2\pi\Delta) - 323C_{\alpha,\Delta} - 128\pi^2\Delta^2)}{2^{13}\pi^3\alpha^3} \\
&- \frac{195\,253C_{\alpha,\Delta} + 128[2\pi^2\Delta^2(79+192\pi^2\Delta^2) + (497-1152\pi^2\Delta^2)\cos(2\pi\Delta) + 2\pi\Delta(497-384\pi^2\Delta^2)\sin(2\pi\Delta)]}{2^{19}\pi^4\alpha^4} \\
&- \frac{7\,579\,195C_{\alpha,\Delta} + 640[2\pi^2\Delta^2(323-192\pi^2\Delta^2) - (899-1152\pi^2\Delta^2)\cos(2\pi\Delta) - 2\pi\Delta(899-384\pi^2\Delta^2)\sin(2\pi\Delta)]}{2^{23}\pi^5\alpha^5} \\
&+ \mathcal{O}(\alpha^{-6})\Bigg],
\end{aligned}
\tag{D10}
$$

where $C_{\alpha,\Delta} = \ln(2\alpha/\Delta) + \mathrm{Ci}(2\pi\Delta)$. Retaining the leading-order terms yields the expression given in Eq. (51).

If we take $c = \pi\sqrt{\Delta^2+\alpha^2}$ and expand in a series in c, then we obtain

$$
\delta = 4\sqrt{c/\pi}e^{-2c}\Bigg[C_{c,\Delta} - \frac{(5-16\pi^2\Delta^2)C_{c,\Delta}}{16c} \tag{D11}
$$

$$
\begin{aligned}
&- \frac{[64\cos(2\pi\Delta) + 128\pi\Delta\sin(2\pi\Delta) + (79+288\pi^2\Delta^2-256\pi^4\Delta^4)C_{c,\Delta} - 128\pi^2\Delta^2]}{2^9c^2} \\
&+ \frac{64(5-16\pi^2\Delta^2)[\cos(2\pi\Delta) + 2\pi\Delta\sin(2\pi\Delta)] - (1615+1904\pi^2\Delta^2+1280\pi^4\Delta^4-2^{12}\pi^6\Delta^6/3)C_{c,\Delta}}{2^{13}c^3} \\
&+ \frac{640\pi^2\Delta^2 - 2^{11}\pi^4\Delta^4}{2^{13}c^3} + \mathcal{O}(c^{-4})\Bigg].
\end{aligned}
\tag{D12}
$$

Retaining just the leading-order terms gives

$$
\delta = 4C_{c,\Delta}\sqrt{c/\pi}e^{-2c}\left[1 - \frac{(5-16\pi^2\Delta^2)}{16c} + \mathcal{O}(c^{-2})\right], \tag{D13}
$$

where $C_{c,\Delta} = \ln(2c/\pi\Delta) + \mathrm{Ci}(2\pi\Delta)$. Now, $C_{c,\Delta}$ takes its minimum value at $\Delta = 1$ but the second term in square brackets above increases with Δ. This implies that for c not too large, the optimal value of Δ will be less than 1, but in the limit of large c the optimal value of Δ approaches 1. That is indeed what is found numerically. This also implies that the first terms for N in terms of ϵ in Eq. (50) are appropriate even when optimizing Δ. In fact, optimizing Δ would make the third term *larger*, with the improvement only in the higher-order terms. This is because the third term is equivalent to $\ln(C_{c,\Delta})/2\epsilon$. It is more accurate to continue using the third term as given, since the true value of N is reduced when optimizing Δ, and using a larger value of $C_{c,\Delta}$ in this term would make the estimate of N larger.

APPENDIX E: SLEPIAN EXPANSION

Here, we explain the details of how to derive the higher-order terms to correct Eq. (4.4) of Slepian [22] for the case $n = 0$. In the notation of that work, $\psi_{m,n}(x)$ is a prolate spheroidal wave function of order m, which is a solution of

$$\frac{d}{dx}(1-x^2)\frac{d\psi}{dx} + \left(\chi - c^2x^2 - \frac{m^2}{1-x^2}\right)\psi = 0, \tag{E1}$$

for an eigenvalue $\chi_{m,n}$. That is, $\psi_{0,0}(x)$ is alternative notation for the function $\mathrm{PS}_{0,0}(c,x)$. Then, for the zeroth-order case, λ_n satisfies

$$\lambda_n\psi_{0,n}(x) = \frac{c}{\pi}\int_{-1}^{1} dz\ \psi_{0,n}(y)\,\mathrm{sinc}(c(x-z)). \tag{E2}$$

This means that λ_n corresponds to the confidence level for a phase measurement. Since the λ_n are given in descending order, we take $n = 0$ for the minimum-error solution. In the notation of Ref. [22], c is the same as we use here, and there are many other quantities such as $P(c), Q(c)$ that are used as intermediate steps in the calculation. We will use those without further definition, as all that is needed is their values for use in deriving the final result.

We start from Eq. (4.3) of Ref. [22], which is

$$\frac{1}{\lambda_n}\frac{\partial\lambda_n}{\partial c} = \frac{2}{c}\left[\psi_{0,n}(1)\right]^2. \tag{E3}$$

According to the expression below Eq. (4.3) of Ref. [22], $[\psi_{0,0}(1)]^2 = N_{0,0}^2 k_3^2$. Now, k_3 is given in Eq. (1.12) of Ref. [22], as

$$k_3 = e^{-c}c^{(l+1)/2}2^{(3l+2)/2}\sqrt{\pi}P(c)Q(c). \tag{E4}$$

Now $l = n - m$, as per the expression below Eq. (1.10) of Ref. [22], and $n \geq m \geq 0$, so with $n = 0$ we have $m = l = 0$. Therefore the expression for k_3 simplifies to

$$k_3 = 2e^{-c}\sqrt{\pi c}\,P(c)Q(c). \tag{E5}$$

For $m = 0$, we have $Q(c) = 1$, according to the explanation below Eq. (1.15) of Ref. [22]. The function $P(c)$ is given in terms of g coefficients in Eq. (1.14) of Ref. [22], which is

$$P(c) = \frac{1 + g_1^1/c + g_2^1/c^2 + \cdots}{1 + g_1^2/c + g_2^2/c^2 + \cdots}. \tag{E6}$$

The values of g are given as per Ref. [22, Table IV, p. 103], which have the simplified form for $l = m = 0$

$$\begin{aligned}
g_1^1 &= -\frac{24}{2^8},\\
g_2^1 &= -\frac{51\,840}{3!\,2^{17}},\\
g_3^1 &= -\frac{857\,226\,240}{3!\,6!\,2^{21}},\\
g_1^2 &= \frac{64}{2^8},\\
g_2^2 &= \frac{165\,888}{3!\times 2^{17}},\\
g_3^2 &= \frac{902\,430\,720}{6!\,2^{22}}.
\end{aligned} \tag{E7}$$

Next, $N_{0,0}^2$ is given by Eq. (1.16) of Ref. [22] as (taking $l = m = 0$)

$$\frac{1}{N_{0,0}^2} = \sqrt{\pi/c}\left(1 + \frac{3}{2^7c^2} + \cdots\right). \tag{E8}$$

To provide a further term, we can use Eq. (1.6) of Ref. [22] (with $m = n = 0$)

$$\psi_{0,0}^1(x) = D_0(x\sqrt{2c}) + \sum_{j\geq 1}\left(\frac{1}{2c}\right)^j\sum_{k=-2j}^{2j}A_k^jD_{2k}(x\sqrt{2c}), \tag{E9}$$

where $D_\nu(x)$ is called the Weber function in Ref. [22] but is more commonly called the parabolic cylinder function (ParabolicCylinderD in *Mathematica*). Using that expression together with the values of A_k^j provided by the recurrence relation in Ref. [22] gives

$$\begin{aligned}
\hat{\psi}_{0,0}(0) &= \psi_{0,0}^1(0)\\
&= 1 - \frac{3}{16c} - \frac{63}{512c^2} - \frac{1449}{8192c^3} + \mathcal{O}(c^{-4}).
\end{aligned} \tag{E10}$$

An alternative formula for the normalized form of $\psi_{0,0}(x)$ is given by Eq. (8.88) of Ref. [63]. That gives, for the normalized function,

$$\begin{aligned}
\psi_{0,0}(0) &= \left(\frac{c}{\pi}\right)^{1/4}\\
&\quad\times\left(1 - \frac{3}{16c} - \frac{159}{2048c^2} - \frac{8049}{65\,536c^3} + \mathcal{O}(c^{-4})\right).
\end{aligned} \tag{E11}$$

Because $\psi_{0,0}(0) = N_{0,0}\hat{\psi}_{0,0}(0)$, the normalization satisfies

$$\frac{1}{N_{0,0}^2} = \left(\frac{\hat{\psi}_{0,0}(0)}{\psi_{0,0}(0)}\right)^2 = \sqrt{\pi/c}\left(1 + \frac{3}{2^7c^2} + \frac{15}{2^8c^3} + \mathcal{O}(c^{-4})\right). \quad \text{(E12)}$$

This gives the extra term not given directly in Ref. [22]. We then obtain

$$\frac{2}{c}\left[\psi_{0,0}(1)\right]^2 = \frac{2}{c}N_{0,0}^2k_3^2 = 8\sqrt{\pi c}\, e^{-2c}\left(1 - \frac{11}{16c} - \frac{147}{512c^2} - \frac{3749}{8192c^3} - \cdots\right). \quad \text{(E13)}$$

Equation (E3) can be rewritten as

$$\frac{d}{dc}\log\lambda_0 = \frac{2}{c}\left[\psi_{0,0}(1)\right]^2, \quad \text{(E14)}$$

and $\lim_{c\to\infty}\lambda_0 = 1$, so

$$-\log\lambda_0 = \int_c^\infty \frac{2}{c'}\left[\psi_{0,0}(1)\right]^2 dc'. \quad \text{(E15)}$$

