Faster Quantum Chemistry Simulations on a Quantum Computer with Improved Tensor Factorization and Active Volume Compilation

Athena Caesura,[1] Cristian L. Cortes,[1] William Pol,[1] Sukin Sim,[1] Mark Steudtner,[1,*] Gian-Luca R. Anselmetti,[2] Matthias Degroote,[2,†] Nikolaj Moll,[2] Raffaele Santagati,[2] Michael Streif,[2] and Christofer S. Tautermann[3]

[1] *PsiQuantum, 700 Hansen Way, Palo Alto, California 94304, USA*

[2] *Quantum Lab, Boehringer Ingelheim, 55218 Ingelheim am Rhein, Germany*

[3] *Medicinal Chemistry, Boehringer Ingelheim Pharma GmbH & Co. KG, 88397 Biberach, Germany*

(Received 21 January 2025; revised 21 May 2025; accepted 22 July 2025; published 28 August 2025)

Electronic structure calculations of molecular systems are among the most promising applications for fault-tolerant quantum computing (FTQC) in quantum chemistry and drug design. However, while recent algorithmic advancements such as qubitization and tensor hypercontraction (THC) have significantly reduced the complexity of such calculations, they do not yet achieve computational runtimes short enough to be practical for industrially relevant use cases. In this work, we combine several advances to electronic structure calculation for molecular systems, resulting in a 2-orders-of-magnitude speedup of estimated runtimes over prior-art algorithms run on comparable quantum devices. One of these advances is a novel framework for block-invariant symmetry-shifted tensor hypercontraction, with which we achieve the tightest Hamiltonian factorizations reported to date. We compile our algorithm for an active volume (AV) architecture, a technical layout that has recently been proposed for fusion-based photonic quantum hardware. AV compilation contributes towards a lower runtime of our computation by eliminating overheads stemming from connectivity issues in the underlying surface code. We present a detailed benchmark of our approach, focusing primarily on the computationally challenging benchmark molecule P450. Leveraging a number of hardware trade-offs in interleaving-based photonic FTQC, we estimate runtimes for the electronic structure calculation of P450 as a function of the device footprint.

DOI: 10.1103/yngp-5fpm

I. INTRODUCTION

High-accuracy quantum chemical calculations have a large number of industrial applications, ranging from the optimization of reaction rates [1,2] to computer-aided drug design [3–6], and battery optimization [7–9]. However, because of the exponential scaling of the computational resources for the most accurate methods and the consequent impractical runtimes on classical computers, most quantum chemical calculations are often limited to approximate approaches, such as density functional theory (DFT), which result in less dependable predictions [10,11]. Consequently, accurate quantum chemical calculations represent one of the most anticipated practical applications of quantum computers.

A prime example is the electronic structure calculation of strongly correlated systems [12–17], such as the iron-molybdenum cofactor (FeMoco) [15] or cytochrome P450 [18], both playing critical roles in biological systems. FeMoco is part of the nitrogenase enzyme, which splits the dinitrogen triple bond, eventually leading to the generation of two ammonia molecules. So, a better understanding of its chemistry could bring new insights to design catalysts for nitrogen fixation. Meanwhile, cytochromes P450 are a group of heme proteins playing a crucial role in the metabolism of drugs. The interplay of drugs with P450 proteins may cause unwanted effects such as drug-drug interactions or accelerated systemic clearance of the active compound from the body. For this reason, they are often considered as antitargets in computer-aided drug design [19]. While both FeMoco and P450 are relevant to industrial applications and exhibit complex electronic structures, this work primarily focuses on the potential speedups in computing the ground-state energy of P450.

*Contact author: msteudtner@psiquantum.com

†Contact author: matthias.degroote@boehringer-ingelheim.com

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

In recent years, there has been an increasing endeavor to enhance the existing methods for improving the efficiency of electronic structure calculations in fault-tolerant quantum computing (FTQC). Since the first resource estimates for FeMoco [15], numerous studies have focused on estimating the upper limits of the computational resources needed, specifically the number of non-Clifford gates and qubits. Thanks to improved quantum algorithms [20–25] and better representations of the quantum chemical Hamiltonians [26–28], we have witnessed a reduction of several orders of magnitude of the quantum resources required for sampling the eigenspectrum of molecular systems.

From an algorithmic perspective, efforts using quantum phase estimation (QPE) have initially focused on estimating the spectra of electronic structure Hamiltonians H, by Trotterizing the time evolution operator $U = \exp(-iHt)$ on a quantum computer [29,30]. More recent research, however, has shifted towards employing *qubitization*: a version of QPE using a quantum walk operator to encode energies E of H into a spectrum of eigenvalues $\pm \arccos(E/\lambda)$, where λ is a parameter referred to as the *1-norm*. Not to be confused with the induced ℓ_1 of the Hamiltonian, the 1-norm is an upper bound on the operator norm of H. On a technical level, the value of the 1-norm arises in the *block encoding*, the part of the walk operator applying the Hamiltonian to the qubits representing the system [21–23]. Qubitization is regarded as the state of the art for electronic structure calculations, and its computational cost is directly proportional to λ. To significantly reduce the runtime of the quantum computation, one may use algorithms that encode factorized Hamiltonians with a lower 1-norm. The quantum algorithms with the current best performance (lowest resource cost) rely either on a double factorization (DF) approach [27] or make use of tensor hypercontraction (THC) [26]. These methods have shown very encouraging improvements in the expected quantum computing runtime for sampling the spectrum of the electronic structure Hamiltonian, estimated to be approximately 10^2 h (or roughly 10^9 non-Clifford gates) required for both the FeMoco and P450 systems [18,26] using a specific superconducting architecture running surface code. On top of these initial proposals, a number of recent works have made additional improvements to the 1-norm by combining DF with the block-invariant shift (BLISS) technique applied to the Hamiltonian [28,31–35].

However, even using the most efficient algorithms available today [26], the wall-clock runtimes of such calculations for industrially relevant systems such as cytochrome P450 [18] are of the order of days. Santagati *et al.* [5] argued that this is incompatible with the speed required for pharmaceutical industrial workflows, where calculations in the range of seconds or faster would be required. This requirement comes from the fact that, in the majority of cases, the calculation of ensemble properties is of interest. Usually, simulations require the sampling of a large number [$O(10^6)$] [36,37] of single-energy evaluations to compute the properties of thermodynamic ensembles [38]. Currently, severe approximations allow the treatment of relevant-sized systems by quantum chemistry, but for strongly correlated systems, such as drugs interacting with the heme center of P450 proteins, these approximations are not applicable. However, adequate descriptions are computationally out of reach of classical hardware. This is why further reductions in computational costs are needed for quantum computers to become relevant to industrial applications. This will likely require combining algorithmic and Hamiltonian decomposition improvements with better hardware and more efficient architectures. Recently, a novel *active volume* (AV) architecture has been proposed for fusion-based photonic FTQC, promising a considerable reduction in the computational runtime by exploiting nonlocal connections between the physical components in the quantum computer [39]. In conventional architectures without active volume capabilities, fault-tolerant operations require the participation of a large number of logical qubits that could otherwise be idle, preventing the compiler from using those qubits for other tasks in parallel. While the performance of AV compilation has been analyzed for RSA factorization [39] and elliptic curve cryptography [40], a study for chemistry problems is still lacking.

In this work, we integrate the BLISS technique with THC. Using a P450 system as a benchmark, we demonstrate that block-invariant symmetry-shifted tensor hypercontraction (BLISS THC), AV compilation, and some modifications of the block-encoding circuit reduce the computational runtime of an electronic structure calculation of P450 by at least a factor of 233, as compared to a THC calculation on an equivalent photonic hardware without AV-capabilities. A breakdown of the different speedups can be found in Table I, and a brief history of previous improvements to electronic structure calculations of P450, leading up to our work, can be found in Table II.

To obtain physical runtimes and space requirements, we provide a pragmatic review of the design features of a fault-tolerant quantum computer based on photonic fusions. For instance, in these types of devices, the number of *interleaving modules* (IMs) becomes a key metric to describe the physical size of the quantum computer. For photonic FTQC, the number of physical qubits, typically referenced in other architectures, has little meaning for the device footprint due to a feature called *interleaving* [41]. By storing entangled photons in a fiber, interleaving facilitates a trade-off between the physical runtime and the number of IMs, thus decoupling the code distance from the physical size of the device.

Our work, therefore, presents physical runtimes as a function of the number of IMs in the quantum computer. For devices with a large amount of interleaving, we obtain the minimum number of interleaving modules required to

TABLE I. Contributions to the speedup of the electronic structure calculation of P450, as compared to the THC-based calculation [26] on a baseline architecture defined in Sec. IV. The three steps, (1) active volume compilation, (2) the incorporation of BLISS within THC, and (3) improvements to the block-encoding circuit, are done in sequence while adding logical qubits saved in the memory to the workspace. Speedups reported reflect the runtime improvements with respect to the previous step. Multiplying the individual improvements hence results in the total speedup, up to rounding errors: all numbers in this table are rounded down, but the total speedup is computed with the exact estimates.

Method	Speedup factor
AV compilation	25.18×
THC ↦ BLISS THC	8.23×
Circuit improvements	1.12×
Total	233.96×

run the computation in a certain time frame, allowing us to compare our results with prior art [18]. What is more, we explore additional runtime improvements using algorithmic trade-offs between the number of qubits assigned to the memory of the computation, and its *workspace*, the group of qubits used for operational tasks.

The remainder of this paper is organized as follows. In Sec. II, we discuss some of the Hamiltonian factorization techniques referenced. Reviewing THC and BLISS, we finally develop BLISS THC. In Sec. III, we introduce the quantum circuit for the block encoding of the new Hamiltonian. After that, we lay the foundation for hardware and runtime calculations in Sec. IV, where we review active volume architectures, interleaving, workspace qubits, and runtime trade-offs. Drawing from the previous sections, we present our results for the P450 benchmark system in Sec. V, where we analyze the BLISS-THC Hamiltonian, obtain relative speedups from logical resource counts, provide wall-clock runtimes, and compute minimal device requirements. Our results feature a number of spacetime trade-offs, in particular with the code distance and size of the workspace, as devices with either smaller code distances or larger workspaces compute faster. This allows us to turn qubit savings due to modifications of the algorithm into speedups. We conclude with proposals for future research directions in Sec. VI.

II. BLISS THC

A. Theoretical overview

Within the standard framework of second quantization, we consider the electronic structure Hamiltonian written in chemist notation,

$$H = \sum_{\substack{pq \\ \sigma}} \left(h_{pq} - \frac{1}{2} \sum_{r} g_{prrq} \right) a^\dagger_{p\sigma} a_{q\sigma} + \frac{1}{2} \sum_{\substack{pqrs \\ \sigma\tau}} g_{pqrs} a^\dagger_{p\sigma} a_{q\sigma} a^\dagger_{r\tau} a_{s\tau}, \tag{1}$$

where $\{\sigma, \tau\}$ are indices for spin configurations $\uparrow, \downarrow$ and $\{p, q, r, s\}$ are indices labeling spatial orbitals. The terms h_{pq} and g_{pqrs} are the conventional one-body and two-body integrals defined with real molecular spatial orbitals $\phi_p(\mathbf{r})$,

$$h_{pq} = \int \mathrm{d}\mathbf{r}_1 \phi_p(\mathbf{r}_1) \left(-\frac{1}{2}\nabla^2 - \sum_I \frac{Z_I}{r_I} \right) \phi_q(\mathbf{r}_1), \tag{2}$$

$$g_{pqrs} = \iint \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 \frac{\phi_p(\mathbf{r}_1)\phi_q(\mathbf{r}_1)\phi_r(\mathbf{r}_2)\phi_s(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}. \tag{3}$$

Given the symmetry-shifting techniques to be introduced in the following sections, we also emphasize the symmetries of the electronic structure Hamiltonian, including total particle number $\hat{N}$, spin projection $\hat{S}_z$, and total

TABLE II. Logical qubit requirements for the device memory, Toffoli counts, active volume count (where applicable), as well as the 1-norm for electronic structure calculations of P450 in a (63e, 58o) active space configuration with respect to DF, symmetry-compressed double factorization (SCDF), THC, and BLISS THC, the product of this work. For our efforts, we have obtained tighter estimates for the costs of the THC circuit with respect to the parameters of prior art. Note that BLISS THC approaches 69.3 E_h, the theoretical 1-norm limit for P450 [42], to within a factor of 2.

Factorization	P450 (63e, 58o)			
	Memory qubits	Toffoli gates	Active volume	λ (E_h)
DF [28]	4922	1.9×10^{10}	–	472.2
SCDF [28]	1706	4.8×10^{9}	–	111.3
THC [18]	1434	7.8×10^{9}	–	388.9
THC [re-estimated for this work]	1357	7.8×10^{9}	1.6×10^{12}	388.9
BLISS THC [this work]	999	1.7×10^{9}	2.3×10^{11}	130.9

spin $\hat{S}^2$,

$$\hat{N} = \sum_p (a^\dagger_{p\uparrow} a_{p\uparrow} + a^\dagger_{p\downarrow} a_{p\downarrow}), \tag{4}$$

$$\hat{S}_z = \frac{1}{2} \sum_p (a^\dagger_{p\uparrow} a_{p\uparrow} - a^\dagger_{p\downarrow} a_{p\downarrow}), \tag{5}$$

$$\hat{S}^2 = \hat{S}_+ \hat{S}_- + \hat{S}_z(\hat{S}_z - 1), \tag{6}$$

where $\hat{S}_+ = \sum_p a^\dagger_{p\uparrow} a_{p\downarrow}$ and $\hat{S}_- = (\hat{S}_+)^\dagger$. While other symmetries may appear in specific problem instances, the symmetries mentioned above are always applicable.

B. Tensor hypercontraction

This work uses the THC factorization technique to decompose the four-index tensor g_{pqrs} [43,44]. THC is currently known to provide the best asymptotic scaling for ground-state energy estimation based on qubitized quantum phase estimation with $\tilde{O}(N)$ logical qubits and $\tilde{O}(N\lambda_{\mathrm{THC}}/\epsilon)$ Toffoli gates. In this framework, the four-index tensor g_{pqrs} is factorized using two-index tensors $\zeta_{\mu\nu}$ and χ^μ_p,

$$g_{pqrs} = \sum_{\mu,\nu=0}^{M-1} \zeta_{\mu\nu} \chi^\mu_p \chi^\mu_q \chi^\nu_r \chi^\nu_s, \tag{7}$$

where $\zeta_{\mu\nu}$ is a real symmetric matrix and M is the factorization rank. The tensor χ^μ_p is also assumed to be normalized such that $\sum_p \chi^\mu_p \chi^\mu_p = 1$. Combining Eqs. (1) and (7) leads to the following expanded form of the Hamiltonian:

$$\begin{aligned} H = &\sum_{\substack{pq \\ \sigma}} \left(h_{pq} - \frac{1}{2} \sum_r g_{prrq} \right) a^\dagger_{p\sigma} a_{q\sigma} \\ &+ \frac{1}{2} \sum_{\substack{\mu\nu \\ \sigma\tau}} \zeta_{\mu\nu} \left(\sum_p \chi^\mu_p a^\dagger_{p\sigma} \right) \left(\sum_q \chi^\mu_q a_{q\sigma} \right) \\ &\left(\sum_r \chi^\nu_r a^\dagger_{r\tau} \right) \left(\sum_s \chi^\nu_s a_{s\tau} \right). \end{aligned} \tag{8}$$

The second term can be simplified considerably by defining a linear transform $\sum_p \chi^\mu_p a_{p\sigma} = U^\dagger_\mu a_{0\sigma} U_\mu$, where the unitaries U_μ are products of Givens rotation operators that we cover later. The Jordan-Wigner transformation then gives rise to the Hamiltonian [26]

$$\begin{aligned} H = &-\frac{1}{2} \sum_{k,\sigma} t_k V^\dagger_k Z_{0,\sigma} V_k \\ &+ \frac{1}{8} \sum_{\substack{\mu\nu \\ \sigma\tau}} \zeta_{\mu\nu} U^\dagger_\mu Z_{0,\sigma} U_\mu U^\dagger_\nu Z_{0,\tau} U_\nu, \end{aligned} \tag{9}$$

where the t_k are the eigenvalues of the one-body matrix $T_{pq} = h_{pq} - \frac{1}{2} \sum_r g_{prrq} + \sum_r g_{pqrr}$, while $Z_{0,\sigma}$ is a Pauli-Z operator with respect to the topmost qubit in the register associated with the spin index σ, and V_k is the orbital rotation operator associated with t_k, defined similarly to U_μ. A block encoding capable of implementing the basis-rotated Z operators would yield the 1-norm

$$\lambda = \sum_k |t_k| + \frac{1}{2} \sum_{\mu\nu} |\zeta_{\mu\nu}|. \tag{10}$$

While this 1-norm expression is correct for the quantum circuit implementation proposed in Ref. [26], it is possible to show that a small modification of the circuit leads to the slightly smaller 1-norm

$$\lambda_\circ = \sum_k |t_k| + \frac{1}{2} \sum_{\mu\nu} |\zeta_{\mu\nu}| - \frac{1}{4} \sum_\mu |\zeta_{\mu\mu}|. \tag{11}$$

This modification involves a special treatment of the diagonal terms $\tilde{\zeta}_{\nu\nu}$ in Eq. (9), ensuring that we do not encode constant terms in our Hamiltonian: for identical spins $\sigma = \tau$, the terms $U^\dagger_\nu Z_{0,\sigma} U_\nu U^\dagger_\nu Z_{0,\tau} U_\nu$ are equal to 1. Constant contributions to the Hamiltonian should be avoided, as they would needlessly increase the 1-norm. Our correction leads to an updated Hamiltonian,

$$\begin{aligned} H_\circ = &-\frac{1}{2} \sum_{k,\sigma} t_k V^\dagger_k Z_{0,\sigma} V_k \\ &+ \frac{1}{8} \sum_{\substack{\mu\nu \\ \sigma\tau}} \zeta_{\mu\nu} U^\dagger_\mu Z_{0,\sigma} U_\mu U^\dagger_\nu Z_{0,\tau} U_\nu - \frac{1}{4} \sum_\mu \zeta_{\mu\mu}. \end{aligned} \tag{12}$$

C. BLISS framework

The BLISS [31,32] framework defines a block-invariant Hamiltonian, H_{BI}, that shifts the original Hamiltonian by a well-behaved symmetry function, $f(\hat{N}, \hat{S}^2, \hat{S}_z)$, leaving the eigenvalue spectrum of the desired symmetry sector unchanged apart from shifts with a constant. This strategy changes the eigenvalues of other symmetry sectors in order to minimize the 1-norm λ. Explicitly, the block-invariant Hamiltonian may be defined as

$$H_{\mathrm{BI}} = H - \alpha_1 \hat{N} - \frac{\alpha_2}{2} \hat{N}^2 - \frac{1}{2} \hat{B}(\hat{N} - \eta), \tag{13}$$

where the second and third terms are equal to one-body and two-body symmetry-shift operators, based on the total particle number operator $\hat{N}$, obeying the eigenvalue relation, $\hat{N}|\psi_\eta\rangle = \eta|\psi_\eta\rangle$, where η is the eigenvalue of the particle number operator, equal to the total number of spin-up and spin-down electrons, and $|\psi_\eta\rangle$ is any electronic wave function defined in the η-electron symmetry sector. The fourth

term in Eq. (13) is the BLISS operator equal to zero in the desired particle number sector and nonzero everywhere else, where $\hat{B}$ is a Hermitian one-body operator defined as

$$\hat{B} = \sum_{\substack{pq \\ \sigma}} \beta_{pq} a^\dagger_{p\sigma} a_{q\sigma}. \tag{14}$$

In total, the coefficients α_1, α_2, and β_{pq} encode a symmetry-based gauge invariance, which helps to reduce the 1-norm compared to what is theoretically achievable with the conventional Hamiltonian [45]. The BLISS method ultimately exploits the fact that the conventional Hamiltonian H inherently includes redundant information, which is only necessary when considering the full Fock space with all its possible symmetry sectors. Within the context of this quantum circuit, the input and output wave functions will generally belong to a single, well-defined symmetry sector, implying that the compiled Hamiltonian must only be equal to the original Hamiltonian within the very same sector. We conclude this section by explicitly writing the block-invariant Hamiltonian as

$$H_{\mathrm{BI}} = \sum_{\substack{pq \\ \sigma}} h^{(\mathrm{BI})}_{pq} a^\dagger_{p\sigma} a_{q\sigma} + \frac{1}{2} \sum_{\substack{pqrs \\ \sigma\tau}} g^{(\mathrm{BI})}_{pqrs} a^\dagger_{p\sigma} a_{q\sigma} a^\dagger_{r\tau} a_{s\tau}, \tag{15}$$

where we define the renormalized block-invariant integrals as

$$h^{(\mathrm{BI})}_{pq} = h_{pq} - \frac{1}{2} \sum_r g_{prrq} - \alpha_1 \delta_{pq} + \frac{1}{2} \beta_{pq} \eta, \tag{16}$$

$$g^{(\mathrm{BI})}_{pqrs} = g_{pqrs} - \alpha_2 \delta_{pq} \delta_{rs} - \tfrac{1}{2}(\beta_{pq} \delta_{rs} + \delta_{pq} \beta_{rs}). \tag{17}$$

In the following section, we build off this work, but perform the analysis in the Majorana representation, which is required to properly implement the THC framework for the block encoding.

