Implementation of measurement reduction for the variational quantum eigensolver

Alexis Ralli,[1,*] Peter J. Love,[2,3,†] Andrew Tranter,[2,4,‡] and Peter V. Coveney[1,5,§]

[1] *Centre for Computational Science, Department of Chemistry, University College London, London WC1H 0AJ, United Kingdom*
[2] *Department of Physics and Astronomy, Tufts University, Medford, Massachusetts 02155, USA*
[3] *Computational Science Initiative, Brookhaven National Laboratory, Upton, New York 11973, USA*
[4] *Cambridge Quantum Computing, 9a Bridge Street, Cambridge CB2 1UB, United Kingdom*
[5] *Informatics Institute, University of Amsterdam, Amsterdam 1098 XH, Netherlands*

(Received 7 December 2020; revised 28 June 2021; accepted 28 July 2021; published 27 August 2021)

One limitation of the variational quantum eigensolver algorithm is the large number of measurement steps required to estimate different terms in the Hamiltonian of interest. Unitary partitioning reduces this overhead by transforming the problem Hamiltonian into one containing fewer terms. We explore two different circuit constructions of the transformation required—one built by a sequence of rotations and the other built by a linear combination of unitaries (LCU). To assess performance, we simulated chemical Hamiltonians and studied the ground states of H_2 and LiH. Both implementations are successful even in the presence of noise. The sequence-of-rotations realization offers the greatest benefit to calculations, whereas the probabilistic nature of LCU reduces its effectiveness. This work also demonstrates an experimental implementation of LCU on quantum hardware.

DOI: 10.1103/PhysRevResearch.3.033195

I. INTRODUCTION

Current quantum computing devices have significant limitations, namely, short coherence times, low qubit numbers, and little to no error correction. These machines are termed noisy intermediate-scale quantum (NISQ) computers [1]. The leading candidate algorithms for use on NISQ devices are variational hybrid quantum-classical algorithms such as the variational quantum eigensolver (VQE) and quantum approximate optimization algorithm (QAOA) [2,3]. VQE estimates Hamiltonian eigenvalues on near-term quantum computers [3]. Many different implementations of the algorithm have been performed utilizing a wide array of different quantum platforms [3–6].

VQE has been widely applied to the electronic structure problem. The second quantized form of the molecular electronic Hamiltonian is converted to a qubit Hamiltonian by the Jordan-Wigner, Bravyi-Kitaev, or related transformations [7–9]. The resulting qubit Hamiltonian is a linear combination of m Pauli operators on n qubits:

$$H_q = \sum_{i=0}^{m-1} c_i P_i = \sum_{i=0}^{m-1} c_i \big(\sigma_0^i \otimes \sigma_1^i \otimes, \dots, \otimes \sigma_{n-1}^i\big), \tag{1}$$

where c_i are real coefficients, P_i are n-qubit Pauli operators, which are n-fold tensor products of 1-qubit Pauli operators, or the 2×2 identity matrix: $\sigma_j^i \in \{X, Y, Z, \mathcal{I}\}\ \forall i$, and j indexes the qubit the operator acts on.

In general, the number of terms in Eq. (1) scales as $\mathcal{O}(N_{\text{orb}}^4)$, where N_{orb} is the number of orbitals [10]. The linearity of expectation values allows

$$E(\vec{\theta}) = \langle H_q \rangle = \sum_{i=0}^{m-1} c_i \langle \psi(\vec{\theta})| P_i |\psi(\vec{\theta})\rangle , \tag{2}$$

where $|\psi(\vec{\theta})\rangle$ is an ansatz state produced by a parametrized quantum circuit. In conventional VQE, the expectation value of each subterm $\langle P_i \rangle$ is determined independently.

An estimate of each term's expectation value $\langle P_i \rangle$ is found by averaging over M_i repeated measurement outcomes $\{s_j^{(i)}\}_{j=1,2,\dots,M_i}$ via [3,11,12]

$$\langle P_i \rangle = c_i \langle \psi(\vec{\theta})| P_i |\psi(\vec{\theta})\rangle \approx c_i \left(\frac{1}{M_i} \sum_{j=1}^{M_i} s_j^{(i)} \right), \tag{3}$$

where $s_j^{(i)} \in \{-1, +1\}$. The above expression is exact as the number of samples $M_i \to \infty$.

A finite number of runs is used to estimate each $\langle P_i \rangle$ term, and thus each outcome will belong to a distribution centered around the expectation value $\langle \psi(\vec{\theta})| P_i |\psi(\vec{\theta})\rangle$ with standard deviation ϵ_i. Each estimate $\langle \psi(\vec{\theta})| P_i |\psi(\vec{\theta})\rangle$ is derived from sums of random variables with finite variance. Due to the central limit theorem, they must converge to a normal distribution [11]. Because we will be comparing Hamiltonians with different numbers of terms, we will take a slightly different approach. We combine a single sample of all terms into a single-shot energy estimate $e_j = \sum_i c_i s_j^{(i)}$. The distribution of

[*] alexis.ralli.18@ucl.ac.uk
[†] peter.love@tufts.edu
[‡] tufts@atranter.net
[§] p.v.coveney@ucl.ac.uk

this estimate determines how many samples are required to achieve a given error on the mean.

