JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC

Perspective

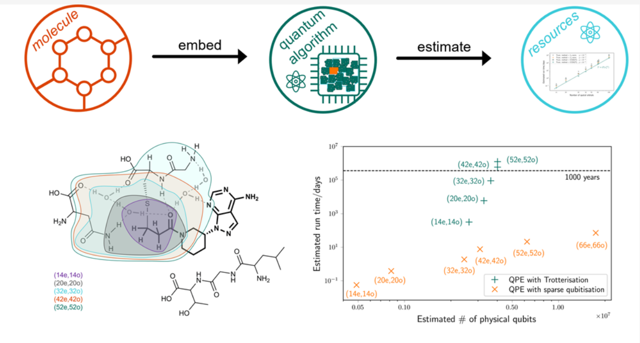
Perspective on the Current State-of-the-Art of Quantum Computing for Drug Discovery Applications

Nick S. Blunt,[§] Joan Camps,[§] Ophelia Crawford,[§] Róbert Izsák,[§] Sebastian Leontica,[§] Arjun Mirani,[§] Alexandra E. Moylett,[§] Sam A. Scivier,[§] Christoph Sünderhauf,[§] Patrick Schopf, Jacob M. Taylor, and Nicole Holzmann*

Cite This: *J. Chem. Theory Comput.* 2022, 18, 7001−7023

Read Online

ACCESS | Metrics & More | Article Recommendations



ABSTRACT: Computational chemistry is an essential tool in the pharmaceutical industry. Quantum computing is a fast evolving technology that promises to completely shift the computational capabilities in many areas of chemical research by bringing into reach currently impossible calculations. This perspective illustrates the near-future applicability of quantum computation of molecules to pharmaceutical problems. We briefly summarize and compare the scaling properties of state-of-the-art quantum algorithms and provide novel estimates of the quantum computational cost of simulating progressively larger embedding regions of a pharmaceutically relevant covalent protein−drug complex involving the drug Ibrutinib. Carrying out these calculations requires an error-corrected quantum architecture that we describe. Our estimates showcase that recent developments on quantum phase estimation algorithms have dramatically reduced the quantum resources needed to run fully quantum calculations in active spaces of around 50 orbitals and electrons, from estimated over 1000 years using the Trotterization approach to just a few days with sparse qubitization, painting a picture of fast and exciting progress in this nascent field.

1. INTRODUCTION

The drug design process is a complex procedure in which computers and wet-lab methods are used together in pursuing new pharmaceuticals. Although most methods of computer-aided drug design (CADD) rely on statistical fitting methods or on classical mechanics,[1] it has been argued that more accurate quantum mechanical methods have an important contribution to make to several aspects of CADD.[2−4] Unfortunately, finding exact or nearly exact solutions for chemically relevant systems becomes intractable for more than ∼30 electrons.[5] Although efficient and accurate approximations exist for much larger systems than that, it remains desirable to find methods that can deliver the exact quantum mechanical solution in a cost-efficient way.

Being quantum systems themselves, quantum computers are naturally suited to simulating quantum mechanical problems without running out of memory exponentially fast. Many aspects of chemical research are expected to benefit from accurate quantum methods, including in the pharmaceutical industry.[6,7] Carrying out such industrially disruptive quantum simulations requires very high fidelity quantum computers.

Quantum computers have seen a significant number of experimental developments over the past several years. Recent trapped ion quantum devices have an average of two-qubit gate fidelities of up to 99.8%.[8] Developments have also been seen in superconducting technologies, most famously shown in the 50−60 qubit devices from Google and USTC, which claim to show quantum computational advantage, the point where a quantum computer is believed to have solved a classically intractable problem, albeit not a problem with applications in quantum chemistry.[9,10] Superconducting devices have also been developed with over 120 qubits, in particular IBM's Eagle processor.[11] These devices are now at a point where it is possible to run noisy intermediate-scale quantum algorithms such as the variational quantum eigensolver,[12] with the largest experimental efforts to date simulating the binding energy of hydrogen chains with up to 12 atoms.[13−15] Along with early applications, experimental groups have started showing initial implementations of quantum error correction, a fundamental

Received: June 1, 2022
Published: November 10, 2022

https://doi.org/10.1021/acs.jctc.2c00574

step in scaling up quantum computing where multiple physical qubits are used to protect a smaller number of logical qubits from errors,[16] thus increasing the effective fidelity. These experiments have been shown to suppress errors while keeping a single logical qubit alive and applying some simple logical gates in a number of platforms, including superconducting devices,[17] trapped ions,[18−21] and nuclear-spin qubits in diamond.[22] These results show the significant progress that has been made in quantum hardware, as well as laying the groundwork to reaching large-scale fault-tolerant quantum computation.

This perspective focuses on the disruption enabled by the large-size complete active space configuration interaction (CASCI) calculations admitted by near-future quantum computers. We discuss the steps involved in running a pharmaceutical application on a quantum computer, from mapping the chemical problem onto quantum memory, selecting a quantum algorithm, and specifying an error-corrected quantum architecture to solve it. We illustrate these steps with an example system: the drug Ibrutinib bound covalently to Brutons tyrosine kinase. We estimate the quantum computational resources needed to fully quantum simulate progressively larger clusters of the binding pocket and the Ibrutinib inhibitor. Our estimates exhibit that quantum algorithmic developments over the past five years have dramatically reduced the quantum resources needed to run fully quantum calculations in active spaces of around 50 orbitals, which could be performed on sufficiently large error-corrected quantum computers with a runtime of just a few days.

This perspective is organized as follows. Section 2 discusses the mapping of the electronic structure problem onto a quantum computer. Section 3 discusses and compares the scaling of the two salient quantum algorithms for finding the ground-state energy of an electronic Hamiltonian—variationa quantum eigensolver (VQE) and quantum phase estimation (QPE). We conclude that QPE scales more favorably, and the rest of the work focuses on this algorithm. Section 4 discusses the aspects of error correction needed for estimating the quantum resources needed to run QPE. The main ingredient of the QPE algorithm is an efficiently implemented unitary operator that is related to the Hamiltonian. Section 5 discusses two methods to construct such unitary operators: Trotterization and qubitization. Section 6 discusses the pharmaceutical system of focus, the computational methods, and the active spaces used. Section 7 contains the results of our resource estimations. We find that qubitization gives much more favorable runtimes than Trotterization. We conclude in Section 8.

2. CHEMISTRY ON A QUANTUM COMPUTER

2.1. Chemistry and the Electronic Structure Problem.

The question of how particles interact had already led the ancient Greek and Roman atomists to talk about “hooked atoms” that could intertwine and hold matter together. After atomism was revived two millennia later, in his Opticks, Newton preferred to hypothesize an attractive force, as yet unknown, that holds atoms together. Accumulating knowledge on electricity and electrochemistry in the 19th century favored explanations featuring electrostatic interactions in this role. The period also saw the rise of the theory of chemical valency that sought to determine the number of partners an atom might have in a compound and eventually led to the characterization of the combining forces as chemical bonds.[23] After the discovery of the electron and the refinement of atom models that culminated in Bohr’s model in 1913, G. N. Lewis put forward his own interpretation of the (covalent) chemical bond as electron pairs shared between atomic nuclei[24] and to give a physical picture of that “hook and eye”, as he put it.[25] But the real breakthrough promising quantitative predictions came with the early application of quantum mechanics to simple chemical systems in the late 1920s, including the works of Burrau on H_2^+,[26] Heitler and London,[27] and later Pauling[28] anticipating valence bond theory, with Hund,[29] Mulliken,[30] and Lennard-Jones[31] laying the foundations for molecular orbital theory. However, starting from the first-principles of quantum mechanics and special relativity leads to equations that are insoluble in all but the simplest of cases, as Dirac lamented in 1929, concluding that more efficient approximate solutions are necessary.[32] Current wave function based methods of quantum chemistry rely on a series of approximations that lead to a computable first guess, and then, whenever possible, various other methods are applied to account for the approximations made or, as computational chemists say, to correct for the various “effects” neglected. To begin with, typically a single molecule in a vacuum is considered, without taking relativity into account and considering static solutions only. To facilitate a quantum mechanical treatment, one must be able to represent the interactions among the electrons and nuclei of molecules as a linear Hermitian operator, the Hamiltonian, the eigensolutions of which represent the possible states of the system and the total energies associated with them. To simplify the problem further, the Born−Oppenheimer approximation[33] posits that electronic and nuclear degrees of freedom can be separated, after which most methods focus on tackling the electronic problem. The resulting time-independent Schrödinger equation reads

$$\hat{H}\Psi_k = E_k\Psi_k, \qquad \hat{H} = \hat{T}_e + \hat{V}_{ee} + \hat{V}_{ne} + \hat{V}_{nn} \tag{1}$$

where Ψ_k and E_k are the electronic wave function and energy of the kth state and $\hat{H}$ is the electronic Hamiltonian consisting of a kinetic energy term of the electrons $(\hat{T}_e)$ and the potential energy terms of the electron−electron $(\hat{V}_{ee})$, nuclear−electron $(\hat{V}_{ne})$, and nuclear−nuclear $(\hat{V}_{nn})$ interactions.

At this stage, the computational problem is still intractable. Further progress was made by assuming that electronic coordinates can also be separated and the total wave function has the form of a Slater determinant[34]

$$\Phi = \hat{\mathcal{A}} \prod_{i=1}^{N} \varphi_i(i) \tag{2}$$

where the antisymmetrizer $\hat{\mathcal{A}}$ permutes the particle labels and sums over terms with the appropriate sign and norm factor. Thus, the exact wave function describing N electrons is approximated as a determinant Φ constructed from functions describing a single electron, the molecular (spin−)orbitals φ_i. Calculating the expectation value of $\hat{H}$ using Φ and minimizing it with respect to the orbitals yields the Hartree−Fock (HF) equations.[35−37] To make the parametrization of this problem easier, the molecular orbitals themselves are expanded as a linear combination of known atomic orbitals. In molecular calculations, a convenient choice for the latter is Gaussian functions and, using these, the Hartree−Fock equations are reduced to a set of algebraic equations for the expansion coefficients of molecular orbitals.[38] While the algebraic Hartree–Fock problem is soluble for molecules containing several hundred atoms, being an effective one-electron theory, it does not account for correlation effects between multiple electrons.[39] However, once the molecular orbitals are obtained in a given atomic orbital basis, a linear combination of all possible Slater determinants will yield the exact solution in that basis. Unfortunately, this full configuration interaction (FCI) solution scales exponentially with the number of electrons and orbitals in the system. The classical solution to the problem is to define less expensive ansätze for the wave function that only scale polynomially,[39] e.g., the coupled-cluster singles and doubles (CCSD) ansatz. When correlation effects are weak, i.e., when HF is a good starting guess, this approach has been extremely successful in many areas of chemistry. For strongly correlated systems, the most straightforward alternative to HF is obtaining the FCI solution within a complete active space (CAS)[40] rather than for the entire orbital space. We will refer to the configuration interaction solution within this active space (for our purposes, without orbital optimization) as CASCI. Unfortunately, this still leaves many important problems outside the reach of quantum chemical methods on the classical computer. For strongly correlated systems, the main difficulty lies in the size of the active spaces required for correctly describing some systems, an area where quantum computers may make a breakthrough.[41] For weakly correlated systems, high-quality results delivered by quantum computers may still yield significant improvements over popular density functional theory (DFT) approaches[42] or even efficient wave function based approaches on the classical computer.

As it was pointed out even in the case of an archetypal strongly correlated complex,[41] outperforming popular density functional methods is an important practical criterion. In contrast, usual definitions of quantum advantage involve formal criteria such as exponential speed-up, the relevance of which to chemistry has been recently questioned.[43] Here, a less ambitious working definition is used: quantum benefit is reached if quantum computers can outperform classical computers in some industrially relevant process. An important part of that is the ability to give better results than DFT at a reasonable cost. Recent work on photochemical processes with simulated quantum computing uses precisely such criteria.[42] Unfortunately, the assessment of potential quantum benefit for pharma remains a difficult task, not in the least because predictive application of quantum mechanics in the drug discovery process is a relatively new trend even using classical computers.[3] Some areas where quantum effects are known to be important, such as the description of weak hydrogen bonds,[44] also stand out as the most likely candidates for quantum benefit. In this perspective, our aim is to provide quantum resource estimates for a protein–drug system in which such interactions play an important role.

2.2. Quantum Computation.

Quantum computers are computational devices that use the laws of quantum mechanics to perform calculations. The theory of quantum computing was first developed in the early 1980s by pioneers including Paul Benioff, Richard Feynman, David Deutsch, and Peter Shor.[45] The motivation for quantum computing comes from the potential to perform calculations efficiently, which can only be performed inefficiently on a digital computer. Here, efficient means that the runtime is polynomial in the size of the problem.

This initial work led in 1994 to the development of Shor's algorithm,[46] which allows the prime factorization of an integer to be performed in polynomial time, compared to the superpolynomial time required by classical algorithms. In 1996, Lov Grover developed an algorithm to search an unsorted database of size N in $O(\sqrt{N})$ time, compared to the $O(N)$ runtime required by classical algorithms.[47] These discoveries demonstrate the potential for improved performance of certain quantum algorithms over classical ones.

The key to efficiently studying chemistry on a quantum computer came in the late 1990s. Alexei Kitaev, building on the work of Shor, introduced quantum phase estimation (QPE) in 1995 to study the Abelian stabilizer problem.[48] In 1998, Cleve et al. extended this QPE approach to estimate the phase of an arbitrary unitary operator;[49] the form of QPE introduced here is identical to that often considered today. The QPE method can be applied to find the eigenvalues of a chemical Hamiltonian to a given accuracy with a runtime that scales polynomially with system size. For this reason, we believe that quantum computers can perform accurate chemical calculations beyond the reach of classical devices.

A quantum computer consists of a register of qubits or quantum bits. Each of these qubits can be in a state $|0\rangle$ or $|1\rangle$. However, following the laws of quantum mechanics, the state can also be an arbitrary superposition of the two

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{3}$$

in addition to possible entanglement between the qubits. Time evolution in quantum mechanics is unitary, and as such the gates performed on the qubits are unitary operations too. In particular, a quantum computer is built to perform a small set of basis unitary operations at the physical level. These operations are designed to be universal; that is, any unitary operator on any number of qubits can be built from these basis gates. This can be achieved using gates that only act on one or two qubits at a time, a fact that is crucial for physical realizations of quantum computers; it is not realistic to perform physical operations that entangle large numbers of qubits simultaneously with high fidelity. Instead, these operations can be built from much simpler physical operations. Finally, state preparation and measurement are important components of quantum computation; qubits are each prepared in state $|0\rangle$ at the start of a computation, and measurement causes wave function collapse according to the Born rule.

One set of universal gates, which will be important for later discussion of quantum error correction, consists of the Hadamard gate (H) and phase gates S and T, defined in matrix form by

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix} \tag{4}$$

and the CNOT gate, defined by

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{5}$$

which flips the state of a second qubit, conditional on the first being in state $|1\rangle$. Programs built from these gates are often represented by circuit diagrams. An example of the QPE circuit diagram is given later in Section 2.4.2. For a thorough introduction to quantum computing, including circuit diagram notation, we refer the reader to ref 50.

2.3. Qubit Hamiltonian. As discussed above, the great promise of quantum computers for chemistry is that they can find eigenvalues of a Hamiltonian with polynomial scaling. This would render a great number of strongly correlated chemical problems amenable to exact quantum mechanical treatment with potential benefits in many branches of the chemical industry.[41] To realize this promise, the Hamiltonian encoding the interactions in the chemical system needs to be represented in a way that the quantum computer will be able to interpret. One possibility for this is the second quantized representation

$$\hat{H} = h_0 + \sum_{p,q} h_p^q a_p^\dagger a_q + \frac{1}{2}\sum_{p,q,r,s} h_{pr}^{sq} a_p^\dagger a_r^\dagger a_q a_s \tag{6}$$

where the Fermionic annihilation (a_q) and creation $(a_p^\dagger)$ operators are summed over the molecular spin−orbital labels p, q, ... within the active space. It is important to emphasize that such active spaces are often chosen to reduce the cost of the calculation and involve the projection of the full Hamiltonian to the CAS space.[51] This causes screening terms to appear in the matrix elements: the constant term h_0 contains the nuclear−nuclear interaction and any screening terms, the one-body term h_p^q includes the kinetic and nuclear−electron attraction as well as any screening terms, and the two-body term consists of the interelectronic repulsion term. Once the Hartree−Fock solution or some other set of molecular orbitals is available to define p, q, ..., it is possible to generate h_0, h_p^q, and h_{pr}^{qs}. The result of a quantum computation using such a CAS Hamiltonian corresponds to a CASCI calculation on the classical computer that becomes identical with the exact (FCI) solution in the limit that the active space includes the entire orbital space in a given basis.

In the next step, the Fermionic operators need to be mapped to qubit operators, whose action on the qubits can be directly calculated. The Jordan−Wigner transformation[52] achieves this using Pauli spin-matrices and requiring that the new representation satisfy the anticommutation rules of Fermion operators. The resulting transformation for creation operators reads

$$a_p^\dagger \to \frac{1}{2}(X_p - iY_p) \bigotimes_{q<p} Z_q \tag{7}$$

and for annihilation operators, it is

$$a_p \to \frac{1}{2}(X_p + iY_p) \bigotimes_{q<p} Z_q \tag{8}$$

where X_p, Y_p, and Z_p are Pauli spin operators acting on the pth qubit. It should be noted that there are alternatives to the Jordan−Wigner transformation, and it has recently been argued that, for larger chemical problems, the one proposed by Bravyi and Kitaev[53,54] will be more advantageous.[55] Whichever method one chooses, the result is a qubit Hamiltonian, i.e., a linear combination of Pauli strings that represent the chemical system for the quantum computer and serve as a starting point for quantum algorithms. We write this as

$$\hat{H} = \sum_{i=1}^{L} w_i P_i \tag{9}$$

where each P_i is a Pauli operator and w_i its corresponding (real) coefficient.

There are two main classes of algorithm for performing computational chemistry calculations on quantum computers—the variational quantum eigensolver (VQE) and quantum phase estimation (QPE). The focus of this work is using the latter to estimate the quantum computational resources required to perform pharmaceutically relevant chemistry calculations.