D. Block-invariant symmetry-shifted tensor hypercontraction

While recent work [42] has shown that previous implementations of THC are within a factor of 2–3 of the conventional spectral bound, which is symmetry agnostic, they are still within a factor of 4–5 away from the bound in the $S = 5/2$ symmetry sector of P450. We improve upon those results by proposing a BLISS THC. To provide a unified framework that works both with the standard THC representation as well as BLISS THC, we present the results of this section using the Majorana representation of the Hamiltonian, where $a_{p\sigma} = (\hat{\gamma}_{p\sigma,0} + i\hat{\gamma}_{p\sigma,1})/2$ and $a^\dagger_{p\sigma} = (\hat{\gamma}_{p\sigma,0} - i\hat{\gamma}_{p\sigma,1})/2$. This representation is preferred due to the properties of Majorana operators, which are Hermitian and self-inverse, and have a clear one-to-one mapping with Pauli strings after performing the Jordan-Wigner transformation. Our main result is the block-invariant Majorana-based electronic structure Hamiltonian

$$H_{\mathrm{BI}} = \frac{i}{2} \sum_{\substack{pq \\ \sigma}} \kappa^{(\mathrm{BI})}_{pq} \hat{\gamma}_{p\sigma,0} \hat{\gamma}_{q\sigma,1} - \frac{1}{8} \sum_{\substack{pqrs \\ \sigma\tau}} g^{(\mathrm{BI})}_{pqrs} \hat{\gamma}_{p\sigma,0} \hat{\gamma}_{q\sigma,1} \hat{\gamma}_{r\tau,0} \hat{\gamma}_{s\tau,1}, \tag{18}$$

where

$$\kappa^{(\mathrm{BI})}_{pq} = h_{pq} - \frac{1}{2} \sum_r g_{prrq} + \sum_r g^{(\mathrm{BI})}_{pqrr} - \alpha_1 \delta_{pq} + \tfrac{1}{2} \beta_{pq} \eta, \tag{19}$$

$$g^{(\mathrm{BI})}_{pqrs} = g_{pqrs} - \alpha_2 \delta_{pq} \delta_{rs} - \tfrac{1}{2}(\beta_{pq} \delta_{rs} + \delta_{pq} \beta_{rs}). \tag{20}$$

These coefficients correctly encode the symmetry-shift and BLISS operations, renormalizing the coefficients that are typically found in the Majorana representation [27]. In the BLISS-THC framework, the block-invariant four-index tensor $g^{(\mathrm{BI})}_{pqrs}$ is factorized as

$$g^{(\mathrm{BI})}_{pqrs} = \sum_{\mu\nu}^{M} \tilde{\zeta}_{\mu\nu} \chi^\mu_p \chi^\mu_q \chi^\nu_r \chi^\nu_s, \tag{21}$$

which, combined with the elimination of constant terms, gives rise to the block-invariant THC Hamiltonian

$$\widetilde{H} = -\frac{1}{2} \sum_{\substack{k \\ \sigma}} \tilde{t}_k V^\dagger_k Z_{0,\sigma} V_k + \frac{1}{8} \sum_{\substack{\mu\nu \\ \sigma\tau}} \tilde{\zeta}_{\mu\nu} U^\dagger_\mu Z_{0,\sigma} U_\mu U^\dagger_\nu Z_{0,\tau} U_\nu - \frac{1}{4} \sum_\mu \tilde{\zeta}_{\mu\mu}, \tag{22}$$

where the $\tilde{t}_k$ are the eigenvalues of the one-body matrix $\kappa^{(\mathrm{BI})}_{pq}$. Since the proposed implementation, as defined above in Eq. (22), does not alter the final form of the Hamiltonian when compared to Eq. (9), we find that the final 1-norm expression of the BLISS-THC Hamiltonian is similarly given by

$$\tilde{\lambda}_{\mathrm{THC}} = \sum_k |\tilde{t}_k| + \frac{1}{2} \sum_{\mu\nu} |\tilde{\zeta}_{\mu\nu}| - \frac{1}{4} \sum_\mu |\tilde{\zeta}_{\mu\mu}|. \tag{23}$$

III. QUANTUM CIRCUITS

In this section, we consider the Hamiltonian block encoding of BLISS THC, a unitary quantum circuit that

acts as the factorized Hamiltonian in a certain subspace of the quantum system. Since BLISS THC and THC produce a Hamiltonian of the same form, we propose a quantum circuit quite similar to that in Ref. [26], but with a few modifications that prove to be beneficial for its complexity. Since the electronic structure calculation consists almost entirely of calls to the walk operator, even small modifications of the block encoding can have an impact on the overall runtime.

In the following, we provide a pedagogical explanation of the circuit, highlight our modifications to its construction, and refer the reader to the literature for further details. The block encoding is a sequence

$$\mathsf{Prepare}^\dagger\ \mathsf{Select}\ \mathsf{Prepare}$$

of the subroutines Select, depicted in Fig. 1, and Prepare, depicted in Fig. 2, as well as the uncomputation of the latter. Let us start with the Prepare routine in Sec. III A. After learning about the routine and the qubit registers featured in this algorithm, we move on to Select, and first focus on the part of the circuit implementing the one- and two-body operators in Sec. III B: circuits for the so-called *Givens rotations* are highlighted orange within Fig. 1. In Sec. III C we consider the full Select circuit as well as some of its variations that can facilitate trade-offs between time and space complexity.

A. Prepare

For the block encoding of BLISS THC, we use the Prepare routine of Lee *et al.* [26] to build an auxiliary state $|\Lambda\rangle$, as well as a Select routine entangling the auxiliary register with the molecular system qubits, such that

$$\langle\Lambda|\mathsf{Select}|\Lambda\rangle = \frac{\widetilde{H}}{\widetilde{\lambda}_{\mathrm{THC}}}. \tag{24}$$

In this work, we represent the electron system with the qubit registers $\mathsf{s}_\uparrow$ and $\mathsf{s}_\downarrow$, associated with spin-up and spin-down orbitals, respectively. A list of all qubits and registers can be found in Fig. 1(b). The auxiliary state takes the form

$$|\Lambda\rangle = \Bigg(\sqrt{p}|1\rangle_{\mathsf{a}} \sum_{\nu=0}^{M}\sum_{\mu=0}^{\nu} \sqrt{|\widehat{\zeta}_{\mu\nu}|}|\mu\rangle_{\mathsf{b}_0}|\nu\rangle_{\mathsf{b}_1}|\delta_{M,\nu}\rangle_{\mathsf{c}} \\ |\mathrm{sign}(\widehat{\zeta}_{\mu\nu})\rangle_{\mathsf{m}} + \sqrt{1-p}|0\rangle_{\mathsf{a}}\ldots\Bigg)|0\rangle_{\mathsf{i}}|0\rangle_{\mathsf{x}}|0\rangle_{\mathsf{y}}|0\rangle_{\mathsf{z}}, \tag{25}$$

where

(1) the coefficients $\widehat{\zeta}_{\mu\nu}$ are normalized versions of the one- and two-body operators, respectively;
(2) p is understood as a success parameter designed to be close to 1: if qubit a were to be measured, p would correspond to the probability of projecting into the subspace where the block encoding is prepared exactly;
(3) the function $x \mapsto \mathrm{sign}(x)$ outputs 0 unless x is negative, in which case it outputs 1;
(4) $|\mu\rangle$ and $|\nu\rangle$ denote computational basis states representing the integers μ and ν in binary representation;
(5) $\delta_{M,\nu}$ is a Kronecker delta.

To achieve the block-encoding property of Eq. (24), the coefficients $\widehat{\zeta}_{\mu\nu}$ are matched to the parameters of Hamiltonian (22) as

$$\widehat{\zeta}_{\mu\nu} = \frac{1}{\widetilde{\lambda}_{\mathrm{THC}}} \times \begin{cases} -\tilde{t}_\mu & \text{for } \nu = M \text{ and } \mu \in [0, N/2-1], \\ \tilde{\zeta}_{\mu\nu} & \text{for } \mu,\nu \in [0,M) \text{ and } \nu > \mu, \\ \tilde{\zeta}_{\nu\nu}/4 & \text{for } \nu \in [0,M) \text{ and } \mu = \nu, \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

The interested reader may find a detailed proof of this mapping in Appendix E. We again encounter a special treatment of the diagonal terms $\tilde{\zeta}_{\nu\nu}$ in Eq. (26), which is due to our version of Select excluding the constant terms, that have been subtracted off in Eq. (22).

Note that, to achieve the 1-norm of Eq. (23), we need to make sure that the success amplitude p is close to unity: $p \mapsto 1$. Later, spurious contributions to the energy from the inexact subspace of $|\Lambda\rangle$ will be eliminated by controlling the Select circuit on qubit a.

To encode $|\Lambda\rangle$, we follow the procedure of Lee *et al.* [26] and utilize a modified *alias sampling* routine [25] to prepare the state

$$\sum_\nu \sum_\mu \sqrt{|\widehat{\zeta}_{\mu\nu}|}|\mu\rangle_{\mathsf{b}_0}|\nu\rangle_{\mathsf{b}_1}|\mathrm{trash}_{\mu\nu}\rangle_{\mathsf{g}},$$

setting the values of qubits c and m in the process. In the resulting version of $|\Lambda\rangle$, registers b_0 and b_1 are entangled with some states $|\mathrm{trash}_{\mu\nu}\rangle$ in a garbage register g, but this is an acceptable overhead and has no consequence for the calculation. Before entering the alias sampling routine, both registers b_0 and b_1 must be prepared in a uniform superposition of all configurations $|\mu\rangle|\nu\rangle$ with $\nu \le M$ and $\mu \le \nu$. Starting from a superposition created by applying Hadamard gates to all qubits in b_0 and b_1, which would certainly include states with $\nu > M$ and $\mu > \nu$, we can obtain the desired superposition after at most two rounds

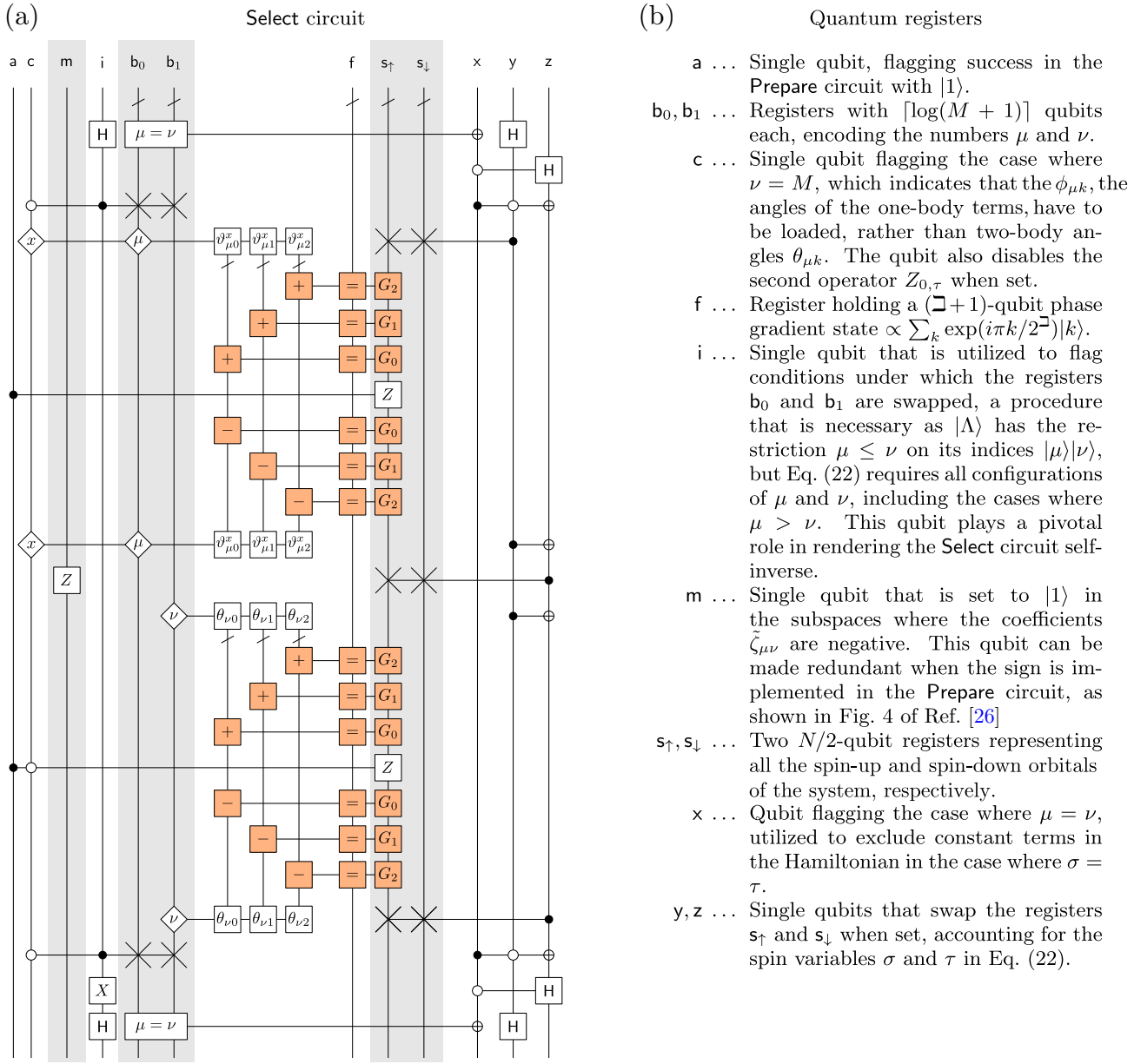


FIG. 1. (a) A small Select circuit of BLISS THC for $N = 8$. The circuit is to be read from top to bottom, where the Z gate refers to a Pauli-Z operator on the topmost qubit of $\mathsf{s}_\uparrow$, the H are Hadamard gates, and the gates labeled "$\mu = \nu$" compare the values stored in both registers and flip the target qubit if the integers match. The gates with the diamond-shaped connectors are dataloaders, loading the angles θ_{jk} and ϑ^x_{jk} into freshly allocated qubits, where we adopt the elbow formalism of Babbush *et al.* [25]. Here the angles ϑ^x_{jk} are the product of a combined register such that the angles $\vartheta^0_{jk} = \theta_{jk}$ are associated with two-body operators U_j and $\vartheta^1_{jk} = \phi_{jk}$ are associated with one-body operators, following Eq. (31). These angles serve as inputs for the programmable Givens rotation circuits, highlighted in orange and depicted in detail within Fig. 3(b), involving a phase gradient state in register f. Note that the X gate applied to qubit i is important to make Select self-inverse, i.e., $(\text{Select})^2 = 1$, an important prerequisite for qubitization. (b) Quantum registers in the Select circuit and the auxiliary state $|\Lambda\rangle$ in Eq. (25), not accounting for the garbage registers of Fig. 2.

of amplitude amplification featuring comparator circuits and partial reflections, where $\sqrt{1-p}$, the amplitude of the residual state with $\nu > M$ and $\mu > \nu$, can be made exponentially small by increasing the bit precision of the synthesized rotations. The Prepare circuit of Lee *et al.* [26] is depicted in Fig. 2.