Evaluating that integral gives

$$-\log\lambda_0 \approx 4\sqrt{\pi c}\, e^{-2c}\left(1 + \frac{1985}{1536c} - \frac{3749}{6144c^2}\right) - \frac{2657\pi\,\mathrm{erfc}(\sqrt{2c})}{192\sqrt{2}}. \quad \text{(E16)}$$

It is appropriate to make the approximation $\delta = 1 - \lambda_0 \approx -\log\lambda_0$, since the difference is of order ce^{-4c}. Expanding an asymptotic series for the erfc function then gives the error outside the confidence interval as

$$\delta \approx 4\sqrt{\pi c}\, e^{-2c}\left(1 - \frac{7}{16c} - \frac{91}{2^9c^2} - \frac{2657}{2^{13}c^3}\right). \quad \text{(E17)}$$

Next, we describe how to derive Eq. (56). In the discrete case, the probability distribution for the phase error is

$$\frac{1}{2\pi}\sum_{n,m=-N}^{N} f(n)f(m)e^{i(n-m)\theta}. \quad \text{(E18)}$$

In the continuous limit, we replace $x = n/(N+1/2)$ and $z = m/(N+1/2)$ and use $\vartheta = (N+1/2)\theta$, to give the probability distribution for the error

$$\frac{1}{2\pi}\int_{-1}^{1}dx\int_{-1}^{1}dz\, f(x)f(z)\, e^{i(x-z)\vartheta}, \quad \text{(E19)}$$

with the convention that the continuous function $f(x)$ is normalized over the interval $[-1,1]$. Then, the integral for the confidence level is

$$\frac{1}{2\pi}\int_{-c}^{c}d\vartheta\int_{-1}^{1}dx\int_{-1}^{1}dz\, f(x)f(z)\, e^{i(x-z)\vartheta} = \frac{c}{\pi}\int_{-1}^{1}dx\int_{-1}^{1}dz\, f(x)f(z)\,\mathrm{sinc}(c(x-z)). \quad \text{(E20)}$$

The solution for $f(x)$ with maximum confidence level then corresponds to an eigenfunction of the maximum eigenvalue, so that

$$(1-\delta)f(x) = \frac{c}{\pi}\int_{-1}^{1}dz\, f(z)\,\mathrm{sinc}(c(x-z)). \quad \text{(E21)}$$

In particular, we have

$$(1-\delta)f(0) = \frac{c}{\pi}\int_{-1}^{1}dz\, f(z)\,\mathrm{sinc}(cz). \quad \text{(E22)}$$

We can use this expression for the function $f(z) = \mathrm{PS}_{0,0}(c,z)$ to give Eq. (56). The function $\mathrm{PS}_{0,0}(c,z)$ is not normalized to 1, but that is unimportant because it appears on both sides of the equation and so the normalization cancels.

APPENDIX F: METHODS OF CALCULATING PROLATE SPHEROIDAL FUNCTIONS

The prolate spheroidal functions are given in various mathematical software programs as follows:

1. In *Mathematica*, $\mathrm{PS}_{0,0}(c,z)$ is given as SpheroidalPS $[0,0,c,z]$ and $S^1_{0,0}(c,1)$ is given as SpheroidalS1 $[0,0,c,1]$. These are normalized according to the Meixner-Schäfke scheme.
2. MATLAB gives the discrete prolate spheroidal sequences (i.e., for finite N) via the function DPSS.
3. In PYTHON, SCIPY.SIGNAL.WINDOWS.DPSS has similar functionality as DPSS in MATLAB. PYTHON also provides SCIPY.SPECIAL with PRO_ANG1 and PRO_RAD1 for the angular and radial prolate spheroidal functions, respectively. Unfortunately, PRO_RAD1 only outputs nan (not a number) in our testing, making it unusable.

Very similar results to those obtained in *Mathematica* are obtained using DPSS in MATLAB, which is a useful independent verification. To be more specific, consider the control state for phase estimation of the form

$$\sum_{n=-N}^{N} f(n)\,|n\rangle\,. \quad \text{(F1)}$$

Applying a phase shift and taking an inner product with the phase state gives

$$\frac{1}{\sqrt{2N+1}}\sum_{n=-N}^{N} f(n)e^{in(\phi-\hat{\phi})}. \tag{F2}$$

The inner product squared is

$$\frac{1}{2N+1}\sum_{n,m=-N}^{N} f(n)f(m)e^{i(n-m)\theta}, \tag{F3}$$

where we have replaced $\phi-\hat{\phi}$ with θ. The usual convention for phase measurements is to consider a slightly different normalization convention with a continuous range of θ values, so the probability distribution for the error is

$$\frac{1}{2\pi}\sum_{n,m=-N}^{N} f(n)f(m)e^{i(n-m)\theta}. \tag{F4}$$

Integrating θ over $[-\pi,\pi]$ then gives 1.

To obtain the probability in the confidence interval $[-c/(N+1/2), c/(N+1/2)]$, the integral is

$$\frac{1}{2\pi}\int_{-c/(N+1/2)}^{c/(N+1/2)} d\theta \sum_{n,m=-N}^{N} f(n)f(m)e^{i(n-m)\theta} = \frac{c}{\pi(N+1/2)}\sum_{n,m=-N}^{N} f(n)f(m)\,\mathrm{sinc}\left(\frac{c(n-m)}{N+1/2}\right). \tag{F5}$$

Given $f(n)$ from DPSS, this expression can be used to determine the value of δ for given N. The function DPSS also gives $1-\delta$ as an output, which can be used instead of performing the explicit sum. The relative error for the value of δ estimated with various values of N is shown in Fig. 22. It can be seen that the results using DPSS are very accurate, even for moderate values of N. The results for $c > 4\pi$ are obtained less accurately, because DPSS gives $1-\delta$ to double precision accuracy and so cannot give accurate values for δ below about 10^{-14}. Nearly identical results are obtained using SCIPY.SIGNAL.WINDOWS.DPSS in PYTHON, although the function breaks down for $N > 2^{15}$.

Note that DPSS is giving the error for a specific value of N, so the relative errors seen in Fig. 22 correspond to the difference between using the continuous window and the sampling of the continuous window for finite N. For phase estimates of relevance to quantum computing, N would be well above 2^{10} and even for that value the relative error is less than about 0.01%. This difference is less than the significant figures usually reported for complexities and so it is reasonable to ignore it in the analysis.

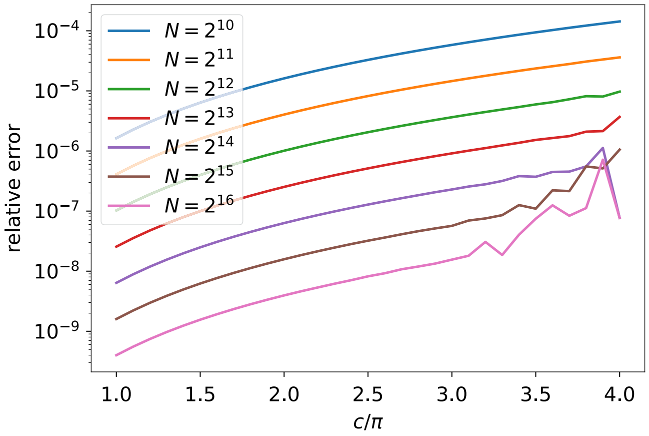

FIG. 22. The relative error for the values of δ estimated with various values of N using DPSS in MATLAB. The lines show the results using $N = 2^{10}$ to $N = 2^{16}$ from top to bottom.

These results can be further checked by determining the maximum eigenvalue of a matrix. The maximization of the confidence level in Eq. (F5) corresponds to finding the maximum eigenvalue of a matrix with entries

$$A_{n,m} = \frac{c}{\pi(N+1/2)}\mathrm{sinc}\left(\frac{c(n-m)}{N+1/2}\right), \tag{F6}$$

for n and m in the range $-N,\ldots,N$. The maximum eigenvalue of this matrix can be determined from the spectral norm. Numerical testing with a range of values of c and N yields results equal to those from DPSS within numerical precision.

APPENDIX G: COMBINATIONS OF EXCITED STATES

To show that the probability of error should not be increased for combinations of excited states with different β, let us first consider a weighted linear combination of excited states, so that $\delta_1 = s\delta_1^a + (1-s)\delta_1^b$ and, similarly, $\delta_2 = s\delta_2^a + (1-s)\delta_2^b$ for δ_1^a, δ_2^a, δ_1^b, and δ_2^b, the values for the individual excited states. It turns out that the maximum error of probability will be for either $s=0$ or $s=1$. For this result, we use the condition $\delta_2 \le 1-\delta_1$ (for each of the excited states). This condition is satisfied for these measurements because $1-\delta_1$ is the probability of the measurement result being *less than* $E_0+\epsilon$ and δ_2 is the probability of the estimate being less than $E_0-\epsilon$ (which of course must be less).