The optimal number of repetitions M_i to achieve a certain precision ϵ is [13,14]

$$M = \sum_{i=0}^{m-1} M_i = \frac{1}{\epsilon^2}\left(\sum_{i=0}^{m-1} |c_i| P_i\right)^2 \leqslant \frac{1}{\epsilon^2}\left(\sum_{i=0}^{m-1} |c_i|\right)^2, \tag{4}$$

where M is the total number of measurements. Since the number of terms in Eq. (1) scales as $\mathcal{O}(N_{\text{orb}}^4)$, the total number of measurements required will scale as $\mathcal{O}(N_{\text{orb}}^6/\epsilon^2)$, where chemical accuracy is defined as $\epsilon = 1$ kcal/mol (1.6 mHa), the accuracy required to match typical thermochemical experiments. Wecker *et al.* showed that obtaining energy estimates for HeH^+, BeH_2, and H_2O requires 10^8–10^9 samples to achieve an error of 1 mHa [13]. This implies that the number of measurements required is an obstacle for experimental implementations of VQE because of the number of qubits currently available on NISQ devices [6]. For example, take the experimental implementation of VQE by Hempel *et al.*, in which it took 20 ms to perform each VQE repetition on a trapped-ion quantum computer [5]. To obtain the ground-state energy of H_2, in a minimal basis to within chemical accuracy, of the order of 14 000 repetitions were needed, and 4.6 min of averaging were required.

Various approaches have been proposed for reducing the total number of samples required by VQE [15–23]. In this paper, we focus on the unitary partitioning procedure independently proposed by Verteletskyi and co-workers [24] and Zhao *et al.* [25]. The main idea of this approach is to partition the qubit Hamiltonian into groups of n-fold Pauli operators whose linear combination is unitary. The overall operator represented by these sums can then be measured at once using additional coherent resources. In this paper, we compare two different circuit realizations of unitary partitioning, as proposed in Ref. [25].

II. UNITARY PARTITIONING

The expectation value of any Hermitian operator can be obtained via a single set of single-qubit measurements, because it can be written in terms of its spectral decomposition. For example, consider the spectral decomposition of a general Hermitian operator A:

$$A = \sum_{a=0}^{d-1} \lambda_a \left|\psi_a\right\rangle \left\langle\psi_a\right|, \tag{5}$$

where d is the dimension of the space and A acts on orthonormal states $|\psi_a\rangle$. Each $|\psi_a\rangle$ is an eigenstate of the operator with corresponding eigenvalue λ_a. As the set of eigenvectors $\{|\psi_a\rangle\}$ form an orthonormal basis, there always exists a unitary R that maps this basis to another: $R\,|\psi_a\rangle = |e_a\rangle$ or $|\psi_a\rangle = R^\dagger\,|e_a\rangle$. The operator A can be written in this basis:

$$A = \sum_{a=0}^{d-1} \lambda_a \left|\psi_a\right\rangle \left\langle\psi_a\right| \tag{6a}$$

$$= \sum_{a=1}^{d} \lambda_a R^\dagger \left|e_a\right\rangle \left\langle e_a\right| R \tag{6b}$$

$$= R^\dagger \left(\sum_{a=1}^{d} \lambda_a \left|e_a\right\rangle \left\langle e_a\right|\right) R \tag{6c}$$

$$= R^\dagger Q R. \tag{6d}$$

The expectation value of A can be found by $\langle A\rangle = \langle\psi|\,A\,|\psi\rangle = \langle\psi|\,R^\dagger Q R\,|\psi\rangle$. Note that Q is a matrix defined by the large parentheses in Eq. (6c). This idea underpins the unitary partitioning method.

Individual n-fold tensor product operators of Pauli operators P_i are Hermitian and unitary; however, a sum of unitary operators is in general not unitary. To make $\sum_j c_j P_j$ unitary, three constraints are imposed [24,25]:

(1) $\{P_i, P_j\} = 2\delta_{i,j}$,
(2) $\sum_j |c_j|^2 = 1$,
(3) $\text{Im}\,(c_j^* c_i) = 0$.

Here, { , } is the anticommutator ($\{A, B\} \equiv AB + BA$). The first condition is satisfied by partitioning the qubit Hamiltonian H_q into m_c sets denoted $\{S_l\}_{l=0,2,\ldots,m_c-1}$. The sub-Hamiltonian corresponding to each anticommuting set H_{S_l} is defined as

$$H_{S_l} = \sum_{\substack{j=0 \\ P_j \in S_l}}^{|S_l|-1} c_j^{(l)} P_j^{(l)}. \tag{7}$$

S_l is the set of P_i terms in H_{S_l}, where $\{P_j, P_i\} = 0$ $\forall P_j \neq P_i \in S_l$. The process of finding such sets is discussed in Refs. [21,24,25] and formulated as a minimum clique cover problem. This is an NP-hard problem [26]; however, heuristic algorithms can provide sufficiently good approximate solutions to this problem [16,24,27]. Condition (2) is satisfied by renormalizing each anticommuting set:

$$H_{S_l} = \gamma_l \sum_{\substack{j=0 \\ P_j \in S_l}}^{|S_l|-1} \beta_j^{(l)} P_j^{(l)}, \tag{8}$$

where $\sum_j (\beta_j^{(l)})^2 = 1$ and $c_j^{(l)} = \gamma_l \beta_j^{(l)}$. The final condition is already satisfied, as all the coefficients c_i in H_q are real.

Using the unitary partitioning method, the qubit Hamiltonian is separated into m_c sets of unitary sums:

$$H_q = \sum_{i=0}^{m-1} c_i P_i = \sum_{l=0}^{m_c-1} H_{S_l} = \sum_{l=0}^{m_c-1} \gamma_l \left(\sum_{\substack{j=0 \\ P_j \in S_l}}^{|S_l|-1} \beta_j^{(l)} P_j^{(l)}\right). \tag{9}$$

Each sum $\sum_{P_j \in S_l} \beta_j^{(l)} P_j^{(l)}$ can be written as $R_l^\dagger Q_l R_l$. The expectation value of the Hamiltonian can therefore be obtained via

$$\langle\psi|\,H_q\,|\psi\rangle = \sum_{l=0}^{m_c-1} \gamma_l \,\langle\psi|\,R_l^\dagger Q_l R_l\,|\psi\rangle\,, \tag{10}$$

and only $m_c \leqslant m$ terms are estimated, at the expense of needing to implement R_l coherently within each circuit [25].