Note that the dataloader, highlighted blue in Fig. 2, is generally made optimal with respect to its magic state cost, by turning the quantum read-only memory (QROM) into a QROAM using the techniques developed in Ref. [46]. With $\aleph$ qubits assigned to register j, which is later absorbed into the garbage register g, Prepare can approximate the

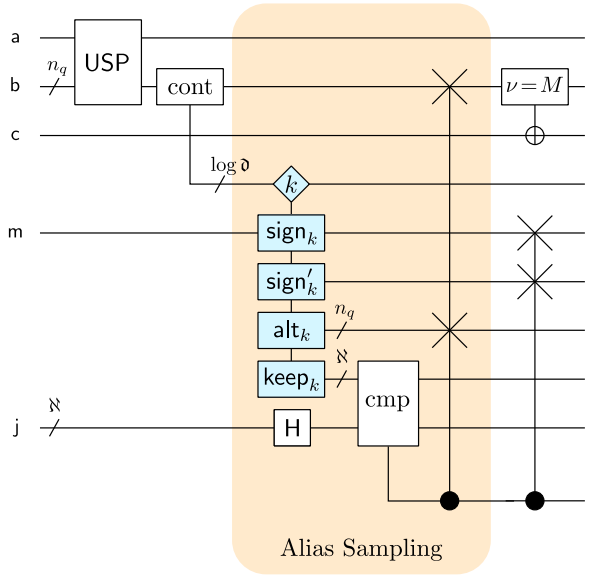


FIG. 2. The Prepare circuit of Ref. [26], where registers b_0 and b_1 are combined into a register b of n_q qubits. The register is then fed into the subroutine USP, which prepares a uniform superposition $\propto \sum_{\mu\nu} |\mu\rangle|\nu\rangle$ over all $\mathfrak{d} = M(M+1)/2 + N/2$ values of $\mu \leq \nu$, where $\widehat{\zeta}_{\mu\nu} \neq 0$. This uniform state preparation uses amplitude amplification, and success is therefore flagged in qubit a. This superposition state is a necessary input for the alias sampling block (orange) of Prepare, but since the basis states of the superposition do not represent numbers from a continuous range, parsing them would be hard for the routine. An arithmetic contiguizer (cont) therefore entangles the basis states $|\mu\rangle|\nu\rangle$ of b with basis states $|k(\mu,\nu)\rangle$ of a temporary register, $|\mu\rangle|\nu\rangle|0\rangle \mapsto |\mu\rangle|\nu\rangle|k(\mu,\nu)\rangle$, such that the numbers $k(\mu,\nu)$ are covering all integers from 0 to $\mathfrak{d}-1$ within the superposition. Inside the alias sampling routine, a dataloader (blue) loads $\aleph$-bit keep_k values and alternative indices $\mathsf{alt}_k = (\iota,\kappa)$ to the index pair (μ,ν) of a contiguous value $k(\mu,\nu)$. The routine also loads sign_k, the sign of the corresponding coefficient $\widehat{\zeta}_{\mu\nu}$ as well as the sign sign'_k of the coefficient $\widehat{\zeta}_{\iota\kappa}$ with respect to the alternative indices. A comparator (cmp) flags the cases where the values $|m\rangle$ of a register j are larger than the keep_k values. In this case, the alternative indices are swapped into b. The values of qubit c are computed after the alias sampling, using a controlled swap and the comparator "$\nu = M$", Here H denotes the application of Hadamard gates to every qubit in a register. The register j and temporary qubits make up the garbage register after the circuit.

coefficients $\widehat{\zeta}_{\mu\nu}$ within the accuracy

$$\frac{2^{-\aleph+1}}{M(M+1)+N}. \tag{27}$$

B. Givens rotations

The Select circuit is based on a subroutine that we shall call $\mathsf{subselect}_i$, performing the operation

$$\mathsf{subselect}_i|\mu\rangle_{\mathsf{b}_i} \otimes |\psi\rangle_{\mathsf{s}_\uparrow} = |\mu\rangle_{\mathsf{b}_i} \otimes U_\mu^\dagger Z_{0,\uparrow} U_\mu |\psi\rangle_{\mathsf{s}_\uparrow}, \tag{28}$$

defined with respect to one of the b_i registers. Rather than multiplexing over operators $\{U_\mu^\dagger Z_{0,\uparrow} U_\mu\}$ directly, $\mathsf{subselect}_i$ multiplexes over a set of parameters that define U_μ: it loads a series of angles $\{\theta_{\mu k}\}_k$ into temporary registers based on index μ. The quantum circuit for the $\mathsf{subselect}_i$ routine is depicted in Fig. 3(a). Let us say that each angle $\theta_{\mu k}$ is represented in $\beth + 1$ bits, i.e., we would store a positive number $\theta < 1$ in its binary fixed-point form to represent an angle $2\pi\theta$. These angles then become inputs for programmable Givens rotations, i.e., circuits that implement Givens rotations $G_k(\theta)$ for numbers $|\theta\rangle$ stored in an input register: $|\theta\rangle|\psi\rangle \mapsto |\theta\rangle G_k(\theta)|\psi\rangle$. Programmable Givens rotations make catalytic use of phase gradient states,

$$|\Psi\rangle = \frac{1}{\sqrt{2^{\beth+1}}} \sum_{k=0}^{2^{\beth+1}-1} e^{i\pi k/2^{\beth}} |k\rangle, \tag{29}$$

that we store in a register f. With such a state $|\Psi\rangle$, as well as a temporary register holding an angle θ, we can perform the phase operation $\mathcal{A}|\theta\rangle|\Psi\rangle = \exp(-i2\pi\theta)|\theta\rangle|\Psi\rangle$ with an in-place adder $\mathcal{A}$, adding the content of the temporary register into $|\Psi\rangle$. Using this phase gradient state addition for rotations $\exp(i2\pi\theta P)$ of some Pauli string P, one must additionally flip the sign of the angle, i.e., $\theta \mapsto 1-\theta$, in the (-1) subspace of P. This can be achieved by flipping all qubits of f conditional on the value of P [47]. However, a single Givens rotation consists of two (commuting) Pauli string rotations acting on consecutive qubits (k and $k-1$ for G_k) with opposite angles, i.e.,

$$G_k(\theta) = \exp(i\pi\theta X_k Y_{k-1}) \exp(-i\pi\theta Y_k X_{k-1}), \tag{30}$$

and so $G_k(\theta)$ has the eigenvalues $\{1, \exp(i2\pi\theta), \exp(-i2\pi\theta)\}$. Hence we can implement $G_k(\theta)$ with two separate adders, or we can fuse the two adders into a single circuit: a controlled adder featuring conditional flips of the phase gradient register. The fused-adder circuit, shown in Fig. 3(b), has a lower AV than the separate-adder circuit, used in Ref. [26]. It even has a lower Toffoli count, considering that the bit strings going into the fused adder correspond to angles θ rather than $\theta/2$, and are therefore one bit shorter. Putting everything together, the $\mathsf{subselect}_i$ operator multiplexes over all possible values $|\mu\rangle$ of register b_i, writing the respective angles $\theta_{\mu k}$ into temporary registers and feeding those into a sequence of programmable Givens rotation circuits. While the set of angles $\{\theta_{\mu k}\}_k$ constructs the operators U_μ associated with the two-body terms, we also define the set of angles $\{\phi_{\mu k}\}_k$ used to construct the one-body operators V_μ:

$$\prod_{k=0}^{N/2-2} G_k(\theta_{\mu k}) = U_\mu, \qquad \prod_{k=0}^{N/2-2} G_k(\phi_{\mu k}) = V_\mu. \tag{31}$$

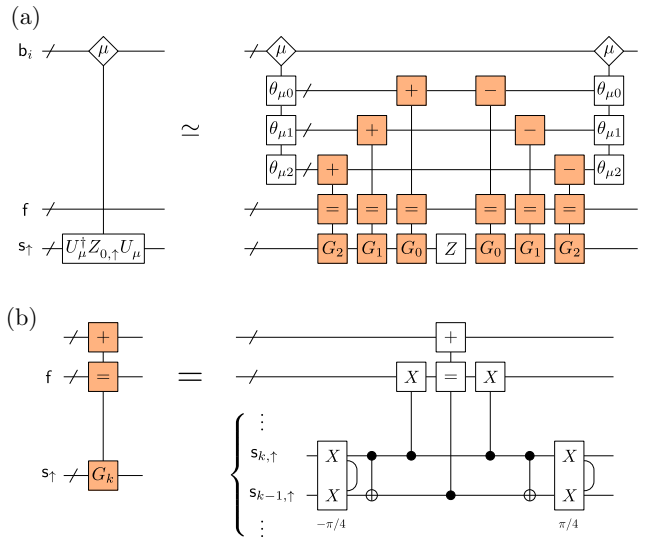


FIG. 3. (a) The subselect$_i$ circuit, depicted on the right-hand side, corresponds to a multiplexed application of $U_\mu^\dagger Z_{0,\uparrow} U_\mu$ (on the left). The circuit loads the angles $\theta_{\mu,k}$ into temporary registers and employs programmable Givens rotation circuits (orange), where both Z and $Z_{0,\uparrow}$ refer to Pauli-Z operators on the topmost qubit of register $\mathsf{s}_\uparrow$. In this minimal example, $\mathsf{s}_\uparrow$ has only $N/2 = 4$ qubits. (b) A programmable Givens rotation subcircuit. Here, the lower register $\mathsf{s}_\uparrow$ fans out into singular qubits $\mathsf{s}_{0,\uparrow}, \mathsf{s}_{1,\uparrow}, \ldots, \mathsf{s}_{N/2-1,\uparrow}$, of which this circuit uses exactly two subsequent qubits, determined by the index k of G_k. The circuit implements the product of rotations $\exp(i\pi\theta X_{\mathsf{s}_{k,\uparrow}} Y_{\mathsf{s}_{k-1,\uparrow}})$ and $\exp(-i\pi\theta Y_{\mathsf{s}_{k,\uparrow}} X_{\mathsf{s}_{k-1,\uparrow}})$ for fixed-point values $\theta < 1$ in the $|\theta\rangle$ subspace of the temporary register on top. This is done under catalytic use of a phase-gradient state in register f. The circuit on the right-hand side features $\pm\pi/4$ rotations of Pauli strings $X_{\mathsf{s}_{k,\uparrow}} X_{\mathsf{s}_{k-1,\uparrow}}$, controlled flips of all qubits in f denoted by the gates "X," and a controlled Gidney adder [47], which adds the binary number stored in the register under the "+" gate into the register under the "=" gate. For the Givens rotation uncomputations, this adder is replaced with an in-place subtractor, symbolized by "−".

The relationship between the rotation angles $\theta_{\mu k}$ and the tensors χ_p^μ are discussed in Appendix B. For the two-body terms, Select must call two instances of subselect$_i$ sequentially, one for $\mathsf{b}_i = \mathsf{b}_0$ and the other on $\mathsf{b}_i = \mathsf{b}_1$.

C. Select

Besides multiplexing over different Givens rotations, the Select circuit in Fig. 1(a) fulfills a number of tasks. It needs to

(i) switch between one- and two-body terms;
(ii) swap registers $\mathsf{s}_\uparrow$ and $\mathsf{s}_\downarrow$ to account for spins σ, τ as in Eq. (22);
(iii) only keep the terms $\sigma \neq \tau$ in the case $\mu = \nu$;
(iv) access the cases $\mu > \nu$ not encoded in $|\Lambda\rangle$; and
(v) make sure that the whole Select circuit is self-inverse.

To accomplish task (i), we use qubit c as an indicator for when to load the angles $\phi_{\mu k}$ rather than $\theta_{\mu k}$. Tasks (ii) and (iii) are fulfilled using qubits x, y, and z. Identifying the spins $\uparrow, \downarrow$ with bit values 0, 1, qubits y and z are prepared in $\propto \sum_\sigma \sum_{\tau(\sigma)} |\sigma\rangle|\tau\rangle$, and qubit x acts as a switch for this state to be either $|+\rangle|+\rangle$ or $(|01\rangle + |10\rangle)/\sqrt{2}$. For task (iv), b_0 and b_1 are swapped controlled on the subspace of qubit i. Select is self-inverse following the argument outlined by Lee *et al.* [26], who suggested that most of what we see in Fig. 1 is just a unitary transform of the actual Select circuit, the (self-inverse) Pauli-X operator applied to qubit i towards the end.

We have two more comments to make about Select. First, while the angle parameters $\theta_{\mu k}$ and $\phi_{\mu k}$ are fixed-point numbers represented with $\beth + 1$ bits of precision, we have found that they can be represented with only $\beth$ bits, as we have the freedom to set their first bit to zero. To our knowledge, this has not been considered in prior art. Our definition of $\beth$ is hence somewhat different from that of Lee *et al.* [26]. For more details, we refer the reader to Appendix B. Second, note that the Select circuit in Fig. 1 is optimized for the lowest magic state count and uses a lot of auxiliary qubits. Specifically, the elbow dataloader needs to reserve $\beth$ bits of precision for each of the $N/2 - 1$ angles, leading to an overhead of 741 qubits for P450 ($N = 116$ and $\beth = 13$). This qubit requirement can be alleviated by a simple modification: instead of loading all $N/2 - 1$ angles at once, one may decide to load only r angles at a time, using $r \times \beth$ auxiliary qubits. After one batch of angles has been fed into r Givens rotations, a subsequent dataloader would load the next batch into the same temporary registers, overwriting their previous values. This process is depicted in Fig. 4. Batching in groups of r angles would require $(N-2)/r$ more dataloaders per subselect, which increases its overall complexity regardless of how we measure it, but it can prove to be useful when the number of qubits is a limiting factor.

IV. FRAMEWORK FOR HARDWARE AND RUNTIME CALCULATIONS

A. Active volume and physical resource estimation

This subsection outlines our methodology for estimating the physical resources required to execute a quantum computation using photonic fusion-based quantum computing hardware. We consider two architectures from the literature: a baseline (BL) interleaved fusion-based quantum computing (FBQC) architecture [41] and the AV architecture [39]. To simplify our resource estimation, we assume that our architectures implement surface codes and that logical operations are carried out using lattice surgery [48].

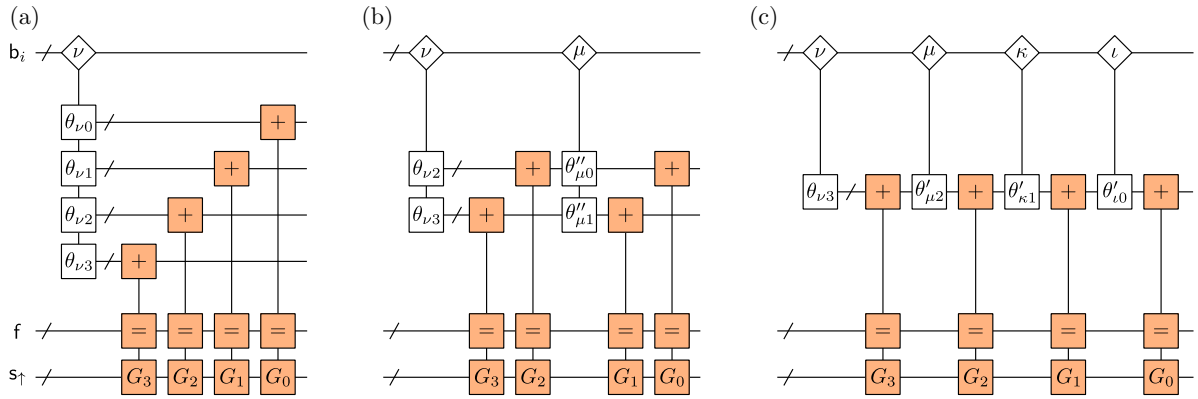


FIG. 4. Batching the angle loading exemplified on a part of the Select circuit. (a) No batching: all angles are loaded at the same time, each into a separate temporary quantum register. Givens rotations then feed on those individual registers. This results in the version of Select depicted in Fig. 1. It has the lowest magic state requirements, but uses the largest number of qubits. (b) Some batching: a number of angles loaded at the same time, right before their use in Givens rotation circuits. This reduces the qubit cost, but requires repeated data loading, increasing the active volume. After the Givens rotation circuits of one batch are done, new angles are loaded into the already-allocated temporary registers. Those registers still hold the values $|\theta_{\nu 2}\rangle$ and $|\theta_{\nu 3}\rangle$, but the data strings $\theta''_{\mu 0}$ and $\theta''_{\mu 1}$ are defined such that overwriting the temporary registers leaves them in states $|\theta_{\nu 0}\rangle$ and $|\theta_{\nu 1}\rangle$. (c) The maximum amount of batching: each angle is loaded right before its respective Givens rotation circuit. This version uses only one temporary register to store angles. After the kth Givens rotation circuit, we are turning state $|\theta_{\nu k}\rangle$ into $|\theta_{\nu(k+1)}\rangle$ by loading the data strings $\theta'_{\nu(k+1)}$.

We first quantify the logical resources, and then translate those estimates into physical resource counts. For both of the aforementioned architectures, the canonical metric for quantifying logical resources is the *spacetime volume*. Roughly speaking, one can calculate the spacetime volume by multiplying the number of logical qubits by the number of logical cycles required to complete the computation. A *logical cycle* comprises d *code cycles*, where d refers to the code distance and a code cycle is the time required to measure every syndrome. We often measure time in units of logical cycles because a logical cycle is the period in which a single lattice surgery operation is implemented [49].

As the spacetime volume is a product of the two main resources used in a computation—the number of qubits and time—trade-offs between these two resources are captured well by this metric. The quantum architecture determines the layout of the logical qubits and how logical operations are implemented, so the choice of architecture can have a profound influence on the spacetime volume. Thus, our first task will be to quantify the spacetime volume for both the baseline and AV architectures.

The baseline architecture assumes $2m$ logical qubits, of which half are memory qubits (we will in fact use m as the number of memory qubits) and the other half are workspace qubits. In addition to memory and workspace qubits, a third group of qubits is reserved for distilling magic states to implement T gates, as shown in Fig. 5(a). In the simplest variant of this architecture, we use enough magic state factories to produce one T gate per logical cycle [50]. Then, while the next T gate is being produced, the workspace qubits can consume the magic state produced in the previous logical cycle. This simple production and consumption strategy ensures that the total computation time depends only on the number of T gates that have to be produced. In fact, we can use the total T count, or n_T, as a representative proxy for the time dimension of the spacetime volume metric. Thus, the spacetime volume of the BL architecture is $2m \times n_T$, where $m \times n_T$ is also referred to as the *circuit volume*.

In the AV architecture, circuit volume is replaced by active volume. Active volume is simply the number of lattice surgery operations used in the computation. The active volume architecture has a total of $n = m + w$ logical qubits: m qubits are allocated for memory, and w qubits are allocated for workspace. Unlike in the BL architecture described above, the number of workspace qubits is not restricted to be equal to the number of memory qubits and we have the option to vary the number of workspace qubits [51]. The workspace qubits carry out magic state distillations and logical operations, so the more workspace qubits we have, the faster we compute. This architecture has a logarithmic number of nonlocal connections among memory qubits and workspace qubits that facilitate operations such as quickswaps to help "pack" each logical cycle with lattice surgery operations; see Fig. 5(b). To emphasize that we are "packing" lattice surgery operations as much as possible, we often refer to lattice surgery operations in the AV architecture as *logical blocks*. "Logical blocks" is thus the unit of AV and circuit volume counts. Optimistically, the additional connections given in the AV architecture allow us to execute w logical blocks in each logical cycle. A quantum computation with a total active volume of b_{AV}

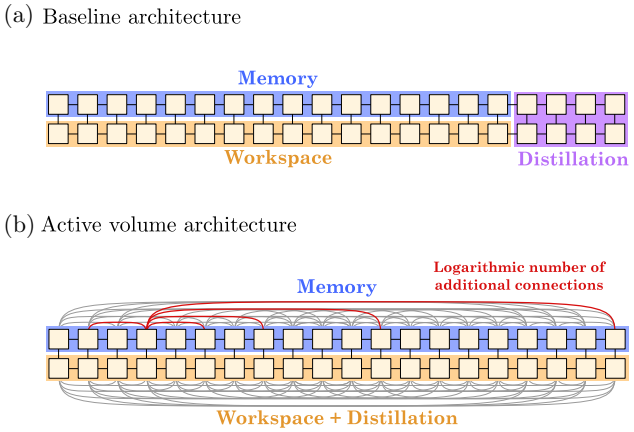


FIG. 5. FBQC architectures considered in this work, where nodes correspond to surface code patches. (a) Baseline architecture, with connectors signifying geometric adjacency, enabling lattice surgery operations, between workspace (yellow), memory (blue), and distillation (purple) qubits. (b) Active volume architecture, in which distillation is done with the workspace qubits. Within the memory and workspace groups, a logarithmic number of additional connections per qubit lets us swap qubits quickly. These additional connections allow workspace qubits to connect with any memory qubit with a small number of quickswap operations. Thus, unlike the baseline architecture, we can vary the number of workspace qubits while maintaining connectivity.

logical blocks results in an estimated spacetime volume of $(w+m)\times(b_{\mathrm{AV}}/w)=(1+m/w)b_{\mathrm{AV}}$. To roughly estimate b_{AV} or the total active volume of an algorithm, one could look up and sum the numbers of logical blocks in an active volume architecture for various operations and subroutines from Table 1 of Ref. [39].