Lemma G1. If $P_{\mathrm{err}}(s)$ is given by, for $n \in \mathbb{N}$,

$$\begin{aligned} P_{\mathrm{err}}(s) &:= [p\delta/2 + (1-p)\delta_1]^n \\ &\quad + 1 - \{1 - [p\delta/2 + (1-p)\delta_2]\}^n, \\ \delta_1 &:= s\delta_1^a + (1-s)\delta_1^b, \\ \delta_2 &:= s\delta_2^a + (1-s)\delta_2^b, \end{aligned} \tag{G1}$$

and $\delta_2^a \leq 1-\delta_1^a$, $\delta_2^b \leq 1-\delta_1^b$, $p \in (0,1)$, $\delta < 1$, then

$$\max_{s\in[0,1]} P_{\mathrm{err}}(s) = \max\left(P_{\mathrm{err}}(0), P_{\mathrm{err}}(1)\right). \tag{G2}$$

Proof. If there were a larger probability of error for some intermediate value of s, then we would need the first derivative of $P_{\mathrm{err}}(s)$ to pass through zero (so that there is a turning point), with the second derivative being negative (so that there is a maximum). The first derivative gives

$$\begin{aligned}\frac{dP_{\mathrm{err}}}{ds} &= n(1-p)(\delta_1^a-\delta_1^b)[p\delta/2+(1-p)\delta_1]^{n-1}\\ &\quad + n(1-p)(\delta_2^a-\delta_2^b)\{1-[p\delta/2+(1-p)\delta_2]\}^{n-1},\end{aligned} \tag{G3}$$

so the first derivative being zero implies

$$\frac{\delta_1^b-\delta_1^a}{\delta_2^a-\delta_2^b} = \frac{\{1-[p\delta/2+(1-p)\delta_2]\}^{n-1}}{[p\delta/2+(1-p)\delta_1]^{n-1}}. \tag{G4}$$

The conditions $\delta_2^a \leq 1-\delta_1^a$ and $\delta_2^b \leq 1-\delta_1^b$ imply that $\delta_2 > 1-\delta_1$, which together with $p\in(0,1)$ and $\delta<1$ gives

$$\begin{aligned}& p\delta+(1-p)\delta_2 < p+(1-p)(1-\delta_1),\\ & p\delta/2+(1-p)\delta_2 < 1-p\delta/2-(1-p)\delta_1,\\ & 1-[p\delta/2+(1-p)\delta_2] > p\delta/2+(1-p)\delta_1,\\ & \frac{1-[p\delta/2+(1-p)\delta_2]}{p\delta/2+(1-p)\delta_1} > 1,\\ & \frac{\{1-[p\delta/2+(1-p)\delta_2]\}^{n-1}}{[p\delta/2+(1-p)\delta_1]^{n-1}} > 1.\end{aligned} \tag{G5}$$

Therefore, if we have a zero derivative, then both the left and right sides of Eq. (G4) are greater than 1. Squaring the left-hand side of Eq. (G4) increases it, whereas taking the right-hand side to the power of $(n-2)/(n-1)$ decreases it, implying that

$$\left(\frac{\delta_1^b-\delta_1^a}{\delta_2^a-\delta_2^b}\right)^2 > \frac{\{1-[p\delta/2+(1-p)\delta_2]\}^{n-2}}{[p\delta/2+(1-p)\delta_1]^{n-2}}. \tag{G6}$$

Now, the second derivative gives

$$\begin{aligned}\frac{d^2P_{\mathrm{err}}}{ds^2} &= n(n-1)(p-1)^2(\delta_1^a-\delta_1^b)^2\\ &\quad \times [p\delta/2+(1-p)\delta_1]^{n-2}\\ &\quad - n(n-1)(p-1)^2(\delta_2^a-\delta_2^b)^2\\ &\quad \times \{1-[p\delta/2+(1-p)\delta_2]\}^{n-2},\end{aligned} \tag{G7}$$

which must be *positive* given Eq. (G6). Therefore, either there is no turning point for $s\in[0,1]$ or there is a minimum; in either case, the maximum must be either for $s=0$ or $s=1$. ■

Thus we find that it is impossible to obtain a larger error probability by taking a combination of two excited states. Moreover, this argument shows that it is impossible to obtain a larger error probability with a combination of any number of excited states. To see that result, we can use δ_1^a, δ_2^a and δ_1^b, δ_2^b to be the probabilities resulting from disjoint sets of eigenstates. The reasoning is that $\delta_2 < 1-\delta_1$ holds regardless of the combination of excited states, because the probability of a measurement result below $E_0+\epsilon$ must always be greater than the probability of a result below $E_0-\epsilon$. Thus, for any combination of excited states, we can always split the set into two such that one of the subsets gives at least as large a value of P_{err}. Repeating this process gives a single excited state with at least as large an error probability. Thus we can obtain the maximum error probability with a single excited state and this result upper bounds the error probability for any spectrum of excited states.

APPENDIX H: FURTHER IMPROVEMENT OF BINARY-SEARCH APPROACH

1. Optimizing the shrinking factor

In the binary-search algorithm that we have discussed above, we have shrunk the interval in which λ_0 is located by 2/3 in each iteration. This is, however, not the optimal shrinking factor. If we shrink by a factor of ω instead of 2/3 at each search step, then using the leading order in the asymptotic expression for the Kaiser and Slepian windows in Eq. (50), we need to use W

$$\begin{aligned}&\underbrace{\frac{1}{1-\omega}}_{\text{search steps}} \times \underbrace{\frac{\omega}{2(2\omega-1)}\frac{\lambda}{\epsilon}\ln\left(\frac{16}{p}\right)}_{\text{QPE}}\\ &\times \underbrace{\frac{4}{3\sqrt{p}}\ln\left(\frac{\log_{1/\omega}(\lambda/\epsilon)}{q}\right)}_{\text{amplitude estimation}}\\ &= \frac{4\omega}{3(2\omega-1)(1-\omega)}\frac{\lambda}{\sqrt{p}\epsilon}\ln\left(\frac{4}{\sqrt{p}}\right)\ln\left(\frac{\log_{1/\omega}(\lambda/\epsilon)}{q}\right)\end{aligned} \tag{H1}$$

times.

We will explain how we have arrived at the above expressions. For each binary-search step, the two cases that we wish to distinguish are now

$$\lambda_0 > (1-\omega)\lambda_L+\omega\lambda_R \quad \text{or} \quad \lambda_0 < \omega\lambda_L+(1-\omega)\lambda_R. \tag{H2}$$

If we have eliminated the first case, then the value of λ_R is mapped as

$$\lambda_R \mapsto (1-\omega)\lambda_L + \omega\lambda_R, \tag{H3}$$

with λ_L unchanged, so the range of values is mapped as

$$\lambda_R - \lambda_L \mapsto (1-\omega)\lambda_L + \omega\lambda_R - \lambda_L = \omega(\lambda_R - \lambda_L). \tag{H4}$$

The reduction in the range is equivalent if we eliminate $\lambda_0 < \omega\lambda_L + (1-\omega)\lambda_R$.

We now need phase error at most

$$\frac{(2\omega-1)(\lambda_R-\lambda_L)}{2\lambda} \le \frac{1}{2}\left[\arccos\left(\frac{\omega\lambda_L + (1-\omega)\lambda_R}{\lambda}\right) - \arccos\left(\frac{(1-\omega)\lambda_L + \omega\lambda_R}{\lambda}\right)\right]. \tag{H5}$$

At each step the range is shrunk by ω, and we start with $\lambda_R - \lambda_L = 2\lambda$, so at step j the error is

$$(2\omega-1)\omega^{j-1}. \tag{H6}$$

The number of queries for the QPE circuit is then approximately, using Eq. (59),

$$\frac{1}{2(2\omega-1)}(1/\omega)^{j-1}\ln(1/\delta_1). \tag{H7}$$

Now, the number of steps L is calculated so that

$$(1/\omega)^L\lambda = \epsilon. \tag{H8}$$

If we sum the above cost over $j = 1$ to L, then we obtain

$$\frac{1}{2(2\omega-1)} \times \frac{(1/\omega)^L - 1}{1/\omega - 1}\ln(1/\delta_1) \approx \frac{\omega}{2(2\omega-1)(1-\omega)}\frac{\lambda}{\epsilon}\ln(1/\delta_1). \tag{H9}$$

To minimize the cost, we therefore aim to maximize $(2\omega - 1)(1/\omega - 1)$. That has a derivative of $1/\omega^2 - 2$, so the turning point is $\omega = 1/\sqrt{2}$.

Note that an extra factor needs to be taken into account for amplitude estimation. This is independent of the choice of ω and therefore identical to the case that we have studied before for $\omega = 2/3$, which is given in Eq. (110). With the parameters γ_1 and γ_2 given in Eq. (114), we thus arrive at the expressions in Eq. (H1) by multiplying this extra factor by Eq. (H9).

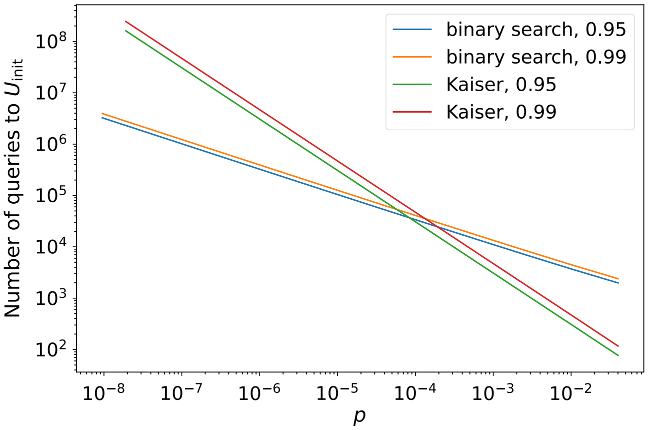

FIG. 23. The number of queries to $U_{\rm init}$ as a function of the squared overlap p for FeMoco with $\lambda = 306$ and $\epsilon = 0.0016$, using both the binary-search method discussed in this appendix and a pure QPE+Kaiser window. At $p = 0.01$, the numbers of queries using the binary-search method are 3683 and 4457 for 95% confidence and 99% confidence, respectively, compared to 309 and 472 using the pure QPE+Kaiser window. The value of ω is chosen to be $1/\sqrt{2}$.

Optimizing ω for Eq. (H1) outside the $\ln\ln$ we have $\omega = 1/\sqrt{2}$ and the number of queries is roughly

$$\frac{7.77\lambda}{\sqrt{p}\epsilon}\ln\left(\frac{4}{\sqrt{p}}\right)\ln\left(\frac{\log_{\sqrt{2}}(\lambda/\epsilon)}{q}\right). \tag{H10}$$

We need to use $U_{\rm init}$

$$\underbrace{\log_{1/\omega}(\lambda/\epsilon)}_{\text{search steps}} \times \underbrace{\frac{4}{3\sqrt{p}}\ln\left(\frac{\log_{1/\omega}(\lambda/\epsilon)}{q}\right)}_{\text{amplitude estimation}} = \frac{4\log_{1/\omega}(\lambda/\epsilon)}{3\sqrt{p}}\ln\left(\frac{\log_{1/\omega}(\lambda/\epsilon)}{q}\right) \tag{H11}$$

times, which is monotonically increasing with respect to ω.