2.4. Algorithms. In this section, we outline the VQE and QPE algorithms.

2.4.1. Variational Quantum Eigensolver. VQE[12] is a hybrid algorithm, making use of both classical and quantum computational resources. A classical optimizer explores some set of quantum states, seeking that with the smallest Hamiltonian expectation value. By the variational principle, any such expectation value is necessarily greater than or equal to the ground-state energy. It is therefore hoped that the smallest expectation value will be close to the ground-state energy. Excited-state energies can also be sought through extensions (e.g., see refs 56−60).

The set of states explored is known as an ansatz. These states are prepared through some parametrized quantum circuit. Having chosen some initial parameter values, the ansatz circuit is run to prepare a particular quantum state and measurements of the state made. Typically the ansatz circuit must be applied many times to obtain sufficient information to estimate the expectation of the Hamiltonian on the quantum state to some desired level of accuracy. On the basis of this expectation, the parameter values are updated by the classical optimizer and the expectation estimation pSocess repeated until some convergence criteria are satisfied. The VQE algorithm is illustrated in Figure 1, and we provide further details of the different aspects of the algorithm in section 3.1.1.

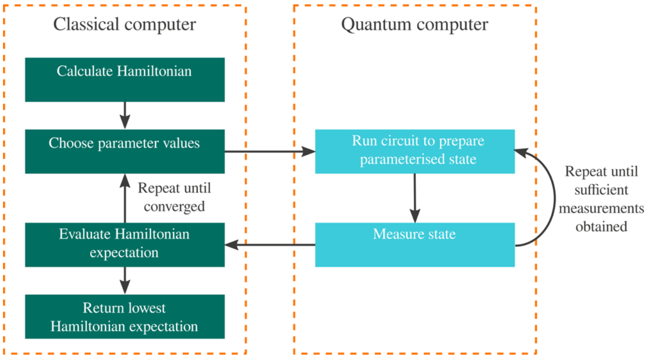

Classical computer
Quantum computer
Calculate Hamiltonian
Choose parameter values
Run circuit to prepare parameterised state
Repeat until converged
Repeat until sufficient measurements obtained
Evaluate Hamiltonian expectation
Measure state
Return lowest Hamiltonian expectation

Figure 1. Outline of the VQE algorithm, indicating which parts occur on the quantum computer and which parts on the classical.

2.4.2. Quantum Phase Estimation. Quantum phase estimation is another algorithm for calculating energies of chemical systems.[48,62] It requires deeper circuits than VQE, but these must be performed typically only a handful of times. Furthermore, QPE does not require an ansatz; it instead calculates eigenvalues of the Hamiltonian directly, up to some level of precision.

QPE makes use of the quantum Fourier transform to estimate the eigenphases of a unitary operator, U. An eigenphase, φ_i, satisfies

$$U|\Psi_i\rangle = e^{i\varphi_i}|\Psi_i\rangle \tag{10}$$

where $|\Psi_i\rangle$ is the corresponding eigenstate. In order to perform computational chemistry calculations, the unitary must be constructed from the Hamiltonian; one choice is $U = e^{-i\hat{H}t}$. The operators U and $\hat{H}$ share eigenstates, and their eigenvalues are related through $E_i t = -\varphi_i$. An outline of the circuit used to perform QPE is shown in Figure 2. There are two sets of qubits; the state register (bottom) used to prepare the eigenstate by the end of the calculation and the data register (top) used to read bits corresponding to $\frac{\varphi_i}{2\pi}$ in binary fraction representation. The precision of the estimate of the energy is thus limited by the number of qubits in the data register.

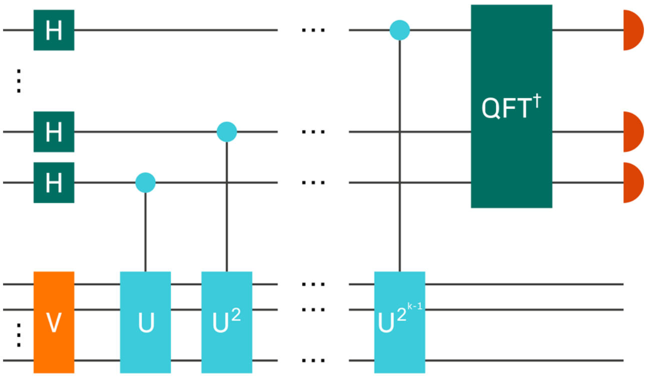


Figure 2. Outline of the circuit used to perform QPE, as discussed in ref 61.

Initially, the state register is prepared to contain a state which is hoped to have significant overlap with the true ground state—a common choice when performing chemistry calculations is the Hartree–Fock state. A sequence of controlled unitaries $U^{2^{k-1}}$ are then applied to the state register, controlled on the kth qubit of the data register, followed by performing the inverse quantum Fourier transform on the data register. Finally, the data register is measured to obtain an estimate of an eigenphase. The probability of the estimate corresponding to a particular eigenphase is given by the overlap probability of the initial state with the corresponding eigenstate.

3. ALGORITHM CHOICES

In this section, we discuss the reasons for our focus on QPE and present some details of our resource estimation calculations. Further details are presented in following sections.

3.1. Algorithm Scaling. In order to motivate our algorithmic choices, we first present simple scaling estimates for the resources required to perform the algorithms outlined in Section 2.4. We present this scaling in terms of a few key parameters, ignoring the coefficients. We assume that the total time scaling takes the form

$$n_t = \frac{n_g n_{rep}}{n_{QPU}} \tag{11}$$

where n_g is the gate depth of a single circuit, n_{rep} is the number of times this circuit must be performed, and n_{QPU} is the number of available quantum processors of suitable size, that is, the number of circuits that can be performed in parallel. The gate depth is the number of layers of gates that must be applied, where a layer of gates is a set of gates that can be applied simultaneously.

We will now outline the form of these components for the two algorithms. We define n_o and n_e to be the number of spin orbitals (which we note is twice the number of spatial orbitals) and number of electrons in the chemical system respectively and

$$E_{max} = \sum_{i=1}^{L} |w_i| \tag{12}$$

which gives the maximum possible value of any energy. We recall the w_i coefficients are those in the qubit form of the Hamiltonian, eq 9, and L is the total number of terms. In both cases, the time scaling will also depend on the desired level of accuracy, ϵ, in the energy estimate.

3.1.1. VQE Resources. In this section, we perform a rough estimate of the time taken to perform a VQE calculation, considering eq 11. For a VQE calculation, the circuit depth n_g will depend on the ansatz. There may also be some additional depth required to, for example, measure the appropriate operators; in general, we aim to make assumptions that are favorable toward VQE, and thus we will ignore this additional circuit depth here. The number of times the circuit must be repeated, n_{rep}, will be the product of two factors—the number of circuit applications required to obtain a single estimate of the Hamiltonian expectation, n_a, and the number of Hamiltonian expectations required in order to optimize the parameters, n_h, so that

$$n_{rep} = n_a n_h \tag{13}$$

A significant degree of parallelization is possible in VQE; we will return to this once we have outlined the QPE scaling too.

We will now outline the form of the components in more detail. For a full review of VQE and its components, see Tilly et al.;[63] a scaling of VQE is also presented there, though different assumptions are made compared to ours.

3.1.1.1. Number of Qubits. We first define the number of qubits, n_q, needed to represent the relevant quantum states on the quantum computer. We will assume that we have one qubit per spin orbital, so

$$n_q^{VQE} = n_o \tag{14}$$

This is the case for both the Jordan–Wigner and Bravyi–Kitaev transformations mentioned in Section 2.3. It is, however, typically possible to reduce this number slightly by conserving symmetries of the chemical system;[64] however, this will not have a large effect on our calculation and so we ignore this possibility here.

3.1.1.2. Number of Parameters. It will next be important to consider the ansatz. The choice of ansatz plays a key role in determining the performance of a VQE calculation. The ideal ansatz would

- enable preparation of a state close to the true ground state;
- require as few parameters as possible, so as to minimize the time required to perform the classical optimization;
- use as few quantum computational resources as possible.

In general, the ansatz circuit depth will thus depend on the accuracy, ϵ, as a deeper circuit will typically allow a state closer to the true ground state to be prepared. However, it is difficult to quantify the relationship between n_g and ϵ. In this section, we will consider a fixed ansatz—the unitary coupled-cluster singles–doubles (UCCSD) ansatz.[65] This is a chemically inspired ansatz, which means we have reason to believe the ansatz space contains chemically relevant systems. The circuit for this ansatz prepares the states

$$|\psi(\boldsymbol{\theta})\rangle = e^{\hat{T}-\hat{T}^{\dagger}}|\mathrm{HF}\rangle \tag{15}$$

where $|\mathrm{HF}\rangle$ is the Hartree−Fock state and

$$\hat{T} = \hat{T}_1 + \hat{T}_2 = \sum_{i,a} \theta_a^i a_a^{\dagger} a_i + \sum_{i,j,a,b} \theta_{ab}^{ij} a_a^{\dagger} a_b^{\dagger} a_j a_i \tag{16}$$

By varying the parameters θ_a^i and θ_{ab}^{ij}, we can produce different quantum states. Here, i, j, ... refer to occupied orbitals in the Hartree−Fock state and a, b, ... refer to virtual orbitals. In the VQE process, the θ parameters are optimized. We can see that the operator $\hat{T}$ contains components corresponding to single and double excitations of electrons from the Hartree−Fock state. For strongly correlated systems, UCCSD may not be able to prepare a state which is suitably close to the ground state due to the limitation on the excitations considered.

The key property of the ansatz that affects the VQE calculation time is the number of parameters, n_{p}. For the UCCSD ansatz, this is

$$n_{\mathrm{p}} \sim n_{\mathrm{e}}^{2}(n_{\mathrm{o}} - n_{\mathrm{e}})^2 \tag{17}$$

which is the scaling of the number of θ_{ab}^{ij} parameters. We will assume that both n_{e} and $n_{\mathrm{o}} - n_{\mathrm{e}}$ scale linearly with n_{o} and so

$$n_{\mathrm{p}} \sim n_{\mathrm{o}}^{4} \tag{18}$$

Finding good ansatze is a topic of ongoing research. More recently proposed ansatz methods, which typically seek to reduce the number of parameters and/or gate depth required for a given level of accuracy, include the k-UpCCGSD[66] ansatz and adaptive ansatz procedures such as ADAPT-VQE.[67] These have been shown to outperform UCC in some cases; however, their performance with larger chemical systems is difficult to predict. Ansatze with further improved behavior may be developed in the future.

3.1.1.3. Number of Hamiltonian Expectations.

The number of Hamiltonian expectations required in a particular VQE calculation is difficult to know in advance as it will depend on the shape of the ansatz parameter space. Here, we make a favorable assumption. We will assume that the number of Hamiltonian expectations required, n_{h}, is simply given by

$$n_{\mathrm{h}} = n_{\mathrm{p}} \tag{19}$$

This, for example, could arise if the optimizer needs only to look in each parameter direction once, perhaps to verify that a minimum has already been found. Needing any fewer evaluations would imply that it was known before the calculations occurred that some parameters were not needed in the ansatz. Typical calculations will require more evaluations than this.

3.1.1.4. Number of Ansatz Circuit Applications.

As mentioned above, the ansatz circuit must typically be applied many times in order to obtain an expectation of the Hamiltonian with respect to a particular state to a sufficient degree of accuracy. The number of applications needed depends on the form of the Hamiltonian and the particular quantum state.

It is typically not possible to obtain measurements of the Hamiltonian directly. However, measurements of Pauli operators can easily be obtained, and so we can make use of eq 9 in calculating the expectation of the Hamiltonian. Assuming measurements of each Pauli are obtained separately, the number of times the ansatz circuit must be performed is given by[68]

$$n_{\mathrm{a}} = \left(\frac{1}{\epsilon} \sum_{i=1}^{L} |w_i| \sqrt{\mathrm{Var}[P_i]} \right)^2 \tag{20}$$

where ϵ is the desired error in the expectation estimate and

$$\mathrm{Var}[P_i] = 1 - \langle \psi(\boldsymbol{\theta})|P_i|\psi(\boldsymbol{\theta})\rangle^2 \tag{21}$$

The maximum value of each variance is 1 and so

$$n_{\mathrm{a}} \leq \left(\frac{E_{\max}}{\epsilon} \right)^2 \tag{22}$$

In the following we take the equality in this expression. This is likely an overestimate in practice, but we believe it is sufficient to demonstrate the challenges faced. We later take a generous assumption in the scaling of $E_{\max}$.

We note that it is possible to reduce this through several different methods. It is possible to improve upon the assumption that we measure each Pauli individually by, for example, measuring commuting Paulis simultaneously[69−72] or factorizing the two-electron integral tensor.[73] Such methods reduce the overall number of measurements required while retaining the scaling in $\left(\frac{1}{\epsilon^2}\right)$. This scaling can also be improved using QPE-inspired methods at the cost of an increased circuit depth;[74,75] however, such increased depths are unlikely to be possible in the noisy intermediate-scale quantum (NISQ) era, before error correction is available.

3.1.1.5. Circuit Depth.

We will assume that it is possible to perform $O(n_{\mathrm{q}})$ parameters per layer of gates and so

$$n_{\mathrm{g}}^{\mathrm{VQE}} \sim n_{\mathrm{o}}^{3} \tag{23}$$

for the UCCSD ansatz outlined above.

3.1.1.6. Summary.

We therefore see that, for VQE, given the assumptions we have made,

$$n_{\mathrm{q}}^{\mathrm{VQE}} \sim n_{\mathrm{o}}, \quad n_{\mathrm{g}}^{\mathrm{VQE}} \sim n_{\mathrm{o}}^{3}, \quad n_{\mathrm{rep}}^{\mathrm{VQE}} \sim \frac{n_{\mathrm{o}}^{4} E_{\max}^{2}}{\epsilon^2} \tag{24}$$

The degree of parallelization will depend on the total number of qubits available; we will discuss this once we have considered QPE.

3.1.2. QPE Resources.

In contrast to VQE, it is possible to make a good estimation of the quantum computational resources required to perform a QPE calculation for a given chemical system. However, this does depend on the probability overlap of the initial state with the ground state, η. In this section, we present a rough scaling of the time taken to perform a QPE calculation.

3.1.2.1. Circuit Depth and Number of Repetitions.

Considering first only the parameters ϵ and η, using textbook phase estimation, one expects[76]

$$n_{\mathrm{g}}^{\mathrm{QPE}} \sim \frac{1}{\epsilon\eta}, \quad n_{\mathrm{rep}}^{\mathrm{QPE}} \sim \frac{1}{\eta} \tag{25}$$

The scaling with the properties of the Hamiltonian depends on the specifics of the quantum phase estimation calculation (see later sections for details). Using the most recent methods,[77] the circuit depth depends on $E_{\max}$, which we recall was defined in eq 12, and n_o, the number of orbital basis functions, so that

$$n_g^{QPE} \sim \frac{E_{\max} n_o}{\epsilon \eta} \tag{26}$$

3.1.2.2. Number of Qubits. Like VQE, QPE requires approximately n_o qubits to store the relevant quantum state. However, QPE typically also requires additional auxiliary qubits. First, such qubits are needed to store the bits corresponding to the energy estimate. For the specific version of QPE outlined in Section 2.4.2, the number of additional bits is $\log_2(1/\epsilon)$; however, this can be reduced to just a single qubit using iterative phase estimation.[62,78] Auxiliary qubits are also required for some methods of implementing the required unitary operators. For the most recent methods,[77] the number of additional qubits required is $\tilde{O}(n_o)$, and so

$$n_q \sim n_o \tag{27}$$

3.1.2.3. Error Correction Overhead. Because the circuits used when performing QPE are very deep, we expect error correction procedures to be required in order to obtain useful results from the calculations. This introduces an overhead, both in the number of qubits (spatial overhead) and the depth of the circuit (temporal overhead). We write these overheads as θ_S and θ_T, respectively. For the surface code, the overheads are determined by the code distance, d, with $\theta_S \sim d^2$ and $\theta_T \sim d$. We can therefore write $\theta_S \sim {\theta_T}^2$. We note that, in order to maintain a constant probability of a logical error occurring, these overheads must increase with increasing logical circuit depth and number of logical qubits; however, they increase logarithmically and so we ignore this here. This can be seen from eq 34, as will be motivated in Section 4.3. We therefore write

$$n_q \sim n_o {\theta_T}^2, \quad n_g^{QPE} \sim \frac{E_{\max} n_o \theta_T}{\epsilon \eta} \tag{28}$$

For further information about the error correction overhead in the context of the surface code, see Section 4.

3.1.2.4. Summary. For QPE, we therefore have

$$n_q^{QPE} \sim n_o {\theta_T}^2, \quad n_g^{QPE} \sim \frac{E_{\max} n_o \theta_T}{\epsilon \eta}, \quad n_{rep}^{QPE} \sim \frac{1}{\eta} \tag{29}$$

3.1.3. Comparison and Discussion. We presented the scaling of the number of qubits, circuit depth, and number of circuit repetitions for VQE and QPE in eqs 24 and 29, respectively. We will now consider n_{QPU} in the two cases. We will assume that the total number of available qubits in the two cases is n_q^{QPE}. We note that, should additional qubits be available, some degree of parallelization is possible for QPE as the procedure must be repeated some number of times. However, as this factor would be the same in both the VQE and QPE scalings, we do not consider it further. Therefore, the degree of parallelization possible for VQE is

$$n_{QPU}^{VQE} = \frac{n_q^{QPE}}{n_q^{VQE}} \sim {\theta_T}^2 \tag{30}$$

Putting everything together, we therefore have, from eq 11,

$$n_t^{VQE} \sim \frac{{n_o}^7 {E_{\max}}^2}{{\theta_T}^2 \epsilon^2}, \quad n_t^{QPE} \sim \frac{E_{\max} n_o \theta_T}{\epsilon \eta^2} \tag{31}$$

and so

$$\frac{n_t^{VQE}}{n_t^{QPE}} \sim \frac{{n_o}^6 E_{\max} \eta^2}{{\theta_T}^3 \epsilon} \tag{32}$$

We will now consider the scaling of some of these terms further. The scaling of $E_{\max}$ with n_o is typically examined numerically and depends on the specific chemical system and, for example, whether the increase in n_o is due to the increasing number of atoms or increasing basis set size. Best estimates find it scales between n_o and ${n_o}^3$.[77] Here we take it to scale as n_o, so as to be favorable toward the VQE scaling. We can consider ϵ and η to be constant. The degree of accuracy in the final energy estimate, ϵ, will typically be taken to be chemical accuracy and thus not depend on the size of the system. While η, the overlap of the initial state with the true ground state, will be system-dependent, it has been suggested that it is typically possible to prepare a simple state with a good degree of overlap.[79] We argued above that the increase in θ_T should be logarithmic and did not consider such logarithmic factors. We therefore find

$$\frac{n_t^{VQE}}{n_t^{QPE}} \sim {n_o}^7 \tag{33}$$

and thus expect QPE to become preferable once the system is large enough, this size being determined by the constant in front of the scaling, which we have ignored in our analysis. This motivates our choice of QPE for the remainder of this work.