In summary, for a BL architecture, the spacetime volume is approximately $2\times$ the circuit volume, while for an AV architecture, the spacetime volume is approximately $(n/w)\times$ the active volume:

$$\begin{aligned}\text{Spacetime volume (STVol)} &= \text{number of logical qubits}\\ &\quad\times\text{number of logical cycles},\\ \mathrm{STVol}_{\mathrm{BL}} &= \underbrace{2m}_{\text{logical qubits}}\times\underbrace{n_T}_{\text{logical cycles}} = 2\times\underbrace{m\times n_T}_{\text{circuit volume}},\end{aligned}\tag{32}$$

$$\begin{aligned}\mathrm{STVol}_{\mathrm{AV}} &= \underbrace{(m+w)}_{\text{logical qubits}}\times\underbrace{b_{\mathrm{AV}}/w}_{\text{logical cycles}} = \frac{n}{w}\times\underbrace{b_{\mathrm{AV}}}_{\text{active volume}}\\ &= \frac{n}{w}\left(\sum_{i\in\text{subroutines}} b_{\mathrm{AV},i}\right),\end{aligned}\tag{33}$$

where $b_{\mathrm{av},i}$ is the number of logical blocks in each subroutine used in the computation. Because of the additional operations that pack or parallelize more logical blocks per logical cycle, an active volume architecture can significantly reduce the overall spacetime volume of a given computation, implying that the active volume can often be significantly lower than the circuit volume [39].

Now that we have estimated the logical resources, we can calculate the physical resources required for executing the computation, including the runtime and *footprint* [52], using additional information about the hardware. For a general surface code–based quantum computation, the physical runtime can be estimated as

$$t_{\mathrm{comp}} = \text{number of code cycles}\times\text{code cycle time},\tag{34}$$

where the number of code cycles is the number of logical cycles times the code distance d. For the baseline architecture, we recall that the number of logical cycles is n_T. For the active volume architecture, the number of logical cycles is b_{AV}/w, where b_{AV} is the total active volume. The code cycle time depends on the hardware. For instance, the Google Quantum AI team often reports the code cycle time of their superconducting circuit-based quantum computers to be of the order of microseconds [18,53–55]. The physical footprint can be estimated based on the number of logical qubits and the code distance.

We now specify the equations for estimating the runtime and footprint for photonic FBQC hardware. At a high level, FBQC involves the generation of entangled few-photon resource states, storing them in optical fiber or delay lines, and implementing entangling two-photon measurements between different resource states that are called fusions [41]. The generation and measurement of resource states are prescribed by a *fusion graph*, which can be mapped from a spacetime diagram, a common representation used to specify a surface code–based quantum computation. This mapping determines fusions by corresponding stabilizer measurements in the spacetime diagram representation.

An FBQC has a number of IMs, which generate and hold these photonic resource states. Active volume architectures use a modified version of the interleaving modules in baseline architectures, due to the different connectivity requirements (see Fig. 4 of Ref. [41] and Fig. 23 of Ref. [39]), but we will regard the two types as comparable. These resource states pass through delay lines connecting interleaving modules over specific durations before participating in measurements called fusions. In this simplified picture, FBQC can be viewed as a game of delaying and routing resource states before fusions—the length of the delay line determines the ratio of logical qubits per IM. In contrast, connections between different interleaving modules and fusions facilitate operations.

The expression to estimate the footprint can be more easily understood by considering how logical qubits are realized in FBQC. The number of logical qubits depends

on the maximum number of resource states present simultaneously in the interleaving module, which is linearly dependent on the length of the longest delay line in the interleaving module l_{delay} and the number of interleaving modules n_{IM}. Assuming a surface code, each logical qubit requires a total of d^2 resource states, despite practical implementations of a surface code typically requiring $2d^2 - 1$ physical qubits. This is because the fusions between resource states act as syndrome measurements, eliminating half the qubits that would typically be needed to hold the syndrome data in a surface code [56].

Thus, the number of logical qubits n in an FBQC at any given time can be written as

$$n = \underbrace{n_{\text{IM}}\, n_{\text{RS per IM}}}_{\substack{\text{number of resource states}\\ \text{over all IMs}}} \times \underbrace{1/d^2}_{\substack{\text{ratio of logical qubits}\\ \text{to resource states}}}, \tag{35}$$

where $n_{\text{RS per IM}}$ is the number of resource states that can be stored in one interleaving module. We can write $n_{\text{RS per IM}}$ as

$$n_{\text{RS per IM}} = r_{\text{IM}} \times \underbrace{\frac{l_{\text{delay}}}{c_{\text{fiber}}}}_{\text{delay time}}, \tag{36}$$

where $r_{\text{IM}} \approx 10^9$ Hz [39] is the rate at which an IM produces resource states, l_{delay} is the length of the fiber optic cable, and $c_{\text{fiber}} \approx 2 \times 10^5$ km/s is the speed of light in fiber optic cables. The footprint can then be estimated by rearranging the equation with respect to the number of IMs:

$$n_{\text{IM}} = \frac{nd^2 c_{\text{fiber}}}{r_{\text{IM}} l_{\text{delay}}}. \tag{37}$$

While our explanation derives the number of IMs based on the memory requirement of a computation, in practice, the number of IMs is fixed.

If we denote the total number of resource states needed to complete the computation as Vol_{comp} then the runtime can be estimated by the time required to produce all of those states:

$$\begin{aligned} t_{\text{comp}} &= \text{Vol}_{\text{comp}} \times \underbrace{(n_{\text{IM}} r_{\text{IM}})^{-1}}_{\text{resource states created in 1 s}} \\ &= \frac{\text{Vol}_{\text{comp}}\, l_{\text{delay}}}{nd^2\, c_{\text{fiber}}}. \end{aligned} \tag{38}$$

This volume can be estimated based on the spacetime volume, which is defined by the architecture

$$\text{Vol}_{\text{comp}} = \text{STVol} \times d^3, \tag{39}$$

where STVol is the spacetime volume as given by Eqs. (32) and (33). Putting the last three equations together with Eqs. (32) and (33), we can calculate the total time for both the baseline and active volume architectures as

$$t_{\text{comp}} = \frac{l_{\text{delay}}\, d}{c_{\text{fiber}}} \times \begin{cases} n_T & \text{for BL architectures,} \\ b_{\text{AV}}/w & \text{for AV architectures.} \end{cases} \tag{40}$$

Although these equations may seem remarkably similar, we will see in the next section that optimizing these two equations to reduce the runtime can be quite different.

B. Architecture optimization

Equations (37) and (38) show a linear spacetime trade-off between the number of IMs and the physical runtime in the baseline architecture. If all the interleaving modules use a shorter delay line length, Eq. (38) dictates that this will proportionally reduce the runtime of the algorithm. However, in order to run the algorithm we must still have n qubits available to satisfy memory requirements. By Eq. (37) we must increase the number of interleaving modules n_{IM} to make up for the decrease in qubits incurred by decreasing l_{delay}. One can see this as an example of a spacetime trade-off where we can decrease the time of the computation by increasing the size of the computer. This spacetime trade-off is a feature of FBQC and is referred to as *interleaving* [41].

In the active volume architecture, we have access to another trade-off worth considering: trading code distance for the number of workspace qubits. If we can decrease the distance and increase the number of logical qubits, those new logical qubits can be added to the workspace and reduce the number of logical cycles.

To demonstrate this trade-off mathematically, let us explicitly calculate the runtime for the case of the AV architecture by combining Eqs. (37) and (40). We start with a quick rearrangement of Eq. (40) and then plug in Eq. (37) for n:

$$\begin{aligned} t_{\text{comp}}^{\text{AV}} &= \frac{l_{\text{delay}}}{c_{\text{fiber}}} \frac{b_{\text{AV}}\, d}{n - m} \\ &= \frac{b_{\text{AV}}\, d^3}{n_{\text{IM}} r_{\text{IM}} - c_{\text{fiber}} m d^2 / l_{\text{delay}}}. \end{aligned} \tag{41}$$

Note that decreasing d will decrease the computation time and, in contrast to the baseline architecture, increasing l_{delay} also decreases the computation time. Lowering the distance decreases the time because it lowers the time it takes to complete a logical cycle. Increasing l_{delay} decreases the computation time because the qubits made available by increasing the delay length can be used to execute logical blocks. However, one cannot simply reduce the distance of the code to achieve an arbitrarily low runtime. The distance has to be kept large enough such that the computation does not incur an error. We derive this minimal distance d_{min}.

It is common to model the probability of an error in a computation using an exponential model [18,55]:

$$\text{probability of failure per logical block} = 10^{-\alpha d/2}. \quad (42)$$

The parameter α is defined by

$$\alpha = \log_{10}\left(\frac{p_{\text{phys}}}{p_{\text{thresh}}}\right), \quad (43)$$

where p_{phys} is the physical error rate and p_{thresh} is the threshold error rate of the quantum error-correction code. To account for varying levels of hardware performance affecting p_{phys}, in the analysis in Sec. V below we report numbers for both $\alpha = 1$ and $\alpha = 1/2$. We expect that $\alpha = 1/2$ is a more realistic estimate for an early FTQC device. However, previous literature [18] has used $\alpha = 1$ for their resource estimates. However, since current developments [55] are at the level of $\alpha = 0.33$, we regard the $\alpha = 1$ case as optimistic and treat the case $\alpha = 0.5$ as a realistic alternative.

We can use this model to relate the distance d to the number of logical blocks that can be executed:

$$\begin{aligned} p_f &= 1 - (1 - 10^{-\alpha d/2})^{\text{STVol}} \\ &\approx \text{STVol} \times 10^{-\alpha d/2} \quad (44) \\ \text{STVol} &\approx p_f\, 10^{\alpha d/2}, \quad (45) \end{aligned}$$

where p_f is the probability of failure for the overall algorithm and STVol is the total spacetime volume of the computation in logical blocks. The above equation is valid for both the active volume and circuit volume: STVol can be identified with STVol_{AV} or STVol_{BL}. We can directly relate this to the active volume by using Eq. (33):

$$b_{\text{AV}} \approx \frac{w}{n} p_f\, 10^{\alpha d/2}. \quad (46)$$

Finally, let us use the total number of qubits $n = w + m$ and Eq. (37) to give

$$b_{\text{AV}} \approx \left(1 - m\frac{d^2 c_{\text{fiber}}}{n_{\text{IM}} r_{\text{IM}} l_{\text{delay}}}\right) p_f\, 10^{\alpha d/2}. \quad (47)$$

The above equation tells us the maximum number of logical blocks b_{AV} we can execute with m memory qubits and a total probability of failure p_f. We can then define d_{min} to be the minimal distance, which satisfies the above relation. There is no closed-form solution for d_{min}, however, since d_{min} will almost certainly be less than 100; it can be found numerically. By inserting this d_{min} into Eq. (41) we obtain an explicit relation for the minimal computational time.

V. RESULTS

In this section, we present logical and physical resource estimates, as well as runtimes of the electronic structure calculation for the P450 cytochrome molecular benchmark system. As in all former studies, we take the heme group from the active site of P450 as a model system for the enzymatically relevant part of the protein and reduce the number of orbitals in the Hamiltonian by selecting a (63e, 58o) active space model in the high-spin ($S = 5/2$) electron configuration [18,57]. In Sec. V A, we discuss the BLISS-THC factorization of P450 [58], obtaining not just the new 1-norm $\tilde{\lambda}_{\text{THC}}$ that gives us an immediate speedup over THC, but also a suitable BLISS-THC rank M as well as the bit precision parameters $\aleph$ and $\beth$. Those numbers are the basis for the circuit compilation in Sec. V B, where we present logical resource estimates along with the combined speedup for BLISS THC, AV compilation, and circuit modifications. In Sec. V C, we estimate wall-clock runtimes using various amounts of interleaving. Because of optimizations of the code distance, the speedup is increased even more. We conclude in Sec. V C with a discussion about the minimal number of IMs for a target runtime. A final attempt to reach a larger speedup is made in Sec. V D, where we discuss how loading angles in batches within the block encoding influences the runtime of the entire calculation.

The analysis that we have subjected P450 to can of course also be applied to other molecular systems. The interested reader may find results on the speedup for the electronic structure calculations of FeMoco in Appendix A.

A. BLISS-THC factorization

Just like other factorization schemes, BLISS THC requires a user-specified error threshold to determine the optimal rank and bit precision requirements for the coefficients $\tilde{\zeta}_{\mu\nu}$ as well as the rotation angles associated with the unitary implementation of χ_p^{μ} [18,26]. However, practical reasons demand that the optimal two-index tensors $\tilde{\zeta}_{\mu\nu}$ and χ_p^{μ} must be found to arbitrary precision first. To that end, we perform a numerical optimization, for which we propose the BLISS-THC cost function

$$\begin{aligned} C(\alpha, \beta, \chi, \tilde{\zeta}) = \frac{1}{2}\sum_{pqrs}\left[g^{(\text{BI})}_{pqrs}(\alpha,\beta) - \sum_{\mu\nu}\tilde{\zeta}_{\mu\nu}\chi_p^{\mu}\chi_q^{\mu}\chi_r^{\nu}\chi_s^{\nu}\right]^2 \\ + \rho\left(\sum_k |\tilde{t}_k| + \frac{1}{2}\sum_{\mu\nu}|\tilde{\zeta}_{\mu\nu}| - \frac{1}{4}\sum_{\mu}|\tilde{\zeta}_{\mu\mu}|\right) \end{aligned} \quad (48)$$

as a modification to the regularized cost function of Goings *et al.* [18], which includes prior knowledge of the computational cost of quantum phase estimation via the 1-norm weighted by the regularizer, ρ. While the original cost

function was only optimized for the tensors $\zeta_{\mu\nu}$ and χ_p^{μ}, we include here the symmetry-shift and BLISS coefficients, denoted α and β, respectively. The numerical optimization of the THC tensors attempts to find an optimal trade-off between minimizing the ℓ_2-norm with the smallest THC rank M while also minimizing the 1-norm. Unlike previous work, we explicitly include the one-body contribution to the 1-norm in the regularization term. Fundamentally, the need for this inclusion arises from the BLISS coefficients β_{pq}, which affect one-body and two-body Hamiltonian operators. Since this is BLISS THC, the conventional four-index tensor g_{pqrs} is replaced with the block-invariant tensor $g^{(\mathrm{BI})}_{pqrs}(\alpha,\beta)$ that is implicitly defined with respect to coefficients α and β. Similar to prior THC workflows [18], our optimization procedure starts with a symmetric canonical polyadic decomposition of the Cholesky vectors to provide the initial guess, which takes around 6 min running on a CPU. This is followed by a regularized optimization of the BLISS-THC tensors, carried out with 15 000 limited-memory Broyden-Fletcher-Goldfarb-Shanno bound-constrained iterations (L-BFGS-B) iterations on a single Nvidia GeForce RTX 4090 consumer-level GPU, which takes another 6 min.

The cost function (48) evaluates the ℓ_2-norm error of the truncated BLISS-THC Hamiltonian (13) with respect to the exact Hamiltonian (1). While this is a suitable error metric for the optimization procedure, it does not provide a reliable proxy to the correlation energy error. We therefore estimate the truncation error $\epsilon_{\mathrm{trunc.}}$ via coupled-cluster singles and doubles with perturbative triples [CCSD(T)] calculations based on the reconstructed one-electron and two-electron integrals using PySCF [59]. From the tabulated results, we choose the final BLISS-THC rank M, satisfying the user-specified threshold $\epsilon_{\mathrm{trunc.}} \leq 0.3$ mE_{h}, similar to Ref. [26]. An expanded discussion of these salient points is provided in Appendix B.

We present the analysis of the THC rank with respect to CCSD(T) values for $\epsilon_{\mathrm{trunc.}}$ in Table III. Notably, we observe an increase of the 1-norm as a function of the THC rank while still maintaining a submillihartree CCSD(T) correlation energy error for nearly all cases. These results establish that the lowest possible rank is $M = 160$, which meets the target error threshold of 0.3 mE_{h}.

With the rank now fixed, we set out to find the number of bits for the precision of the alias sampling state preparation and Givens rotations, $\aleph$ and $\beth$. In Fig. 6, we present a two-dimensional (2D) heat map of the CCSD(T) correlation energy error defined with respect to $\aleph$ and $\beth$. Similar to prior art, we observe a nontrivial oscillatory behavior in the CCSD(T) correlation energy error. To adhere to the error threshold of 0.3 mE_{h}, we select the bit precision parameters $(\aleph, \beth) = (13, 13)$. This choice is confirmed by the density matrix renormalization group (DMRG) results, where the error to the correlation energy introduced by the BLISS-THC procedure is 0.08 mE_{h} for a bond dimension of 1500. For details on the DMRG results, we refer the reader to Appendix C.

TABLE III. BLISS-THC results for the factorization rank M, the ℓ_2-norm of the difference tensor $g_{pqrs} - \widetilde{g}_{pqrs}$ (where $\widetilde{g}_{pqrs}$ is the version of g_{pqrs} reconstructed from the factorization), the CCSD(T) error, and the BLISS-THC 1-norm $\tilde{\lambda}_{\mathrm{THC}}$. We ultimately choose the THC rank $M = 160$ (highlighted blue) due its CCSD(T) error being below the threshold of 0.3 mE_{h}. Recall that Goings *et al.* [18] found a rank of $M = 320$, which is exactly twice the rank for BLISS THC, as well as a 1-norm of 388.9 E_{h}, roughly 3 times as high as ours.

Rank M	ℓ_2-error (E_{h})	CCSD(T) error (mE_{h})	1-norm (E_{h})
120	0.309	−2.14	132.3
140	0.219	−1.15	132.5
160	0.202	0.25	130.9
180	0.122	0.11	134.4
200	0.106	−0.03	133.3
220	0.065	0.32	139.3
240	0.054	0.25	137.4
260	0.042	0.29	138.1
280	0.033	0.26	138.9
300	0.032	−0.04	138.9
320	0.024	0.28	139.1

B. From factorization to active volume, circuit volume, and speedups

With all parameters of the factorization fixed, we can now compile the quantum phase estimation routine central to this electronic structure calculation, while neglecting any overhead that might be incurred in the initial state preparation, as well as repetitions of the entire algorithm to increase the statistical confidence of the results. The computational volume of the routine, as well as its qubit *highwater* [60], will critically depend on which Select circuit we pick, and we have several versions to choose from: the Select developed in Sec. III differs somewhat from prior art [26] due to our use of fused adders, tighter analysis for the bit precision, and the treatment of diagonal terms in the THC matrix. Also, the possibility of batching offers a number of variations on each version of Select. For now, let us focus on versions of the circuit where all angles are loaded in a single batch: the Select with our modifications as well as the literature version for comparison. To correct some inaccuracies in the logical resource counts of the literature reference, we have re-estimated the resources of the THC algorithm with respect to the parameters $\aleph = 10$, $\beth = 18$ [61], $\lambda = 388.9\ E_{\mathrm{h}}$, and $M = 320$ from Ref. [18].