In Fig. 23, we numerically compute the complexity in terms of queries to $U_{\rm init}$ of the binary-search method and the direct-sampling approach. In this case, the crossover of query complexity is significantly lower, around $p \sim 10^{-4}$. This is because the binary-search approach uses many low-accuracy phase estimates, reducing the queries to W but not $U_{\rm init}$.

2. Trading $\mathcal{W}$ queries for $U_{\rm init}$ queries

We may improve the number of queries to $\mathcal{W}$ slightly at the expense of queries to $U_{\rm init}$. Instead of a uniform failure probability as in Eq. (112) for amplitude estimation across all L steps of fuzzy bisection, we may choose a nonuniform schedule where in the lth step, we choose the failure probability $\delta_{l,2}$ to be some monotonically increasing function of

$l=1,2,\ldots,L$. For instance, let $\delta_{l,2}=q_0/(L-l+1)^\alpha$ for some constant $\alpha>1$. Then, by a union bound, the failure probability of amplitude estimation after all L steps is at most

$$\sum_{l=1}^{L}\delta_{l,2}\le\sum_{l=1}^{\infty}\frac{q_0}{l^\alpha}=q_0\zeta(\alpha)=q, \tag{H12}$$

where ζ is the Riemann zeta function, and the last equality follows by choosing $q_0=q/\zeta(\alpha)$.

At step l, we perform phase estimation to a precision that is at least $(2/3)^l$ and with a failure probability $\delta_1=\Theta(p)$. Subsequently, we apply amplitude estimation to a precision $\Theta(\sqrt{p})$ with a failure probability $\delta_{l,2}$. The numbers of queries $C_{l,W}$ and $C_{l,U_{\text{init}}}$ at step l to W and U_{init}, respectively, are then

$$\begin{aligned}C_{l,W}&=Q((2/3)^l,\Theta(p))Q(\Theta(\sqrt{p}),\delta_{l,2})\\&=\Theta\left(\frac{1}{\sqrt{p}}\left(\frac{2}{3}\right)^l\ln\left(\frac{1}{\sqrt{p}}\right)\ln\left(\frac{1}{\sqrt{\delta_{l,2}}}\right)\right),\end{aligned}\tag{H13}$$

$$C_{l,U_{\text{init}}}=Q(\Theta(\sqrt{p}),\delta_{l,2})=\Theta\left(\frac{1}{\sqrt{p}}\ln\left(\frac{1}{\sqrt{\delta_{l,2}}}\right)\right). \tag{H14}$$

The total number of queries to W across all steps is then

$$\begin{aligned}C_W&=\sum_{l=1}^{L}C_{l,W}=\Theta\left(\frac{1}{\sqrt{p}}\ln\left(\frac{1}{\sqrt{p}}\right)\right.\\&\quad\left.\times\sum_{l=1}^{L}\left(\frac{2}{3}\right)^l\ln\left(\frac{\sqrt{\zeta(\alpha)(L-l+1)^\alpha}}{\sqrt{q}}\right)\right)\\&=\mathcal{O}\left(\frac{1}{\sqrt{p}}\ln\left(\frac{1}{\sqrt{p}}\right)\ln\left(\frac{1}{\sqrt{q}}\right)\left(\frac{2}{3}\right)^L\right)\\&=\mathcal{O}\left(\frac{\lambda}{\sqrt{p}\epsilon}\ln\left(\frac{1}{\sqrt{p}}\right)\ln\left(\frac{1}{\sqrt{q}}\right)\right),\end{aligned}\tag{H15}$$

which improves on Eq. (118) by a factor of $\ln\ln(\lambda/\epsilon)$. Above, the second line follows from the sum identities

$$\begin{aligned}\sum_{l\in[L]}x^{l-1}(L-l+1)&=\frac{L(1-x)+x(x^L-1)}{(x-1)^2}\\&=\frac{x^{L+1}}{(x-1)^2}-\mathcal{O}\left(\frac{L(x-1)+x}{(x-1)^2}\right)\end{aligned}\tag{H16}$$

and that $(2/3)^l\ln(L-l+1)=\mathcal{O}((2/3)^l(L-l+1))$. Similarly, the number of queries

$$\begin{aligned}C_{U_{\text{init}}}&=\sum_{l=1}^{L}C_{l,U_{\text{init}}}\\&=\Theta\left(\frac{1}{\sqrt{p}}\sum_{l=1}^{L}\ln\left(\frac{\sqrt{\zeta(\alpha)(L-l+1)^\alpha}}{\sqrt{q}}\right)\right)\\&=\Theta\left(\frac{1}{\sqrt{p}}\sum_{l=1}^{L}\ln\left(\frac{1}{\sqrt{q}}\right)+\frac{\alpha}{2}\ln\left(\zeta(\alpha)(L-l+1)\right)\right)\\&=\mathcal{O}\left(\frac{L}{\sqrt{p}}\left(\ln\left(\frac{1}{\sqrt{q}}\right)+\frac{\alpha}{2}\ln\left(L\right)\right)\right)\\&=\mathcal{O}\left(\frac{1}{\sqrt{p}}\ln\left(\frac{\lambda}{\epsilon}\right)\ln\left(\frac{\ln\left(\lambda/\epsilon\right)}{\sqrt{q}}\right)\right).\end{aligned}\tag{H17}$$

Note that $C_{U_{\text{init}}}$ is asymptotically similar to Eq. (H11) but will have a larger constant factor that grows with α.

APPENDIX I: QUANTUM MULTIPLE-EIGENVALUE GAUSSIAN FILTERED SEARCH (QMEGS)

Although QMEGS [47] was originally designed for the simultaneous estimation of multiple eigenvalues, the algorithm can also be directly applied to estimate the ground-state energy. In this appendix, we assume quantum oracle access to a Hadamard test circuit that implements controlled-Hamiltonian evolution $\exp(-iHn)$ for $n\in\mathbb{Z}$, as depicted in Fig. 24. Although the theoretical results in Ref. [47] focus on the case of Hamiltonian evolution using rational powers, $\exp(-iHt)$ for $t\in\mathbb{R}$, these results can be readily extended to the case of integer powers, as discussed in Ref. [47, Appendix C]. The algorithm contains two steps—"Step 1. Data generation" (the quantum part) and "Step 2. Filtering and searching" (the classical part)—as discussed in the following.

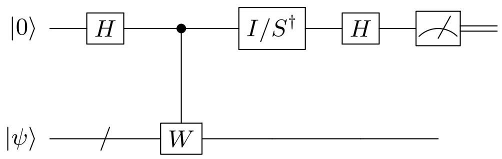

FIG. 24. The Hadamard test circuit is used to estimate $\langle\psi|W|\psi\rangle$. The Hadamard gate, denoted by H, is applied to an ancilla qubit. By choosing either the I or $S^\dagger$ operator on the ancilla qubit, we can estimate the real or imaginary parts of $\langle\psi|W|\psi\rangle$, respectively. In QMEGS, W is often chosen as $W=\exp(-iHt)$, where $t\in\mathbb{R}$ or $t=n\tau$ with $n\in\mathbb{Z}$ and $\tau>0$.

Step 1: Data generation. We implement the Hadamard test quantum circuit as shown in Fig. 24 to obtain the data set. Specifically, given $t \in \mathbb{R}$, we set $W = \exp(-iHt)$, apply I (or $S^\dagger$, with S being the phase gate) on the ancilla qubits, measure the ancilla qubit, and define a random variable X (or Y) such that $X = 1$ (or $Y = 1$) if the outcome is 0 and $X = -1$ (or $Y = -1$) if the outcome is 1. Then,

$$\mathbb{E}(X_t + iY_t) = \langle \psi | \exp(-itH) | \psi \rangle \, . \tag{I1}$$

We first generate a set of integer time points $\{t_n = \lfloor \widetilde{t}_n \rfloor\}_{n=1}^{N} \subset \mathbb{Z}^n$, with $\{\widetilde{t}_n\}$ independently drawn from a truncated Gaussian with density

$$a(t) = \left(1 - \int_{-\sigma T}^{\sigma T} \frac{1}{\sqrt{2\pi} T} \exp\left(-\frac{s^2}{2T^2}\right) \mathbf{1}_{[-\sigma T, \sigma T]}(s) \mathrm{d}s\right) \delta_0(t) + \frac{1}{\sqrt{2\pi} T} \exp\left(-\frac{t^2}{2T^2}\right) \mathbf{1}_{[-\sigma T, \sigma T]}(t), \tag{I2}$$

where $\sigma, T > 0$ and δ_0 is the delta function. For each t_n, we apply Hadamard tests *twice* to generate the data set

$$\mathcal{D}_H = \{(t_n, Z_n)\}_{n=1}^{N} := \{(t_n, X_n + iY_n)\}_{n=1}^{N} \, . \tag{I3}$$

Each evaluation of X_n (or Y_n) only requires running the Hadamard test circuit with I (or $S^\dagger$) *once* at $t = t_n$. The detailed data-generation process can be found in Ref. [47, Algorithm 1].

Step 2: Filtering and searching. After generating the data set in Eq. (I3), we construct the filtering function

$$G(\theta) = \left| \frac{1}{N} \sum_{n=1}^{N} Z(t_n) \exp(i\theta t_n) \right| \tag{I4}$$

and solve the optimization problem to find the approximation $\widetilde{\lambda}_0$ to λ_0:

$$\widetilde{\lambda}_0 = \operatorname{argmax}_\theta \left| \frac{1}{N} \sum_{n=1}^{N} Z(t_n) \exp(i\theta t_n) \right| \approx \lambda_0. \tag{I5}$$

All computations in the second step are performed on a classical computer, with *no additional quantum cost*. In this case, Eq. (I5) can be solved by a brute-force search over a finite set of discrete points. Since we are only concerned with estimating the ground-state energy, no filtering is needed. The detailed algorithm for this procedure is provided in Ref. [47, Algorithm 2].