In this paper, the operator R_l required by unitary partitioning is implemented by either a sequence of rotations or a linear combination of unitaries (LCU) [28]. These methods follow

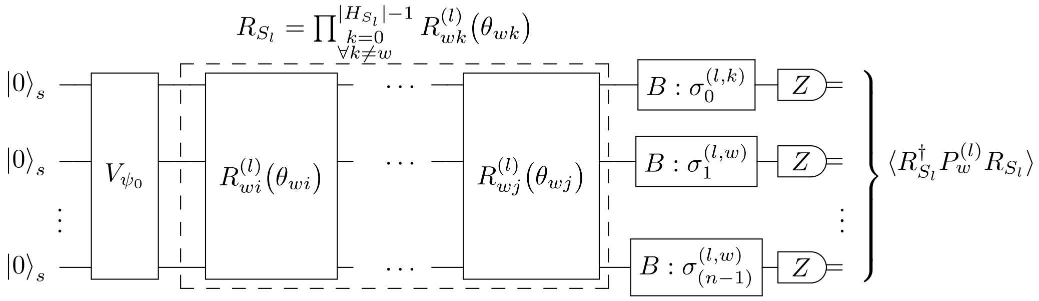

$R_{S_l} = \prod_{k=0,\ \forall k\neq w}^{|H_{S_l}|-1} R_{wk}^{(l)}(\theta_{wk})$
$|0\rangle_s$
$|0\rangle_s$
$|0\rangle_s$
V_{ψ_0}
$R_{wi}^{(l)}(\theta_{wi})$
$R_{wj}^{(l)}(\theta_{wj})$
$B : \sigma_0^{(l,k)}$
$B : \sigma_1^{(l,w)}$
$B : \sigma_{(n-1)}^{(l,w)}$
Z
Z
Z
$\langle R_{S_l}^\dagger P_w^{(l)} R_{S_l}\rangle$

FIG. 1. General quantum circuit implementation of unitary partitioning constructed as a sequence of rotations. The subscript s denotes system qubits. V_{ψ_0} is a unitary gate that prepares the ansatz state. To measure the qubits in the computational basis, the single-qubit gates $B \in \{H, R_x(-\pi/2), \mathcal{I}\}$ are required to perform a change of basis dependent on the Pauli operator $P_w^{(l)}$ measured.

the constructions outlined in Ref. [25]. The relevant details for each method are summarized in Secs. II A and II B. It will be shown that the new operator Q_l is simply a particular n-fold Pauli operator that we denote as $P_w^{(l)}$.

A. Ordered sequence-of-rotations approach

In this section, R_l is constructed such that it maps a completely anticommuting set H_{S_l} [Eq. (7)] to a single Pauli operator via conjugation: Formally, $R_l(\frac{H_{S_l}}{\gamma_l})R_l^\dagger = P_w^{(l)}$. To begin, a particular $P_j \in S_l$ is selected to be reduced to. This term is denoted by the index w and written as $P_w^{(l)}$, where l indexes the set. Once chosen, this operator is used to define the following set of Hermitian self-inverse operators [25]:

$$\left\{\mathcal{X}_{wk}^{(l)} = iP_w^{(l)}P_k^{(l)} \,\middle|\, \forall P_k \in S_l, \quad \text{where } k \neq w\right\}; \tag{11}$$

note that the coefficients β_w and β_k are not present. As every P_j operator in S_l anticommutes with all other operators in the set by definition, it is clear from Eq. (11) that $\mathcal{X}_{wk}^{(l)}$ will commute with all $\{P_j | \forall P_j \in S_l, \text{ where } j \neq w, k\}$ and anticommute with $P_w^{(l)}$ and $P_k^{(l)}$. This property is the crux of this conjugation approach.

The adjoint rotation generated by $\mathcal{X}_{wk}^{(l)}$ can be written [25]

$$R_{wk}^{(l)} = e^{(-i\frac{\theta_{wk}}{2}\mathcal{X}_{wk}^{(l)})}, \tag{12}$$

whose action on H_{S_l} is [25]

$$\begin{aligned} R_{wk}^{(l)}\left(\frac{H_{S_l}}{\gamma_l}\right)R_{wk}^{\dagger(l)} &= (\beta_k \cos\theta_{wk} - \beta_w \sin\theta_{wk})P_k^{(l)} \\ &\quad + (\beta_k \sin\theta_{wk} + \beta_w \cos\theta_{wk})P_w^{(l)} \\ &\quad + \sum_{\substack{P_j \in S_l \\ \forall j \neq w,k}} \beta_j P_j. \end{aligned} \tag{13}$$

The coefficient of $P_k^{(l)}$ can be made to go to 0, by setting $\beta_k \cos\theta_{wk} = \beta_w \sin\theta_{wk}$. This approach removes the term with index k and increases the coefficient of $P_w^{(l)}$ as $\beta_w \mapsto \sqrt{\beta_w^2 + \beta_k^2}$. This process is repeated over all indices excluding $k = w$ until only the $P_w^{(l)}$ term remains. This procedure can be concisely written using the following operator:

$$R_{S_l} = \prod_{\substack{k=0 \\ \forall k\neq w}}^{|H_{S_l}|-1} R_{wk}^{(l)}(\theta_{wk}), \tag{14}$$

which is simply a sequence of rotations. The angle θ_{wk} is defined iteratively at each step of the removal process, as the coefficient of $P_w^{(l)}$ increases at each step and thus must be taken into account. Importantly, the correct solution for θ_{wk} must be chosen given the signs of β_w and β_k [25]. The overall action of this sequence of rotations is

$$R_{S_l}\left(\frac{H_{S_l}}{\gamma_l}\right)R_{S_l}^\dagger = P_w^{(l)}. \tag{15}$$

Figure 1 shows the general circuit for unitary partitioning implementation as a sequence of rotations.