From only their logical resource estimates, we can derive relative speedups between two quantum calculations without explicitly computing their respective runtimes. For the total runtime speedup, taking into account the effects of circuit modifications, BLISS THC, and AV

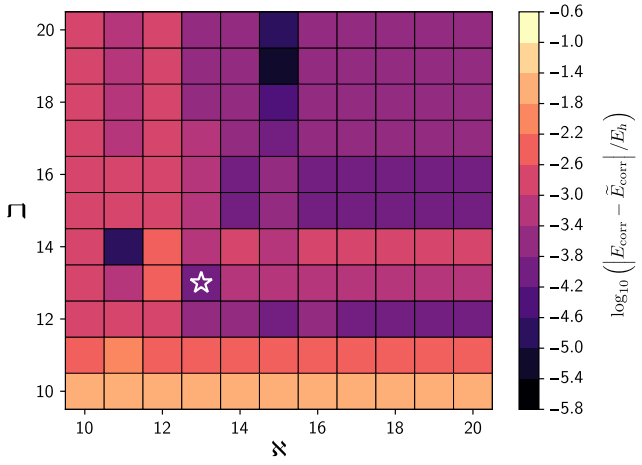


FIG. 6. Heat map of the error between the CCSD(T) correlation energy calculated with the uncompressed Hamiltonian and the BLISS-THC compressed Hamiltonian as a function of fixed-point precisions ℵ and ⊐. The star indicates the combination $(ℵ, ⊐) = (13, 13)$ selected for the BLISS THC of P450. We have also performed a DMRG calculation with bond dimension 1500 for the tuples $(ℵ, ⊐) = (13, 13)$ and $(ℵ, ⊐) = (11, 14)$, confirming errors in the correlation energy of 0.08 and 0.10 mE_h, respectively. For comparison, Goings *et al.* [18] found a tuple of $(ℵ, ⊐) = (10, 18)$ with respect to our definition of ⊐; however, we believe that their report has neglected the preparation of one-body terms, which would have changed the landscape of both ℵ and ⊐.

compilation, the following two quantum programs are relevant: computation 1 is our version of the algorithm run on an AV architecture using BLISS THC, and computation 0 is the literature version of the electronic structure calculation for reference. Let us say that we run both computations on devices with the same footprint, which, for now, will mean identical code distances, interleaving lengths, and qubit counts. The latter would pose a condition to be enforced. Following Eq. (40), the speedup $t_{\mathrm{comp},0}/t_{\mathrm{comp},1}$ with respect to the individual runtimes $t_{\mathrm{comp},i}$ of computations $i = 0, 1$ is

$$\frac{t_{\mathrm{comp},0}}{t_{\mathrm{comp},1}} = \frac{n_{T,0}\, w_1}{b_{\mathrm{AV},1}}, \tag{49}$$

where $b_{\mathrm{AV},1}$ and w_1 are the AV count and workspace size of computation 1, and $n_{T,0}$ is the T count of computation 0. Because of being run in a baseline architecture, computation 0 has a 1:1 ratio between the sizes of the workspace and memory, while the number of workspace qubits can be freely chosen for computation 1. Through circuit modifications and BLISS, the memory of computation 1 will be somewhat relaxed, and so we reassign the freed-up memory qubits to the workspace, such that the sum of memory and workspace qubits remains constant for both computations. To obtain a lower bound on the runtime speedup, we will not reassign the distillation qubits from computation 0 in computation 1. With $w_1 = 2m_0 - m_1$, where m_0 and m_1 are the memory qubit highwaters of computation 0 and 1, respectively, the speedup in Eq. (49) becomes

$$\frac{t_{\mathrm{comp},0}}{t_{\mathrm{comp},1}} = \left(2 - \frac{m_1}{m_0}\right)\frac{b_{\mathrm{cv},0}}{b_{\mathrm{AV},1}}, \tag{50}$$

where $b_{\mathrm{cv},0}$ is the circuit volume of computation 0. While the circuit volume is easily accounted for, we must obtain an upper bound on the total active volume by adding the AV costs of the algorithm's subcomponents. A repository for the active volume costs of all subroutines relevant to this case can be found in Table 1 of Ref. [39]. We present the results of our logical resource estimations for computations featuring THC and BLISS-THC Hamiltonians and with and without our modifications to Select in Table IV. Substituting the highlighted values for the circuit volume and active volume into Eq. (50), it becomes apparent that we have achieved a total speedup of 233.9× over the runtime of the THC algorithm on a baseline architecture. While our modifications to the Select circuit have a small, yet positive effect on the AV, it is not the only way in which they reduce the runtime. Through the modifications, Select also calls fewer qubits, making room for a larger workspace. This reduction in the memory is due to encoding each of the 57 Givens rotation angles using ⊐ rather than ⊐ + 1 qubits. An even larger reduction in memory is achieved by switching from THC to the BLISS-THC Hamiltonian, where ⊐ is 13 rather than 18. While ⊐ has the biggest impact on the global qubit highwater, the Prepare circuit allocates roughly $M\sqrt{ℵ/2 + \log M}$ qubits for a QROAM dataloader in alias sampling. This local qubit highwater is almost on par with that of Select.

Before we move on to runtimes, we present a breakdown of AV costs between the algorithmic subroutines. The call graph in Fig. 7 provides insight into the relative cost between the different parts of a single BLISS-THC block encoding. In computation 1 40% of the total AV is spent on Prepare and 10% on Prepare†, while the Select routine accounts for the remaining 50%. Inverse routines are usually cheaper, as they tend to feature measurement-based uncomputation, such as in right elbows and inverse dataloaders. It is worth noting that AV compilation shifts the relative costs of some subroutines: in previous cost metrics, the adders associated with the Givens rotations account for roughly $4N⊐$ Toffoli gates in a single instance of Select, which dwarves the roughly $2M + N$ Toffoli gates in the dataloaders by an order of magnitude. However, the AV costs of adders and dataloaders are quite similar. This is likely due to the cost associated with the Clifford gates: writing and overwriting $(2M + N)(N/2 - 1)$ different angles of ⊐ bits into temporary quantum registers requires many conditional bit flips.

TABLE IV. Toffoli count, memory qubit highwater, the circuit volume, as well as the active volume for the electronic structure calculation of P450 with respect to the THC Hamiltonian of Ref. [26] and the BLISS-THC Hamiltonian developed in this work. For both Hamiltonians, we offer the choice between the Select circuit of Ref. [26] and our modified Select in Fig. 1. In this table, we have highlighted the circuit volume $b_{\mathrm{cv},0}$ with respect to the prior-art Select [26] and the THC Hamiltonian of Goings *et al.* [18] in blue, and $b_{\mathrm{AV},1}$, the value for the active volume of our modified Select using the BLISS-THC Hamiltonian, in orange. The ratio $b_{\mathrm{cv},0}/b_{\mathrm{AV},1}$ is important for the computation of the relative speedup factor in Eq. (50). Note that we have effectively re-estimated the logical resources for the circuit of Goings *et al.* with respect to their reported rank, 1-norm, and precision parameters.

Hamiltonian	Circuit mods	Toffoli gates	Memory	Circuit volume	Active volume
THC	No	7.786×10^9	1357	42.264×10^{12}	1.678×10^{12}
	Yes	7.647×10^9	1298	39.703×10^{12}	1.594×10^{12}
BLISS THC	No	1.761×10^9	1058	7.450×10^{12}	0.249×10^{12}
	Yes	1.714×10^9	999	6.848×10^{12}	0.228×10^{12}

C. From computational volume to runtimes and device footprint

For Eq. (50), we have assumed the code distances of both computations to be the same, but we can do better: due to its lower computational volume, the AV-based BLISS-THC computation requires a lower code distance. While this would give the runtime an additional boost according to Eq. (40), lowering the distance of every logical qubit would also reduce the required footprint. Following Sec. IV B, we minimize the code distance and compute the runtimes after assigning freed-up resources to the workspace. In doing so, we can ensure a direct comparison between the two different computations, by keeping the number of IMs fixed. Note that the runtime improvement

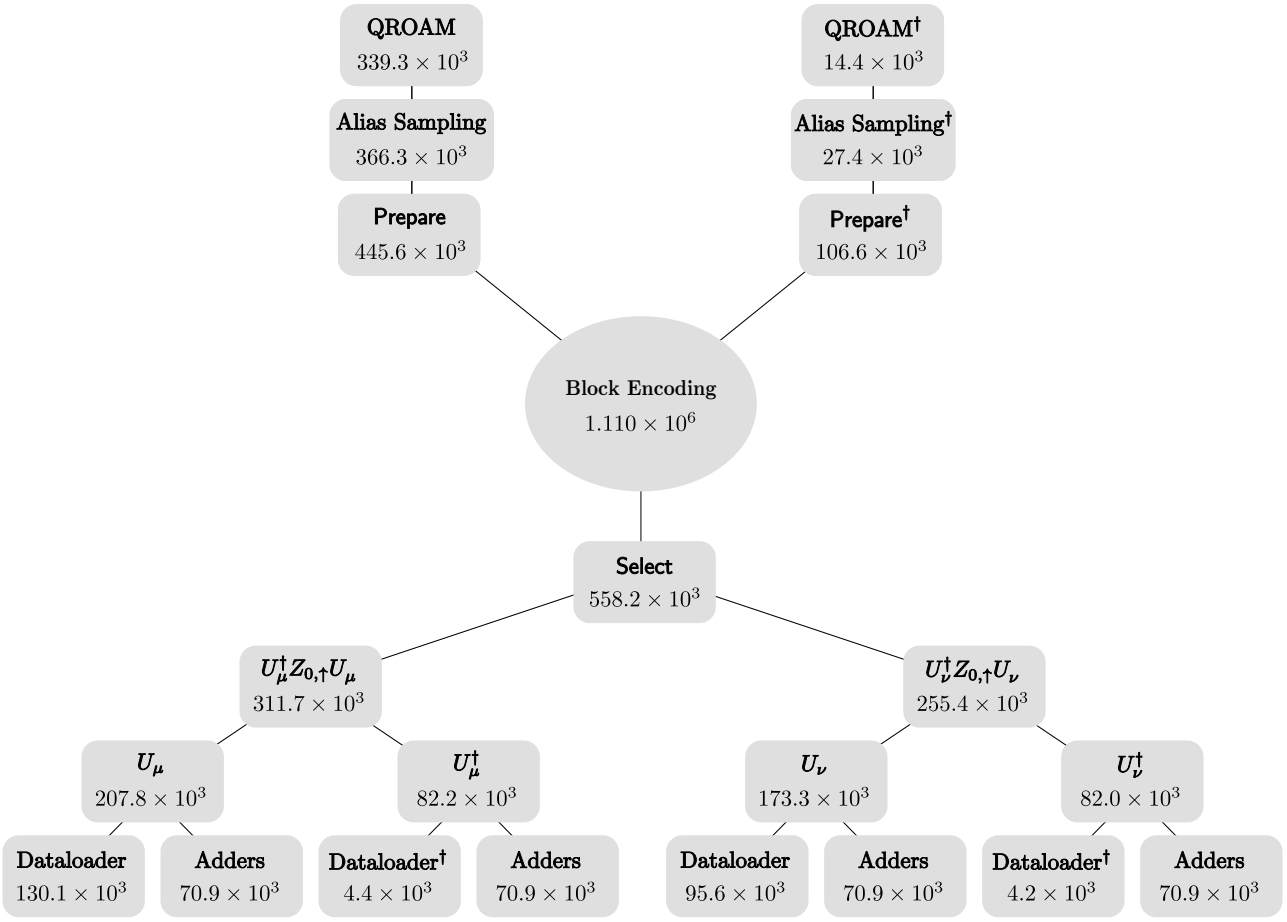


FIG. 7. A call graph showing the AV for a hierarchy of relevant subroutines emanating from the BLISS-THC block encoding of P450 (center). The subroutines labeled $U_\mu^\dagger Z_{0,\uparrow} U_\mu$ and $U_\nu^\dagger Z_{0,\uparrow} U_\nu$ are versions of subselect, multiplexing over indices μ and ν, where the former also multiplexes over one-body operators. The basis rotations U_μ and U_ν include Givens rotations and dataloaders.

TABLE V. Numbers of interleaving modules, code distances, and runtimes for the electronic structure calculation of P450 as a function of the delay length (l_{delay}) as a tunable parameter for interleaving. The table considers two computations: (1) the BLISS-THC calculation with circuit modifications run on an AV architecture and (2) the THC calculation without circuit modifications run on a BL architecture. For a direct comparison, the number of IMs for the AV calculation is matched to the IM requirements of the BL run, and code distances are computed for a logical error probability of $\leq$10% with respect to the whole algorithm. Additionally, we distinguish between the cases $\alpha = 0.5$ and $\alpha = 1$. Runtimes are presented in the format hours:minutes:seconds.

			1-m delay		10-m delay		100-m delay		1000-m delay	
Algorithm (architecture)	α	d	IMs	Time	IMs	Time	IMs	Time	IMs	Time
THC (BL)	0.5	60	1 954 080	2:35:44	195 408	25:57:16	19 541	259:32:38	1955	2595:26:18
BLISS THC (AV)	0.5	50	1 954 080	0:00:20	195 408	0:03:17	19 541	0:32:42	1955	5:26:47
THC (BL)	1	30	488 520	1:17:52	48 852	12:58:38	4886	129:46:19	489	1297:43:09
BLISS THC (AV)	1	25	488 520	0:00:10	48 852	0:01:39	4886	0:16:21	489	2:43:17

of 233×, obtained in the previous subsection, is retained as a lower bound on the speedup, independent of the physical error rate and the threshold. Both of these parameters are captured in the error suppression constant α of Eq. (43). In Goings *et al.* [18], runtimes are obtained for $\alpha = 1$, an optimistic assumption for early fault-tolerant quantum computers that we would like to contrast with a more realistic scenario of $\alpha = 0.5$.

In Table V, we finally present runtimes, IM numbers, and code distances for the two relevant computations: the THC algorithm of Ref. [26] run on a baseline architecture, as well as our BLISS-THC algorithm run on an AV architecture.

For both computations, we keep the number of IMs fixed, and distinguish the optimistic scenario of $\alpha = 1$ from the realistic scenario of $\alpha = 0.5$. In all cases, the total wall-clock speedup is roughly 476×. Using the length of the optical fiber to tune the amount of interleaving, footprints are traded off against runtimes for the computations considered. The shorter the delay length, the lower the ratio of resource states per interleaving module, which makes the computation faster, but increases the number of interleaving modules required.

When we want to keep the quantum computer as small as possible in physical size, the delay lengths need to be maximized. To still reduce the runtime, we can then change the ratio of workspace-to-memory qubits: the more qubits we dedicate to the workspace, the more logical blocks can be executed in parallel.

In the following example, we want to compute the minimum number of IMs required to run our BLISS-THC algorithm in a specific time frame. Let us fix the interleaving length to 2 km—a large value at which we start to notice the effects of fiber loss [41]. As the number of memory qubits is fixed by the algorithm, the minimum number of IMs is determined by how many qubits we will need in the workspace. In Table VI, we provide the results for a runtime of 73 h, which is the runtime reported by Goings *et al.* [18] for an algorithm with 4.6×10^6 physical qubits. We also consider the runtime of 1 h as an intermediate milestone on the way to industrial utility. For both timescales, Table VI presents the minimum number of IMs required under error threshold parameters of $\alpha = 1$ and $\alpha = 0.5$, along with updated runtimes for the modified Select operators discussed in the following subsection.

D. Batching

In a final attempt to speed up the computation, we utilize a trade-off between qubit highwater and AV counts within the algorithm. Previously, we had decided to only consider versions of Select that load all Givens rotation angles at once—those circuits are optimal with respect to Toffoli and AV counts, but have a large memory qubit highwater. In fact, the local highwater of Select dominates the qubit requirements of the entire algorithm for P450. Allowing circuits that load the angles in batches, as depicted in Fig. 4, we reduce the qubit requirements of the entire algorithm at the expense of a higher AV. Once the qubits are freed up in memory, they can then be reassigned to the workspace, allowing for a possible runtime speedup, assuming that the AV hike is overcome in the process.

TABLE VI. Number of interleaving modules required for corresponding runtimes of BLISS THC and BLISS-THC-b for four hypothetical devices targeting a 73-h and a 1-h operation of BLISS THC at optimistic ($\alpha = 1$) and realistic ($\alpha = 1/2$) assumptions regarding the error model. The 73-h number is set to match the benchmark of Goings *et al.* [18], where the optimistic assumption of $\alpha = 1$ is used. Here, the delay length is kept as long as possible at 2 km, at which the loss threshold is argued to be only somewhat affected [41]. Note that, for the smallest device, we cannot achieve 73 h directly, so we report the largest runtime below the target.

# IMs	α	BLISS-THC runtime (h)	BLISS-THC-b runtime (h)
8184	0.5	1	1
1055	1	1	1
395	0.5	73	41
88	1	54	26

To avoid confusion, let us refer to any BLISS-THC calculation with a Select using batched dataloaders as a BLISS-THC-b calculation. There is one more modification that we have to make for BLISS-THC-b: in Prepare, the dataloader associated with the alias sampling routine, highlighted blue in Fig. 2, needs to be relaxed. By adjusting the lambda parameter of its LKS construction [46], we can force the QROAM back into a QROM, and reduce the number of qubits allocated by the routine from roughly $M\sqrt{\aleph/2 + \log M}$ to $\aleph + 2\log M$ while approximately doubling its AV. Note that this qubit trade-off always increases the AV of Prepare. However, if we did not make this modification, then reducing the local highwater of Select would at some point have no effect on the global highwater, as the qubits freed up in Select would still be needed in Prepare.

The AV numbers for various instances of BLISS-THC-b with different batch sizes are presented in Fig. 8, which also shows the qubit minimum that a magic-state-optimal QROAM in alias sampling would have imposed.

From this plot, we chose the instance of BLISS-THC-b that minimizes the runtime. Table VI contains not just the number of IMs for targeted runtimes, but also provides the lowest runtimes for the corresponding BLISS-THC-b calculations. For the devices with lower IM numbers, we find that BLISS-THC-b is able to provide a speedup, while for the devices with a larger number of IMs, the runtime is unchanged. For the latter, the optimization defaults to the case where all angles are loaded in a single batch, turning BLISS-THC-b back to BLISS THC as no other number of batches provides a runtime speedup. In the two devices with the smallest IM counts, the optimal runtime is achieved when loading the angles in seven batches.

As an alternative to batching, we have also considered converting Hamiltonian data into a unary representation, and so relax the angle loading. While one would expect this to come with a large qubit highwater, it is only narrowly beaten by batching, as we discuss in Appendix D.

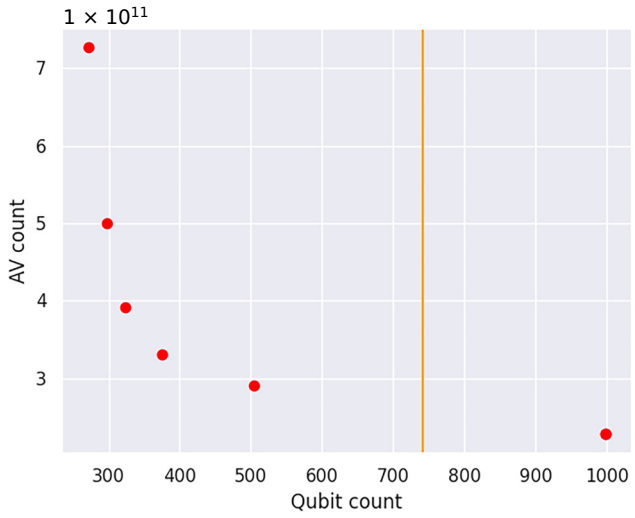


FIG. 8. Active volume and logic qubit count of BLISS-THC-b for various batch sizes. BLISS-THC-b is a version of our algorithm for BLISS THC, where the angles for the programmable Givens rotations are loaded in batches, as indicated in Fig. 4. As an additional modification, we turn the QROAM [46] in Prepare back into a QROM. If we did not do this, we would not be able to access the qubit numbers left of the orange separator, as the reduction in the qubit count of Select would be overtowered by the local qubit highwater of Prepare and all total qubit counts left of the orange line would lie on the line instead. While clearly not optimal with respect to magic state counts, this modification has a smaller impact on the AV, which we can now trade off for the number of logical qubits. The lowest number of qubits achievable is 271, at an AV of approximately 7.26×10^{11} logical blocks for the algorithm in which we load every angle before the corresponding Givens rotation, as displayed in Fig. 4(c). The degree of batching is not made explicit here, but higher qubit numbers correspond to larger batches, and the rightmost point represents the standard BLISS-THC version, where all angles are loaded at once.