APPENDIX J: THRESHOLD ANALYSIS FOR RESOURCE ESTIMATES

In this appendix, we provide data used for determining truncation thresholds for double-factorized and tensor-hypercontraction representations of the two-electron integrals used in DF and THC block encodings. The phase-estimation cost is based upon an uncertainty of $\sigma_{\mathrm{PEA}} = 1$ millihartree, consistent with prior work (allowing for additional error in other parts of the algorithm). To numerically determine truncation thresholds, we use high-spin unrestricted coupled-cluster singles and doubles with perturbative triples (UCCSD(T)) as a metric for the correlation energy of truncated Hamiltonians. All energy differences in the following tables are in millihartree.

TABLE III. Analysis of the THC rank versus the accuracy of coupled-cluster singles and doubles with perturbative triples (CCSD(T)) for $\mathrm{Fe_2S_2}$ with (FeIII, FeIII) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 8$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF.

n_{THC}	λ_{THC}	$\lVert V - V_{\mathrm{THC}} \rVert$	$E_{\mathrm{corr}} - E^*_{\mathrm{corr}}$	C_{BE}	QPE	Logical qubits
60	104.4673	0.1431	3.7846	3903	4.0030×10^8	575
80	105.3320	0.0400	4.9146	4222	4.3660×10^8	580
100	105.4446	0.0219	6.9844×10^{-1}	4414	4.5694×10^8	581
120	105.5165	0.0124	-1.4958×10^{-1}	4626	4.7921×10^8	581
140	105.5543	0.0082	-4.2948×10^{-1}	4951	5.1306×10^8	598
160	105.5910	0.0069	-3.8850×10^{-1}	5208	5.3988×10^8	598
180	106.0045	0.0070	-3.1358×10^{-1}	5418	5.6385×10^8	1045

1. Fe_2S_2

TABLE IV. Analysis of the THC rank versus the accuracy of CCSD(T) for Fe_2S_2 with (FeIII, FeIII) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 8$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF. The LCU 1-norm λ is computed with a number-operator symmetry shift computed as the median of $\{f_i\}$, where the f_i are eigenvalues of the one-body operator being block encoded.

n_{THC}	λ_{THC}	$\lVert V - V_{\mathrm{THC}} \rVert$	$E_{\mathrm{corr}} - E^*_{\mathrm{corr}}$	C_{BE}	QPE	Logical qubits
60	62.7590	0.1431	3.7846×10^{0}	3903	2.4048×10^{8}	573
80	63.6236	0.0400	4.9146×10^{0}	4222	2.6372×10^{8}	578
100	63.7363	0.0219	6.9769×10^{-1}	4414	2.7620×10^{8}	579
120	63.8082	0.0124	-1.3363×10^{-1}	4626	2.8979×10^{8}	579
140	63.8460	0.0082	-4.1996×10^{-1}	4951	3.1033×10^{8}	596
160	63.8827	0.0069	-3.8494×10^{-1}	5208	3.2663×10^{8}	596
180	64.2961	0.0070	-3.1175×10^{-1}	5418	3.4200×10^{8}	1043
200	72.3297	0.0069	-5.5725×10^{-1}	5621	3.9915×10^{8}	1047

TABLE V. Analysis of the accuracy of double factorization using the CCSD(T) correlation energy (in mE_{h}) for Fe_2S_2 with (FeIII, FeIII) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 8$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF. The results are shown with no shift ($\alpha_2 = 0$) and with a shift ($\alpha_2 = 0.1$).

$\alpha_2 = 0$				$\alpha_2 = 0.1$			
DF threshold	n_{DF}	$\lVert V - V_{\mathrm{DF}} \rVert$	$\lvert E_{\mathrm{corr}} - E^*_{\mathrm{corr}} \rvert$	DF threshold	n_{DF}	$\lVert V - V_{\mathrm{DF}} \rVert$	$\lvert E_{\mathrm{corr}} - E^*_{\mathrm{corr}} \rvert$
1.0×10^{-2}	62	0.0473	7.864	1.0×10^{-2}	62	0.0471	8.250
3.0×10^{-3}	80	0.0154	8.804×10^{-1}	1.0×10^{-3}	80	0.0159	1.606
1.0×10^{-3}	93	0.0044	2.496×10^{-1}	1.0×10^{-3}	93	0.0046	6.849×10^{-1}
3.0×10^{-4}	111	0.0014	1.424×10^{-1}	1.0×10^{-4}	111	0.0014	1.164×10^{-1}
1.0×10^{-4}	131	0.0005	6.639×10^{-2}	1.0×10^{-4}	131	0.0005	4.667×10^{-2}
3.0×10^{-5}	152	0.0001	6.824×10^{-2}	1.0×10^{-5}	152	0.0001	3.560×10^{-3}

2. Fe_4S_4

TABLE VI. Analysis of the THC rank versus the accuracy of CCSD(T) for Fe_4S_4 with (2 FeII, 2 FeIII) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 16$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF.

n_{THC}	λ_{THC}	$\lVert V - V_{\mathrm{THC}} \rVert$	$E_{\mathrm{corr}} - E^*_{\mathrm{corr}}$	C_{BE}	QPE	Logical qubits
108	271.5315	1.2684	1.9652×10^{1}	6888	1.8362×10^{9}	937
144	292.3270	0.1294	2.8155×10^{0}	7384	2.1191×10^{9}	942
180	292.8878	0.0796	1.9821×10^{0}	7804	2.2440×10^{9}	1081
216	293.5319	0.0431	7.5063×10^{-1}	8175	2.3558×10^{9}	1083
252	293.7534	0.0301	-1.9193×10^{-1}	8573	2.4724×10^{9}	1083
288	293.8331	0.0273	-1.7791×10^{-1}	9120	2.6308×10^{9}	1151
324	293.9603	0.0217	6.7870×10^{-2}	9590	2.7676×10^{9}	1151
360	293.9936	0.0198	2.2911×10^{-1}	10 031	2.8952×10^{9}	2110

TABLE VII. Analysis of the THC rank versus the accuracy of CCSD(T) for Fe_4S_4 with (2 FeII, 2 FeIII) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 16$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF. The LCU 1-norm λ is computed with a number-operator symmetry shift computed as the median of $\{f_i\}$, where the f_i are eigenvalues of the one-body operator being block encoded.

$n_{\rm THC}$	$\lambda_{\rm THC}$	$\|V - V_{\rm THC}\|$	$E_{\rm corr} - E^*_{\rm corr}$	$C_{\rm BE}$	QPE	Logical qubits
144	166.9151	0.1294	2.8155×10^0	7384	1.2100×10^9	940
180	167.4759	0.0796	1.9821×10^0	7804	1.2831×10^9	1079
216	168.1200	0.0431	7.5063×10^{-1}	8175	1.3493×10^9	1081
252	168.3416	0.0301	-1.9193×10^{-1}	8573	1.4169×10^9	1081
288	168.4212	0.0273	-1.7791×10^{-1}	9120	1.5080×10^9	1149
324	168.5485	0.0217	6.7870×10^{-2}	9590	1.5869×10^9	1149
360	168.5817	0.0198	2.2911×10^{-1}	10 031	1.6602×10^9	2108

TABLE VIII. Analysis of the accuracy of double factorization using the CCSD(T) correlation energy for Fe_4S_4 with (2 FeII, 2 FeIII) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 16$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF. The results are shown with no shift ($\alpha_2 = 0$) and with a shift ($\alpha_2 = 0.1$).

$\alpha_2 = 0$				$\alpha_2 = 0.1$			
DF thresh.	$n_{\rm DF}$	$\|V - V_{\rm DF}\|$	$\|E_{\rm corr} - E^*_{\rm corr}\|$	DF threshold	$n_{\rm DF}$	$\|V - V_{\rm DF}\|$	$\|E_{\rm corr} - E^*_{\rm corr}\|$
3.0×10^{-3}	152	0.0217820	1.223	3.0×10^{-3}	152	0.0221000	1.444
1.0×10^{-3}	181	0.0070033	1.579×10^{-1}	1.0×10^{-3}	181	0.0070700	3.610×10^{-1}
3.0×10^{-4}	222	0.0020602	1.719×10^{-1}	3.0×10^{-4}	222	0.0020400	2.576×10^{-1}
1.0×10^{-4}	260	0.0006839	4.648×10^{-2}	1.0×10^{-4}	260	0.0006840	5.879×10^{-2}
3.0×10^{-5}	312	0.0002120	6.671×10^{-3}	3.0×10^{-5}	312	0.0002130	1.180×10^{-2}
1.0×10^{-5}	365	0.0000736	9.582×10^{-3}	1.0×10^{-5}	365	0.0000731	1.067×10^{-2}

TABLE IX. Analysis of the THC rank versus the accuracy of CCSD(T) for Fe_4S_4 with Fe-4(III) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 16$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF.