Our scaling analysis is not intended to be definitive, and we acknowledge that there may be possible improvements to VQE. However, VQE presents some more general difficulties. As mentioned above, the choice of ansatz is key to determining how close it is possible to get to the true ground state—it is difficult to guarantee that the ansatz allows preparation of a state that is close to the true ground state without having a large number of parameters and circuit depth. Furthermore, even if the ansatz can describe the desired state, there is no guarantee that the optimizer will find it—it may instead converge to a local minimum.

In this work, we have considered only the scaling of VQE and compared it to that of QPE. Other works have performed resource estimations for VQE and find runtimes to be prohibitively large.[80−82] For example, Gonthier et al.[81] estimate runtimes of approximately 1−100 days to perform a single expectation evaluation for systems requiring approximately 100−300 qubits.

3.2. QPE in This Work.

Having motivated our choice of QPE, we now present some details of the algorithmic choices made in this work. Further choices are discussed in the following sections.

As mentioned above, in contrast to VQE, it is possible to make a good estimation of the quantum computational resources required to perform a QPE calculation for a given chemical system. In this work, we present results of resource calculations, given the chemical system and desired accuracy as inputs and further allow for several algorithmic choices to be made.

After a single run of phase estimation, an energy estimate is obtained. However, this estimate may not be within the desired accuracy of the true energy. This can be for several reasons. First, it is possible that the estimate is of an energy other than the desired ground-state energy. Second, as the true energies can typically not be represented exactly in the finite number of

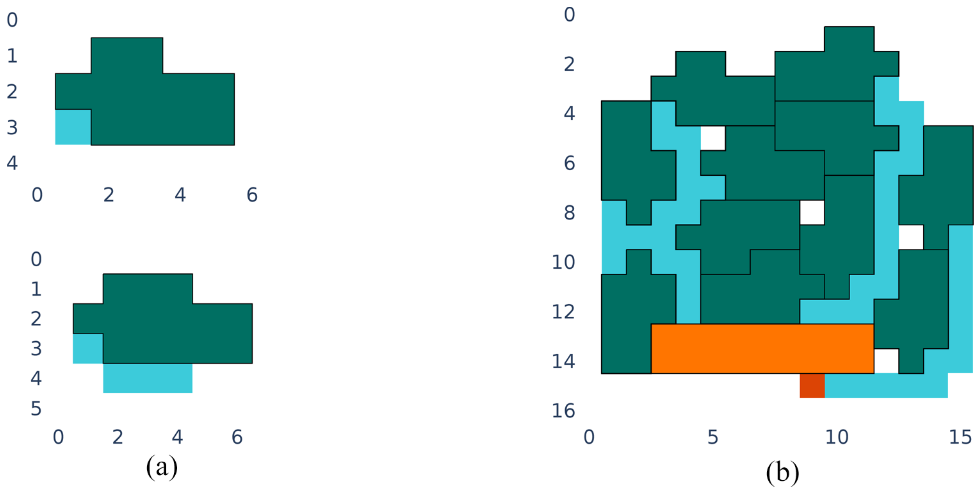


Figure 3. (a) Layouts for 15-to-1 (top) and 20-to-4 (bottom) magic-state factories. These consist of 11 and 14 logical qubits, respectively (green). The magic states produced are stored in the blue spaces. (b) Factory which distills 225 imperfect magic states to one higher quality magic state. Eleven first-level 15-to-1 factories (green) are used to produce 15 refined magic states, which are in turn used by the second-level 15-to-1 factory (orange) to produce one magic state of even higher quality (red). Blue lines are used to store and transport lower-quality magic states. White spaces are unused logical qubits.

chosen bits, there is some probability that an estimate, even of the correct eigenstate, will not be to the desired level of accuracy. Third, it is possible that our error correction procedure failed and so a logical error occurred, making any result obtained inaccurate. It is thus necessary to repeat the phase estimation procedure several times, the number depending on the overall desired probability of success. In this work, we do not explicitly calculate the number of repetitions needed but outline one possible method for doing so in Appendix A. Alternative methods exist in the literature.[76,83]

4. IMPLEMENTING ERROR-CORRECTED QUANTUM ALGORITHMS

4.1. Quantum Error Correction and the Surface Code.

The theory of error correction is vital to practical computing schemes. All physical computers are subject to noise, and this noise can cause arbitrary errors that must be detected and corrected to ensure accurate results. In classical computers an error can flip a bit '0' to a bit '1' or vice versa, which can be corrected by a variety of schemes. In quantum computing, the task of correcting errors is dramatically more challenging. Errors on quantum computers are continuous in the general case; a qubit state $|\psi\rangle$ can in theory be transformed to any new state $|\psi'\rangle$ by noise. Noise can also entangle multiple qubits. Furthermore, measuring the state of a system to directly check for errors will cause the wave function to collapse, thus losing information if not done carefully. Quantum error correction (QEC) is designed to overcome these challenges.

One might wonder if we can manage without QEC by improving the accuracy of devices further. However, as we will see later, useful quantum circuits may contain over 10^{10} logical gates; the error rate of each of these gates would need to be unrealistically small with iterations of current technology to perform the full circuit without error, and proceeding without QEC is not an option for large-scale quantum computing applications.

Ultimately, QEC schemes exist that can protect against arbitrary errors, provided sufficient resources are available. In practice, this is done by encoding many *physical* qubits into a *logical* qubit.[16] The quantum threshold theorem then states that if the error rate on the physical qubits is below a certain threshold, the error rate on the logical qubits can be made arbitrarily small.[84−87] In general, the more physical qubits available, the larger the logical qubit and the better the protection that can be achieved. As such, resource estimation for future QPE calculations must carefully include the effect of QEC.

There is a wide family of techniques for QEC. Here we shall consider the surface code, which represents each logical qubit by a $d \times d$ grid of physical qubits.[88] Protecting all d^2 of these qubits is not possible. Instead, we seek to define just a single qubit as a protected subspace. This subspace is known as the *codespace*. The state of the logical qubit is forced to reside in the codespace by measuring operators known as *stabilizers*. The measurement of these stabilizers allows one to check and correct errors, without destroying information encoded in the logical qubit. A further d^2 *syndrome* qubits are present to allow efficient measurement of the stabilizers. This leads to a total of $2d^2$ physical qubits per logical qubit.

The surface code has a number of useful properties for an error correcting code: first, the physical qubits are arranged on a 2D grid and only require nearest-neighbor connectivity; and second, the surface code can tolerate a relatively high probability of errors occurring on the physical qubits. Specifically, for a probability p of an error occurring on each physical qubit per operation, the probability of an error on a logical qubit is approximately $0.1(100p)^{(d+1)/2}$,[89] for each given logical operation. Note that this suggests an error threshold of 1%, below which the error rate of the logical qubit is decreased with increasing d.

4.2. Magic-State Factories and the QPU Architecture.

One challenge in QEC schemes is the Eastin−Knill theorem, which says that no QEC code can trivially implement a universal gate set.[90] For example, the surface code can only encode Clifford gates, a collection of quantum gates which implement elements of the Clifford group. The Clifford group can be defined as the set of operations that map Paulis to other Paulis and can be generated by the H, S, and CNOT gates, as defined in eqs 4 and 5. In order to achieve universal quantum computation, we need an extra non-Clifford operation. In the surface code this is often taken to be the T gate, also defined in eq 4. A T gate can be performed outside of the surface code by generating and consuming a specific quantum state, known as a *magic state*. Circuits to create high-fidelity magic states are known as magic-state factories,[89,91] and the process of creating these states is known as *magic-state distillation*. This process works by taking some number of noisy magic states as input and producing a smaller number of magic states, which are of higher quality, as output. The number of input and output states, the probability of success, and the time taken for distillation all vary depending on the choice of factory. For example, the 15-to-1 factory from ref 89 uses 11 logical qubits, takes 15 magic states as input, and after 11 time steps produces a single magic state. Here, a time step corresponds to a single error-corrected logical operation. If the input magic states have probability p of error, then the probability of the distilled magic state failing is $35p^3$. In comparison, the 20-to-4 factory from ref 89 uses 14 logical qubits, runs in 17 time steps, and has probability $22p^2$ of any one of the four output magic states not being a magic state. Figure 3a shows these two factories for comparison. There are also larger factories, such as the 116-to-12 factory in refs 89 and 92, which uses 81 tiles and produces 12 magic states after 50 time steps, such that the probability of a failed state is $41.25p^4$.

These factories can be concatenated to create even higher-quality magic states, such that the magic states produced from a lower-level factory are used as input for a higher-level factory. In Figure 3b we show how the magic states produced by 11 15-to-1 factories can be used as input for another 15-to-1 factory. This produces a 225-to-1 factory, which uses significantly more magic states and logical qubits, but in 15 time steps produces a magic state with failure probability $35(35p^3)^3 = 1500625p^9$. It is through these techniques that we can design factories which produce magic states with an arbitrarily small probability of failure.

Once a magic state is created, we need to ensure that it can be routed to the logical qubits which are involved in the logical quantum circuit, which we shall refer to as *data* qubits. Litinski describes a layout called the fast block, where data qubits are arranged in a 2D grid with additional *auxiliary* qubits between each row of data qubits.[89] This arrangement allows for one magic state to be consumed by any data qubits within one time step. A time step is d code cycles, where a code cycle is the time required to measure all stabilizers. Magic-state factories can then be arranged around this data block to form an architecture for our quantum computer. Note that we want to design this architecture in such a way that magic states produced from each factory can reach the data block and, at the same time, try to minimize the number of additional unused logical qubits—that is, logical qubits which are not being used for data, generating magic states or routing, yet exist on the 2D grid.

4.3. Error-Corrected Resource Estimation.

We are now ready to explain how to calculate the overhead of QEC, using techniques described by Litinski.[89] As explained in that reference, the execution of quantum circuits can be reduced to just the application of T-like gates and measurements, by commuting the Clifford gates past all T gates and the measurements at the end of the circuit. Thus, we focus first on magic-state distillation. The number of magic states to be distilled is equal to the number of T gates to be performed, denoted N_{T}. This number depends on the details of the algorithm used and will be discussed in detail in Section 5. If we wish to achieve a total failure probability of P_{df} for distillation, then the failure rate for each individual distillation should be no greater than $P_{\mathrm{df}}/N_{\mathrm{T}}$. We therefore choose a magic-state factory whose failure rate satisfies this requirement. Several possible factories have been described, such as by Litinski[89,91] and by Haah and Hastings.[93] The failure rate of a particular factory is denoted by q. Then we choose the factory with the largest q that satisfies $q \leq P_{\mathrm{df}}/N_{\mathrm{T}}$.

We next decide the size of the fast block needed. As described above, this is the region of the quantum computer where algorithmic operations are performed on the data qubits. The data qubits are interspersed with auxiliary qubits. For n data qubits, the fast block uses approximately $2n + \sqrt{8n} + 1$ logical qubits in total. If $\sqrt{8n}$ is an integer, then this number of qubits is exact and the fast block is exactly square. If $\sqrt{8n}$ is not an integer, then additional qubits are added or removed to the final column, as needed.

We now consider how many magic-state factories are needed. The fast block can consume one magic state per time step. We therefore choose the number of magic-state factories such that one magic state can be produced per time step on average. For example, the 15-to-1 factory produces a single magic state every 11 time steps, so we would include 11 such factories in our setup; the 116-to-12 factory produces 12 factories every 50 timesteps and would require five factories; and the 225-to-1 factory produces a single magic state every 15 time steps, so we would require 15 factories.

Next we discuss how to arrange the magic-state factories around the data block. Our aim is to arrange all factories around the data block such that each factory is connected to the data block and that the number of unused logical qubits is minimized. Problems of this nature are commonly referred to as 2D packing problems, many variants of which are NP-Hard to solve,[94] and therefore it is unlikely that an optimal solution can be found efficiently. Instead, we use a greedy algorithm, which uses a heuristic to place each individual factory in a reasonable spot based on the arrangement of the ones before it. Thus, for each factory we look at every position we could place the factory and check which ones have a path connecting the factory to the data block. We then choose the best placement for this factory on the basis of which position minimizes our heuristic. Once all factories have been placed, the algorithm is complete. A pseudocode description of the algorithm is shown in Algorithm 1.

Algorithm 1 A greedy algorithm for allocating magic state factories around a data block.

```
Input  $DataBlockShape$: Shape of the data block
Input  $FactoryShape$ Shape of the magic state factories
Input  $NumFactoriesTotal$: Number of magic state factories to place
Output  Arrangement of magic state factories around data block

  $NumFactoriesPlaced \leftarrow 0$
  $Arrangement \leftarrow DataBlock$
  while $NumFactoriesPlaced < NumFactoriesTotal$ do
      $BestPlacementFound \leftarrow \emptyset$
      $BestPlacementCost \leftarrow \infty$
      for all $Placement \in \mathrm{NewPlacements}(Arrangement, FactoryShape)$ do
          $Path \leftarrow \mathrm{FindShortestPath}(Placement, DataBlock)$
          if $Path \neq \emptyset$ then
              $PlacementCost \leftarrow \mathrm{CalculatePerimeter}(Placement)$
              if $PlacementCost < BestPlacementCost$ then
                  $BestPlacementFound \leftarrow Placement$
                  $BestPlacementCost \leftarrow PlacementCost$
              end if
          end if
      end for
      $Arrangement \leftarrow Arrangement \cup BestPlacementFound$
      $NumFactoriesPlaced \leftarrow NumFactoriesPlaced + 1$
  end while
  return $Arrangement$
```

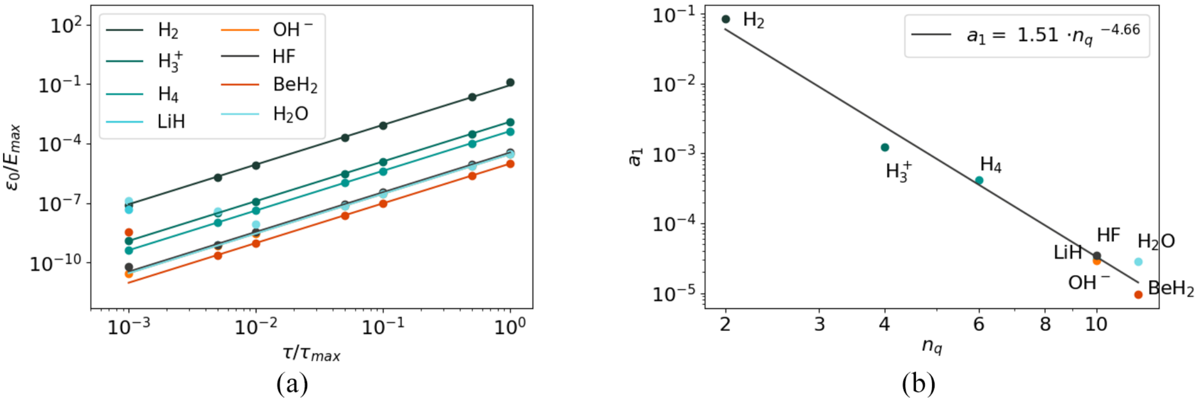


Figure 4. Fit of empirical law for our set of molecules. The fit is done in two steps. In the first step (left), for each of the molecules, we generate δE_0 for $\tau/\tau_{\max} = [1.0, 0.5, 0.1, 0.05, 0.01, 0.005, 0.001]$ and do a one-parameter fit of $\epsilon_0 = E_{\max}a_1\left(\frac{\tau}{\tau_{\max}}\right)^2$. Note that, for larger molecules, ϵ_0 for small values of τ appear to deviate from the quadratic behavior. We attribute this to numerical error and exclude these values from the fits. In the second step (right), we plot a_1 for each molecule and fit $a_1 = a(n_q)^b$, obtaining the parameters of the empirical law in eq 38: $a = 1.51 \pm 0.84$; $b = -4.66 \pm 0.27$.

A key question is the choice of heuristic we optimize each placement over. One option is to minimize the number of additional logical qubits. However, there are many placements which might lead to the same number of additional logical qubits, by placing the factory around the edge of the current arrangement. Furthermore, this can lead to awkward arrangements around a data block, with a lot of wasted qubits which would be hard to use with other computations happening in parallel. Instead, we aim to minimize the perimeter of the arrangement. This heuristic ensures the arrangement remains relatively well packed by minimizing gaps between factories. We also use a second heuristic to ensure that the arrangements form a roughly rectangular shape so that other computations can be more easily arranged around it. Example layouts created by this scheme are presented in Section 7, in Figure 8.

Finally, we choose the surface code distance, d. As noted above, in the surface code the error rate per logical qubit per code cycle is approximately $0.1(100p)^{(d+1)/2}$. There are d code cycles per time step, and the fast block consumes an average of one magic state per time step. The total number of magic states to consume is N_T, and the total number of logical qubits is N_L. Thus, for an overall target failure rate of P_{target}, we require that

$$N_T \times N_L \times d \times 0.1(100p)^{(d+1)/2} \leq P_{\text{target}} \tag{34}$$

Solving this equation for d gives us the required surface code distance. This allows the total number of physical qubits to be calculated, as each logical qubit consists of $2d^2$ physical qubits.

Since, following Litinski,[89] we have reduced the quantum circuit to just the application of T gates, the total runtime can be estimated as $N_T \times d \times t$, where t is the time to perform one code cycle, and d code cycles are performed per time step.

For resource estimates in this work, we set the distillation failure probability as $P_{\text{df}} = 1 \times 10^{-3}$ and the surface code failure probability as $P_{\text{target}} = 9 \times 10^{-3}$.

5. TROTTERIZATION VS QUBITIZATION

5.1. Trotterization. As explained in Section 2.4.2, the QPE algorithm estimates an eigenvalue of a unitary operator U. A natural choice is to take U to be the evolution operator for some time t

$$U = U(t) = \mathrm{e}^{-i\hat{H}t} \tag{35}$$

Given a Hamiltonian $\hat{H}$, producing its corresponding evolution operator U is generally a difficult task. One can, at best, aim for a good approximation to U. When using U to estimate an energy to a desired level of accuracy—say, chemical accuracy—it is paramount to control the error due to this approximation. This is usually referred as the problem of "Hamiltonian simulation".[95]

The "Trotter approximation" is a widespread strategy for approximating U, given a Hamiltonian written as a sum of terms

$$\hat{H} = \sum_{j=1}^{L} \hat{H}_j \tag{36}$$

each of which is easy to exponentiate—that is, we can construct $e^{-i\hat{H}_j t}$ for all j. Examples of such Hamiltonians include those found in chemistry.