VI. CONCLUSION

New algorithms and methods will find application in an industrial setting only if they can deliver accurate results for systems of relevant sizes in time frames compatible with industrial development cycles. This means that to gauge the utility of quantum computing for industrial applications, actual runtimes must be obtained, not just gate complexities or asymptotic scaling analyses. The accurate evaluation of runtimes, however, is a complex task for which many aspects must be considered. First of all, runtime estimates are developed at the focal point of not only quantum algorithms and compilation, but also quantum error correction and quantum system design. Suddenly, features like the code threshold, code distance, and logical error rate come into play, along with essential questions about qubit connectivity, magic state distillation protocols, and more. In this work, the above considerations are taken into account, providing a holistic picture for us to present the latest improvements to the electronic structure calculations of molecular systems on the example of the heme group occurring in cytochrome P450. One such improvement is the introduction of BLISS THC, the new state of the art in a chain of cascading advancements in quantum simulation of molecular systems, and applicable to all quantum architectures. To make BLISS THC numerically attainable, we have drastically driven down the runtime of the classical preprocessing of all flavors of THC. Factorizing a single P450 Hamiltonian with THC currently takes approximately 6 min (plus 6 min of optimization warm-up) on a single Nvidia GeForce RTX 4090 consumer-level GPU. As a result, we were able to subject the P450 Hamiltonian to a much tighter factorization procedure, with BLISS THC not only improving

the Hamiltonian 1-norm, but also the factorization rank of the tensor. Furthermore, we have reduced the classical precompute so much that it has become insignificant compared to prior art [26,28]. This cancels a big disadvantage of cost function–based methods compared to diagonalization-based methods like DF. We believe that BLISS THC, on account of it being a strict improvement over THC in terms of compression, will be beneficial to all kinds of molecular systems, and find applications beyond electronic structure calculations, such as the simulation of time dynamics, or the estimation of molecular observables [17].

The most significant contribution to our speedups comes from switching to AV compilation. This mode of running quantum programs is amenable to only some quantum hardware types, and the fusion-based photonic architecture is one of them. However, any platform that allows for flying qubits or nonlocal physical gates could be amenable to active volume compilation, and the comparison can be made to a reference architecture with immovable qubits or limited connectivity. Accounting for different execution rates of logical blocks, even cross-platform comparisons are possible.

With some small algorithmic changes, BLISS THC, and AV, we manage to improve the runtime of the electronic structure calculation by a factor of 233 with respect to a reference THC algorithm on the same platform.

This speedup is agnostic to savings in the code distance, for the account of which we require additional information about the error model. Optimizing the code distance for fusion-based photonic hardware the speedup would even jump up to 476×.

We have provided runtimes for several sizes of photonic fusion-based quantum computers, where interleaving is used to trade the runtime against the number of IMs. Note however that we cannot increase the number of IMs indefinitely. When considering larger numbers of IMs, the system may encounter additional bottlenecks not examined in this paper, such as the reaction limit [40]. We leave examination of the high-IM configurations to future work. We also show that qubit trade-offs between using a single or more batches in the loading of Hamiltonian data for the BLISS-THC algorithm, which conventionally slow down the computation in the baseline architecture, can sometimes be utilized to speed it up.

Some improvements in this work not only lower the computational volume, but also reduce the qubit highwater. With the goal of reducing the wall-clock runtime, we have repeatedly converted these space savings into savings of time by enlarging the workspace of the AV architecture. Thus, every resource saved other than time has been put back into the device, just so we can have a larger speedup over the algorithms used in prior art. However, perhaps we should also consider device footprints. Many trade-offs in this work could be used to fit the electronic structure calculation on a quantum computer smaller than the baseline architecture required for the reference computation. In that spirit, we have derived minimal device footprints for computations of a fixed duration, by varying the number of workspace qubits. Even larger space savings can be achieved by changing the algorithm: the original THC routine requires a number of auxiliary qubits that exceeds by far the number of qubits necessary to represent the system. While BLISS THC gives us an immediate reduction of 299 qubits for P450, we have discovered that more substantial space reductions of up to a highwater of only 271 qubits increase the AV by only a factor of 3.2. While we have shown how this trade-off could be utilized for runtime reductions, accepting the AV penalty could be reasonable for a smaller quantum computer.

This paper certainly does not mark the last round of runtime improvements for electronic structure computations on a quantum computer. Many promising avenues remain for future speedups. While we have increased the performance of THC with BLISS and reduced the 1-norm of P450 from 389 to 130.9 E_h, we have not achieved the theoretical 1-norm limit of 69.3 E_h [42]. Other Hamiltonian factorization techniques [31,32] might, however, get us closer. Introducing AV is a paradigm shift in algorithmic costing and has led us to change how we think about certain subroutines. In the future, we expect to be able to make profound modifications to quantum circuits that reflect an optimal choice with respect to the new cost model. For instance, the fact that Clifford gates are now a dominant cost in dataloaders might open a whole new optimization space to be explored. In that spirit, we should also consider costing subroutines like dataloaders with respect to real data rather than considering their worst-case counts. In fact, this approach could go far beyond dataloaders: the runtimes of all subroutines might be reduced by making quantum programs more concrete. What is more, we are currently considering upper bounds on the AV of subroutines. A tighter analysis of AV costs would have an immediate impact on projected runtimes without changes to the algorithms.

We are optimistic that the resource estimates in this work will be improved. To this end, we depend on the contributions of many disciplines on all levels of the quantum computing stack. The impact of novel developments in quantum error correction [62–65], for instance, would need to be considered within the cost model. By continuously improving the cost of this use case, it is our aim to bring practical quantum computing closer and to contribute to the path to industrial value.

ACKNOWLEDGMENTS

A.C., C.C., W.P., S.S., and M.S. would like to thank S. Pallister and D. Litinski for insightful discussions on the subject matter of this project, as well as S. Greenaway, S.

Morley-Short, and others for their advice and support with respect to the utilized resource estimation tools. G.L.A., M.D., N.M., R.S., M.S., and C.T. thank Clemens Utschig-Utschig for his comments on the manuscript and support during the project.

DATA AVAILABILITY

The data that support the findings of this article are openly available [58].

APPENDIX A: SPEEDUPS FOR THE ELECTRONIC STRUCTURE CALCULATION OF FeMoCo

The iron-molybdenum cofactor, FeMoco (Fe_7MoS_9C), is the active site of nitrogenase, an enzyme playing a crucial role in the biological reduction of nitrogen to ammonia. This iron-molybdenum cluster presents a significant challenge for accurate theoretical characterization. Its large size, open-shell electronic configuration requiring a multireference description, and the presence of static correlations make conventional computational methods inadequate [66]. For these reasons, FeMoco is a perfect candidate for quantum chemical calculations on quantum computers, becoming a well-known benchmark system in many studies [15,26,27]. In this appendix, we expand the results presented in the main text with an equivalent analysis for the electronic structure calculation of FeMoco, using the Hamiltonian of Li *et al.* [67]. We obtain equivalent versions of Tables I–IV as well as Fig. 6 for FeMoco. In Table VII we report the factorization rank M together with the errors and the 1-norm for the sampling of the eigenspectrum of the FeMoco Hamiltonian. In blue, we highlight the selected factorization that gives the best performance for the target error. Results to the finite-bit analysis of ℵ and ב are presented as a heat map in Fig. 9. Logical resource estimates as well as speedups are presented in Tables VIII and IX, where we compare against the re-estimated resources of two factorized Hamiltonians: the original THC version of Ref. [26] and the partially symmetry-shifted THC* version of Ref. [33], finding a speedup of 427× and 278×, respectively. A brief summary of all previous and current factorization attempts to the Li Hamiltonian are collated in Table X.

APPENDIX B: ERROR METRICS AND THE ROUNDING PROCEDURE FOR APPROXIMATE TENSORS

This appendix describes the algorithm performance model used to choose all algorithm parameters. The overall target accuracy ϵ is taken to be 0.0016 hartree (chemical accuracy) for all of the resource estimates in this manuscript. To achieve this target, we consider the total errors given by the sum of the quantum phase estimation error ϵ_{PEA} and Hamiltonian approximation error ϵ_{THC},

$$\epsilon \geq \epsilon_{\mathrm{PEA}} + \epsilon_{\mathrm{THC}}. \tag{B1}$$

Here, ϵ_{PEA} is the error due to measurement in the phase estimation procedure, including, for instance, spectral leakage effects. In contrast, ϵ_{THC} corresponds to the total approximation and compilation error that arises from

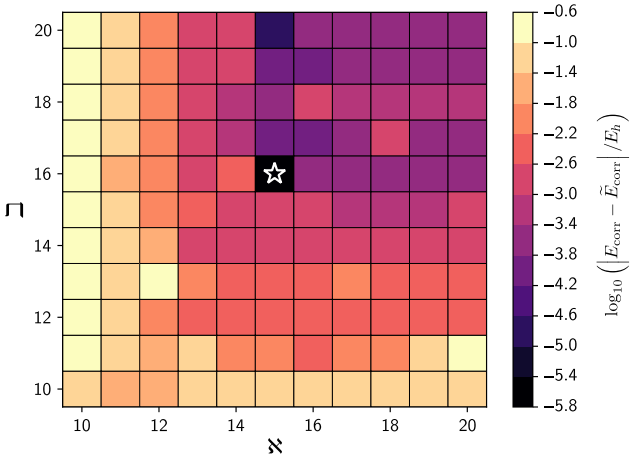


FIG. 9. Heat map of the CCSD(T) correlation energy error calculated with the uncompressed Hamiltonian and the BLISS-THC compressed Hamiltonian as a function of fixed-point precisions ℵ and ב. The star indicates the final selected combination $(ℵ, ב) = (15, 16)$. For comparison, Lee *et al.* [26] found a tuple of $(ℵ, ב) = (10, 18)$ with respect to our definition of ב; however, we believe that their report neglected the preparation of one-body terms, which would have changed the CCSD(T) correlation energy error landscape of both ℵ and ב.

TABLE VII. FeMoco BLISS-THC results as a function of the factorization rank M, ℓ_2-error norm, CCSD(T) correlation energy error, and 1-norm. We ultimately choose the THC rank $M = 290$ (highlighted blue) due its CCSD(T) correlation error being below the threshold of 0.3 mE_{h}. The CCSD(T) calculation uses a high-spin ($S = 35/2$) reference using unrestricted CCSD(T) in PySCF. For reference, the THC rank M reported in Ref. [26] was equal to 450.

Rank M	ℓ_2 error (E_{h})	CCSD(T) error (mE_{h})	1-norm (E_{h})
250	0.146	−0.87	197.9
270	0.112	0.61	198.7
290	0.086	−0.26	198.9
310	0.068	−0.48	198.8
330	0.056	−0.26	199.6
350	0.045	−0.21	198.9
370	0.039	0.09	199.0
390	0.035	0.07	202.3
410	0.030	0.16	200.7
430	0.025	−0.20	200.8
450	0.022	0.21	202.5

TABLE VIII. Toffoli count, number of memory qubits, and the circuit and active volumes for the electronic structure calculation of FeMoco with respect to the Select circuit of Ref. [26] and our modified Select in Fig. 1, as well as the choice between the THC Hamiltonian of Ref. [26], the THC* Hamiltonian of Ref. [33], and the novel BLISS-THC Hamiltonian. Note that we have effectively re-estimated the circuit of Ref. [26] with respect to the reported ranks, 1-norms, and precision parameters reported in Refs. [26,33].

Hamiltonian	Circuit mods	Toffoli gates	Memory	Circuit volume	Active volume
THC	No	31.767×10^9	2163	274.850×10^{12}	8.762×10^{12}
	Yes	31.201×10^9	2163	269.951×10^{12}	8.392×10^{12}
THC*	No	20.671×10^9	2163	178.841×10^{12}	5.701×10^{12}
	Yes	20.302×10^9	2163	175.654×10^{12}	5.460×10^{12}
BLISS THC	No	4.375×10^9	1589	27.809×10^{12}	0.888×10^{12}
	Yes	4.282×10^9	1512	25.895×10^{12}	0.837×10^{12}

directly implementing the THC Hamiltonian on the quantum computer. Note that this error may be bounded as

$$\epsilon_{\mathrm{THC}} \leq \epsilon_{\mathrm{trunc.}} + \epsilon_{\mathrm{coeff}} + \epsilon_{\mathrm{rot}}, \tag{B2}$$

where $\epsilon_{\mathrm{trunc.}}$ is the truncation error based on the BLISS-THC factorization procedure with a maximum tensor rank, $\epsilon_{\mathrm{coeff}}$ is the error that arises from coherent alias sampling based on implementing the BLISS-THC Hamiltonian coefficients $\tilde{\zeta}_{\mu\nu}$ with a finite-bit representation, and ϵ_{rot} is the approximation error that arises from implementing the individual Givens rotations needed for χ_p^{μ}.

While strict analytical bounds on the truncation error, coefficient error, and rotation error may be used to estimate ϵ_{THC}, these bounds are often quite loose. To circumvent this, we use the procedure that was initially proposed by Lee *et al.* [26], where we reconsider the purpose of the quantum phase estimation algorithm as providing an energy estimate of the *correlation* energy, E_{corr}, defined by the standard ground-state energy E_{G} decomposition, $E_{\mathrm{G}} = E_{\mathrm{HF}} + E_{\mathrm{corr}}$, where E_{HF} is the Hartree-Fock energy, which can be computed with floating-point (64-bit) precision classically. This is quite different from the standard assumption where QPE is considered to provide an estimate of the ground-state energy E_{G} with absolute accuracy. Ultimately, this change in mindset relaxes the stringent requirements for the THC rank or bits of precision that would have been necessary otherwise.

The main challenge with this approach lies in having a reliable assessment of the correlation energy error, which now becomes the proper error metric for ϵ_{THC}. In previous work, Lee *et al.* [26] proposed estimating this error with two separate CCSD(T) calculations. The proposed use of CCSD(T) is due to its broadly recognized status as the "gold standard" for computational chemistry. It provides sufficient scalability for large system sizes while preserving size extensivity. In this work, we advocate for CCSD(T) as a scalable approach for providing correlation energy error estimates for the BLISS-THC tensors. Furthermore, we provide numerical validation of the CCSD(T) correlation energy error metric based on DMRG calculations, which we discuss in the results section in greater detail.

TABLE IX. Contributions to the speedup of the electronic structure calculation for the Li *et al.* Hamiltonian of FeMoco [67], as compared to both the THC-based calculation [26] and the symmetry-shifted THC* [33], on a baseline architecture defined in Sec. IV. The three steps, (1) active volume compilation, (2) the incorporation of BLISS, and (3) improvements to the block-encoding circuit, are done in sequence while adding logical qubits saved in the memory to the workspace. The partial speedups reported are improvements to the runtime with respect to the previous step. Multiplying the individual improvements hence results in the total speedup, up to rounding errors: all numbers in this table are rounded down, but the total speedup is computed with the exact estimates.

	Speedup with respect to	
Method	Ref. [26]	Ref. [33]
AV compilation	31.36×	31.36×
THC ↦ BLISS THC	12.48×	8.12×
Circuit improvements	1.09×	1.09×
Total	427.42×	278.11×

1. Numerical evaluation of error metrics

In the following, we provide explicit details on how the BLISS-THC Hamiltonian error metric is computed, considering both rank truncation and bit precision effects. As mentioned above, we first bound the BLISS-THC Hamiltonian approximation error by the absolute difference between two CCSD(T) calculations,

$$\epsilon_{\mathrm{THC}} \geq |E_{\mathrm{CCSD(T)}}(\mathrm{exact}) - E_{\mathrm{CCSD(T)}}(M, \aleph, \beth)|, \tag{B3}$$

where $E_{\mathrm{CCSD(T)}}(\mathrm{exact})$ corresponds to the correlation energy estimate based on an exact implementation of the one-body and two-body integrals with full floating-point (64-bit) precision, while $E_{\mathrm{CCSD(T)}}(M, \aleph, \beth)$ corresponds to the correlation energy estimate based on the reconstructed one-body and two-body integrals. The one-body integrals are reconstructed based on the standard eigendecomposition of the T_{pq} matrix, while the two-body integrals are

TABLE X. Logical qubit requirements, Toffoli counts, active volume, as well as the 1-norm for electronic structure calculations of FeMoco in a (113e, 76o) active space configuration defined with respect to sparse, symmetry-shifted double factorization (DF*), THC, symmetry-shifted tensor hypercontraction (THC*), and BLISS THC. The Toffoli count of the symmetry-shifted versions of DF and THC (denoted DF* and THC*) account for the qubitization block-encoding cost multiplied by the total number of repetitions required for the QPE algorithm, taken from Table II of Ref. [33]. Note that BLISS THC approaches 108.9 E_h, the theoretical 1-norm limit for FeMoco [42], also to within a factor of 2.