Applying R_{S_l} on $(\frac{H_{S_l}}{\gamma_l})$ by conjugation maps the unitary sum H_{S_l} to a single Pauli operator $P_w^{(l)}$, and unitary partitioning has been achieved. To summarize, in order to measure the expectation value of the Hamiltonian the following set of measurements are required:

$$\begin{aligned} H_q &= \sum_{i=0}^{m-1} c_i P_i = \sum_{l=0}^{m_c-1} H_{S_l} = \sum_{l=0}^{m_c-1} \gamma_l \left(\frac{H_{S_l}}{\gamma_l}\right) \\ &= \sum_{l=0}^{m_c-1} \gamma_l R_{S_l}^\dagger P_w^{(l)} R_{S_l}, \end{aligned} \tag{16}$$

where the number of terms in the qubit Hamiltonian requiring separate measurement is reduced as $m \mapsto m_c$. Note that $R_{S_l} \equiv R_l$; we use this notation to differentiate this assembly from the LCU construction.

B. Linear-combination-of-unitaries method

1. LCU technique overview

Given a complex operator as a linear combination of d unitary operators,

$$A = \sum_{j=0}^{d-1} \alpha_j U_j, \quad \|A\| \leqslant \|\vec{\alpha}\|_1 = \sum_{j=1}^{d} |\alpha_j|, \tag{17}$$

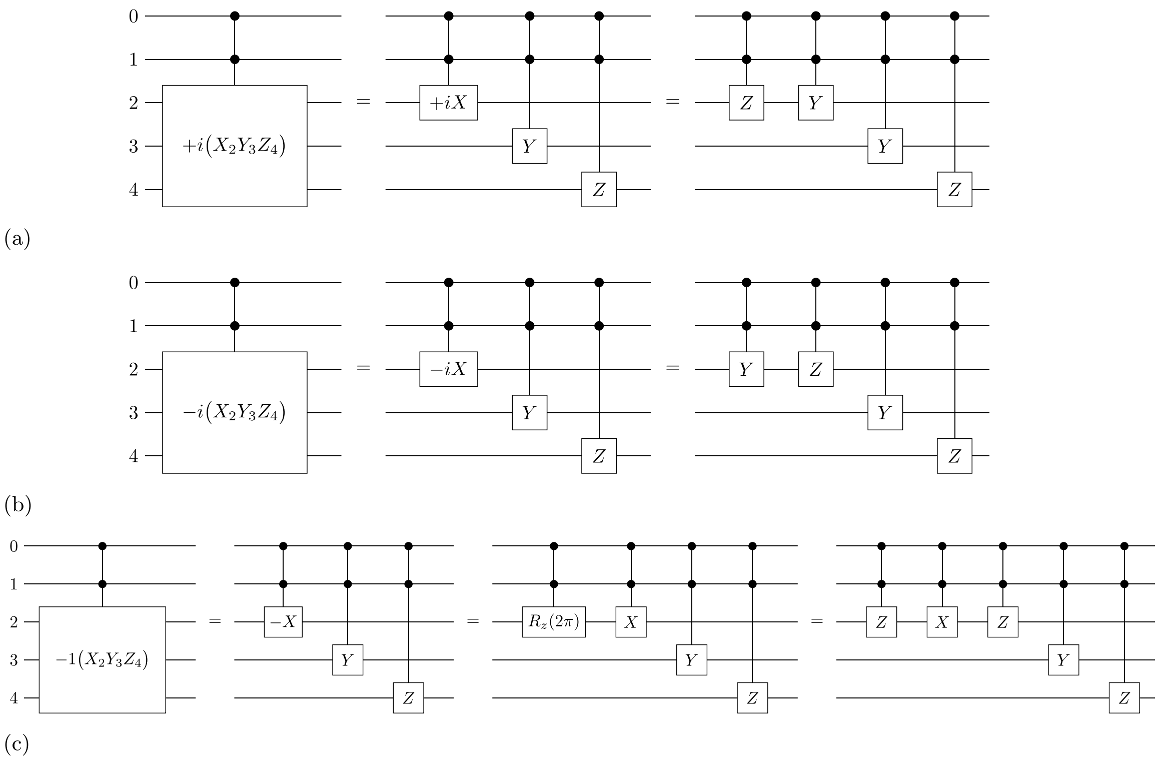


FIG. 2. Example implementations of n-fold tensor products of Pauli operators with complex phase of (a) $+i$, (b) $-i$, and (c) -1. Note that for (a), other possibilities are equally valid: $+i(X_2Y_3Z_4) = X_2(Z_3X_3)Z_4 = X_2Y_3(X_4Y_4)$. Likewise, for (b), $-i(X_2Y_3Z_4) = X_2(X_3Z_3)Z_4 = X_2Y_3(Y_4X_4)$, and for (c), $-1(X_2Y_3Z_4) = X_2(Y_3R_z[2\pi])Z_4 = X_2Y_3(Z_4R_z[2\pi]) = (Y_2X_2Y_2)Y_3Z_4 = X_2(X_3Y_3X_3)Z_4 = X_2(Z_3Y_3Z_3)Z_4 = X_2Y_3(X_4Z_4X_4) = X_2Y_3(Y_4Z_4Y_4)$.