$n_{\rm THC}$	$\lambda_{\rm THC}$	$\|V - V_{\rm THC}\|$	$E_{\rm corr} - E^*_{\rm corr}$	$C_{\rm BE}$	QPE	Logical qubits
108	261.8652	1.1938	3.1102×10^1	6888	1.7708×10^9	935
144	280.7802	0.1477	7.2471×10^0	7384	2.0354×10^9	942
180	282.0736	0.0691	-6.4487×10^{-1}	7804	2.1611×10^9	1081
216	282.4604	0.0464	-6.7771×10^{-1}	8175	2.2670×10^9	1083
252	282.7991	0.0280	-2.0255×10^{-1}	8573	2.3802×10^9	1083
288	282.8973	0.0225	-3.2773×10^{-1}	9120	2.5329×10^9	1151
324	282.9653	0.0199	-7.2980×10^{-1}	9590	2.6641×10^9	1151
360	283.0203	0.0175	-4.2608×10^{-1}	10 031	2.7872×10^9	2110

TABLE X. Analysis of the THC rank versus the accuracy of CCSD(T) for Fe_4S_4 with Fe-4(III) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 16$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF. The LCU 1-norm λ is computed with a number-operator symmetry shift computed as the median of $\{f_i\}$, where the f_i are eigenvalues of the one-body operator being block encoded.

$n_{\rm THC}$	$\lambda_{\rm THC}$	$\|V - V_{\rm THC}\|$	$E_{\rm corr} - E^*_{\rm corr}$	$C_{\rm BE}$	QPE	Logical qubits
108	143.1948	1.1938	3.1102×10^1	6888	9.6833×10^8	935
144	162.1098	0.1477	7.2422×10^0	7384	1.1752×10^9	940
180	163.4031	0.0691	-6.4487×10^{-1}	7804	1.2519×10^9	1079
216	163.7900	0.0464	-6.8280×10^{-1}	8175	1.3145×10^9	1081
252	164.1287	0.0280	-2.0255×10^{-1}	8573	1.3814×10^9	1081
288	164.2269	0.0225	-3.2750×10^{-1}	9120	1.4704×10^9	1149
324	164.2949	0.0199	-7.2980×10^{-1}	9590	1.5468×10^9	1149
360	164.3499	0.0175	-4.2608×10^{-1}	10 031	1.6185×10^9	2108

TABLE XI. Analysis of the accuracy of double factorization using CCSD(T) for Fe_4S_4 with Fe-4(III) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 16$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF. The results are shown with no shift ($\alpha_2 = 0$) and with a shift ($\alpha_2 = 0.1$).

$\alpha_2 = 0$				$\alpha_2 = 0.1$			
DF threshold	n_{DF}	$\lVert V - V_{\mathrm{DF}} \rVert$	$\lvert E_{\mathrm{corr}} - E^*_{\mathrm{corr}} \rvert$	DF threshold	n_{DF}	$\lVert V - V_{\mathrm{DF}} \rVert$	$\lvert E_{\mathrm{corr}} - E^*_{\mathrm{corr}} \rvert$
3.0×10^{-3}	154	0.0228617	1.920	3.0×10^{-3}	154	0.0228752	2.908
1.0×10^{-3}	185	0.0075771	6.028×10^{-1}	1.0×10^{-3}	185	0.0075597	5.635×10^{-1}
3.0×10^{-4}	226	0.0020788	7.355×10^{-2}	3.0×10^{-4}	226	0.0020692	1.129×10^{-1}
1.0×10^{-4}	265	0.0006872	8.432×10^{-2}	1.0×10^{-4}	265	0.0006910	7.745×10^{-2}
3.0×10^{-5}	316	0.0002163	7.598×10^{-3}	3.0×10^{-5}	316	0.0002147	1.807×10^{-2}
1.0×10^{-5}	369	0.0000738	1.080×10^{-3}	1.0×10^{-5}	369	0.0000740	3.570×10^{-3}

TABLE XII. Analysis of the THC rank versus the accuracy of CCSD(T) for Fe_4S_4 with 2Fe(III)2Fe(II) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 16$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF.

n_{THC}	λ_{THC}	$\lVert V - V_{\mathrm{THC}} \rVert$	$E_{\mathrm{corr}} - E^*_{\mathrm{corr}}$	C_{BE}	QPE	Logical qubits
108	280.7099	0.7869	3.0641×10^{1}	6888	1.8982×10^{9}	937
144	291.3346	0.1912	5.4497×10^{0}	7384	2.1120×10^{9}	942
180	293.2292	0.0799	-3.8260×10^{0}	7804	2.2466×10^{9}	1081
216	293.8136	0.0466	2.2190×10^{0}	8175	2.3581×10^{9}	1083
252	294.1204	0.0286	1.8563×10^{0}	8573	2.4755×10^{9}	1083
288	294.2313	0.0206	5.1673×10^{-2}	9120	2.6344×10^{9}	1151
324	294.2698	0.0192	3.0606×10^{-1}	9590	2.7705×10^{9}	1151
360	294.3255	0.0168	-7.6609×10^{-1}	101,031	2.8985×10^{9}	2110

TABLE XIII. Analysis of the THC rank versus the accuracy of CCSD(T) for Fe_4S_4 with 2Fe(III)2Fe(II) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 16$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF. The LCU 1-norm λ is computed with a number-operator symmetry shift computed as the median of $\{f_i\}$, where the f_i are eigenvalues of the one-body operator being block encoded.

n_{THC}	λ_{THC}	$\lVert V - V_{\mathrm{THC}} \rVert$	$E_{\mathrm{corr}} - E^*_{\mathrm{corr}}$	C_{BE}	QPE	Logical qubits
108	155.1929	0.7869	3.0641×10^{1}	6888	1.0495×10^{9}	935
144	165.8176	0.1912	5.4497×10^{0}	7384	1.2021×10^{9}	940
180	167.7122	0.0799	-3.8260×10^{0}	7804	1.2849×10^{9}	1079
216	168.2966	0.0466	2.2190×10^{0}	8175	1.3507×10^{9}	1081
252	168.6034	0.0286	1.8563×10^{0}	8573	1.4191×10^{9}	1081
288	168.7143	0.0206	5.1673×10^{-2}	9120	1.5106×10^{9}	1149
324	168.7528	0.0192	3.0606×10^{-1}	9590	1.5888×10^{9}	1149
360	168.8085	0.0168	-7.6609×10^{-1}	10 031	1.6624×10^{9}	2108
396	168.8265	0.0174	-3.3251×10^{-2}	10 386	1.7214×10^{9}	2110
432	168.8812	0.0082	-3.6832×10^{-2}	10 757	1.7835×10^{9}	2110
468	168.9190	0.0073	3.4664×10^{-1}	11 156	1.8501×10^{9}	2110
504	168.9603	0.0078	-6.8758×10^{-1}	11 561	1.9177×10^{9}	2110
540	168.9765	0.0058	-1.3886×10^{-1}	12 167	2.0184×10^{9}	2242

TABLE XIV. Analysis of the accuracy of double factorization using CCSD(T) for Fe_4S_4 with 2Fe(III)2Fe(II) oxidation states [31] using a high-spin $n_\uparrow - n_\downarrow = 16$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF. The results are shown with no shift ($\alpha_2 = 0$) and with a shift ($\alpha_2 = 0.1$).

$\alpha_2 = 0$				$\alpha_2 = 0.1$			
DF threshold	n_{DF}	$\|V - V_{DF}\|$	$\|E_{corr} - E^*_{corr}\|$	DF threshold	n_{DF}	$\|V - V_{DF}\|$	$\|E_{corr} - E^*_{corr}\|$
3.0×10^{-3}	152	0.0221358	7.301×10^{-1}	3.0×10^{-3}	152	0.0224483	2.619×10^{-1}
1.0×10^{-3}	183	0.0069379	2.493×10^{-1}	1.0×10^{-3}	181	0.0070526	6.233×10^{-1}
3.0×10^{-4}	222	0.0020665	8.187×10^{-2}	3.0×10^{-4}	222	0.0020752	1.715×10^{-2}
1.0×10^{-4}	260	0.0006887	4.144×10^{-2}	1.0×10^{-4}	260	0.0006901	1.664×10^{-2}
3.0×10^{-5}	312	0.0002146	1.359×10^{-2}	3.0×10^{-5}	312	0.0002151	2.107×10^{-2}
1.0×10^{-5}	365	0.0000741	4.842×10^{-3}	1.0×10^{-5}	365	0.0000736	2.165×10^{-2}

3. FeMoco

TABLE XV. Analysis of the accuracy of double factorization using the CCSD(T) correlation energy for FeMoco [31] using a high-spin $n_\uparrow - n_\downarrow = 35$ reference for CCSD(T). The CCSD(T) calculations are performed using UCCSD(T) in PySCF. The results are shown with no shift ($\alpha_2 = 0$) and with a shift ($\alpha_2 = 0.1$).

$\alpha_2 = 0$				$\alpha_2 = 0.1$			
DF threshold	n_{DF}	$\|V - V_{DF}\|$	$\|E_{corr} - E^*_{corr}\|$	DF threshold	n_{DF}	$\|V - V_{DF}\|$	$\|E_{corr} - E^*_{corr}\|$
5.00×10^{-3}	312	0.0578095	2.414	5.00×10^{-3}	312	1.706	0.0580414
2.50×10^{-3}	344	0.0289546	1.470	2.50×10^{-3}	344	1.621	0.0289732
1.25×10^{-3}	394	0.0144793	5.330×10^{-2}	1.25×10^{-3}	394	3.872×10^{-1}	0.0144793
5.00×10^{-4}	470	0.0057589	4.048×10^{-1}	5.00×10^{-4}	472	4.500×10^{-1}	0.0057756
2.50×10^{-4}	526	0.0028081	6.060×10^{-2}	2.50×10^{-4}	522	6.970×10^{-2}	0.0028139
1.25×10^{-4}	589	0.0013772	3.989×10^{-2}	1.25×10^{-4}	585	7.884×10^{-3}	0.0013765
5.00×10^{-5}	679	0.0005509	1.028×10^{-2}	5.00×10^{-5}	679	7.430×10^{-3}	0.0005514

APPENDIX K: VALIDATION OF OVERLAP-EXTRAPOLATION PROTOCOL

Here, we show several examples that demonstrate the validity of the overlap-extrapolation protocol for the Fe_2S_2 and Fe_4S_4 systems. For the Fe_2S_2 system, the exact wave function ($\Phi(\infty)$) for the active-space model of CAS(30e,20o) is accessible. We have obtained the exact MPS and other MPSs with several bond dimensions and calculated the overlap between them. As a preliminary study, in Fig. 25 we show the extrapolations for the three energies using data obtained with the reverse-schedule DMRG, showing that there is linear behavior of the energy. In Figs. 26(a) and 26(b), we show the plots corresponding to Eqs. (126) and (127), respectively. We can see that both empirical linear relations hold remarkably well.