	FeMoco (113e, 76o)			
Factorization	Logical qubits	Toffoli gates	Active volume	λ (E_h)
Sparse [26]	2489	$4,4 \times 10^{10}$	–	1547.3
DF* [33]	6402	3.2×10^{10}	–	582.4
THC [26]	2196	3.2×10^{10}	–	1201.5
THC* [33]	2194	2.1×10^{10}	–	781.8
THC [re-estimated for this work]	2163	3.2×10^{10}	8.8×10^{12}	1201.5
THC* [re-estimated for this work]	2163	2.1×10^{10}	5.7×10^{12}	781.8
BLISS THC [this work]	1512	4.3×10^{9}	8.4×10^{11}	198.9

reconstructed based on the BLISS-THC expansion with finite rank M. Finite bit precision requirements are also imposed onto coefficients t_k, $\widehat{\zeta}_{\mu\nu}$ and the rotation angles needed to implement the unitary implementation of χ_p^μ and eigenvectors of T_{pq}. The parameters $\aleph$ and $\beth$ indicate the finite bit precision parameters. Explicitly, the rounding procedure for $\tilde{\zeta}$ is given by

$$\tilde{\tilde{\zeta}}_{\mu\nu} = \begin{cases} u_\zeta^0 \times \text{round}(\tilde{\zeta}_{\mu\nu}/u_\zeta^0 + x) & \text{for } \mu \neq \nu, \\ u_\zeta^d \times \text{round}(\tilde{\zeta}_{\mu\nu}/u_\zeta^d + x) & \text{for } \mu = \nu, \end{cases} \tag{B4}$$

where $u_\zeta^0 = \tilde{\lambda}_{\text{THC}}/(\mathfrak{d}2^\aleph)$ and $u_\zeta^d = \tilde{\lambda}_{\text{THC}}/(\mathfrak{d}2^{\aleph-2})$ with $\mathfrak{d} = M(M+1)/2 + N/2$, and x is a small constant that is used to ensure the proper normalization $\widehat{\zeta}_{\mu\nu}$ [26]. In this work, we parameterize the rotation vector $\vec{\chi}^\mu$ by the set of angles $\{\theta_{\mu p}\}_{p=0}^{N/2-2}$ defined as

$$\chi_p^\mu[\vec{\theta}_\mu] = \begin{cases} \cos(2\pi\theta_{\mu p}) \prod_{q<p} \sin(2\pi\theta_{\mu q}) & \text{for } p < N/2-1, \\ \prod_{q=0}^{N/2-2} \sin(2\pi\theta_{\mu q}) & \text{for } p = N/2-1, \end{cases} \tag{B5}$$

with

$$\widetilde{\theta}_{\mu p} = u_\theta \times \text{round}(\theta_{\mu p}/u_\theta), \tag{B6}$$

where $u_\theta = 1/2^{\beth+1}$, meaning that the angles are defined with $\beth + 1$ bits of precision. However, we can always set the most significant bit to zero, i.e., restrict the angles to $[0, \pi)$. This parametrization corresponds to the standard Euclidean space mapping of a multidimensional spherical coordinate system. The absolute value of all χ_p^μ can be set with angles in the range $[0, \pi/2]$ where all sines and cosines are positive. The larger range $[0, \pi]$ allows cosines to be negative, and so all χ_p^μ for $p < N/2-1$ may acquire a sign. The only coordinate that would need the full range $[0, 2\pi)$ to be negative is $\chi_{N/2-1}^\mu$, and as we can remove a global minus sign, we can always choose $\chi_{N/2-1}^\mu$ to be positive. With a similar argument, we can exclude the point π in the range $[0, \pi)$. This parametrization is a departure from the definitions in prior art [26,27]. Because of that and another factor of 2 from Eq. (49) of Ref. [26], our definition of $\beth$ differs from that of Lee *et al.* [26] by $\beth_{\text{Lee}} = \beth_{\text{here}} + 2$. Equation (B5) can then be used to build an approximate representation of the one-body and two-body integrals, and the correlation energy $E_{\text{CCSD(T)}}(M, \aleph, \beth)$ can be calculated from these reconstructed integrals.

APPENDIX C: DMRG RESULTS FOR P450

To validate the BLISS-THC truncation parameters, DMRG calculations were performed using Block2 based on the reconstructed one-body and two-body integrals

TABLE XI. Breakdown of the ℓ_2 error, restricted open-shell Hartree-Fock (ROHF) energy, CCSD(T) correlation energy, and DMRG correlation energy for reference P450 active space integrals as well as reconstructed integrals based on $(M, \aleph, \beth) = (160, 11, 14)$ and $(M, \aleph, \beth) = (160, 13, 13)$ truncation parameter settings.

Hamiltonian	ℓ_2 error (E_h)	E_{ROHF} (E_h)	$E_{\text{CCSD(T)}} - E_{\text{ROHF}}$ (E_h)	$E_{\text{DMRG}} - E_{\text{ROHF}}$ (E_h)
Reference	0	−419.408 07	−0.448 64	−0.450 03
$M = 160$, $(\aleph, \beth) = (11, 14)$	0.207	−418.640 88	−0.448 66	−0.449 93
$M = 160$, $(\aleph, \beth) = (13, 13)$	0.202	−419.030 84	−0.448 72	−0.449 95

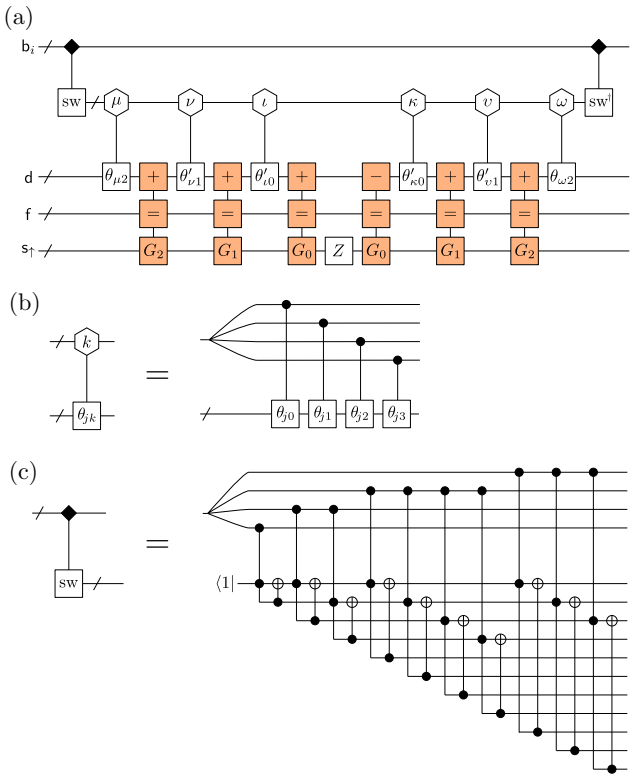


FIG. 10. Unary encoding in the Select circuit. (a) Selecting the operators $U_\mu Z_0 U_\mu^\dagger$ with a unary circuit encoding index μ. This depiction uses the maximum amount of batching; here primed angles θ'_{jk} are chosen such that they write that data θ_{jk} into a register already dirtied with the values $|\theta_{j\,(k-1)}\rangle$. At the beginning, the binary numbers held in registers b_i are converted into unary numbers. Register d is an additional resource, storing the angle. The gates used in this circuit are outlined in the next panels. (b) Unary version of a multiplexor. Here the data strings θ_{jk} are written into the lower register when the kth qubits of the upper register are set. This circuit is composed of only Clifford gates. (c) Out-of-place binary-to-unary conversion using left elbows. This conversion performs the operation $|k\rangle|0\rangle \mapsto |k\rangle|2^{k+1}\rangle$, meaning that the $(k+1)$th qubit of the second register is in $|1\rangle$, while the others are in $|0\rangle$. This circuit is very similar to an inverse QRAM: the topmost qubit has been set to $|1\rangle$, and conditional swap operations move it to the respective position. Using the fact that the lower register starts in the collective $|0\rangle$ state, one can replace the conditional swaps with left elbows and Clifford gates.

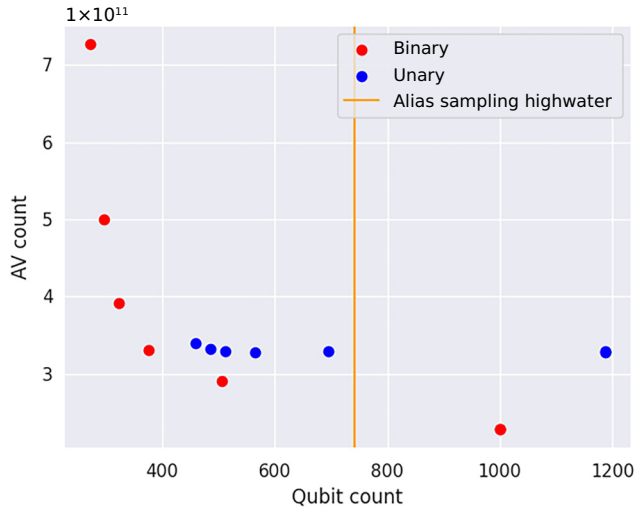


FIG. 11. Active volume and logic qubit count of the entire algorithm with respect to various degrees of batching, and distinguished by the version of Select that it uses. Instances in which the binary version of Select is used are denoted by red dots, while the blue data points represent instances in which the b_i registers have been converted to unary. As an additional modification, we have turned the QROAM dataloader in the alias sampling procedure of Prepare back into a QROM. In this plot, instances of the binary Select seem much more preferable than the unary Select—the blue data points seem to lie above the curve along which we find the red data points. This is happening not for the high AV of the unary circuit, but for its overhead of $M + N/2$ qubits.

[68]. Results are shown in Table XI, highlighting the Hartree-Fock ground-state energy contribution, CCSD(T) correlation energy, and DMRG correlation energy in the last column using a bond dimension of 1500. Interestingly, while the Hartree-Fock energy contribution, and hence the absolute energy, is observed to change up to 0.77 E_h in one of the two cases, the correlation energy error computed with both CCSD(T) and DMRG methods remains below the 0.3-mE_h threshold for both truncation parameter settings. Moreover, although our DMRG-based results show that the CCSD(T) correlation energy error remains a good proxy for the P450 benchmark system, further studies on a broader set of benchmarks are needed to determine whether this holds in general or whether alternative metrics should be employed. This question remains open and warrants additional investigation.

APPENDIX D: UNARY ENCODING

A modification of Select sees some internal data be converted from binary to unary. In the registers b_0 and b_1, the numbers μ are encoded as binary numbers, i.e., by $|\mu\rangle$ we mean

$$|\mu\rangle = |\mu_{\varkappa-1}\rangle \otimes \cdots \otimes |\mu_1\rangle \otimes |\mu_0\rangle$$

for a register with $\varkappa$ qubits, and with bit values $\mu_x \in \{0,1\}$ such that $\mu = \sum_{x=0}^{\varkappa-1} \mu_x 2^x$. There is a quantum circuit that, considering a fresh register with $2^\kappa + 1$ qubits in $|0\rangle$, achieves an out-of-place conversion of μ into unary: $|\mu\rangle|0\rangle \mapsto |\mu\rangle|2^{\mu+1}\rangle$, where $|2^{\mu+1}\rangle$ is a bit string with only the $(\mu+1)$th (least significant) qubit in $|1\rangle$ and the other qubits in $|0\rangle$, i.e., $|00010\rangle$ for $\mu = 1$. In Select, we can

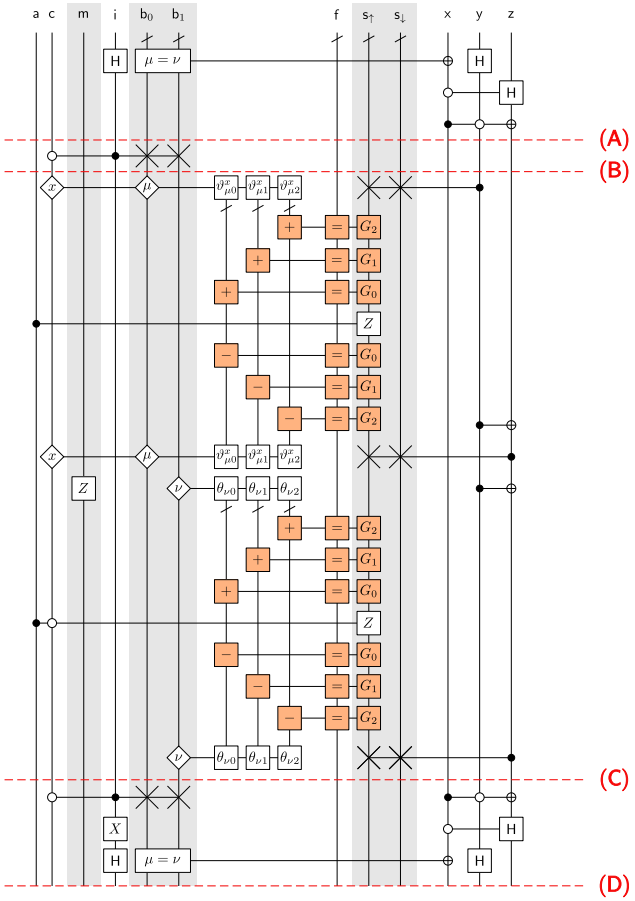


FIG. 12. Select in Fig. 1 divided into sections with waypoints (A), (B), (C), and (D).

use the binary-to-unary conversion for register b_i: shifting the dataloaders from a binary b_i register to the respective entangled unary register allows us to nullify the circuit's magic state cost. This allows us to combine the binary-to-unary conversion with batching, as depicted in Fig. 10. The caveat is that the binary-to-unary conversion circuit has roughly the magic state cost of one dataloader. However, with a maximum of $2^{\varkappa}+1 = M + N/2 = 218$ qubits, hundreds of auxiliary qubits can be saved with batching. A subselect circuit, along with binary-to-unary conversion circuits and unary dataloading, is depicted in Fig. 10. Unfortunately, it turns out that, in the AV world, the unary-encoding version of Select is not more effective than Select of BLISS-THC-b, as is illustrated in Fig. 11. While the AV costs of the algorithm are decent, the unary data points in the figure (corresponding to various batch sizes after the unary conversion) lie strictly above the curve defined by the BLISS-THC-b algorithm. If unary did not have the overhead of $M + N/2$ qubits, the points would be shifted past the curve to the left. Perhaps future work will address these shortcomings by hybridizing unary and binary encodings.

APPENDIX E: BLOCK ENCODING OF THE HAMILTONIAN

In this appendix, we prove the block-encoding property in Eq. (24), using the Select circuit of Fig. 1 and justify the choice for the coefficients $\widehat{\zeta}_{\mu\nu}$ in Eq. (26). Let us consider state

$$|\Lambda_1\rangle = |1\rangle_{\mathsf{a}} \sum_{\nu=0}^{M} \sum_{\mu=0}^{\nu} \sqrt{|\widehat{\zeta}_{\mu\nu}|}\, |\mu\rangle_{\mathsf{b}_0} |\nu\rangle_{\mathsf{b}_1} \times |0\rangle_{\mathsf{i}} |\delta_{M,\nu}\rangle_{\mathsf{c}} |\mathrm{sign}(\widehat{\zeta}_{\mu\nu}))\rangle_{\mathsf{m}} |0\rangle_{\mathsf{x}} |0\rangle_{\mathsf{y}} |0\rangle_{\mathsf{z}}, \tag{E1}$$

a $p = 1$ version of state $|\Lambda\rangle$ in Eq. (25). We now compare Hamiltonian $H' = \langle\Lambda_1|\mathsf{Select}|\Lambda_1\rangle$, encoded in the quantum computer, to Hamiltonian to $\widetilde{H}$ of Eq. (22). To this end, we compute $\mathsf{Select}|\Lambda\rangle$ in four stages, represented by the states at waypoints (A)–(D) in Fig. 12. In the first stage, we apply a Hadamard gate to qubit i and compute the qubits x, y, as well as z, thus separating the cases $\nu = \mu$. Omitting $|1\rangle_{\mathsf{a}}$ from here on, $|\Lambda_1\rangle$ becomes

$$\begin{aligned}(\mathsf{A}):\quad & \sum_{\nu=0}^{M} \sum_{\mu=0}^{\nu-1} \sqrt{|\widehat{\zeta}_{\mu\nu}|}\, |\mu\rangle_{\mathsf{b}_0} |\nu\rangle_{\mathsf{b}_1} |\delta_{M,\nu}\rangle_{\mathsf{c}} |\mathrm{sign}(\widehat{\zeta}_{\mu\nu})\rangle_{\mathsf{m}} \\ & \times |0\rangle_{\mathsf{x}} |+\rangle_{\mathsf{y}} |+\rangle_{\mathsf{z}} \\ & + \frac{1}{\sqrt{2}} \sum_{\nu=0}^{M} \sqrt{|\widehat{\zeta}_{\nu\nu}|}\, |\nu\rangle_{\mathsf{b}_0} |\nu\rangle_{\mathsf{b}_1} |\delta_{M,\nu}\rangle_{\mathsf{c}} |\mathrm{sign}(\widehat{\zeta}_{\mu\nu})\rangle_{\mathsf{m}} \\ & \times |1\rangle_{\mathsf{x}} (|0\rangle_{\mathsf{y}} |1\rangle_{\mathsf{z}} + |1\rangle_{\mathsf{y}} |0\rangle_{\mathsf{z}})\end{aligned} \tag{E2}$$

at waypoint (A). In the next stage, registers b_0 and b_1 are swapped, but only in the subspace reserved for two-body operators: the $|0\rangle$ subspace of c. At waypoint (B) the state becomes

$$\begin{aligned}(\mathsf{B}):\ & \frac{1}{\sqrt{2}} \sum_{\nu=0}^{M-1} \sum_{\mu=0}^{\nu-1} \sqrt{|\widehat{\zeta}_{\mu\nu}|} (|\mu\rangle_{\mathsf{b}_0} |\nu\rangle_{\mathsf{b}_1} |0\rangle_{\mathsf{i}} + |\nu\rangle_{\mathsf{b}_0} |\mu\rangle_{\mathsf{b}_1} |1\rangle_{\mathsf{i}}) \\ & \times |0\rangle_{\mathsf{c}} |\mathrm{sign}(\widehat{\zeta}_{\mu\nu})\rangle_{\mathsf{m}} |0\rangle_{\mathsf{x}} |+\rangle_{\mathsf{y}} |+\rangle_{\mathsf{z}} \\ & + \frac{1}{\sqrt{2}} \sum_{\nu=0}^{M} \sqrt{|\widehat{\zeta}_{\nu\nu}|}\, |\nu\rangle_{\mathsf{b}_0} |\nu\rangle_{\mathsf{b}_1} |+\rangle_{\mathsf{i}} |\delta_{M,\nu}\rangle_{\mathsf{c}} \\ & \times |\mathrm{sign}(\widehat{\zeta}_{\mu\nu})\rangle_{\mathsf{m}} |1\rangle_{\mathsf{x}} (|0\rangle_{\mathsf{y}} |1\rangle_{\mathsf{z}} + |1\rangle_{\mathsf{y}} |0\rangle_{\mathsf{z}}) \\ & + \sum_{\mu=0}^{N/2-1} \sqrt{|\widehat{\zeta}_{\mu M}|}\, |\mu\rangle_{\mathsf{b}_0} |M\rangle_{\mathsf{b}_1} |+\rangle_{\mathsf{i}} |1\rangle_{\mathsf{c}} |\mathrm{sign}(\widehat{\zeta}_{\mu M})\rangle_{\mathsf{m}} \\ & \times |0\rangle_{\mathsf{x}} |+\rangle_{\mathsf{y}} |+\rangle_{\mathsf{z}}.\end{aligned} \tag{E3}$$

The following stage applies a Z operator on the sign qubit m, as well as the (modified) subselect operators

$$\sum_{\sigma}\sum_{\mu}|\mu\rangle\langle\mu|_{\mathsf{b}_0}\otimes|\sigma\rangle\langle\sigma|_{\mathsf{y}}\otimes(|0\rangle\langle 0|_{\mathsf{c}}\otimes\widetilde{Z}_{\mu,\sigma}+|1\rangle\langle 1|_{\mathsf{c}}\otimes\widetilde{T}_{\mu,\sigma}) \tag{E4}$$

and

$$\sum_{\tau}\sum_{\nu}|\nu\rangle\langle\nu|_{\mathsf{b}_1}\otimes|\tau\rangle\langle\tau|_{\mathsf{z}}\otimes\widetilde{Z}_{\nu,\tau}, \tag{E5}$$

where

$$\widetilde{Z}_{\mu,\sigma}=U_\mu^\dagger Z_{0,\sigma}U_\mu \quad\text{and}\quad \widetilde{T}_{\mu,\sigma}=V_\mu^\dagger Z_{0,\sigma}V_\mu. \tag{E6}$$