The Trotter approximation divides the time t into intervals of duration τ and considers a simple approximation to the evolution operator for each of these intervals:

$$U(t) \approx U_{\text{Trot}}(t) = \left(\prod_{j=1}^{L} \mathrm{e}^{-i\tau/2\hat{H}_j} \prod_{j=L}^{1} \mathrm{e}^{-i\tau/2\hat{H}_j}\right)^{t/\tau} \tag{37}$$

with t an integer multiple of τ. [The expression presented here is the so-called second order Trotter approximation, in which each operator $\hat{H}_j$ appears twice per time step. Other Trotter orders exist—they differ in the number of times $\hat{H}_j$ appear per time step.] The error in this approximation goes to zero for τ → 0. However, the cost of implementing this approximation increases as τ decreases. To do well in the trade-off of resources vs accuracy, one would choose the largest τ that gives the desired accuracy.

Given a Hamiltonian, rigorous error bounds for the Trotterization of its evolution operator are available for finite τ,[96] but in practice these bounds tend to be very generous. For tighter error estimates, one can use an empirical law for the error inferred from small systems—small systems are a numerical necessity when the error is estimated via exact diagonalization of the Trotter and Hamiltonian operators (see, e.g., refs 95 and 97 for other empirical approaches to the Trotter error). There are a number of choices to make. For starters, there are a variety of notions of error to quantify. We choose ϵ_0, the difference between the ground-state energies of the original Hamiltonian $\hat{H}$ and its Trotterized evolution operator $U_{\text{Trot}}(\tau)$ (it is apparent from eq 37 that the energy spectrum of $U_{\text{Trot}}(t)$, defined via its eigenvalues $\{e^{-iE_i t}\}$, is a function of τ).

We inferred an empirical law from the difference between ground-state energies of $\hat{H}$ (E_0) and $U_{\text{Trot}}(\tau)$ (E_{Trot}) for a data set composed of small molecules (H_2, H_3^+, H_4, LiH, OH^-, HF, BeH_2, and H_2O) in the STO-3G basis, in the symmetry-conserving Bravyi−Kitaev encoding—$\hat{H}_j$ in eq 37 being the Pauli strings of the Hamiltonian in that encoding. For each molecule, this difference $\epsilon_0 = E_{\text{Trot}} - E_0$ is well modeled by a quadratic monomial of τ. The coefficient of this monomial depends on the size of the molecule, which we characterize by the number of logical qubits needed to represent it, n_q. In symmetry-conserving Bravyi−Kitaev, this is two less than the number of spin−orbitals, $n_{\text{q}} = n_{\text{o}} - 2$. The following law results in a good fit:

$$\epsilon_0 = E_{\text{max}} a(n_{\text{q}})^b \left(\frac{\tau}{\tau_{\text{max}}}\right)^2 \tag{38}$$

where E_{max} is a certain bound on the maximum eigenenergy of the Hamiltonian that has been defined in eq 12; $\tau_{\text{max}} \equiv \pi/E_{\text{max}}$; and n_{q} is the number of qubits used to represent the active space of $\hat{H}$. Figure 4 describes the fit, resulting in $a = 1.51 \pm 0.84$ and $b = -4.66 \pm 0.27$.

One finds that, even with empirical laws for the error committed, the Trotter approximation is an expensive method for the Hamiltonian simulation step of QPE. For example, as quoted in Section 7 below, for simulations of active spaces of (32e, 32o), the compilation of the Trotter operator into Clifford and T gates[98] gives T-gate counts of around 4×10^{14}, for chemically accurate Trotterization. These T-gate counts result in impractically long runtimes on current and projected quantum computers. We note that while improvements to the Trotterization circuits can be made,[99] these will not reduce the runtimes by the required orders of magnitude. More modern methods, encompassed under the names of "qubitization" and "linear combination of unitaries" result in more moderate T-gate counts, of around 10^{10}, with projected runtimes on the order of a few days.

5.2. Qubitization.

In the quest to find the eigenenergies of a Hamiltonian, it is actually not necessary to solve the problem of Hamiltonian simulation and implement the time evolution operator $U = e^{-i\hat{H}t}$ with eigenvalues $e^{-iE_j t}$. Instead, qubitization methods[100,101] facilitate the implementation of a so-called walk operator W with eigenvalues

$$\text{spectrum}(W) = \{e^{\pm i \arccos(E_j/E_{\text{max}})}\} \tag{39}$$

The Hamiltonian's energies E_j can then readily be retrieved by performing QPE on the walk operator W. The upside is that W can be implemented with many fewer T gates than the Trotterized time evolution U_{Trot}, for a given Hamiltonian $\hat{H}$ and target accuracy.

The walk operator W is simply related to the Hamiltonian $\hat{H}$. The starting point for the implementation of the walk operator is a decomposition of the Hamiltonian into a linear combination of unitaries (LCU):

$$\hat{H} = \sum_{i=1}^{L} w_i U_i \tag{40}$$

The individual U_i should be unitary and simple enough to be readily implementable on a quantum computer. Then the LCU decomposition can be implemented in a block-encoded fashion by using the PREPARE/SELECT framework.[100,101] In its most basic and simplified form, the LCU implementation is based on a state

$$|\text{PREPARE}\rangle = \frac{1}{\sqrt{E_{\text{max}}}} \sum_{i=1}^{L} \sqrt{|w_i|}\,|i\rangle, \qquad E_{\text{max}} = \sum_{i=1}^{L} |w_i| \tag{41}$$

and an operator

$$\text{SELECT} = \sum_{i=1}^{L} |i\rangle\langle i| \otimes U_i \tag{42}$$

which selects one of the U_i on the basis of the value of the auxiliary qubit register $|i\rangle$. Put together, these give

$$\hat{H} \propto \langle \text{PREPARE}|\,\text{SELECT}\,|\text{PREPARE}\rangle \tag{43}$$

Any signs of w_i have been absorbed into the U_i. Qubitization then shows how to construct the walk operator W with the eigenvalues in eq 39 from these PREPARE and SELECT operators.

There are several flavors of qubitization.[77,101−103] On the one hand, by transforming or factorizing the chemical Hamiltonian in different ways, they arrive at different LCU decompositions in eq 40 that promise better efficiency. On the other hand, the flavors introduce new ways to implement the PREPARE and SELECT operators, which improve upon previous approaches but can also be very tailored to their specific factorization of the Hamiltonian.

Throughout, we consider the sparse method first presented in[102] and further improved in Appendix A of ref 77. The method is tailored to the Jordan−Wigner Fermion-to-qubit encoding. We select this qubitization flavor on the basis of its simplicity and flexible applicability to a wide range of systems. We consider a total error $\epsilon = 1.6$ mHa of chemical accuracy.[77] The total error is made up of three parts $\epsilon = \epsilon_{\text{TRUNC}} + \epsilon_{\text{PREP}} + \epsilon_{\text{QPE}}$,[77] which we chose as $\epsilon_{\text{TRUNC}} = \epsilon_{\text{PREP}} = 0.3$ mHa and $\epsilon_{\text{QPE}} = 1$ mHa. In the following we explain how they arise, and which parameters of the algorithm can be adjusted to reach our total error budget of 1.6 mHa.

First of all, the chemical Hamiltonian $\hat{H}$ is not decomposed exactly into an LCU in eq 40. The sparse method exploits (approximate) sparsity in the Hamiltonian $\hat{H}$ by truncating small terms. We denote the truncated Hamiltonian $\hat{H}_{\text{TRUNC}}$ and perform the LCU on $\hat{H}_{\text{TRUNC}}$ instead of $\hat{H}$. This reduces the number of terms in the LCU decomposition, yielding a faster quantum computation. We consider two criteria to truncate the Hamiltonian according to a given error budget. First, a truncation based on the L_2-norm of $\hat{H}$. In this, the truncation threshold for the two-body coefficients is chosen such that $\|\hat{H} - \hat{H}_{\text{TRUNC}}\|_{L_2} \leq \epsilon_{\text{TRUNC}}$.[103] Note that the L_2-norm must be taken with respect to the LCU coefficients. Then, we consider a truncation based on CCSD(T).[77] Specifically, we calculate the CCSD(T) energy with the original Hamiltonian ($E_{\text{CCSD(T)}}$) and a truncated Hamiltonian ($E_{\text{CCSD(T) trunc}}$) and find the largest truncation for which $|E_{\text{CCSD(T)}} - E_{\text{CCSD(T) trunc}}| \leq \epsilon_{\text{TRUNC}}$. This reduces the number of terms in the Hamiltonian by up to ~90%, which lowers the cost of implementation significantly.

Second, an error ϵ_{PREP} occurs when implementing the LCU decomposition with the PREPARE/SELECT machinery. The quantum circuit for PREPARE cannot implement the coefficients $\sqrt{|w_i|/E_{\max}}$ in eq 41 to infinite precision, resulting in the rounding error ϵ_{PREP}. It can be controlled by the bit length $\aleph = \lceil \log(E_{\max}/(2\epsilon_{\mathrm{PREP}})) \rceil$ (eq A12 of ref 77) for the coherent alias sampling procedure.[101] [Apart from the bit length $\aleph$ in coherent alias sampling, the bit length b_r used in amplitude amplification of an equal superposition state in PREPARE also contributes to ϵ_{PREP}. However, the contribution of b_r to the total cost is subleading and we take b_r as constant.[77]] The LCU can be implemented much more precisely than the evolution operator in Trotterization; the dependence of gate count on allowable error ϵ_{PREP} is much smaller in qubitization. The LCU is a more efficient approximation than Trotterization. The main reason for this is that while Trotterization targets the approximate implementation of $U_{\mathrm{Trot}} \approx e^{-i\hat{H}t}$, the LCU directly targets the approximate implementation of $\hat{H}$, avoiding any approximation in the expansion of the exponential.

Finally, as in Trotterization, an error ϵ_{QPE} occurs due to the final QPE, which has a finite accuracy as discussed in Section 2.4.2. Qubitization methods typically use alternative phase estimation methods[101,104] to slightly improve on standard textbook QPE. An error ϵ_{QPE} implies that the walk operator W, and hence the qubitization procedure, needs to be repeated $\lceil \pi E_{\max}/(2\epsilon_{\mathrm{QPE}}) \rceil$ (eq 44 of ref 77) times in a phase estimation.

Subsequently, we will assess the time and logical qubit number needed for a quantum computer to be of aid in chemical applications. The article[77] has taken great effort in deriving the number of logical T gates and logical qubits needed for a given Hamiltonian and parameters determining the errors. The number of logical T gates in the sparse qubitization algorithm can be found by multiplying the cost (eq A17 of ref 77) of a single iteration with the number of iterations (eq 44 of ref 77) in the phase estimation. The number of logical qubits is given by eq A18 of ref 77. We combine these results with error correction (see Section 4.3) for an estimate of the physical resources required. These results for the sparse qubitization algorithm can be directly compared to the runtime requirements of the Trotterization algorithm. Figure 7 highlights the tremendous runtime advantage of qubitization algorithms compared to Trotterization.

6. DRUG DESIGN METHODS AND THE MODEL SYSTEM

6.1. Computational Chemistry in Pharma.

The interaction between drug molecules and various proteins is vital for the understanding of pharmaceutically relevant mechanisms. Unfortunately, a protein–drug system within its physiological environment may easily consist of hundreds of thousands of atoms, which makes the full quantum mechanical treatment of such systems out of reach for quantum and classical computers alike. As a consequence, the most widely used computational techniques in pharma rely on a classical (Newtonian) parametrization of the various interactions via empirical force fields. Current methods of rational drug design broadly belong to either ligand-based or structure-based design approaches. While the former focuses on structural features of ligands, the latter considers drug molecules within a protein environment. Especially for the latter, an accurate description of the forces involved in protein–ligand binding is vital and the necessary force-field parameters may be obtained from quantum mechanical calculations.[105−108] However, while classical force fields capture most prominently the bond lengths, bond angles, and dihedrals, as well as nonbonded electrostatic and van der Waals interactions, their traditional formulation does not account for finer electronic effects such as polarization, charge-transfer phenomena, aromatic stacking interactions, or interactions with metal ions, although extensions do exist that attempt to treat the latter phenomena with varying degrees of success.[109−111] The fact that only atom types and not electrons and nuclei are considered in force-field parametrization also renders force-field approaches incapable of describing covalent interactions and reaction mechanisms that involve bond breaking, finding transition-state structures, and making spectroscopic predictions. Yet despite the continued improvements in computer power and speed, the routine application of steeply scaling quantum mechanical methods in the drug design process is still very limited and mainly reserved for the study of small-molecule properties and small-molecule conformations. While using semiempirical methods such as DFTB (tight-binding density functional theory)[112] and HF-3c[113] reduces the cost significantly, these methods are often considerably less accurate than fully quantum mechanical methods. Thus, when more accurate treatment is required, embedding techniques are typically used. These methods either partition the molecule into small fragments and assemble the whole from fragment calculations, or build layers with one of them treated at a high level and the others more approximately. The great variety of these methods is reviewed elsewhere;[114,115] we only remark here that some of them have been applied to protein systems containing more than 20000 atoms.[116] Here we are concerned with two typical choices: hybrid QM/MM and QM cluster.[117,118]

Since the ground-breaking work by Warshel and Levitt in 1976,[119] the idea of partitioning a chemical system into layers treated with methods of different sophistication has been a field of intense research.[120,121] In drug design the approach is traditionally used to describe selected residues of the binding pocket and the drug with a quantum mechanical (QM) method while the remainder of the system is simulated using molecular mechanics (MM). These hybrid QM/MM methods are generally divided into subtractive methods where the MM energy of the active site is subtracted from a sum of the QM energy of the active site and the MM energy of the entire system, and additive methods that only consider the MM energy of the environment and account for the interaction between the two systems by adding an electrostatic coupling term.[122] The latter describes interactions either (a) solely on the MM force-field level and without any polarization of the QM region (mechanical embedding), (b) by incorporating point charges from the MM region in the QM Hamiltonian (electrostatic embedding), or (c) by mutual polarization of the regions requiring a polarizable MM force field (polarizable embedding).[117]

In a QM-cluster approach the active site is physically cut out of its environment, only considering the drug and the nearest interacting amino acids. Cross-sections are saturated by usually hydrogen atoms or methyl groups and constraints are added to ensure the rigidity imposed by the protein surroundings. Electrostatic effects are compensated for by using continuum solvation and a dielectric constant.[123,124] Both the hybrid QM/

MM and the QM-cluster method are used to gain insight into the drug−protein binding and electronic processes in the binding-pocket-like electronic excitations[125] or mechanisms of binding or action.[126] However, both methods are restricted to a few hundred atoms at most, depending on the level of description, which is not enough to describe, e.g., effects of ligand binding at other sites than the binding pocket (allosteric) or other large-scale mechanisms.

Finally, it should be mentioned that a number of embedding methods have already been proposed for use with quantum computers in an attempt to reduce the heavy resource requirements. Local approaches to active space construction have been recently proposed[127] and applied to quantum computing.[128] The quantum variants of dynamical mean field theory[129] and density matrix embedding theory (DMET)[130] were published some years ago, and costing studies are also available for DMET.[131] Energy-weighted DMET[132] and Gutzwiller variational embedding[133] approaches have been tested on current quantum processors. Pharmaceutical model systems have also been studied, including a study of protein−ligand interactions using DMET[134] and our own work on a multilayer embedding approach.[135]

6.2. Active Space Selection. In both the QM/MM and QM-cluster approaches, a central QM region is selected to be treated at the highest level of theory. Unfortunately, this region is typically still too large to be treated directly on a quantum device. To construct the molecular Hamiltonian within this region, an active space of orbitals must be selected in a manner reminiscent of frozen core[136,137] and complete active space[40] approaches. In our previous work, we suggested a way for how the active orbitals may be selected using local fragment occupied orbitals and a corresponding set of natural orbitals obtained from perturbation theory.[135] We also outlined a secondary subtractive embedding process to take care of correlation effects outside the active space. For the purposes of resource estimation, this second step is not necessary.

6.3. Model System. As a model system for the subsequent resource estimation within a QM-cluster approach, we have chosen the drug Ibrutinib which was approved for treatment of non-Hodgkin lymphoma by the U.S. Food and Drug Administration (FDA) in 2015.[138] It inhibits Bruton's tyrosine kinase (BTK)—a vital element of the B-cell receptor signaling pathway—and thus induces apoptosis in B-cell tumors.[139] It covalently binds to cysteine 481 in BTK via a Michael addition reaction.[140] Successful binding of a drug to a target depends on many factors in both the binding pocket and its environment. In order to design drugs efficiently, we need to gain a thorough understanding of the binding process. In the first step a covalent drug binds in the same manner as a noncovalent drug, namely, via weak interactions. If a reactive electrophilic group on the drug is then positioned in proximity and favorable arrangement to a nucleophilic group on the protein, the covalent bond can be formed via an electronic rearrangement. The latter cannot be described by most commonly used computational drug design methods.[141]

The size of the cluster has do be decided depending on the residues, ligand groups, and water molecules or ions contributing to the binding or mechanism. To ensure the correct atomic arrangements and to represent the rigidity of the binding pocket, it is crucial that the underlying crystal structure is well resolved and that the bonds cut and the atoms fixed are carefully chosen.[142] A cluster containing the ligand, all neighboring residues (within 5 Å of ligand), and water molecules would account for over 400 atoms and thus be too big for our purposes. Instead, a medium sized cluster was selected in which the ligand was cut beyond the pyrazolo pyrimidine moiety and which also included residues Leu408, Gly409, Thr410, Gly480, Cys481, and Asn484 and four water molecules. The cluster contained 129 atoms and was fixed at position 3 of the piperidine ring (Figure 5). There have been numerous studies in the past that have used sizes similar to the cluster size considered here, although the treatment of a larger system would have been computationally feasible.[143−145] The cluster represents the product structure of the binding mechanism of the formation of the covalent bond between the ligand and Cys481 of the protein.[140] In our approach, the selection of occupied orbitals corresponds to selecting fragments. We selected active fragments in five different sizes as shown in Figure 5. The smallest chosen area comprises the ligand warhead up to the carbonyl group, sulfur of cysteine and an OH^- unit from the interacting water molecule (purple) and represents the minimum of the reacting and primarily interacting atoms for the boond formation step. With incremental addition of more interacting atoms from the binding pocket and larger parts of the ligand and the respective amino acid, the largest chosen area captures all relevant interactions and includes the ligand's warhead moiety up to the two closest CH_2 and the coordinating pyrazolo pyridine nitrogen atom, four water molecules, Gly480 and Cys481 excluding the saturation groups, and the functional tail and carbonyl oxygen of Asn484 (green).