However, the exact wave functions for the complete active-space models of the Fe_4S_4 systems, namely, CAS(54e,36o) and CAS(52e,36o), are not accessible. Therefore, while we are unable to directly confirm the empirical linear relations for these systems, in Fig. 27 we provide strong evidence supporting the validity of the overlap extrapolation. Each empty triangle corresponds to $|\langle\Phi(M')|\Phi(\infty)\rangle|$ for each value of $M' = 20, 40$, and 60, and has been obtained by fitting the blue, yellow, and green triangles just below it, as discussed in Fig. 18. Based on these $|\langle\Phi(M')|\Phi(\infty)\rangle|$ values, we have applied the linear relation of Eq. (127) to predict the values of $|\langle\Phi(M')|\Phi(M'' = 8000)\rangle|$, corresponding to the empty circles in Fig. 27. On the other hand, we have obtained the MPS for $M'' = 8000$ using DMRG and directly computed the overlaps between this MPS and those for $M' = 20, 40$, and 60, corresponding to the red circles. The excellent agreement between the red and empty circles provides indirect validation of $|\langle\Phi(M')|\Phi(\infty)\rangle|$ for $M' = 20, 40$, and 60, indicating the reliability of the extrapolation.

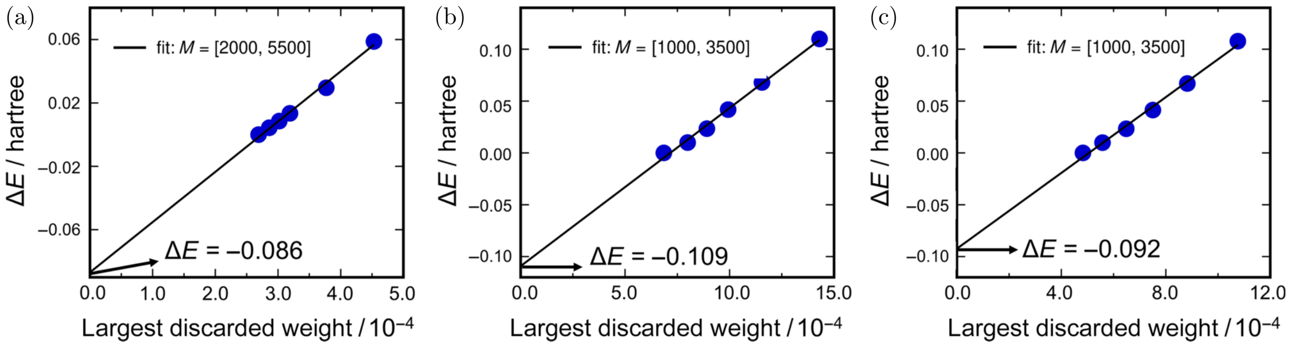

FIG. 25. The extrapolated DMRG energy for FeMoco with (a) MPS1, (b) MPS2, and (c) MPS3 with respect to discarded weight, assuming a linear relationship between discarded weight and energy. The energy differences (ΔE) in hartree are represented relative to the DMRG energy at the highest bond dimension used for each extrapolation, which are $-22\,140.270$, $-22\,140.249$, and $-22\,140.223$ for MPS1 ($M = 5500$), MPS2 ($M = 3500$), and MPS3 ($M = 3500$), respectively.

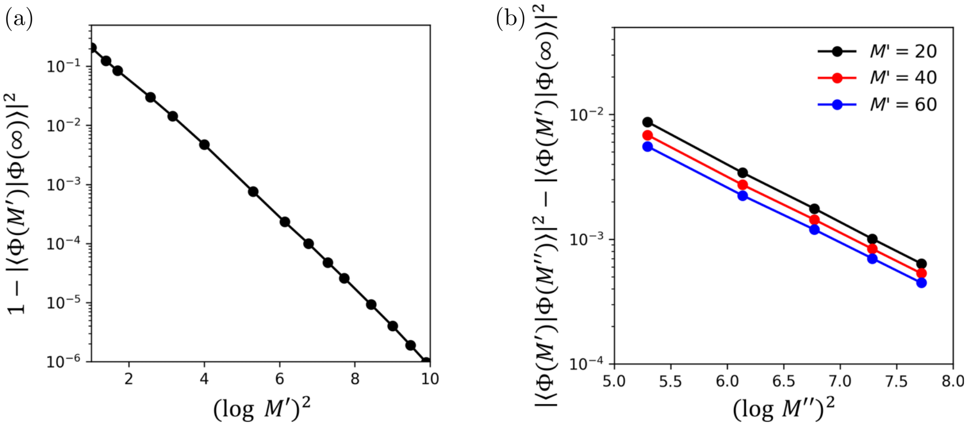

FIG. 26. Two empirical linear relations for the Fe_2S_2 system. We demonstrate the first linear relation (a) using the overlap between MPSs for $M' = [10, 1400]$ and the exact MPS, while the second linear relation (b) is demonstrated using the overlap between MPSs for $M' = [20, 60]$, MPSs for $M'' = [200, 600]$, and the exact MPS.

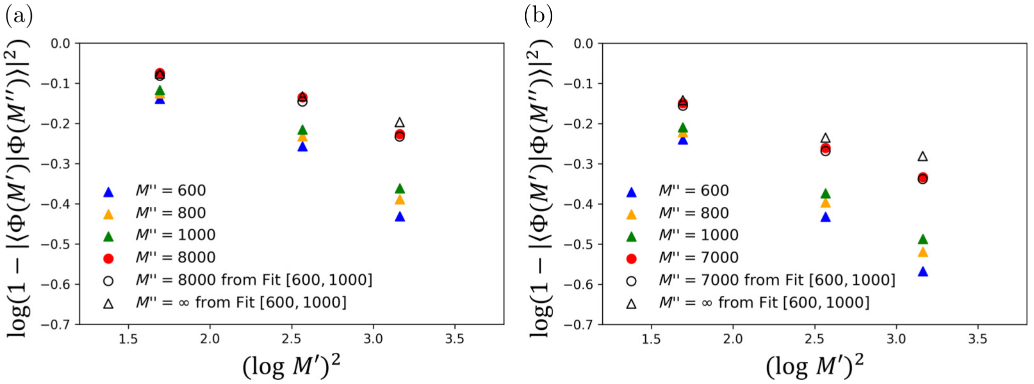

FIG. 27. Validation of the overlap extrapolation for the Fe_4S_4 systems: (a) Fe_2(III)Fe_2(II); (b) Fe_4(III). The data from the overlaps of the MPSs for $M' = 20, 40, 60$ and $M'' = 600, 800, 1000, 8000$, obtained by DMRG, are represented by the colored dots, while the data obtained through extrapolation based on the linear relation in Eq. (127) are represented by the empty dots. The close agreement between the empty circles and red circles demonstrates the robustness of the extrapolation protocol.

[1] Y. Xiong, S. X. Ng, G.-L. Long, and L. Hanzo, Dual-frequency quantum phase estimation mitigates the spectral leakage of quantum algorithms, IEEE Signal Process. Lett. **29**, 1222 (2022).

[2] S. Greenaway, W. Pol, and S. Sim, A case study against QSVT: Assessment of quantum phase estimation improved by signal processing techniques, arXiv:2404.01396.

[3] N. M. Tubman, C. Mejuto-Zaera, J. M. Epstein, D. Hait, D. S. Levine, W. Huggins, Z. Jiang, J. R. McClean, R. Babbush, M. Head-Gordon, and K. B. Whaley, Postponing the orthogonality catastrophe: Efficient state preparation for electronic structure simulations on quantum devices, arXiv:1809.05523.

[4] P. J. Ollitrault, C. L. Cortes, J. F. Gonthier, R. M. Parrish, D. Rocca, G.-L. Anselmetti, M. Degroote, N. Moll, R. Santagati, and M. Streif, Enhancing initial state overlap through orbital optimization for faster molecular electronic ground-state energy estimation, Phys. Rev. Lett. **133**, 250601 (2024).

[5] M. Erakovic, F. Witteveen, D. Harley, J. Günther, M. Bensberg, O. R. Meitei, M. Cho, T. Van Voorhis, M. Reiher, and M. Christandl, High ground state overlap via quantum embedding methods, PRX Life **3**, 013003 (2025).

[6] S. Fomichev, K. Hejazi, M. S. Zini, M. Kiser, J. Fraxanet, P. A. M. Casares, A. Delgado, J. Huh, A.-C. Voigt, J. E. Mueller, and J. M. Arrazola, Initial state preparation for quantum chemistry on quantum computers, PRX Quantum **5**, 040339 (2024).

[7] R. Haghshenas, J. Gray, A. C. Potter, and G. K.-L. Chan, Variational power of quantum circuit tensor networks, Phys. Rev. X **12**, 011047 (2022).

[8] G. H. Low, V. Kliuchnikov, and L. Schaeffer, Trading T gates for dirty qubits in state preparation and unitary synthesis, Quantum **8**, 1375 (2024).

[9] S.-J. Ran, Encoding of matrix product states into quantum circuits of one- and two-qubit gates, Phys. Rev. A **101**, 032310 (2020).

[10] D. Malz, G. Styliaris, Z.-Y. Wei, and J. I. Cirac, Preparation of matrix product states with log-depth quantum circuits, Phys. Rev. Lett. **132**, 040404 (2024).

[11] C. Schön, E. Solano, F. Verstraete, J. I. Cirac, and M. M. Wolf, Sequential generation of entangled multiqubit states, Phys. Rev. Lett. **95**, 110503 (2005).