where U_j are unitary operators (that are assumed to be easy to implement) and α_j are real positive coefficients. Without loss of generality, phase factors and signs can be absorbed into the unitaries U_j to make all $\alpha_j \geqslant 0$. Figure 2 shows how to do this for n-fold Pauli operators. The linear-combination-of-unitaries method offers a way to probabilistically implement such an operator using the two unitary operators G and U_{LCU} [28,29]:

$$U_{\text{LCU}} = \sum_{j=0}^{d-1} |j\rangle_a \langle j|_a \otimes U_j, \tag{18}$$

$$G = \sum_{j=0}^{d-1} \sqrt{\frac{\alpha_j}{\|\vec{\alpha}\|_1}} |j\rangle_a \langle 0|_a + \cdots, \tag{19}$$

where subscript a denotes the ancilla register. U_{LCU} is sometimes known as the "select" operator, and G is sometimes known as the "prepare" operator. The most important property of the unitary operator G is that the coefficients α_j only define the first column of the matrix—resulting in $G|0\rangle_a \mapsto |G\rangle_a$. The rest of the columns must be orthogonal but can have any values: Hence there is a freedom of choice when defining G. A practical note on this is if one finds a quantum circuit that performs $|0\rangle_a \mapsto |G\rangle_a$, then its action on other basis states will automatically be accounted for and G is completely defined (provided the quantum circuit is composed as a product of unitaries). To summarize the LCU method, first G is used to initialize the ancilla register: $G|0\rangle_a \mapsto |G\rangle_a$. The controlled unitary U_{LCU} is then applied across the system and ancilla registers, resulting in [29]

$$\begin{aligned} U_{\text{LCU}} |G\rangle_a |\psi\rangle_s \mapsto{}& |G\rangle_a \frac{A}{\|\vec{\alpha}\|_1} |\psi\rangle_s \\ &+ |G^{\perp}\rangle_a \sqrt{1 - \left\| \frac{A}{\|\vec{\alpha}\|_1} |\psi\rangle_s \right\|^2} |\psi\rangle_s . \end{aligned} \tag{20}$$

If $|G\rangle_a$ is measured in the ancilla register, then the state will be projected onto $\frac{A}{\|\vec{\alpha}\|_1} |\psi\rangle_s$, and A was successfully applied to the system state $|\psi\rangle_s$. If any other state in the ancilla register is measured (orthogonal complement $\in \mathcal{H}_{G^{\perp}}$), then the quantum state is projected into the wrong part of the Hilbert space, and $\frac{A}{\|\vec{\alpha}\|_1} |\psi\rangle_s$ is not performed.

The LCU method gives a probabilistic implementation of the matrix A, which has a probability of success given by

$$\begin{aligned} P_{\text{success}} &= \langle\psi|_s \langle G|_a \frac{A^{\dagger}}{\|\vec{\alpha}\|_1} \frac{A}{\|\vec{\alpha}\|_1} |G\rangle_a |\psi\rangle_s \\ &= \left(\frac{1}{\|\vec{\alpha}\|_1}^2\right) \langle\psi|_s A^{\dagger}A |\psi\rangle_s . \end{aligned} \tag{21}$$

As A can be a nonunitary matrix, $A^{\dagger}A$ does not necessarily result in the identity matrix. In general the success probably therefore depends on both $|\psi\rangle$ and $(\frac{1}{\|\vec{\alpha}\|_1}^2)$. For the special case when A is a unitary matrix, the success probability is given by $(\frac{1}{\|\vec{\alpha}\|_1}^2)$, due to $A^\dagger A = \mathcal{I}$.

To increase the probability of success, different techniques such as oblivious amplitude amplification [30,31] and amplitude amplification [32,33] can be used. However, extra coherent resources are required.

An alternate way to describe the LCU method is to note that the matrix A is in general not unitary. To build a unitary form, the normalized matrix A can be embedded into a larger Hilbert space by putting it into the upper left block of a unitary matrix [34]:

$$U = \begin{bmatrix} \frac{A}{\|\vec{\alpha}\|_1} & \cdot \\ \cdot & \cdot \end{bmatrix} \Rightarrow A = \|\vec{\alpha}\|_1((\langle 0|_a \otimes \mathcal{I}_s)U(\mathcal{I}_s \otimes |0\rangle_a)), \tag{22}$$

where each $[\cdot]$ represents a matrix of arbitrary elements. The reason to divide by $\|\vec{\alpha}\|_1$ is that unitary matrices must have eigenvalues in the form $e^{i\theta}$, ensuring a norm of 1. If, for example, $A = 1000Y$, putting this directly in the top left block of U would result in $\|U\| \neq 1$. In contrast, embedding $A/1000 = Y$ ensures that U can have a norm of 1 ($\|U\| = 1$)—now dependent on the remaining $[\cdot]$ parts.

The operator U is a probabilistic implementation of A and is commonly known as a "block encoding" of A. Overall the LCU method encodes the desired matrix as [29]

$$\langle G|_a U_{\text{LCU}} |G\rangle_a = \frac{A}{\|\vec{\alpha}\|_1}. \tag{23}$$

A final note on notation. When a matrix, e.g., A, is block encoded using U, we usually say "the unitary U gives a (α, k, ϵ)-block encoding of A." Here, α is the l_1 norm of the matrix to be block encoded. k is the extra ancilla qubits required to perform the block encoding. This depends on the number of operators in the linear combination of unitaries [Eq. (17)] and scales logarithmically as $k = \lceil \log_2(|A|) \rceil$, where $|A|$ is the number of operators in the linear combination.

Finally, ϵ is the error of the block encoding and is determined by [34]

$$\epsilon = ||A - \|\vec{\alpha}\|_1((\langle 0|_a \otimes \mathcal{I}_s)U(\mathcal{I}_s \otimes |0\rangle_a))||. \tag{24}$$

The LCU technique can be used to implement nonunitary operations [28,35], such as matrix inversion. This is achieved by constructing the required operator as a linear combination of unitaries. As the n-fold tensor product of Pauli operators including the n-fold identity operation form a complete operator basis, any $(2^n \times 2^n)$ complex operator can be built by different linear combinations of these unitary operators.