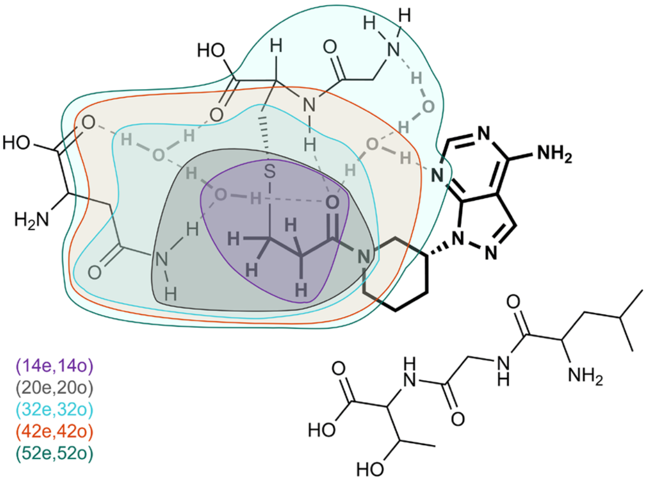


Figure 5. Cluster containing part of the binding pocket and the Ibrutinib inhibitor. The various fragments in which the active space orbitals were selected are indicated using various colors.

6.4. Computational Methods. Input structures for all calculations were based on the crystal structure of a covalently bound Ibrutinib/BTK complex by Bender et al. (PDB ID: 5P9J, 1.08 Å resolution).[146] With use of Maestro's[147] Protein Preparation Wizard,[148] missing residues were filled in using Prime[149,150] and hydrogen atoms were added and refined with PROPKA.[151,152] The Cα atoms were constrained for every terminal amino acid included in the cluster and one atom in the ligand to account for their positions in the X-ray structure. Geometry optimizations were carried out with Jaguar 11.2[153] using the DFT functional B3LYP,[154,155] Grimme's dispersion correction D3,[156,157] and the 6-31G+** basis set. The CPCM (conductor-like polarizable continuum) solvation model[158]

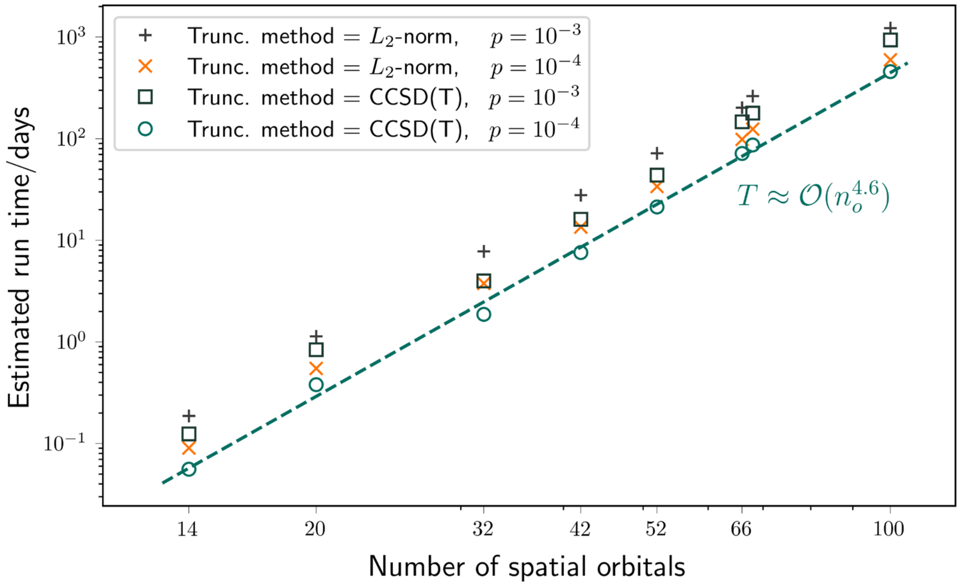


Figure 6. Runtime to perform QPE using sparse qubitization. Active spaces from (14e,14o) to (100e,100o) are considered. It is assumed that one time step takes 1 μs to perform. Physical error rates, p, of 0.01 and 0.1% are considered. The Hamiltonian is either truncated using an L_2-norm criterion or a CCSD(T) criterion. In each case, the runtime scales as approximately $n_o^{4.6}$ with the number of active orbitals.

with the dielectric constant $\varepsilon = 4$ was used to describe the effect of the global protein environment.[159,160] A frequency calculation was carried out to confirm that the only imaginary modes present are small and result from the atomic constraints, confirming the structure to be a minimum. The active space integrals were calculated using the ORCA program package.[161] The def2-TZVP basis set was used.[162] The occupied orbitals were localized using the Pipek−Mezey approach[163] and were mapped to fragments using intrinsic atomic orbitals[164] and the criterion that the orbital charge on the fragment be larger than 0.95.[135] The same number of active virtuals were selected using perturbation theory as the number of active occupied orbitals obtained in the previous step.[135]

7. RESULTS

We now present results of our resource estimations. We consider the molecule and active spaces discussed in Section 6, which have sizes from (14e,14o) to (100e,100o).

We perform resource estimation for the two QPE approaches described in Section 5. In the first approach we consider the textbook QPE algorithm using Trotterization to a precision of 1.6 mHa. We estimate the error from Trotterization using the empirical law described in eq 38. In the second approach we consider the Heisenberg-limited QPE algorithm described by Lee et al.[77,101] using qubitization, specifically the sparse qubitization method as described in Section 5. The overall precision is again taken to be 1.6 mHa. We refer to these two approaches as “QPE with Trotterization” and “QPE with sparse qubitization” in the following, although it is important to emphasize that improvements in the latter are not solely due to the use of qubitization.

Physical error rates of 0.01 and 0.1% ($p = 10^{-4}$ and $p = 10^{-3}$) are considered. The code cycle duration is taken to be 1 μs, which is believed to be realistic for future superconducting quantum processors.

Figure 6 presents the runtime for QPE with sparse qubitization as a function of active space size, considering both error rates ($p = 10^{-4}$ and $p = 10^{-3}$) and both Hamiltonian truncation approaches defined in Section 5. On this log−log plot a reasonable power law fit is evident. The runtime is found to scale as roughly $T \sim O(n_o^{4.6})$, where n_o is the number of spatial orbitals. The power law plotted is fit using data with $p = 10^{-4}$ and CCSD(T) truncation only, but the same scaling is observed for each set of data. A power law with respect to the number of orbitals was already anticipated in Appendix D of ref 102, and the exponent found is in approximate agreement with our observations. For the trivial (14e,14o) we estimate a runtime of 1.3 or 3.0 h with $p = 10^{-4}$ and $p = 10^{-3}$, respectively (and using CCSD(T) to assess the truncation criterion). For (32e,32o) the corresponding runtimes are 1.9 or 4.0 days. For (100e,100o), we estimate respective runtimes of 1.3 and 2.6 years. These runtimes are high, but are likely to reduce with further algorithmic developments.

We also considered using both CCSD(T) and L_2-norm to assess the truncation criterion. In both cases, we aimed for a Hamiltonian truncation error of 0.3 mHa or less. As can be seen in Figure 6, the CCSD(T) metric allows more Hamiltonian terms to be truncated, resulting in fewer T gates overall. The number of T gates is typically within a factor of 1.2 to 2.0 between these two approaches, for the active spaces studied here. The estimated numbers of T gates in each approach are presented in Table 1. Note that the runtime to perform CCSD(T) is negligible compared to the estimated QPE runtime for the active spaces considered. This CCSD(T) cost is not included in the presented runtimes. If QPE algorithms become efficient enough that thousands of orbitals can be treated, then this situation may eventually change, in which case using L_2-norm may be preferable.

In Figure 7 we compare the runtime and total number of physical qubits between the two QPE approaches, defined above, with a physical error rate of $p = 10^{-4}$. It is seen that the Trotterized approach is dramatically more expensive than the sparse qubitization method and has significantly steeper scaling in runtime with active space size. For example, the (32e,32o) example, which takes 1.9 days in the latter method, is estimated to take roughly 250 years in the Trotterized algorithm.

Table 1. Required Number of *T* Gates to Perform QPE for Various Active Spaces[a]

	qubitization		Trotterization
no. of spatial orbitals	L_2-norm truncation	CCSD(T) truncation	no truncation
14	5.6×10^8	3.7×10^8	1.6×10^{12}
20	3.2×10^9	2.3×10^9	2.7×10^{13}
32	2.0×10^{10}	1.1×10^{10}	4.0×10^{14}
42	6.9×10^{10}	4.1×10^{10}	2.4×10^{15}
52	1.7×10^{11}	1.1×10^{11}	5.2×10^{15}
66	4.7×10^{11}	3.4×10^{11}	-
100	2.7×10^{12}	2.1×10^{12}	-

[a] No truncation of the Hamiltonian is performed for QPE with Trotterization. For QPE using qubitization the Hamiltonian is truncated using both CCSD(T) and the L_2-norm to assess the error incurred, with a target truncation error of 0.3 mHa or less. The CCSD(T) criterion truncates more terms, resulting in a lower estimate for the required number of T gates.

To calculate the number of physical qubits on the QPU, we consider the full layout of the fast block, magic-state factories, and routing qubits. Making an assumption that the overall QPU will be rectangular, we then find the smallest rectangle that encloses the fast block and all magic-state factories. The number of qubits within this rectangle defines the total qubit count in our results. It is seen that the total number of physical qubits is increased in the Trotterized algorithm. This is interesting as there is a significant data qubit overhead associated with performing the qubitization algorithm. However, the increased number of T gates in the Trotterized algorithm necessitates a higher surface code distance, such that the number of physical qubits is increased overall. Moreover, while the number of data qubits in the fast block is much lower when performing Trotterization, the QPU architecture may be dominated by several large magic-state factories. Note that the number of physical qubits is the same for both (42e,42o) and (52e,52o) active spaces in Figure 7, when using Trotterization. This is because the same factory arrangements were used for both and the required surface code distance is also found to be the same.

To present a specific example in more detail, we again consider the (32e,32o) active space using the Trotterization QPE approach. Here the number of required data qubits is 82. The required number of T gates is $N_T = 3.96 \times 10^{14}$. In order to perform magic-state distillation for all such T gates with the desired success probability (see Section 4.3), we use 225-to-1 magic-state factories, whose layout is presented in Figure 3b. This factory produces one magic state every 15 time steps. Thus, we include 15 magic-state factories in order to produce one magic state per time step. Note that there may be smaller magic-state factories that suffice and which we have not considered here. However, the 225-to-1 factory is optimal from those considered in this work. Using the approach presented in Algorithm 1, we generate a layout for the device, presented in Figure 8. The total number of logical qubits for the fast block, for magic-state factories, and for routing is 3226. We then solve eq 34 with $p = 10^{-4}$, $N_L = 3226$, and N_T as above, giving a required code distance of $d = 20$. The smallest rectangular region which contains the above layout consists of 4536 tiles in total. Lastly we note that there are $2d^2 = 800$ physical qubits per logical qubit. Thus, the total number of physical qubits is estimated as $4536 \times 800 = 3.6 \times 10^6$, as plotted in Figure 7. The total runtime is estimated as $N_T \times d \times 10^{-6}$ s $= 7.9 \times 10^9$ s.

In the QPE approach with sparse qubitization, the same (32e,32o) problem requires 2207 logical qubits, but only $N_T = 1.1 \times 10^{10}$ T gates. In this case the 116-to-12 factory suffices. The device layout is again presented in Figure 8. A calculation

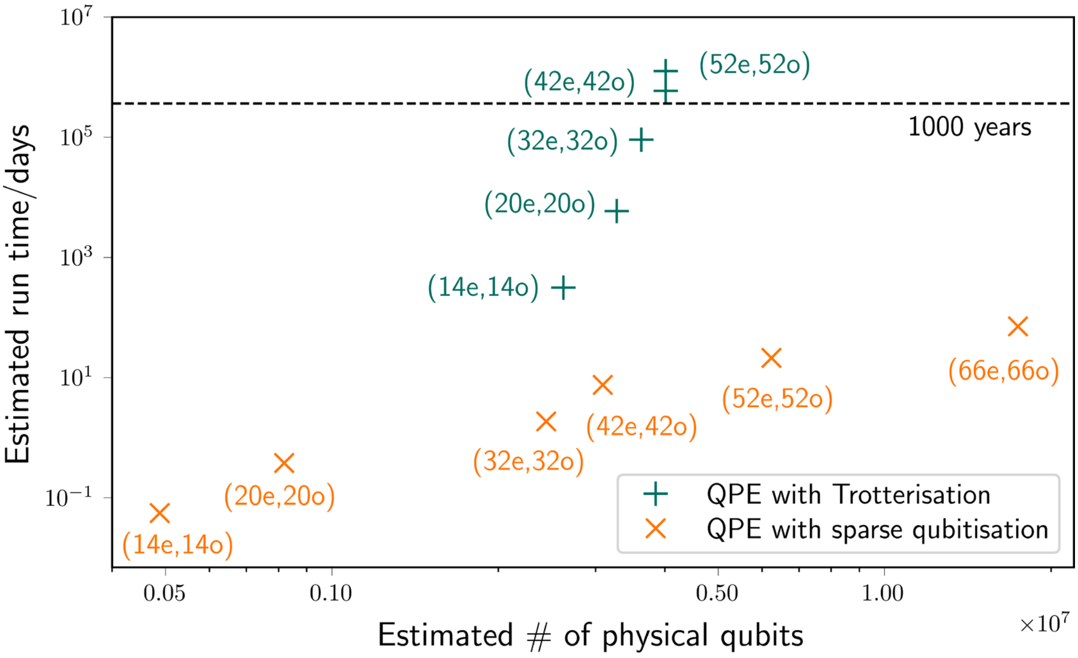


Figure 7. Comparison of resources (runtime and total number of physical qubits) using two QPE algorithms. The first (orange) used qubitization, and the Hamiltonian was truncated to remove small terms up to an error budget. The second (green) used textbook QPE with Trotterization and no truncation of the Hamiltonian. The latter algorithm has a much steeper scaling in runtime. Even for a (14e,14o) active space the runtime is multiple orders of magnitude more expensive.

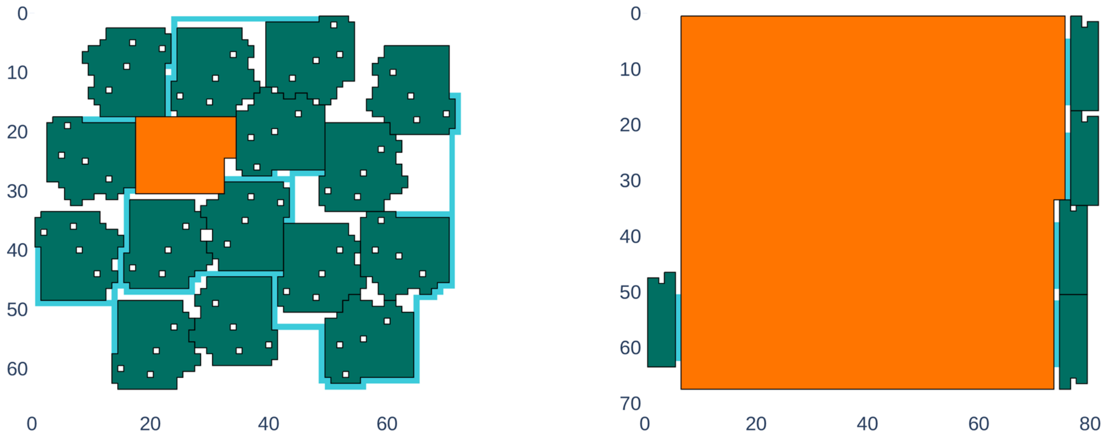


Figure 8. QPU layouts used to perform QPE experiments on the (32e,32o) active space example. Left: layout used for QPE with Trotterization. Right: Layout used for QPE with qubitization. Data block qubits are orange, magic-state factory qubits are green, and routing and storage qubits are blue. Qubitization uses many more data qubits such that the data block is much larger. However, the higher T-gate count in QPE with Trotterization necessitates larger magic-state factories (225-to-1) compared to those in qubitization (116-to-12). Axes are included to indicate the total number of logical qubits in both layouts, with each logical qubit having size 1-by-1. However, note that the code distance is higher in QPE with Trotterization (see Table 2) so that these are not to physical scale.

similar to that above leads to the lower runtime and number of physical qubits as in Figure 7. This dramatic reduction in runtime emphasizes the importance of recent developments and the potential value of similar developments in the future.

Lastly we investigate the required surface code distance in each case, as presented in Table 2. The qubitization results here used CCSD(T) as the Hamiltonian truncation criterion (using the L_2-norm criterion makes no significant difference to the required distance). The main factor affecting the required code distance is the physical error rate. For example, for the (32e,32o) active space the code distance increases from $d = 15$ to $d = 32$ as p is increased from 10^{-4} to 10^{-3}, in QPE with qubitization. For a fixed p the code distance is less sensitive to the T-gate count. For (32e,32o) the code distance increases from $d = 15$ to $d = 20$ between the two QPE approaches, although the number of T gates increases significantly from 1.1×10^{10} to 4.0×10^{14}. This emphasizes that improvements in device fidelities can significantly reduce the challenge of performing an error-corrected algorithm in practice.

Table 2. Required Surface Code Distances for Various Active Spaces[a]

	qubitization		Trotterization	
no. of spatial orbitals	$p = 10^{-4}$	$p = 10^{-3}$	$p = 10^{-4}$	$p = 10^{-3}$
14	13	29	17	36
20	14	31	19	39
32	15	32	20	41
42	16	34	21	43
52	17	35	21	43
66	18	37		
100	19	39		

[a] We consider QPE performed using Trotterization and the full Hamiltonian, and QPE using qubitization and truncating small Hamiltonian elements. Physical error rates (p) of 10^{-4} and 10^{-3} are considered.