[12] S. Lee, J. Lee, H. Zhai, Y. Tong, A. M. Dalzell, A. Kumar, P. Helms, J. Gray, Z.-H. Cui, W. Liu, M. Kastoryano, R. Babbush, J. Preskill, D. R. Reichman, E. T. Campbell, E. F. Valeev, L. Lin, and G. K.-L. Chan, Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry, Nat. Commun. **14**, 1952 (2023).

[13] G. K. Chan, Quantum chemistry, classical heuristics, and quantum advantage, arXiv:2407.11235.

[14] L. Lin and Y. Tong, Heisenberg-limited ground state energy estimation for early fault-tolerant quantum computers, PRX Quantum **3**, 010318 (2022).

[15] R. Zhang, G. Wang, and P. Johnson, Computing ground state properties with early fault-tolerant quantum computers, Quantum **6**, 761 (2022).

[16] K. Wan, M. Berta, and E. T. Campbell, Randomized quantum algorithm for statistical phase estimation, Phys. Rev. Lett. **129**, 030503 (2022).

[17] Y. Dong, L. Lin, and Y. Tong, Ground-state preparation and energy estimation on early fault-tolerant quantum computers via quantum eigenvalue transformation of unitary matrices, PRX Quantum **3**, 040305 (2022).

[18] G. Wang, D. S. França, R. Zhang, S. Zhu, and P. D. Johnson, Quantum algorithm for ground state energy estimation using circuit depth with exponentially improved dependence on precision, Quantum **7**, 1167 (2023).

[19] Z. Ding and L. Lin, Simultaneous estimation of multiple eigenvalues with short-depth quantum circuit on early fault-tolerant quantum computers, Quantum **7**, 1136 (2023).

[20] H. Ni, H. Li, and L. Ying, On low-depth algorithms for quantum phase estimation, Quantum **7**, 1165 (2023).

[21] H. Li, H. Ni, and L. Ying, Adaptive low-depth quantum algorithms for robust multiple-phase estimation, Phys. Rev. A **108**, 062408 (2023).

[22] D. Slepian, Some asymptotic expansions for prolate spheroidal wave functions, J. Math. Phys. **44**, 99 (1965).

[23] H. Imai and M. Hayashi, Fourier analytic approach to phase estimation in quantum systems, New J. Phys. **11**, 043034 (2009).

[24] D. Patel, S. J. S. Tan, Y. Subasi, and A. T. Sornborger, Optimal coherent quantum phase estimation via tapering, arXiv:2403.18927.

[25] D. W. Berry, Y. Su, C. Gyurik, R. King, J. Basso, A. D. T. Barba, A. Rajput, N. Wiebe, V. Dunjko, and R. Babbush, Analyzing prospects for quantum advantage in topological data analysis, PRX Quantum **5**, 010319 (2024).

[26] L. Lin and Y. Tong, Near-optimal ground state preparation, Quantum **4**, 372 (2020).

[27] J. Lee, D. W. Berry, C. Gidney, W. J. Huggins, J. R. McClean, N. Wiebe, and R. Babbush, Even more efficient quantum computations of chemistry through tensor hypercontraction, PRX Quantum **2**, 030305 (2021).

[28] Z. Li, S. Guo, Q. Sun, and G. K.-L. Chan, Electronic landscape of the P-cluster of nitrogenase as revealed through many-electron quantum wavefunction simulations, Nat. Chem. **11**, 1026 (2019).

[29] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Encoding electronic spectra in quantum circuits with linear T complexity, Phys. Rev. X **8**, 041015 (2018).

[30] D. W. Berry, C. Gidney, M. Motta, J. R. McClean, and R. Babbush, Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization, Quantum **3**, 208 (2019).

[31] Z. Li and G. K.-L. Chan, Spin-projected matrix product states: Versatile tool for strongly correlated systems, J. Chem. Theory Comput. **13**, 2681 (2017).

[32] D. W. Berry, M. Kieferová, A. Scherer, Y. R. Sanders, G. H. Low, N. Wiebe, C. Gidney, and R. Babbush, Improved techniques for preparing eigenstates of fermionic Hamiltonians, npj Quantum Inf. **4**, 22 (2018).

[33] D. Poulin, A. Kitaev, D. S. Steiger, M. B. Hastings, and M. Troyer, Quantum algorithm for spectral measurement with a lower gate count, Phys. Rev. Lett. **121**, 010501 (2018).

[34] A. Luis and J. Peřina, Optimum phase-shift estimation and the quantum description of the phase difference, Phys. Rev. A **54**, 4564 (1996).

[35] J. Kaiser and R. Schafer, On the use of the I0-sinh window for spectrum analysis, IEEE Trans. Acoust. Speech Signal Process. **28**, 105 (1980).

[36] D. Slepian and H. O. Pollak, Prolate spheroidal wave functions, Fourier analysis and uncertainty—I, Bell Syst. Tech. J. **40**, 43 (1961).

[37] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).

[38] DLMF, *NIST Digital Library of Mathematical Functions*, https://dlmf.nist.gov/, Release 1.2.3 of December 15, 2024, F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds.

[39] W. R. Clements, P. C. Humphreys, B. J. Metcalf, W. S. Kolthammer, and I. A. Walmsley, Optimal design for universal multiport interferometers, Optica **3**, 1460 (2016).

[40] V. L. Pastor, J. Lundeen, and F. Marquardt, Arbitrary optical wave evolution with Fourier transforms and phase masks, Opt. Express **29**, 38441 (2021).

[41] M. Reck, A. Zeilinger, H. J. Bernstein, and P. Bertani, Experimental realization of any discrete unitary operator, Phys. Rev. Lett. **73**, 58 (1994).

[42] Y. R. Sanders, D. W. Berry, P. C. Costa, L. W. Tessler, N. Wiebe, C. Gidney, H. Neven, and R. Babbush, Compilation of fault-tolerant quantum heuristics for combinatorial optimization, PRX Quantum **1**, 020312 (2020).

[43] C. Gidney, Halving the cost of quantum addition, Quantum **2**, 74 (2018).

[44] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, Contemp. Math. **305**, 53 (2002).

[45] L. Lin and Y. Tong, Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers, PRX Quantum **3**, 010318 (2022).

[46] Z. Ding and L. Lin, Even shorter quantum circuit for phase estimation on early fault-tolerant quantum computers with applications to ground-state energy estimation, PRX Quantum **4**, 020331 (2023).

[47] Z. Ding, H. Li, L. Lin, H. Ni, L. Ying, and R. Zhang, Quantum Multiple Eigenvalue Gaussian filtered Search: An efficient and versatile quantum phase estimation method, Quantum **8**, 1487 (2024).

[48] A. Katabarwa, K. Gratsea, A. Caesura, and P. D. Johnson, Early fault-tolerant quantum computing, PRX Quantum **5**, 020101 (2024).

[49] A. Y. Kitaev, A. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (American Mathematical Society, Providence, Rhode Island, 2002), Vol. 47.

[50] Z. Li, J. Li, N. S. Dattani, C. Umrigar, and G. K. Chan, The electronic complexity of the ground-state of the FeMo cofactor of nitrogenase as relevant to quantum simulations, J. Chem. Phys. **150**, 024302 (2019).

[51] H. Zhai, H. R. Larsson, S. Lee, Z.-H. Cui, T. Zhu, C. Sun, L. Peng, R. Peng, K. Liao, J. Tölle, J. Yang, S. Li, and G. K.-L. Chan, BLOCK2: A comprehensive open source framework to develop and apply state-of-the-art DMRG algorithms in electronic structure and beyond, J. Chem. Phys. **159**, 234801 (2023).

[52] H. Zhai and G. K.-L. Chan, Low communication high performance *ab initio* density matrix renormalization group algorithms, J. Chem. Phys. **154**, 224116 (2021).

[53] L. Cao and U. Ryde, Influence of the protein and DFT method on the broken-symmetry and spin states in nitrogenase, Int. J. Quantum Chem. **118**, e25627 (2018).

[54] Ö. Legeza and G. Fáth, Accuracy of the density-matrix renormalization-group method, Phys. Rev. B **53**, 14349 (1996).

[55] R. Olivares-Amaya, W. Hu, N. Nakatani, S. Sharma, J. Yang, and G. K. Chan, The *ab-initio* density matrix renormalization group in practice, J. Chem. Phys. **142**, 034102 (2015).

[56] C. Xiang, W. Jia, W.-H. Fang, and Z. Li, Distributed multi-GPU *ab initio* density matrix renormalization group algorithm with applications to the P-cluster of nitrogenase, J. Chem. Theory Comput. **20**, 775 (2024).

[57] M. P. Harrigan, T. Khattar, C. Yuan, A. Peduri, N. Yosri, F. D. Malone, R. Babbush, and N. C. Rubin, Expressing and analyzing quantum algorithms with QUALTRAN, arXiv:2409.04643.

[58] V. von Burg, G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, Quantum computing enhanced computational catalysis, Phys. Rev. Res. **3**, 033055 (2021).

[59] I. Loaiza, A. M. Khah, N. Wiebe, and A. F. Izmaylov, Reducing molecular electronic Hamiltonian simulation cost for linear combination of unitaries approaches, Quantum Sci. Technol. **8**, 035019 (2023).

[60] Q. Sun *et al.*, Recent developments in the PySCF program package, J. Chem. Phys. **153**, 024109 (2020).

[61] Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K.-L. Chan, PySCF: The PYTHON-based simulations of chemistry framework, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **8**, e1340 (2018).

[62] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Elucidating reaction mechanisms on quantum computers, Proc. Natl. Acad. Sci. **114**, 7555 (2017).

[63] A. Osipov, V. Rokhlin, and H. Xiao, *Prolate Spheroidal Wave Functions of Order Zero*, Mathematical Tools for Bandlimited Approximation, Vol. 187, Springer, New York, 2013).