A toy example of the LCU method is given in the next section to illustrate the practical implementation of this technique. This can be skipped without loss of continuity.

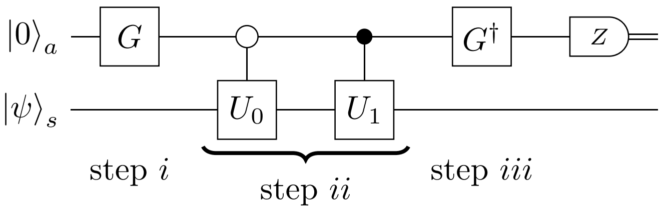


FIG. 3. Quantum circuit to implement $H = \alpha_0 U_0 + \alpha_1 U_1$ as a linear combination of unitaries.

2. Toy LCU example

Consider the Hamiltonian $H = \alpha_0 U_0 + \alpha_1 U_1$, where $\alpha_{0,1} \geqslant 0$ and $\|\vec{\alpha}\|_1 = |\alpha_0| + |\alpha_1|$. To implement this operator as a linear combination of unitaries, G [Eq. (19)] and U_{LCU} [Eq. (18)] must be defined. The construction of these operators is given by the definition of H. In this case, they will be

$$U_{\text{LCU}} = (|0\rangle_a \langle 0|_a \otimes U_0) + (|1\rangle_a \langle 1|_a \otimes U_1), \tag{25}$$

$$\begin{aligned} G &= \sqrt{\frac{\alpha_0}{\|\vec{\alpha}\|_1}} |0\rangle_a \langle 0|_a + \sqrt{\frac{\alpha_1}{\|\vec{\alpha}\|_1}} |1\rangle_a \langle 0|_a \\ &\quad + x\,|0\rangle_a \langle 1|_a + y\,|1\rangle_a \langle 1|_a \\ &= \begin{bmatrix} \sqrt{\frac{\alpha_0}{\|\vec{\alpha}\|_1}} & x \\ \sqrt{\frac{\alpha_1}{\|\vec{\alpha}\|_1}} & y \end{bmatrix}. \end{aligned} \tag{26}$$

The values of x and y can be anything that ensures that the columns of G are orthogonal. The following can be used:

$$G = \begin{bmatrix} \sqrt{\frac{\alpha_0}{\|\vec{\alpha}\|_1}} & -\sqrt{\frac{\alpha_1}{\|\vec{\alpha}\|_1}} \\ \sqrt{\frac{\alpha_1}{\|\vec{\alpha}\|_1}} & \sqrt{\frac{\alpha_0}{\|\vec{\alpha}\|_1}} \end{bmatrix}, \tag{27}$$

as discussed in Ref. [35]. The quantum circuit given in Fig. 3 shows the probabilistic implementation of H using the LCU method. Stepping through this circuit, we find

(i) $|0\rangle_a |\psi\rangle_s \overset{G\otimes\mathcal{I}_s}{\mapsto} \frac{1}{\sqrt{\|\vec{\alpha}\|_1}}(\sqrt{\alpha_0}\,|0\rangle_a + \sqrt{\alpha_1}\,|1\rangle_a)\,|\psi\rangle_s\,,$

(ii) $\overset{|0\rangle_a\langle 0|_a\otimes U_0}{\mapsto} \frac{1}{\sqrt{\|\vec{\alpha}\|_1}}(\sqrt{\alpha_0}\,|0\rangle_a U_0 |\psi\rangle_s + \sqrt{\alpha_1}\,|1\rangle_a |\psi\rangle_s),$

$\overset{|1\rangle_a\langle 1|_a\otimes U_1}{\mapsto} \frac{1}{\sqrt{\|\vec{\alpha}\|_1}}(\sqrt{\alpha_0}\,|0\rangle_a U_0 |\psi\rangle_s + \sqrt{\alpha_1}\,|1\rangle_a U_1 |\psi\rangle_s),$

(iii) $\overset{G^\dagger\otimes\mathcal{I}_s}{\mapsto} \frac{1}{\|\vec{\alpha}\|_1}(|0\rangle_a (\alpha_0 U_0 + \alpha_1 U_1)\,|\psi\rangle_s + \sqrt{\alpha_0\alpha_1}\,|1\rangle_a$
$(U_1 - U_0)\,|\psi\rangle_s).$

Measuring $|0\rangle_a$ on the ancilla line will project the state onto $(\alpha_0 U_0 + \alpha_1 U_1)\,|\psi\rangle_s = H\,|\psi\rangle_s$ up to a normalization and heralds a successful implementation of the linear-combination-of-unitaries method. If $|1\rangle_a$ is measured, then the state is projected into the wrong subspace, and H is not performed. Overall this approach gives a $(\|\vec{\alpha}\|_1, 1, 0)$-block encoding of H.

If H is a unitary operator, then the success probability [Eq. (21)] is $(\frac{1}{\|\vec{\alpha}\|_1})^2$. However, if H is nonunitary, then the success probability depends on $(\frac{1}{\|\vec{\alpha}\|_1})^2 \langle\psi|_s (\alpha_0 U_0^\dagger + \alpha_1 U_1^\dagger)(\alpha_0 U_0 + \alpha_1 U_1)\,|\psi\rangle_s$. Consider the case of $H = \alpha_0 U_0 + \alpha_1 U_1 = \frac{1}{2}\mathcal{I} + \frac{1}{2}Z = |0\rangle\langle 0|$. This matrix defines the projector onto the all-zero state on the system register's qubit and is a nonunitary operation. Clearly, the success probability of block-encoding this operator will depend heavily on $|\psi\rangle_s$ (and whether it has overlap with $|0\rangle_s$).