8. CONCLUSIONS

In this work we have presented an overview of resource estimation for quantum computing calculations in pharmaceutical applications. This has focused on quantum phase estimation (QPE), which was first introduced in the 1990s, but has recently undergone a number of significant developments to reduce its practical cost. We have also performed a detailed costing of quantum error correction (QEC) in QPE applications, particularly the surface code, which will be crucial to performing quantum computation in practical problems.

We performed QPE resource estimation for several active spaces of the drug Ibrutinib. QPE was costed using two techniques: Trotterization with the full Hamiltonian, and qubitization using a truncated Hamiltonian. We find a dramatic improvement with the latter technique; calculating the ground-state energy in a (42e,42o) active space is estimated to take over 1000 years using Trotterization, which is reduced to around 7.6 days using the sparse qubitization approach (assuming a physical error rate of 0.01%, and code cycle duration of 1 μs). This emphasizes that algorithmic improvements can reduce the cost of quantum computing by several orders of magnitude and are transformative to the potential power of quantum computers. Some of the runtimes remain high; for example, obtaining the ground-state energy for a (100e,100o) active space is estimated to take over a year. This emphasizes that further algorithmic improvements are important. Given the dramatic reduction in runtime seen above, we expect such improvements to occur. For example, our costing assumed that all T gates are performed in serial, whereas the runtime can be reduced in theory through parallel execution.[89] Further truncation of the Hamiltonian may be possible through techniques such as tensor hypercontraction.[77] QEC is also an extremely active area of research, and improvements here may further reduce the resources required for large-scale applications. It should be noted that current quantum computers have a low qubit count compared to those presented in our resource estimates. For example, IBM's Eagle processor has 127 qubits.[165] An experiment by Google Quantum AI has recently been performed which demonstrates decreasing logical error rate with increasing qubit number.[166] However, the authors caution that their error rates are still close to the code threshold and must be reduced further to facilitate "practical scaling". Thus, the state-of-the-art is still some way from performing nontrivial QPE calculations.

In assessing the usefulness of quantum computers for pharma, several factors must be considered. In this work, we focused on demonstrating that QPE running on fault-tolerant quantum computers will be able to handle large active spaces. It remains important to ensure that the accurate treatment of this quantum region is coupled with a balanced treatment of the environment lest the errors coming from the latter overwhelm the potential improvements delivered by the quantum computer. Thus, using an appropriate embedding technique will be inevitable in future applications. Furthermore, because weakly correlated systems are more common in pharma, methods on a quantum computer must be compared to DFT in terms of accuracy and efficiency. If the method of choice is QPE, there is obvious advantage in obtaining the exact solution, while for ansatz-based approaches, a comparison to classical wave function based approaches might be appropriate. In terms of efficiency, quantum computers must not introduce a significant overhead compared to DFT so that the improvements in the accuracy of results will come at a reasonable cost. Despite the overall success of DFT, the constant call for better methods indicates that fault-tolerant quantum computers have a significant contribution to make in many areas of chemistry.

It remains to note that even DFT is not widely applied throughout industrial computer-aided drug design workflows. We demonstrated the applicability of quantum computing algorithms for a realistic QM-cluster approach, which similarly as the above-described QM/MM method does indeed utilize quantum mechanics to gain insight into drug−protein binding mechanisms. However, both methods are usually used either for bespoke bits at the end of the computational drug design funnel or in academic pharmaceutical research. Current high-throughput workflows are devised to allow the processing of hundreds of thousands of structures with the limited classical computing resources available, which renders even the usage of DFT with relatively cheap functionals unfeasible throughout most of the computational drug design pipeline. Rather than attempting to simply substitute existing steps in the workflows it will be a challenge for computational chemists and algorithmic researchers to rethink the computer-aided drug design processes while the hardware matures in the next years.

Yet, the thrilling perspectives for chemistry offered by quantum computing cannot be realized today. Even as different actors are racing to build and integrate larger and larger numbers of qubits,[9−11,167,168] significant practical challenges in scaling-up the size of quantum computers remain. We have based our resource estimates on tomorrow's hardware that will have overcome these challenges and today's algorithms. The high resource estimates thus show that tremendous effort must go also into improving algorithms and quantum error correction, improvements which have already allowed reduction in resources by orders of magnitude. As hardware developments and algorithmic requirements continue to draw closer to each other, it is also important to not only improve resource estimates but look at implementing these aspects of the quantum computing stack in practice. One example of this is the recent demonstrations of quantum error correction on physical hardware,[17−22] but other levels must be developed as well. To unlock the potential of quantum computing, along with the physical engineering challenge one must address challenges across the entire stack.

APPENDIX A

A.1. Number of Repetitions of QPE

As discussed in the main text, we will typically have to repeat the QPE procedure several times in order to obtain an estimate of the ground-state energy to the desired level of precision. In this Appendix, we outline one possible method for deciding how many repetitions to perform and how to obtain an energy estimate from the results.

In order that our final energy estimate is to the desired accuracy with the desired probability, we repeat the phase estimation procedure l times and take the median of the lowest k measurement outcomes. These values are determined on the basis of the details of the chemical system and calculation. Taking such a median reduces the effect of outliers.

In order to calculate l and k, we make use of P_0, the probability that an eigenvalue estimate is within 2^{-t} of the true desired eigenvalue, assuming we are measuring the desired eigenstate to t bits of precision. The derivation of this probability is given in Section A.2. We also make use of η, an estimate of the overlap probability of the initial state with the desired eigenstate. Finally, we require P_f, the probability of the error correction procedure failing on a single run of phase estimation, that is, the probability of an undetected error in magic-state distillation or a logical error (see Section 4.3 for further details).

We then assume that all measurements of an *excited* state where a logical error does not occur will result in an estimate *above* the desired range, measurements of the *ground* state where a logical error does not occur will result in an estimate *within* the desired range with probability P_0, and all other measurements result in an estimate *below* the desired range. The last choice in particular is likely to be worse than the true situation; this is deliberate so as to avoid underestimating the number of repetitions required. We then calculate the smallest value of l and a corresponding value of k such that the probability of the median estimate being within the desired range is at least the specified desired success probability, P_s. We find $k = 2\left\lfloor \frac{\eta m}{2} \right\rfloor + 1$ to perform well, though there can be improved values of k. We note that choosing k in this way means the overall success probability is not necessarily larger if the true value of η is greater than the value used to calculate k; as a result, it may be preferable to increase l and/or k to ensure the overall success probability is above the desired value for a full range of desired η values.

A.2. QPE Probabilities

We want to find the probability of a single eigenvalue estimate being within $\epsilon = 2^{-t}$ of the true desired eigenvalue. We assume that each eigenvalue, E_j, satisfies $0 \leq E_j < 1$ and so can write the jth eigenvalue as

$$E_j = \sum_{p=1}^{m} \phi_{jp} 2^{-p} \delta_j 2^{-m} \tag{44}$$

where each ϕ_{jp} has a value of 0 or 1 and $0 \leq \delta_j < 1$. We will find it useful to define $b_j \times 2^{-m} = \sum_{p=1}^{m} \phi_{jp} \times 2^{-p}$. We let p_l, where $-2^{m-1} < l \leq 2^{m-1}$, be the probability of obtaining measurements $(\theta_{l,m}, \theta_{l,m-1}, ..., \theta_{l,1})$ such that $\sum_{p=1}^{m} \theta_{lp} \times 2^{m-p} = (b_n + l)$ mod 2^m, where n is the index of the desired eigenvalue, assuming the desired eigenvalue is measured. Following standard manipulations,[61,169] we see that

$$
\begin{aligned}
p_l &= \frac{\sin^2\{\pi[2^m E_n - (b_n + l)]\}}{\sin^2\{\pi[E_n - 2^{-m}(b_n + l)]\}} && (45)\\
&= \frac{1}{2^{2m}} \frac{\sin^2[\pi(\delta_n - l)]}{\sin^2[2^{-m}\pi(\delta_n - l)]} && (46)\\
&= \frac{1}{2^{2m}} \frac{\sin^2[\pi(\delta_n)]}{\sin^2[2^{-m}\pi(\delta_n - l)]} && (47)\\
&= \frac{\sin^2[\pi(\delta_n)]}{\pi^2(\delta_n - l)^2 \operatorname{sinc}^2[2^{-m}\pi(\delta_n - l)]} && (48)
\end{aligned}
$$

(45)

This is a decreasing function of m. Taking the limit as $m \to \infty$, we see that

$$p_l \geq \frac{\sin^2(\pi\delta_n)}{\pi^2(\delta_n - l)^2} \tag{49}$$

We sum over these probabilities to find the probability, P_{m-t}, that the estimated eigenvalue is within a certain distance, 2^{-t}, of the true eigenvalue, given that we measure m bits of precision. This is given by

$$
\begin{aligned}
P_{m-t} &= \sum_{l=-2^{m-t}+1}^{2^{m-t}} p_l \geq |c_n|^2 \sum_{l=-2^{m-t}+1}^{2^{m-t}} \frac{\sin^2(\pi\delta_n)}{\pi^2(\delta_n - l)^2} && (50)\\
&= |c_n|^2 \frac{\sin^2(\pi\delta_n)}{\pi^2} \sum_{l=0}^{2^{m-t}-1} \left[\frac{1}{(\delta_n + l)^2} + \frac{1}{(\delta_n - l - 1)^2}\right] && (51)
\end{aligned}
$$

(50)

This is minimized when $\delta_n = 1/2$. Therefore,

$$P_{m-t} \geq \frac{8}{\pi^2} \sum_{l=0}^{2^{m-t}-1} \frac{1}{(2l + 1)^2} \tag{52}$$

This has a number of terms that is exponential in the value of $m - t$; however, it will not be prohibitive to evaluate the sum for small values. For larger values, we can derive a bound which does not involve a sum. From the Basel problem, we have

$$\sum_{l=0}^{\infty} \frac{1}{(2l + 1)^2} = \frac{\pi^2}{8} \tag{53}$$

and so

$$
\begin{aligned}
P_{m-t} &\geq \left(1 - \frac{8}{\pi^2} \sum_{l=2^{m-t}}^{\infty} \frac{1}{(2l + 1)^2}\right) && (54)\\
&\geq \left(1 - \frac{8}{\pi^2} \int_{l=2^{m-t}-1}^{\infty} \frac{1}{(2l + 1)^2}\, dl\right) && (55)\\
&= \left(1 - \frac{4}{\pi^2} \frac{1}{(2^{m-t+1} - 1)}\right) && (56)
\end{aligned}
$$

(54)

AUTHOR INFORMATION

Corresponding Author

Nicole Holzmann − *Riverlane, Cambridge CB2 3BZ, United Kingdom; Astex Pharmaceuticals, Cambridge CB4 0QA, United Kingdom;* orcid.org/0000-0001-5717-1984; Email: nicole.holzmann@riverlane.com

Authors

Nick S. Blunt − *Riverlane, Cambridge CB2 3BZ, United Kingdom;* orcid.org/0000-0002-2284-6969

Joan Camps − *Riverlane, Cambridge CB2 3BZ, United Kingdom*

Ophelia Crawford − *Riverlane, Cambridge CB2 3BZ, United Kingdom*

Róbert Izsák − *Riverlane, Cambridge CB2 3BZ, United Kingdom;* orcid.org/0000-0001-9236-9718

Sebastian Leontica − *Riverlane, Cambridge CB2 3BZ, United Kingdom;* orcid.org/0000-0002-4852-1919

Arjun Mirani − *Riverlane, Cambridge CB2 3BZ, United Kingdom*

Alexandra E. Moylett − *Riverlane, Cambridge CB2 3BZ, United Kingdom;* orcid.org/0000-0003-0163-5262

Sam A. Scivier − *Riverlane, Cambridge CB2 3BZ, United Kingdom;* orcid.org/0000-0003-1635-9475

Christoph Sünderhauf − *Riverlane, Cambridge CB2 3BZ, United Kingdom*

Patrick Schopf − *Astex Pharmaceuticals, Cambridge CB4 0QA, United Kingdom*

Jacob M. Taylor − *Riverlane, Cambridge CB2 3BZ, United Kingdom*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.2c00574

Author Contributions

§N.S.B., J.C., O.C., R.I., S.L., A.M., A.E.M., S.A.S, and C.S. contributed equally to this work.

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

This work was performed as part of Astex's Sustaining Innovation Postdoctoral Program. We acknowledge funding from Innovate UKs Sustainable Innovation Fund via SBRI. O.C.'s contribution to the work was in part supported by an Innovate UK grant (Quantum Enhanced Design for Materials and Chemistry, Project No. 105622). We thank David Plant and James R. Cruise for useful discussions and their valuable input to this work.

REFERENCES

(1) Yu, W.; MacKerell, A. D. Computer-Aided Drug Design Methods. In *Antibiotics*; Springer, 2017; pp 85−106. DOI: 10.1007/978-1-4939-6634-9_5.

(2) Tautermann, C. S. Current and Future Challenges in Modern Drug Discovery. In *Quantum Mechanics in Drug Discovery*; Springer, 2020; pp 1−17. DOI: 10.1007/978-1-0716-0282-9_1.

(3) Lam, Y.-h.; Abramov, Y.; Ananthula, R. S.; Elward, J. M.; Hilden, L. R.; Nilsson Lill, S. O.; Norrby, P.-O.; Ramirez, A.; Sherer, E. C.; Mustakis, J.; Tanoury, G. J. Applications of Quantum Chemistry in Pharmaceutical Process Current State and Opportunities. *Org. Process Res. Dev.* **2020**, *24*, 1496−1507.

(4) Heifetz, A., Ed. *Quantum Mechanics in Drug Discovery*; Methods in Molecular Biology, Vol. *2114*; Springer, 2020. DOI: 10.1007/978-1-0716-0282-9.

(5) Hermann, J.; Schätzle, Z.; Noé, F. Deep-neural-network solution of the electronic Schrödinger equation. *Nat. Chem.* **2020**, *12* (10), 891−897.

(6) What Happens When 'If' Turns to 'When' in Quantum Computing?. Boston Consulting Group (BCG), 2021; https://www.bcg.com/publications/2021/building-quantum-advantage (accessed 2022-03-02).

(7) Pharma's digital Rx: Quantum computing in drug research and development. McKinsey, 2021; https://www.mckinsey.com/industries/life-sciences/our-insights/pharmas-digital-rx-quantum-computing-in-drug-research-and-development (accessed 2022-03-02).

(8) Quantinuum Announces Quantum Volume 4096 Achievement. Quantinuum, 2022; https://www.quantinuum.com/pressrelease/quantinuum-announces-quantum-volume-4096-achievement (accessed 2022-03-02).

(9) Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J. C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F. G. S. L.; Buell, D. A.; Burkett, B.; Chen, Y.; Chen, Z.; Chiaro, B.; Collins, R.; Courtney, W.; Dunsworth, A.; Farhi, E.; Foxen, B.; Fowler, A.; Gidney, C.; Giustina, M.; Graff, R.; Guerin, K.; Habegger, S.; Harrigan, M. P.; Hartmann, M. J.; Ho, A.; Hoffmann, M.; Huang, T.; Humble, T. S.; Isakov, S. V.; Jeffrey, E.; Jiang, Z.; Kafri, D.; Kechedzhi, K.; Kelly, J.; Klimov, P. V.; Knysh, S.; Korotkov, A.; Kostritsa, F.; Landhuis, D.; Lindmark, M.; Lucero, E.; Lyakh, D.; Mandrá, S.; McClean, J. R.; McEwen, M.; Megrant, A.; Mi, X.; Michielsen, K.; Mohseni, M.; Mutus, J.; Naaman, O.; Neeley, M.; Neill, C.; Niu, M. Y.; Ostby, E.; Petukhov, A.; Platt, J. C.; Quintana, C.; Rieffel, E. G.; Roushan, P.; Rubin, N. C.; Sank, D.; Satzinger, K. J.; Smelyanskiy, V.; Sung, K. J.; Trevithick, M. D.; Vainsencher, A.; Villalonga, B.; White, T.; Yao, Z. J.; Yeh, P.; Zalcman, A.; Neven, H.; Martinis, J. M. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505−510.

(10) Wu, Y.; Bao, W.-S.; Cao, S.; Chen, F.; Chen, M.-C.; Chen, X.; Chung, T.-H.; Deng, H.; Du, Y.; Fan, D.; Gong, M.; Guo, C.; Guo, C.; Guo, S.; Han, L.; Hong, L.; Huang, H.-L.; Huo, Y.-H.; Li, L.; Li, N.; Li, S.; Li, Y.; Liang, F.; Lin, C.; Lin, J.; Qian, H.; Qiao, D.; Rong, H.; Su, H.; Sun, L.; Wang, L.; Wang, S.; Wu, D.; Xu, Y.; Yan, K.; Yang, W.; Yang, Y.; Ye, Y.; Yin, J.; Ying, C.; Yu, J.; Zha, C.; Zhang, C.; Zhang, H.; Zhang, K.; Zhang, Y.; Zhao, H.; Zhao, Y.; Zhou, L.; Zhu, Q.; Lu, C.-Y.; Peng, C.-Z.; Zhu, X.; Pan, J.-W. Strong Quantum Computational Advantage Using a Superconducting Quantum Processor. *Phys. Rev. Lett.* **2021**, *127*, 180501.

(11) Collins, H.; Easterly, K. IBM Unveils Breakthrough 127-Qubit Quantum Processor. IBM, 2021; https://newsroom.ibm.com/2021-11-16-IBM-Unveils-Breakthrough-127-Qubit-Quantum-Processor [accessed 2022-03-02].

(12) Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.-H.; Zhou, X.-Q.; Love, P. J.; Aspuru-Guzik, A.; O'Brien, J. L. A variational eigenvalue solver on a quantum processor. *Nat. Commun.* **2014**, *5*, 4213.

(13) Hempel, C.; Maier, C.; Romero, J.; McClean, J.; Monz, T.; Shen, H.; Jurcevic, P.; Lanyon, B. P.; Love, P.; Babbush, R.; Aspuru-Guzik, A.; Blatt, R.; Roos, C. F. Quantum Chemistry Calculations on a Trapped-Ion Quantum Simulator. *Phys. Rev. X* **2018**, *8*, 031022.

(14) Chen, M.-C.; Gong, M.; Xu, X.-S.; Yuan, X.; Wang, J.-W.; Wang, C.; Ying, C.; Lin, J.; Xu, Y.; Wu, Y.; Wang, S.; Deng, H.; Liang, F.; Peng, C.-Z.; Benjamin, S. C.; Zhu, X.; Lu, C.-Y.; Pan, J.-W. Demonstration of Adiabatic Variational Quantum Computing with a Superconducting Quantum Coprocessor. *Phys. Rev. Lett.* **2020**, *125*, 180501.