At waypoint (C), the state thus becomes

$$\begin{aligned}(\mathsf{C}):\quad &\frac{1}{\sqrt{8}}\sum_{\nu=0}^{M-1}\sum_{\mu=0}^{\nu-1}\sqrt{|\widehat{\zeta}_{\mu\nu}|}\sum_{\sigma\tau}(|\mu\rangle_{\mathsf{b}_0}|\nu\rangle_{\mathsf{b}_1}|0\rangle_{\mathsf{i}}\widetilde{Z}_{\nu,\tau}\widetilde{Z}_{\mu,\sigma}+|\nu\rangle_{\mathsf{b}_0}|\mu\rangle_{\mathsf{b}_1}|1\rangle_{\mathsf{i}}\widetilde{Z}_{\mu,\tau}\widetilde{Z}_{\nu,\sigma})|0\rangle_{\mathsf{c}}Z_{\mathsf{m}}|\mathrm{sign}(\widehat{\zeta}_{\mu\nu})\rangle_{\mathsf{m}}|0\rangle_{\mathsf{x}}|\sigma\rangle_{\mathsf{y}}|\tau\rangle_{\mathsf{z}}\\ &+\frac{1}{\sqrt{2}}\sum_{\nu=0}^{M-1}\sqrt{|\widehat{\zeta}_{\nu\nu}|}|\nu\rangle_{\mathsf{b}_0}|\nu\rangle_{\mathsf{b}_1}|+\rangle_{\mathsf{i}}|0\rangle_{\mathsf{c}}Z_{\mathsf{m}}|\mathrm{sign}(\widehat{\zeta}_{\mu\nu})\rangle_{\mathsf{m}}|1\rangle_{\mathsf{x}}(|0\rangle_{\mathsf{y}}|1\rangle_{\mathsf{z}}\widetilde{Z}_{\nu,\downarrow}\widetilde{Z}_{\nu,\uparrow}+|1\rangle_{\mathsf{y}}|0\rangle_{\mathsf{z}}\widetilde{Z}_{\nu,\uparrow}\widetilde{Z}_{\nu,\downarrow})\\ &+\frac{1}{\sqrt{2}}\sum_{\mu=0}^{N/2-1}\sqrt{|\widehat{\zeta}_{\mu M}|}|\mu\rangle_{\mathsf{b}_0}|M\rangle_{\mathsf{b}_1}|+\rangle_{\mathsf{i}}|1\rangle_{\mathsf{c}}Z_{\mathsf{m}}|\mathrm{sign}(\widehat{\zeta}_{\mu M})\rangle_{\mathsf{m}}|0\rangle_{\mathsf{x}}(|0\rangle_{\mathsf{y}}\widetilde{T}_{\mu,\uparrow}+|1\rangle_{\mathsf{y}}\widetilde{T}_{\mu,\downarrow})|+\rangle_{\mathsf{z}}.\end{aligned} \tag{E7}$$

In the last stage, we undo the swap of the second stage, flip qubit m, and uncompute i, x, y, and z. Using the fact that $\mathsf{H}|x\rangle=2^{-1/2}\sum_y(-1)^{xy}|y\rangle$ for bits $x,y\in\{0,1\}$, the state at waypoint (D) is

$$\begin{aligned}(\mathsf{D}):\quad &\frac{1}{8}\sum_{iyz}\sum_{\nu=0}^{M-1}\sum_{\mu=0}^{\nu-1}\sqrt{|\widehat{\zeta}_{\mu\nu}|}\,|\mu\rangle_{\mathsf{b}_0}|\nu\rangle_{\mathsf{b}_1}|i\rangle_{\mathsf{i}}\sum_{\sigma\tau}((-1)^i\widetilde{Z}_{\nu,\tau}\widetilde{Z}_{\mu,\sigma}+\widetilde{Z}_{\mu,\tau}\widetilde{Z}_{\nu,\sigma})|0\rangle_{\mathsf{c}}Z_{\mathsf{m}}|\mathrm{sign}(\widehat{\zeta}_{\mu\nu})\rangle_{\mathsf{m}}|0\rangle_{\mathsf{x}}(-1)^{y\sigma+z\tau}|y\rangle_{\mathsf{y}}|z\rangle_{\mathsf{z}}\\ &+\frac{1}{2}\sum_{y}\sum_{\nu=0}^{M-1}\sqrt{|\widehat{\zeta}_{\nu\nu}|}\,|\nu\rangle_{\mathsf{b}_0}|\nu\rangle_{\mathsf{b}_1}|0\rangle_{\mathsf{i}}|0\rangle_{\mathsf{c}}Z_{\mathsf{m}}|\mathrm{sign}(\widehat{\zeta}_{\mu\nu})\rangle_{\mathsf{m}}(\widetilde{Z}_{\nu,\downarrow}\widetilde{Z}_{\nu,\uparrow}+(-1)^y\widetilde{Z}_{\nu,\uparrow}\widetilde{Z}_{\nu,\downarrow})|0\rangle_{\mathsf{x}}|y\rangle_{\mathsf{y}}|0\rangle_{\mathsf{z}}\\ &+\frac{1}{2}\sum_{y}\sum_{\mu=0}^{N/2-1}\sqrt{|\widehat{\zeta}_{\mu M}|}|\mu\rangle_{\mathsf{b}_0}|M\rangle_{\mathsf{b}_1}|0\rangle_{\mathsf{i}}|1\rangle_{\mathsf{c}}Z_{\mathsf{m}}|\mathrm{sign}(\widehat{\zeta}_{\mu M})\rangle_{\mathsf{m}}(\widetilde{T}_{\mu,\uparrow}+(-1)^y\widetilde{T}_{\mu,\downarrow})|0\rangle_{\mathsf{x}}|y\rangle_{\mathsf{y}}|0\rangle_{\mathsf{z}}.\end{aligned} \tag{E8}$$

Multiplying this state from the left-hand side with $\langle\Lambda_1|\cdot$ finally gives us

$$H'=\frac{1}{8}\sum_{\mu=0}^{M-1}\sum_{\nu\neq\mu}\widehat{\zeta}_{\mu\nu}\sum_{\sigma\tau}\widetilde{Z}_{\nu,\tau}\widetilde{Z}_{\mu,\sigma}+\frac{1}{2}\sum_{\nu=0}^{M-1}\widehat{\zeta}_{\nu\nu}\sum_{\sigma}\widetilde{Z}_{\nu,\sigma}\widetilde{Z}_{\nu,\overline{\sigma}}+\frac{1}{2}\sum_{k=0}^{N/2-1}\widehat{\zeta}_{kM}\sum_{\sigma}\widetilde{T}_{k,\sigma},$$

where $\overline{\sigma}$ is the flipped version of σ. This is the Hamiltonian we have encoded, while the Hamiltonian we were aiming to encode is

$$\widetilde{H}=\frac{1}{8}\sum_{\mu=0}^{M-1}\sum_{\nu\neq\mu}\tilde{\zeta}_{\mu\nu}\sum_{\sigma\tau}\widetilde{Z}_{\nu,\tau}\widetilde{Z}_{\mu,\sigma}+\frac{1}{8}\sum_{\nu=0}^{M-1}\tilde{\zeta}_{\nu\nu}\sum_{\sigma}\widetilde{Z}_{\nu,\sigma}\widetilde{Z}_{\nu,\overline{\sigma}}-\frac{1}{2}\sum_{k=0}^{N/2-1}\tilde{t}_k\sum_{\sigma}\widetilde{T}_{k,\sigma}, \tag{E9}$$

where constant terms were eliminated with respect to Eq. (22). We thus choose $\widehat{\zeta}_{\mu\nu}$ as in Eq. (26).

[1] R. M. Langer, The quantum mechanics of chemical reaction, Phys. Rev. **34**, 92 (1929).

[2] D. G. Truhlar, B. C. Garrett, and S. J. Klippenstein, Current status of transition-state theory, J. Phys. Chem. **100**, 12771 (1996).

[3] U. Ryde and P. Söderhjelm, Ligand-binding affinity estimates supported by quantum-mechanical methods, Chem. Rev. **116**, 5520 (2016).

[4] Y.-H. Lam, Y. Abramov, R. S. Ananthula, J. M. Elward, L. R. Hilden, S. O. Nilsson Lill, P.-O. Norrby, A. Ramirez, E. C. Sherer, J. Mustakis, and G. J. Tanoury, Applications of quantum chemistry in pharmaceutical process development: Current state and opportunities, Org. Process Res. Dev. **24**, 1496 (2020).

[5] R. Santagati, A. Aspuru-Guzik, R. Babbush, M. Degroote, L. González, E. Kyoseva, N. Moll, M. Oppel, R. M. Parrish, N. C. Rubin, M. Streif, C. S. Tautermann, H. Weiss, N. Wiebe, and C. Utschig-Utschig, Drug design on quantum computers, Nat. Phys. **20**, 549 (2024).

[6] A. Baiardi, M. Christandl, and M. Reiher, Quantum computing for molecular biology, ChemBioChem **24**, e202300120 (2023).

[7] E. W. C. Spotte-Smith, S. M. Blau, X. Xie, H. D. Patel, M. Wen, B. Wood, S. Dwaraknath, and K. A. Persson, Quantum chemical calculations of lithium-ion battery electrolyte and interphase species, Sci. Data **8**, 203 (2021).

[8] I. H. Kim, Y.-H. Liu, S. Pallister, W. Pol, S. Roberts, and E. Lee, Fault-tolerant resource estimate for quantum chemical simulations: Case study on Li-ion battery electrolyte molecules, Phys. Rev. Res. **4**, 023019 (2022).

[9] A. Delgado, P. A. M. Casares, R. dos Reis, M. S. Zini, R. Campos, N. Cruz-Hernández, A.-C. Voigt, A. Lowe, S. Jahangiri, M. A. Martin-Delgado, J. E. Mueller, and J. M. Arrazola, Simulating key properties of lithium-ion batteries with a fault-tolerant quantum computer, Phys. Rev. A **106**, 032428 (2022).

[10] C. J. Jacobsen, S. Dahl, A. Boisen, B. S. Clausen, H. Topsøe, A. Logadottir, and J. K. Nørskov, Optimal catalyst curves: Connecting density functional theory calculations with industrial reactor design and catalyst selection, J. Catal. **205**, 382 (2002).

[11] N. Mardirossian and M. Head-Gordon, Thirty years of density functional theory in computational chemistry: An overview and extensive assessment of 200 density functionals, Mol. Phys. **115**, 2315 (2017).

[12] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, Quantum chemistry in the age of quantum computing, Chem. Rev. **119**, 10856 (2019).

[13] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, Quantum computational chemistry, Rev. Mod. Phys. **92**, 015003 (2020).

[14] A. Aspuru-Guzik, Simulated quantum computation of molecular energies, Science **309**, 1704 (2005).

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

[16] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, Quantum algorithms for quantum chemistry and quantum materials science, Chem. Rev. **120**, 12685 (2020).

[17] M. Steudtner, S. Morley-Short, W. Pol, S. Sim, C. L. Cortes, M. Loipersberger, R. M. Parrish, M. Degroote, N. Moll, R. Santagati, and M. Streif, Fault-tolerant quantum computation of molecular observables, Quantum **7**, 1164 (2023).

[18] J. J. Goings, A. White, J. Lee, C. S. Tautermann, M. Degroote, C. Gidney, T. Shiozaki, R. Babbush, and N. C. Rubin, Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers, Proc. Natl. Acad. Sci. **119**, e2203533119 (2022).

[19] F. P. Guengerich, A history of the roles of cytochrome P450 enzymes in the toxicity of drugs, Toxicol. Res. **37**, 1 (2021).

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

[21] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Info. Comput. **12**, 901 (2012).

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

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

[24] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019 (Association for Computing Machinery, New York, NY, USA, 2019), p. 193–204.

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

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

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

[28] D. Rocca, C. L. Cortes, J. F. Gonthier, P. J. Ollitrault, R. M. Parrish, G.-L. Anselmetti, M. Degroote, N. Moll, R. Santagati, and M. Streif, Reducing the runtime of fault-tolerant quantum simulations in chemistry through symmetry-compressed double factorization, J. Chem. Theory Comput. **20**, 4639 (2024).

[29] A. Y. Kitaev, Quantum measurements and the Abelian stabilizer problem, arXiv:quant-ph/9511026.

[30] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[31] I. Loaiza and A. F. Izmaylov, Block-invariant symmetry shift: Preprocessing technique for second-quantized Hamiltonians to improve their decompositions to linear combination of unitaries, J. Chem. Theory Comput. **19**, 8201 (2023).

[32] S. Patel, A. Sankar Brahmachari, J. T. Cantin, L. Wang, and A. F. Izmaylov, Global minimization of electronic Hamiltonian 1-norm via linear programming in the block invariant symmetry shift (BLISS) method, arXiv:2409.18277.
[33] D. W. Berry, Y. Tong, T. Khattar, A. White, T. In Kim, G. H. Low, S. Boixo, Z. Ding, L. Lin, S. Lee, G. Kin-Lic Chan, R. Babbush, and N. C. Rubin, Rapid initial-state preparation for the quantum simulation of strongly correlated molecules, PRX Quantum **6**, 020327 (2025).
[34] O. Oumarou, M. Scheurer, R. M. Parrish, E. G. Hohenstein, and C. Gogolin, Accelerating quantum computations of chemistry through regularized compressed double factorization, Quantum **8**, 1371 (2024).
[35] K. Deka and E. Zak, Simultaneously optimizing symmetry shifts and tensor factorizations for cost-efficient fault-tolerant quantum simulations of electronic Hamiltonians, J. Chem. Theory Comput. **21**, 4458 (2025).
[36] S. A. Adcock and J. A. McCammon, Molecular dynamics: Survey of methods for simulating the activity of proteins, Chem. Rev. **106**, 1589 (2006).
[37] M. C. Zwier and L. T. Chong, Reaching biological timescales with all-atom molecular dynamics simulations, Curr. Opin. Pharmacol. **10**, 745 (2010).
[38] T. Ginex, J. Vázquez, C. Estarellas, and F. Luque, Quantum mechanical-based strategies in drug discovery: Finding the pace to new challenges in drug design, Curr. Opin. Struct. Biol. **87**, 102870 (2024).
[39] D. Litinski and N. Nickerson, Active volume: An architecture for efficient fault-tolerant quantum computers with limited non-local connections, arXiv:2211.15465.
[40] D. Litinski, How to compute a 256-bit elliptic curve private key with only 50 million Toffoli gates, arXiv:2306.08585.
[41] H. Bombin, I. H. Kim, D. Litinski, N. Nickerson, M. Pant, F. Pastawski, S. Roberts, and T. Rudolph, Interleaving: Modular architectures for fault-tolerant photonic quantum computing, arXiv:2103.08612.
[42] C. L. Cortes, D. Rocca, J. F. Gonthier, P. J. Ollitrault, R. M. Parrish, G.-L. R. Anselmetti, M. Degroote, N. Moll, R. Santagati, and M. Streif, Assessing the query complexity limits of quantum phase estimation using symmetry-aware spectral bounds, Phys. Rev. A **110**, 022420 (2024).
[43] R. M. Parrish, E. G. Hohenstein, T. J. Martínez, and C. D. Sherrill, Tensor hypercontraction. II. Least-squares renormalization, J. Chem. Phys. **137**, 224106 (2012).
[44] E. G. Hohenstein, R. M. Parrish, and T. J. Martínez, Tensor hypercontraction density fitting. I. Quartic scaling second- and third-order Møller-Plesset perturbation theory, J. Chem. Phys. **137**, 044103 (2012).
[45] I. Loaiza, A. M. Khah, N. Wiebe, and A. F. Izmaylov, Reducing molecular electronic Hamiltonian simulation cost for linear combination of unitaries approaches, Quantum Sci. Technol. **8**, 035019 (2023).
[46] G. H. Low, V. Kliuchnikov, and L. Schaeffer, Trading T gates for dirty qubits in state preparation and unitary synthesis, Quantum **8**, 1375 (2024).
[47] C. Gidney, Halving the cost of quantum addition, Quantum **2**, 74 (2018).
[48] D. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, Surface code quantum computing by lattice surgery, New J. Phys. **14**, 123011 (2012).
[49] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, Quantum **3**, 128 (2019).
[50] We also assume that each individual workspace qubit is used infrequently enough that we can rotate rough and smooth edges to face the workspace qubits when each one is needed.
[51] In the architecture presented in the original AV paper the authors assumed that $m = w$. This was largely for simplicity.
[52] We occasionally refer to the total number of interleaving modules as the footprint, as it determines the physical size of the quantum computer and due to the term "size" being inaccurate. If the number of IMs is to increase, do we need to build a larger machine or do we need to make the quantum computer smaller? We say that devices with more (less) interleaving modules have a bigger (smaller) footprint.
[53] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Surface codes: Towards practical large-scale quantum computation, Phys. Rev. A **86**, 032324 (2012).
[54] C. Gidney and M. Ekerå, How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits, Quantum **5**, 433 (2021).
[55] R. Acharya *et al.*, Quantum error correction below the surface code threshold, Nature **638**, 920 (2024).
[56] S. Bartolucci, P. Birchall, H. Bombín, H. Cable, C. Dawson, M. Gimeno-Segovia, E. Johnston, K. Kieling, N. Nickerson, M. Pant, F. Pastawski, T. Rudolph, and C. Sparrow, Fusion-based quantum computation, Nat. Commun. **14**, 912 (2023).
[57] J. J. Goings, A. White, C. S. Tautermann, M. Degroote, C. Gidney, T. Shiozaki, R. Babbush, and N. C. Rubin, Data for "Reliably assessing the electronic structure of cytochrome P450 on today's classical computers and tomorrow's quantum computers" (2022),
[58] A. Caesura, C. Cortes, W. Pol, S. Sim, M. Steudtner, G.-L. Anselmetti, M. Degroote, N. Moll, M. Streif, and C. Tautermann, Data for "Faster quantum chemistry simulations on a quantum computer with improved tensor factorization and active volume compilation" (2025),.
[59] Q. Sun *et al.*, Recent developments in the PySCF program package, J. Chem. Phys. **153**, 024109 (2020).
[60] We occasionally refer to the number of memory qubits as highwater, due to its nonadditivity: the highwater is determined by the largest number of qubits used in all the subroutines of the algorithm. The term is mostly used when resources are estimated, i.e., when we are trying to determine the size of the memory register.
[61] This is indeed the value of prior art after our redefinition of ⊐.
[62] C. Gidney, N. Shutty, and C. Jones, Magic state cultivation: Growing T states as cheap as CNOT gates, arXiv:2409.17595.
[63] S. Bravyi, A. W. Cross, J. M. Gambetta, D. Maslov, P. Rall, and T. J. Yoder, High-threshold and low-overhead fault-tolerant quantum memory, Nature **627**, 778 (2024).
[64] K. Sahay, J. Jin, J. Claes, J. D. Thompson, and S. Puri, High-threshold codes for neutral-atom qubits with biased erasure errors, Phys. Rev. X **13**, 041013 (2023).
[65] C. T. Hann, K. Noh, H. Putterman, M. H. Matheny, J. K. Iverson, M. T. Fang, C. Chamberland, O. Painter, and

F. G. S. L. Brandão, Hybrid cat-transmon architecture for scalable, hardware-efficient quantum error correction, PRX Quantum **6**, 030305 (2025).

[66] J. M. Montgomery and D. A. Mazziotti, Strong electron correlation in nitrogenase cofactor, FeMoco, J. Phys. Chem. A **122**, 4988 (2018).

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

[68] H. Zhai, H. R. Larsson, S. Lee, Z.-H. Cui, T. Zhu, C. Sun, L. Peng, R. Peng, K. Liao, J. Tölle, J. Yang, S. Li, and G. K.-L. Chan, Block2: A comprehensive open source framework to develop and apply state-of-the-art DMRG algorithms in electronic structure and beyond, J. Chem. Phys. **159**, 234801 (2023).