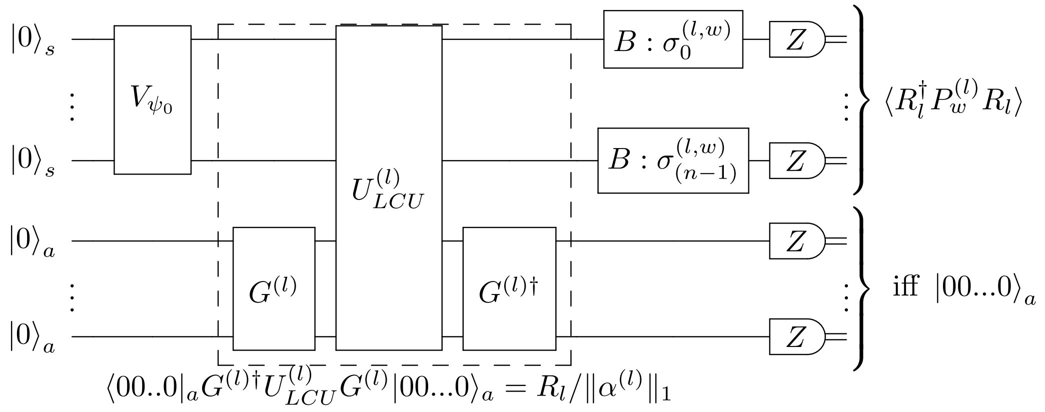

$|0\rangle_s$
$|0\rangle_s$
$|0\rangle_a$
$|0\rangle_a$
V_{ψ_0}
$U^{(l)}_{LCU}$
$G^{(l)}$
$G^{(l)\dagger}$
$B:\sigma_0^{(l,w)}$
$B:\sigma_{(n-1)}^{(l,w)}$
Z
$\langle R_l^\dagger P_w^{(l)} R_l\rangle$
iff $|00...0\rangle_a$
$\langle 00..0|_a G^{(l)\dagger} U^{(l)}_{LCU} G^{(l)} |00...0\rangle_a = R_l/\|\alpha^{(l)}\|_1$

FIG. 4. General quantum circuit implementation of unitary partitioning constructed as a LCU. The subscripts s and a denote the system and ancilla registers, respectively. V_{ψ_0} is a unitary gate that prepares the ansatz state. To measure the qubits in the computational basis, the single-qubit gates $B \in \{H, R_x(-\pi/2), \mathcal{I}\}$ are required to perform a change of basis dependent on the Pauli operator $P_w^{(l)}$ measured.

3. LCU approach to unitary partitioning

Analogous to the sequence-of-rotations method (Sec. II A), R_l will be applied by conjugation to an anticommuting set H_{S_l} to give a single Pauli operator. However, unlike the previous method, where R_l was achieved by a sequence of rotations [Eq. (14)], here it is built via a linear combination of unitaries (LCU; Fig. 4). An overview of the LCU method is given in Sec II B 2.

To construct R_l via LCU, we first need to manipulate each of the m_c anticommuting sets (H_{S_l}) that the qubit Hamiltonian was partitioned into [Eq. (9)]. As before, a particular Pauli operator $P_j \in S_l$ in each set is selected to be reduced to. Again this will be denoted by the index w and written as $P_w^{(l)}$, where l indexes the set. At this point, constructions of R_l diverge. To begin, we define the operator $H_{S_l \setminus \{P_w^{(l)}\}}$:

$$H_{S_l \setminus \{P_w^{(l)}\}} = \sum_{\forall k \neq w}^{|H_{S_l}|-1} \delta_k P_k, \tag{28a}$$

$$\text{where} \quad \sum_{\forall k \neq w}^{|H_{S_l}|-1} \delta_k^2 = 1. \tag{28b}$$

Taking each normalized set H_{S_l} [Eq. (8)] and rewriting them with the term we are reducing to ($\beta_w^{(l)} P_w^{(l)}$) outside the sum,

$$\frac{H_{S_l}}{\gamma_l} = \beta_w^{(l)} P_w^{(l)} + \sum_{\substack{j=0 \\ \forall j \neq w}}^{|H_{S_l}|-2} \beta_j^{(l)} P_j^{(l)}; \tag{29}$$

by renormalizing the remaining sum in Eq. (29),

$$\frac{H_{S_l}}{\gamma_l} = \beta_w^{(l)} P_w^{(l)} + \Omega_l \sum_{\substack{j=0 \\ \forall j \neq w}}^{|H_{S_l}|-2} \delta_j^{(l)} P_j^{(l)}, \tag{30a}$$

where

$$\sum_{\substack{j=0 \\ \forall j \neq w}}^{|H_{S_l}|-2} \delta_j^{2(l)} = 1 \tag{30b}$$

and

$$\beta_j^{(l)} = \Omega_l \delta_j^{(l)}. \tag{30c}$$

We can substitute Eq. (28a) into Eq. (30a):

$$\frac{H_{S_l}}{\gamma_l} = \beta_w^{(l)} P_w^{(l)} + \Omega_l H_{S_l \setminus \{P_w^{(l)}\}}, \tag{31}$$

where $\beta_w^{2(l)} + \Omega_l^2 = 1$. In this form we can use the trigonometric identity $\cos^2(\theta) + \sin^2(\theta) = 1$ to define the following operator:

$$H_w^{(l)} = \cos\left(\phi_w^{(l)}\right) P_w^{(l)} + \sin\left(\phi_w^{(l)}\right) H_{S_l \setminus \{P_w^{(l)}\}}. \tag{32}$$