(15) Google AI Quantum and Collaborators; Arute, F.; Arute, F.; Aryae, K.; Babbush, R.; Bacon, D.; Bardin, J. C.; Barends, R.; Boixo, S.; Broughton, M.; et al. Hartree-Fock on a superconducting qubit quantum computer. *Science* **2020**, *369*, 1084−1089.

(16) Roffe, J. Quantum error correction: an introductory guide. *Contemp. Phys.* **2019**, *60*, 226−245.

(17) Chen, Z.; Satzinger, K. J.; Atalaya, J.; Korotkov, A. N.; Dunsworth, A.; Sank, D.; Quintana, C.; McEwen, M.; Barends, R.; Klimov, P. V.; et al. Exponential suppression of bit or phase errors with cyclic error correction. *Nature* **2021**, *595*, 383−387.

(18) Nguyen, N. H.; Li, M.; Green, A. M.; Huerta Alderete, C.; Zhu, Y.; Zhu, D.; Brown, K. R.; Linke, N. M. Demonstration of Shor Encoding on a Trapped-Ion Quantum Computer. *Phys. Rev. Appl.* **2021**, *16*, 024057.

(19) Egan, L.; Debroy, D. M.; Noel, C.; Risinger, A.; Zhu, D.; Biswas, D.; Newman, M.; Li, M.; Brown, K. R.; Cetina, M.; Monroe, C. Fault-Tolerant Operation of a Quantum Error-Correction Code. *arXiv Preprint (Quantum Physics)*, 2020. arXiv:2009.11482. https://doi.org/10.48550/arXiv.2009.11482.

(20) Pino, J. M.; Dreiling, J. M.; Figgatt, C.; Gaebler, J. P.; Moses, S. A.; Allman, M. S.; Baldwin, C. H.; Foss-Feig, M.; Hayes, D.; Mayer, K.; Ryan-Anderson, C.; Neyenhuis, B. Demonstration of the trapped-ion quantum CCD computer architecture. *Nature* **2021**, *592*, 209−213.

(21) Postler, L.; Heußen, S.; Pogorelov, I.; Rispler, M.; Feldker, T.; Meth, M.; Marciniak, C. D.; Stricker, R.; Ringbauer, M.; Blatt, R.; Schindler, P.; Müller, M.; Monz, T. Demonstration of fault-tolerant universal quantum gate operations. *Nature* **2022**, *605*, 675−680.

(22) Abobeih, M. H.; Wang, Y.; Randall, J.; Loenen, S. J. H.; Bradley, C. E.; Markham, M.; Twitchen, D. J.; Terhal, B. M.; Taminiau, T. H. Fault-tolerant operation of a logical qubit in a diamond quantum processor. *arXiv Preprint (Quantum Physics)*, 2021. arXiv:2108.01646. https://doi.org/10.48550/arXiv.2108.01646.

(23) Partington, J. R. *A short history of chemistry*; Courier: London, 1989.

(24) Lewis, G. N. the Atom and the Molecule. *J. Am. Chem. Soc.* **1916**, *38*, 762−785.

(25) Lewis, G. N. *Valence and the Structure of Atoms and Molecules*; Chemical Catalog, 1923.

(26) Burrau, Ø. Berechnung des Energiewertes des Wasserstoffmolekel-Ions (H2.) im Normalzustand. *Naturwissenschaften* **1927**, *15*, 16−17.

(27) Heitler, W.; London, F. Wechselwirkung neutraler Atome und homöopolare Bindung nach der Quantenmechanik. *Z. Phys.* **1927**, *44*, 455−472.

(28) Pauling, L. The Nature of the Chemical Bond. Application of Results Obtained from the Quantum Mechanics and from a Theory of Paramagnetic Susceptibility to the Structure of Molecules. *J. Am. Chem. Soc.* **1931**, *53*, 1367−1400.

(29) Hund, F. Zur Deutung der Molekelspektren. I. *Z. Phys.* **1927**, *40*, 742−764.

(30) Mulliken, R. S. The Assignment of Quantum Numbers for Electrons in Molecules. I. *Phys. Rev.* **1928**, *32*, 186−222.

(31) Lennard-Jones, J. E. The electronic structure of some diatomic molecules. *Trans. Faraday Soc.* **1929**, *25*, 668−686.

(32) Dirac, P. A. M.; Fowler, R. H. Quantum mechanics of many-electron systems. *Proc. R. Soc. London. Ser. A* **1929**, *123*, 714−733.

(33) Born, M.; Oppenheimer, R. Zur Quantentheorie der Molekeln. *Ann. Phys.* **1927**, *389*, 457−484.

(34) Slater, J. C. The Theory of Complex Spectra. *Phys. Rev.* **1929**, *34*, 1293−1322.

(35) Hartree, D. R. The Wave Mechanics of an Atom with a Non-Coulomb Central Field. Part I. Theory and Methods. *Math. Proc. Cambridge Philos. Soc.* **1928**, *24*, 89−110.

(36) Slater, J. C. Note on Hartree's Method. *Phys. Rev.* **1930**, *35*, 210−211.

(37) Fock, V. Näherungsmethode zur Lösung des quantenmechanischen Mehrkörperproblems. *Z. Phys.* **1930**, *61*, 126−148.

(38) Roothaan, C. C. J. New Developments in Molecular Orbital Theory. *Rev. Mod. Phys.* **1951**, *23*, 69−89.

(39) Knowles, P. J.; Schütz, M.; Werner, H.-J. Ab Initio Methods for Electron Correlation in Molecules. In *Modern Methods and Algorithms Quantum Chemitry*; Grotendorst, J., Ed.; John von Neumann Institute for Computing (NIC): Jülich, Germany, 2000; Vol. *1*, pp 69−151.

(40) Roos, B. O.; Taylor, P. R.; Sigbahn, P. E. A complete active space SCF method (CASSCF) using a density matrix formulated super-CI approach. *Chem. Phys.* **1980**, *48*, 157−173.

(41) Reiher, M.; Wiebe, N.; Svore, K. M.; Wecker, D.; Troyer, M. Elucidating reaction mechanisms on quantum computers. *Proc. Natl. Acad. Sci. U. S. A.* **2017**, *114*, 7555−7560.

(42) Genin, S. N.; Ryabinkin, I. G.; Paisley, N. R.; Whelan, S. O.; Helander, M. G.; Hudson, Z. M. Estimating Phosphorescent Emission Energies in IrIII Complexes Using Large-Scale Quantum Computing Simulations. *Angew. Chem., Int. Ed.* **2022**, *134*, e202116175.

(43) Lee, S.; Lee, J.; Zhai, H.; Tong, Y.; Dalzell, A. M.; Kumar, A.; Helms, P.; Gray, J.; Cui, Z.-H.; Liu, W.; et al. Is there evidence for exponential quantum advantage in quantum chemistry?. *arXiv Preprint (Chemical Physics)*, 2022. arXiv:2208.02199. https://doi.org/10.48550/arXiv.2208.02199.

(44) Anighoro, A. Underappreciated Chemical Interactions in Protein−Ligand Complexes. In *Quantum Mechanics in Drug Discovery*; Heifetz, A., Ed.; Springer: New York, NY, USA, 2020; pp 75−86. DOI: 10.1007/978-1-0716-0282-9_5.

(45) Shor, P. W. The Early Days of Quantum Computation. *arXiv Preprint (Quantum Physics)*, 2022. arXiv:2208.09964v1. https://doi.org/10.48550/arXiv.2208.09964.

(46) Shor, P. Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*; IEEE, 1994; pp 124−134. DOI: 10.1109/SFCS.1994.365700.

(47) Grover, L. K. A Fast Quantum Mechanical Algorithm for Database Search. *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*. New York, NY, USA, 1996; pp 212−219.

(48) Kitaev, A. Y. Quantum measurements and the Abelian Stabilizer problem. *arXiv Preprint (Quantum Physics)*, 1995. arXiv:quant-ph/9511026. https://doi.org/10.48550/arXiv.quant-ph/9511026.

(49) Cleve, R.; Ekert, A.; Macchiavello, C.; Mosca, M. Quantum algorithms revisited. *Proc. R. Soc. London, A* **1998**, *454*, 339−354.

(50) Nielsen, M. A.; Chuang, I. L. *Quantum computation and quantum information*, 10th ed.; Cambridge University Press, 2011. DOI: 10.1017/CBO9780511976667.

(51) Dyall, K. G. The choice of a zeroth-order Hamiltonian for second-order perturbation theory with a complete active space self-consistent-field reference function. *J. Chem. Phys.* **1995**, *102*, 4909−4918.

(52) Jordan, P.; Wigner, E. Über das Paulische Äquivalenzverbot. *Z. Phys.* **1928**, *47*, 631−651.

(53) Bravyi, S. B.; Kitaev, A. Y. Fermionic Quantum Computation. *Ann. Phys.* **2002**, *298*, 210−226.

(54) Seeley, J. T.; Richard, M. J.; Love, P. J. The Bravyi-Kitaev transformation for quantum computation of electronic structure. *J. Chem. Phys.* **2012**, *137*, 224109.

(55) Tranter, A.; Love, P. J.; Mintert, F.; Coveney, P. V. A Comparison of the Bravyi−Kitaev and Jordan−Wigner Transformations for the Quantum Simulation of Quantum Chemistry. *J. Chem. Theory Comput.* **2018**, *14*, 5617−5630.

(56) Higgott, O.; Wang, D.; Brierley, S. Variational Quantum Computation of Excited States. *Quantum* **2019**, *3*, 156.

(57) McClean, J. R.; Kimchi-Schwartz, M. E.; Carter, J.; de Jong, W. A. Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states. *Phys. Rev. A* **2017**, *95*, 042308.

(58) Motta, M.; Sun, C.; Tan, A. T.; O'Rourke, M. J.; Ye, E.; Minnich, A. J.; Brandao, F. G.; Chan, G. K. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nat. Phys.* **2020**, *16*, 205−210.

(59) Zhang, D.-B.; Yuan, Z.-H.; Yin, T. Variational quantum eigensolvers by variance minimization. *arXiv Preprint (Quantum Physics)*, 2020. arXiv:2006.15781. https://doi.org/10.48550/arXiv.2006.15781.

(60) Ryabinkin, I. G.; Genin, S. N.; Izmaylov, A. F. Constrained variational quantum eigensolver: Quantum computer search engine in the Fock space. *J. Chem. Theory Comput.* **2019**, *15*, 249−255.

(61) Nielsen, M. A.; Chuang, I. *Quantum computation and quantum information*; Cambridge University Press, 2002.

(62) Kitaev, A. Y.; Shen, A.; Vyalyi, M. N.; Vyalyi, M. N. *Classical and quantum computation*; American Mathematical Society, 2002.

(63) Tilly, J.; Chen, H.; Cao, S.; Picozzi, D.; Setia, K.; Li, Y.; Grant, E.; Wossnig, L.; Rungger, I.; Booth, G. H.; et al. The Variational Quantum Eigensolver: a review of methods and best practices. *arXiv Preprint (Quantum Physics)*, 2021. arXiv:2111.05176. https://doi.org/10.48550/arXiv.2111.05176.

(64) Bravyi, S.; Gambetta, J. M.; Mezzacapo, A.; Temme, K. Tapering off qubits to simulate fermionic Hamiltonians. *arXiv Preprint (Quantum Physics)*, 2017. arXiv:1701.08213. https://doi.org/10.48550/arXiv.1701.08213.

(65) Romero, J.; Babbush, R.; McClean, J. R.; Hempel, C.; Love, P. J.; Aspuru-Guzik, A. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Sci. Technol.* **2019**, *4*, 014008.

(66) Lee, J.; Huggins, W. J.; Head-Gordon, M.; Whaley, K. B. Generalized unitary coupled cluster wave functions for quantum computation. *J. Chem. Theory Comput.* **2019**, *15*, 311−324.

(67) Grimsley, H. R.; Economou, S. E.; Barnes, E.; Mayhall, N. J. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nat. Commun.* **2019**, *10*, 3007.

(68) Wecker, D.; Hastings, M. B.; Troyer, M. Progress towards practical quantum variational algorithms. *Phys. Rev. A* **2015**, *92*, 042303.

(69) McClean, J. R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **2016**, *18*, 023023.

(70) Yen, T.-C.; Verteletskyi, V.; Izmaylov, A. F. Measuring all compatible operators in one series of single-qubit measurements using unitary transformations. *J. Chem. Theory Comput.* **2020**, *16*, 2400−2409.

(71) Gokhale, P.; Angiuli, O.; Ding, Y.; Gui, K.; Tomesh, T.; Suchara, M.; Martonosi, M.; Chong, F. T. Minimizing State Preparations in Variational Quantum Eigensolver by Partitioning into Commuting Families. *arXiv Preprint (Quantum Physics)*, 2019. arXiv:1907.13623. https://doi.org/10.48550/arXiv.1907.13623.

(72) Crawford, O.; van Straaten, B.; Wang, D.; Parks, T.; Campbell, E.; Brierley, S. Efficient quantum measurement of Pauli operators in the presence of finite sampling error. *Quantum* **2021**, *5*, 385.

(73) Huggins, W. J.; McClean, J.; Rubin, N.; Jiang, Z.; Wiebe, N.; Whaley, K. B.; Babbush, R. Efficient and Noise Resilient Measurements for Quantum Chemistry on Near-Term Quantum Computers. *arXiv Preprint (Quantum Physics)*, 2019. arXiv:1907.13117. https://doi.org/10.48550/arXiv.1907.13117.

(74) Wang, D.; Higgott, O.; Brierley, S. Accelerated variational quantum eigensolver. *Phys. Rev. Lett.* **2019**, *122*, 140504.

(75) Wang, G.; Koh, D. E.; Johnson, P. D.; Cao, Y. Minimizing Estimation Runtime on Noisy Quantum Computers. *PRX Quantum* **2021**, *2*, 010346.

(76) Lin, L.; Tong, Y. Heisenberg-limited ground state energy estimation for early fault-tolerant quantum computers. *arXiv Preprint (Quantum Physics)*, 2021. arXiv:2102.11340. https://doi.org/10.48550/arXiv.2102.11340.

(77) Lee, J.; Berry, D. W.; Gidney, C.; Huggins, W. J.; McClean, J. R.; Wiebe, N.; Babbush, R. Even more efficient quantum computations of chemistry through tensor hypercontraction. *arXiv Preprint (Quantum Physics)*, 2020. arXiv:2011.03494. https://doi.org/10.48550/arXiv.2011.03494.

(78) Kitaev, A. Y. Quantum measurements and the Abelian stabilizer problem. *Electronic Colloquium on Computational Complexity*, Paper TR96-003; Computational Complexity Foundation, 1996.

(79) Tubman, N. M.; Mejuto-Zaera, C.; Epstein, J. M.; Hait, D.; Levine, D. S.; Huggins, W.; Jiang, Z.; McClean, J. R.; Babbush, R.; Head-Gordon, M.; et al. Postponing the orthogonality catastrophe: efficient state preparation for electronic structure simulations on quantum devices. *arXiv Preprint (Quantum Physics)*, 2018. arXiv:1809.05523. https://doi.org/10.48550/arXiv.1809.05523.

(80) Elfving, V. E.; Broer, B. W.; Webber, M.; Gavartin, J.; Halls, M. D.; Lorton, K. P.; Bochevarov, A. How will quantum computers provide an industrially relevant computational advantage in quantum chemistry?. *arXiv Preprint (Quantum Physics)*, 2020. arXiv:2009.12472. https://doi.org/10.48550/arXiv.2009.12472

(81) Gonthier, J. F.; Radin, M. D.; Buda, C.; Doskocil, E. J.; Abuan, C. M.; Romero, J. Identifying challenges towards practical quantum advantage through resource estimation: the measurement roadblock in the variational quantum eigensolver. *arXiv Preprint (Quantum*

Physics), 2020. arXiv:2012.04001. https://doi.org/10.48550/arXiv.2012.04001.

(82) Johnson, P. D.; Kunitsa, A. A.; Gonthier, J. F.; Radin, M. D.; Buda, C.; Doskocil, E. J.; Abuan, C. M.; Romero, J. Reducing the cost of energy estimation in the variational quantum eigensolver algorithm with robust amplitude estimation. *arXiv Peprint (Quantum Physics)*, 2022. arXiv:2203.07275. https://doi.org/10.48550/arXiv.2203.07275.

(83) Campbell, E. A random compiler for fast Hamiltonian simulation. *arXiv Preprint (Quantum Physics)*, 2018. arXiv:1811.08017. https://doi.org/10.48550/arXiv.1811.08017.

(84) Shor, P. Fault-tolerant quantum computation. *Proceedings of 37th Conference on Foundations of Computer Science*; IEEE, 1996. DOI: 10.1109/SFCS.1996.548464

(85) Aharonov, D.; Ben-Or, M. Fault-Tolerant Quantum Computation with Constant Error Rate. *SIAM J. Comput.* **2008**, *38*, 1207−1282.

(86) Knill, E.; Laflamme, R.; Zurek, W. H. Resilient Quantum Computation. *Science* **1998**, *279*, 342−345.

(87) Kitaev, A. Y. Fault-tolerant quantum computation by anyons. *Ann. Phys.* **2003**, *303*, 2−30.

(88) Fowler, A. G.; Mariantoni, M.; Martinis, J. M.; Cleland, A. N. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A* **2012**, *86*, 032324.

(89) Litinski, D. A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery. *Quantum* **2019**, *3*, 128.

(90) Eastin, B.; Knill, E. Restrictions on Transversal Encoded Quantum Gate Sets. *Phys. Rev. Lett.* **2009**, *102*, 110502.

(91) Litinski, D. Magic State Distillation: Not as Costly as You Think. *arXiv Preprint (Quantum Physics)*, 2019. arXiv:1905.06903. https://doi.org/10.48550/arXiv.1905.06903.

(92) Haah, J.; Hastings, M. B. Codes and Protocols for Distilling *T*, controlled-*S*, and Toffoli Gates. *Quantum* **2018**, *2*, 71.

(93) Haah, J.; Hastings, M. B. Codes and Protocols for Distilling T, controlled-S, and Toffoli Gates. *Quantum* **2018**, *2*, 71.

(94) Lodi, A.; Martello, S.; Monaci, M. Two-dimensional packing problems: A survey. *Eur. J. Oper. Res.* **2002**, *141*, 241−252.

(95) Childs, A. M.; Maslov, D.; Nam, Y.; Ross, N. J.; Su, Y. Toward the first quantum simulation with quantum speedup. *Proc. Natl. Acad. Sci. U. S. A.* **2018**, *115*, 9456−9461.

(96) Childs, A. M.; Su, Y.; Tran, M. C.; Wiebe, N.; Zhu, S. Theory of trotter error with commutator scaling. *Phys. Rev. X* **2021**, *11*, 011020.

(97) Reiher, M.; Wiebe, N.; Svore, K. M.; Wecker, D.; Troyer, M. Elucidating reaction mechanisms on quantum computers. *Proc. Natl. Acad. Sci. U. S. A.* **2017**, *114*, 7555−7560.

(98) Bocharov, A.; Roetteler, M.; Svore, K. M. Efficient synthesis of universal repeat-until-success quantum circuits. *Phys. Rev. Lett.* **2015**, *114*, 080502.

(99) Wecker, D.; Hastings, M. B.; Wiebe, N.; Clark, B. K.; Nayak, C.; Troyer, M. Solving strongly correlated electron models on a quantum computer. *Phys. Rev. A* **2015**, *92*, 062318.

(100) Low, G. H.; Chuang, I. L. Hamiltonian Simulation by Qubitization. *Quantum* **2019**, *3*, 163.

(101) Babbush.; et al. R. Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity. *Phys. Rev. X* **2018**, *8*, 041015.

(102) Berry, D. W.; Gidney, C.; Motta, M.; McClean, J. R.; Babbush, R. Qubitization of Arbitrary Basis Quantum Chemistry by Low Rank Factorization. *arXiv Preprint (Quantum Physics)*, 2019. arXiv:1902.02134. https://doi.org/10.48550/arXiv.1902.02134.

(103) von Burg, V.; Low, G. H.; Häner, T.; Steiger, D. S.; Reiher, M.; Roetteler, M.; Troyer, M. Quantum computing enhanced computational catalysis. *arXiv Preprint (Quantum Physics)*, 2020. arXiv:2007.14460. https://doi.org/10.48550/arXiv.2007.14460.

(104) Luis, A.; Perina, J. Optimum phase-shift estimation and the quantum description of the phase difference. *Phys. Rev. A* **1996**, *54*, 4564−4570.

(105) Visscher, K. M.; Geerke, D. P. Deriving Force-Field Parameters from First Principles Using a Polarizable and Higher Order Dispersion Model. *J. Chem. Theory Comput.* **2019**, *15*, 1875−1883.

(106) Jing, Z.; Liu, C.; Cheng, S. Y.; Qi, R.; Walker, B. D.; Piquemal, J.-P.; Ren, P. Polarizable force fields for biomolecular simulations: Recent advances and applications. *Annu. Rev. Biophys.* **2019**, *48*, 371−394.

(107) Nerenberg, P. S.; Head-Gordon, T. New developments in force fields for biomolecular simulations. *Curr. Opin. Struct. Biol.* **2018**, *49*, 129−138.

(108) Xu, P.; Guidez, E. B.; Bertoni, C.; Gordon, M. S. Perspective: *Ab initio* force field methods derived from quantum mechanics. *J. Chem. Phys.* **2018**, *148*, 090901.

(109) Sakharov, D. V.; Lim, C. Force fields including charge transfer and local polarization effects: Application to proteins containing multi/heavy metal ions. *J. Comput. Chem.* **2009**, *30*, 191−202.

(110) Cieplak, P.; Dupradeau, F.-Y.; Duan, Y.; Wang, J. Polarization effects in molecular mechanical force fields. *J. Phys.: Condens. Matter* **2009**, *21*, 333102.

(111) Li, P.; Merz, K. M., Jr Metal ion modeling using classical mechanics. *Chem. Rev.* **2017**, *117*, 1564−1686.

(112) Seifert, G.; Joswig, J.-O. Density-functional tight binding—an approximate density-functional theory method. *WIREs Comput. Mol. Sci.* **2012**, *2*, 456−465.

(113) Sure, R.; Grimme, S. Corrected small basis set Hartree-Fock method for large systems. *J. Comput. Chem.* **2013**, *34*, 1672−1685.

(114) Goez, A.; Neugebauer, J. Embedding Methods in Quantum Chemistry. *Frontiers of Quantum Chemistry*; Springer, 2018; pp 139−179. DOI: 10.1007/978-981-10-5651-2_7.

(115) Sun, Q.; Chan, G. K.-L. Quantum embedding theories. *Acc. Chem. Res.* **2016**, *49*, 2705−2712.

(116) Ikegami, T.; Ishida, T.; Fedorov, D. G.; Kitaura, K.; Inadomi, Y.; Umeda, H.; Yokokawa, M.; Sekiguchi, S. Full electron calculation beyond 20,000 atoms: ground electronic state of photosynthetic proteins. *SC'05 Proceedings of the 2005 ACM/IEEE Conference on Supercomputing*; IEEE, 2005; p 10. DOI: 10.1109/SC.2005.28.

(117) Senn, H. M.; Thiel, W. QM/MM Methods for Biomolecular Systems. *Angew. Chem., Int. Ed.* **2009**, *48*, 1198−1229.

(118) Wu, L.; Qin, L.; Nie, Y.; Xu, Y.; Zhao, Y.-L. Computer-aided understanding and engineering of enzymatic selectivity. *Biotechnol. Adv.* **2022**, *54*, 107793.

(119) Warshel, A.; Levitt, M. Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *J. Mol. Biol.* **1976**, *103*, 227−249.

(120) Cui, Q.; Pal, T.; Xie, L. Biomolecular QM/MM Simulations: What Are Some of the "Burning Issues"? *J. Phys. Chem. B* **2021**, *125*, 689−702.

(121) Lu, X.; Fang, D.; Ito, S.; Okamoto, Y.; Ovchinnikov, V.; Cui, Q. QM/MM free energy simulations: recent progress and challenges. *Mol. Simul.* **2016**, *42*, 1056−1078.

(122) Cao, L.; Ryde, U. On the Difference Between Additive and Subtractive QM/MM Calculations. *Front. Chem.* **2018**, *6*, 89.

(123) Himo, F. Recent Trends in Quantum Chemical Modeling of Enzymatic Reactions. *J. Am. Chem. Soc.* **2017**, *139*, 6780−6786.

(124) Cerqueira, N. M. F. S. A.; Fernandes, P. A.; Ramos, M. Protocol for Computational Enzymatic Reactivity Based on Geometry Optimisation. *ChemPhysChem* **2018**, *19*, 669−689.

(125) Boulanger, E.; Harvey, J. N. QM/MM methods for free energies and photochemistry. *Curr. Opin. Struct. Biol.* **2018**, *49*, 72−76.

(126) Hu, L.; Söderhjelm, P.; Ryde, U. Accurate Reaction Energies in Proteins Obtained by Combining QM/MM and Large QM Calculations. *J. Chem. Theory Comput.* **2013**, *9*, 640−649.

(127) Pandharkar, R.; Hermes, M. R.; Cramer, C. J.; Gagliardi, L. Localized Active Space State Interaction: A Multireference Method For Chemical Insight. *ChemRxiv Preprint (Theoretical and Computational Chemistry)*, 2022. DOI: 10.26434/chemrxiv-2022-jdzlr

(128) Otten, M.; Hermes, M. R.; Pandharkar, R.; Alexeev, Y.; Gray, S. K.; Gagliardi, L. *Localized Quantum Chemistry on Quantum*

Computers. arXiv Preprint (Quantum Physics), 2022. arXiv:2203.02012. https://doi.org/10.48550/arXiv.2203.02012

(129) Bauer, B.; Wecker, D.; Millis, A. J.; Hastings, M. B.; Troyer, M. Hybrid Quantum-Classical Approach to Correlated Materials. *Phys. Rev. X* **2016**, *6*, 031045.

(130) Rubin, N. C. A hybrid classical/quantum approach for large-scale studies of quantum systems with density matrix embedding theory. *arXiv Preprint (Quantum Physics)*, 2016. arXiv:1610.06910. https://doi.org/10.48550/arXiv.1610.06910.

(131) Yamazaki, T.; Matsuura, S.; Narimani, A.; Saidmuradov, A.; Zaribafiyan, A. Towards the practical application of near-term quantum computers in quantum chemistry simulations: A problem decomposition approach. *arXiv Preprint (Quantum Physics)*, 2018. arXiv:1806.01305. https://doi.org/10.48550/arXiv.1806.01305.

(132) Tilly, J.; Sriluckshmy, P. V.; Patel, A.; Fontana, E.; Rungger, I.; Grant, E.; Anderson, R.; Tennyson, J.; Booth, G. H. Reduced density matrix sampling: Self-consistent embedding and multiscale electronic structure on current generation quantum computers. *Phys. Rev. Res.* **2021**, *3*, 033230.

(133) Yao, Y.; Zhang, F.; Wang, C.-Z.; Ho, K.-M.; Orth, P. P. Gutzwiller hybrid quantum-classical computing approach for correlated materials. *Phys. Rev. Res.* **2021**, *3*, 013184.

(134) Kirsopp, J. J. M.; Paola, C. D.; Manrique, D. Z.; Krompiec, M.; Greene-Diniz, G.; Guba, W.; Meyder, A.; Wolf, D.; Strahm, M.; Ramo, D. M. Quantum Computational Quantification of Protein-Ligand Interactions. *Int. J. Quantum Chem.* **2021**, e26975.

(135) Izsak, R.; Riplinger, C.; Blunt, N. S.; de Souza, B.; Holzmann, N.; Crawford, O.; Camps, J.; Neese, F.; Schopf, P. Quantum Computing in Pharma: A Multilayer Embedding Approach for Near Future Applications. *arXiv Preprint (Chemical Physics)*, 2022. arXiv:2202.04460. https://arxiv.org/pdf/2202.04460.pdf.

(136) Fock, V.; Veselov, M.; Petrashen', M. Incomplete separation of variables for divalence atoms. *Zh. Eksp. Teor. Fiz.* **1940**, *10*, 723−739.

(137) Huzinaga, S.; Cantu, A. A. Theory of separability of many-electron systems. *J. Chem. Phys.* **1971**, *55*, 5543−5549.

(138) FDA expands approved use of Imbruvica for rare form of non-Hodgkin lymphoma. *FDA News Release*, U.S. Department of Health and Human Services, 2015;https://wayback.archive-it.org/7993/20170112222810/http://www.fda.gov/NewsEvents/Newsroom/PressAnnouncements/ucm432123.htm (accessed 2022-03-02).

(139) Honigberg, L. A.; Smith, A. M.; Sirisawad, M.; Verner, E.; Loury, D.; Chang, B.; Li, S.; Pan, Z.; Thamm, D. H.; Miller, R. A.; Buggy, J. J. The Bruton tyrosine kinase inhibitor PCI-32765 blocks B-cell activation and is efficacious in models of autoimmune disease and B-cell malignancy. *Proc. Natl. Acad. Sci.* **2010**, *107*, 13075−13080.

(140) Voice, A. T.; Tresadern, G.; Twidale, R. M.; van Vlijmen, H.; Mulholland, A. J. Mechanism of covalent binding of ibrutinib to Bruton's tyrosine kinase revealed by QM/MM calculations. *Chem. Sci.* **2021**, *12*, 5511−5516.

(141) Lodola, A.; Callegari, D.; Scalvini, L.; Rivara, S.; Mor, M. Design and SAR Analysis of Covalent Inhibitors Driven by Hybrid QM/MM Simulations. In *Quantum Mechanics in Drug Discovery*; Heifetz, A., Ed.; Methods in Molecular Biology, Vol. *2114*; Springer, 2020; pp 307−337. DOI: 10.1007/978-1-0716-0282-9_19.

(142) Ahmadi, S.; Barrios Herrera, L.; Chehelamirani, M.; Hostaš, J.; Jalife, S.; Salahub, D. R. Multiscale modeling of enzymes: QM-cluster, QM/MM, and QM/MM/MD: A tutorial review. *Int. J. Quantum Chem.* **2018**, *118*, e25558.

(143) Cerqueira, N. M. F. S. A.; Moorthy, H.; Fernandes, P. A.; Ramos, M. J. The mechanism of the Ser-(cis)Ser-Lys catalytic triad of peptide amidases. *Phys. Chem. Chem. Phys.* **2017**, *19*, 12343−12354.

(144) Prejanó, M.; Marino, T.; Russo, N. QM Cluster or QM/MM in Computational Enzymology: The Test Case of LigW-Decarboxylase. *Front. Chem.* **2018**, *6*, 249.

(145) Blomberg, M. R. A. The structure of the oxidized state of cytochrome c oxidase - experiments and theory compared. *J. Inorg. Biochem.* **2020**, *206*, 111020.

(146) Bender, A. T.; Gardberg, A.; Pereira, A.; Johnson, T.; Wu, Y.; Grenningloh, R.; Head, J.; Morandi, F.; Haselmayer, P.; Liu-Bujalski, L. Ability of Bruton's Tyrosine Kinase Inhibitors to Sequester Y551 and Prevent Phosphorylation Determines Potency for Inhibition of Fc Receptor but not B-Cell Receptor Signaling. *Mol. Pharmacol.* **2017**, *91*, 208−219.

(147) *Maestro*, Release 2021-3; Schrödinger: New York, NY, 2021.

(148) Madhavi Sastry, G.; Adzhigirey, M.; Day, T.; Annabhimoju, R.; Sherman, W. Protein and ligand preparation: parameters, protocols, and influence on virtual screening enrichments. *J. Comput.-Aided Mol. Des.* **2013**, *27*, 221−234.

(149) Jacobson, M. P.; Pincus, D. L.; Rapp, C. S.; Day, T. J. F.; Honig, B.; Shaw, D. E.; Friesner, R. A. A hierarchical approach to all-atom protein loop prediction. *Proteins: Struct., Funct., Bioinf.* **2004**, *55*, 351−367.

(150) Jacobson, M. P.; Friesner, R. A.; Xiang, Z.; Honig, B. On the Role of the Crystal Environment in Determining Protein Side-chain Conformations. *J. Mol. Biol.* **2002**, *320*, 597−608.

(151) Søndergaard, C. R.; Olsson, M. H. M.; Rostkowski, M.; Jensen, J. H. Improved Treatment of Ligands and Coupling Effects in Empirical Calculation and Rationalization of pKa Values. *J. Chem. Theory Comput.* **2011**, *7*, 2284−2295.

(152) Olsson, M. H. M.; Søndergaard, C. R.; Rostkowski, M.; Jensen, J. H. PROPKA3: Consistent Treatment of Internal and Surface Residues in Empirical pKa Predictions. *J. Chem. Theory Comput.* **2011**, *7*, 525−537.

(153) Bochevarov, A. D.; Harder, E.; Hughes, T. F.; Greenwood, J. R.; Braden, D. A.; Philipp, D. M.; Rinaldo, D.; Halls, M. D.; Zhang, J.; Friesner, R. A. Jaguar: A high-performance quantum chemistry software program with strengths in life and materials sciences. *Int. J. Quantum Chem.* **2013**, *113*, 2110−2142.

(154) Becke, A. D. Density-functional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.* **1993**, *98*, 5648−5652.

(155) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B* **1988**, *37*, 785−789.

(156) Grimme, S.; Antony, J.; Ehrlich, S.; Krieg, H. A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu. *J. Chem. Phys.* **2010**, *132*, 154104.

(157) Grimme, S.; Ehrlich, S.; Goerigk, L. Effect of the damping function in dispersion corrected density functional theory. *J. Comput. Chem.* **2011**, *32*, 1456−1465.

(158) Cossi, M.; Rega, N.; Scalmani, G.; Barone, V. Energies, structures, and electronic properties of molecules in solution with the C-PCM solvation model. *J. Comput. Chem.* **2003**, *24*, 669−681.

(159) Siegbahn, P. E. M.; Blomberg, M. R. A. Transition-Metal Systems in Biochemistry Studied by High-Accuracy Quantum Chemical Methods. *Chem. Rev.* **2000**, *100*, 421−438.

(160) Siegbahn, P. E. M.; Himo, F. The quantum chemical cluster approach for modeling enzyme reactions. *WIREs Comput. Mol. Sci.* **2011**, *1*, 323−336.

(161) Neese, F. Software update: the ORCA program system, version 4.0. *WIREs Comput. Mol. Sci.* **2017**, *8*, e1327.

(162) Weigend, F.; Ahlrichs, R. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy. *Phys. Chem. Chem. Phys.* **2005**, *7*, 3297−305.

(163) Pipek, J.; Mezey, P. G. A fast intrinsic localization procedure applicable for ab initio and semiempirical linear combination of atomic orbital wave functions. *J. Chem. Phys.* **1989**, *90*, 4916−4926.

(164) Knizia, G. Intrinsic atomic orbitals: An unbiased bridge between quantum theory and chemical concepts. *J. Chem. Theory Comput.* **2013**, *9*, 4834−4843.

(165) IBM Quantum breaks the 100-qubit processor barrier. IBM, 2021; https://research.ibm.com/blog/127-qubit-quantum-processor-eagle (accessed 2022-03-02).

(166) Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *arXiv Preprint (Quantum Physics)*, 2022. arXiv:2207.06431. https://doi.org/10.48550/arXiv.2207.06431.

(167) Wright, K.; Beck, K. M.; Debnath, S.; Amini, J. M.; Nam, Y.; Grzesiak, N.; Chen, J.-S.; Pisenti, N. C.; Chmielewski, M.; Collins, C.; Hudek, K. M.; Mizrahi, J.; Wong-Campos, J. D.; Allen, S.; Apisdorf, J.; Solomon, P.; Williams, M.; Ducore, A. M.; Blinov, A.; Kreikemeier, S. M.; Chaplin, V.; Keesan, M.; Monroe, C.; Kim, J. Benchmarking an 11-qubit quantum computer. *Nat. Commun.* **2019**, *10*, 5464.

(168) Erhard, A.; Poulsen Nautrup, H.; Meth, M.; Postler, L.; Stricker, R.; Stadler, M.; Negnevitsky, V.; Ringbauer, M.; Schindler, P.; Briegel, H. J.; Blatt, R.; Friis, N.; Monz, T. Entangling logical qubits with lattice surgery. *Nature* **2021**, *589*, 220−224.

(169) Dobšíček, M.; Johansson, G.; Shumeiko, V.; Wendin, G. Arbitrary accuracy iterative quantum phase estimation algorithm using a single ancillary qubit: A two-qubit benchmark. *Phys. Rev. A* **2007**, *76*, 030306.