Comparing Eqs. (31) and (32), it is clear that $\cos(\phi_w^{(l)}) = \beta_w^{(l)}$ or $\sin(\phi_w^{(l)}) = \Omega_l$. Next using the definition of $H_w^{(l)}$ in Eq. (32), it was shown in Ref. [25] that one can consider rotations of H_w around an axis that is Hilbert-Schmidt orthogonal to both $H_{S_l \setminus \{P_w^{(l)}\}}$ [Eqs. (28a) and (28b)] and $P_w^{(l)}$:

$$\mathcal{X}^{(l)} = \frac{i}{2}\left[H_{S_l \setminus \{P_w^{(l)}\}}, P_w^{(l)}\right] = i \sum_{\substack{k=0 \\ \forall k \neq w}}^{|H_{S_l}|-2} \delta_k^{(l)} P_k^{(l)} P_w^{(l)}. \tag{33}$$

$\mathcal{X}^{(l)}$ anticommutes with $H_w^{(l)}$, is self-inverse, and has the following action [25]:

$$\mathcal{X}^{(l)} H_w^{(l)} = i\left(-\sin\phi_w^{(l)} P_w^{(l)} + \cos\phi_w^{(l)} H_{S_l \setminus \{P_w^{(l)}\}}\right). \tag{34}$$

This defines the rotation

$$R_l = e^{\left(-i\frac{\alpha^{(l)}}{2}\mathcal{X}^{(l)}\right)} = \cos\left(\frac{\alpha^{(l)}}{2}\right)\mathcal{I} - i\sin\left(\frac{\alpha^{(l)}}{2}\right)\mathcal{X}^{(l)} \tag{35a}$$

$$= \cos\left(\frac{\alpha^{(l)}}{2}\right)\mathcal{I} - i\sin\left(\frac{\alpha^{(l)}}{2}\right)\left(i\sum_{\substack{k=0 \\ \forall k \neq w}}^{|H_{S_l}|-2} \delta_k^{(l)} P_k^{(l)} P_w^{(l)}\right) \tag{35b}$$

$$= \cos\left(\frac{\alpha^{(l)}}{2}\right)\mathcal{I} + \sin\left(\frac{\alpha^{(l)}}{2}\right)\sum_{\substack{k=0 \\ \forall k \neq w}}^{|H_{S_l}|-2} \delta_k^{(l)} P_{kw}^{(l)}, \tag{35c}$$

where $P_k^{(l)}P_w^{(l)} = P_{kw}^{(l)}$. Importantly, $P_{kw}^{(l)}$ will be another tensor product of Pauli operators, as products of n-fold Pauli operators will yield another operator in the Pauli group. The adjoint action of this rotation on $H_w^{(l)}$ is

$$R_l H_w^{(l)} R_l^\dagger = \sin\left(\phi_w^{(l)} - \theta^{(l)}\right) H_{S_l \setminus \{P_w^{(l)}\}} + \cos\left(\phi_w^{(l)} - \theta^{(l)}\right) P_w^{(l)}. \tag{36}$$

By setting $\theta^{(l)} = \phi_w^{(l)}$, the coefficient of $H_{S_l \setminus \{P_w^{(l)}\}}$ will go to zero, and we achieve the intended result of $R_l H_w^{(l)} R_l^\dagger = P_w^{(l)}$. To build R_l by the LCU method, we use its definition in Eq. (35c). In practice, it is easier to rewrite Eqs. (35a)–(35c) using the fact that all P_{kw} and $\mathcal{I}$ are in the Pauli group. The terms can thus be combined into a single sum:

$$R_l = \alpha\mathcal{I} + \sum_{\substack{k=0 \\ \forall k \neq w}}^{|H_{S_l}|-2} \alpha_k P_{kw}^{(l)} \tag{37a}$$

$$= \sum_{\substack{q=0 \\ \forall q \neq w}}^{|H_{S_l}|-1} \alpha_q P_q^{(l)}. \tag{37b}$$

Note that all α_q must be real and $\alpha_q \geqslant 0\ \forall q$. This is achieved by absorbing any signs and complex phases into $P_{kw}^{(l)}$; hence these operators are n-fold tensor Pauli operators up to a complex phase. When written in this form, it is easy to define the operators G [Eq. 19] and U_{LCU} [Eq. (18)],

$$G^{(l)} = \sum_{q=0}^{|H_{S_l}|-1} \sqrt{\frac{\alpha_q}{\|\vec{\alpha}_q\|_1}}\, |q\rangle_a \langle 0|_a + \cdots, \tag{38}$$

$$U_{\mathrm{LCU}}^{(l)} = \sum_{q=0}^{|H_{S_l}|-1} |q\rangle_a \langle q|_a \otimes P_q^{(l)}, \tag{39}$$

that are required to perform R_l as a LCU. Overall the operator is encoded as

$$\langle 0|_a G^{(l)\dagger} U_{\mathrm{LCU}}^{(l)} G^{(l)} |0\rangle_a = \langle G^{(l)}|_a U_{\mathrm{LCU}}^{(l)} |G^{(l)}\rangle_a = \frac{R_l}{\|\vec{\alpha}_q\|_1}. \tag{40}$$

Without using amplitude amplification, as R_l is unitary, the probability of success is given by the square of the l_1 norm of R_l. Note that the l_1 norm is defined as $\|\vec{\alpha}_q\|_1 = \sum_{q=0}^{|H_{S_l}|-1} |\alpha_q|$.

III. NUMERICAL STUDY

The ability of the unitary partitioning measurement reduction strategy is dependent on the problem Hamiltonian. To assess the performance of each implementation, we investigate the application to Hamiltonians of interest in quantum chemistry.

A. Method

We consider Hamiltonians for H_2 and LiH molecules employing the STO-3G and STO-6G basis sets, respectively. These were calculated using OPENFERMION-PYSCF and converted into the qubit Hamiltonian using the Bravyi-Kitaev transformation in OPENFERMION [36,37].

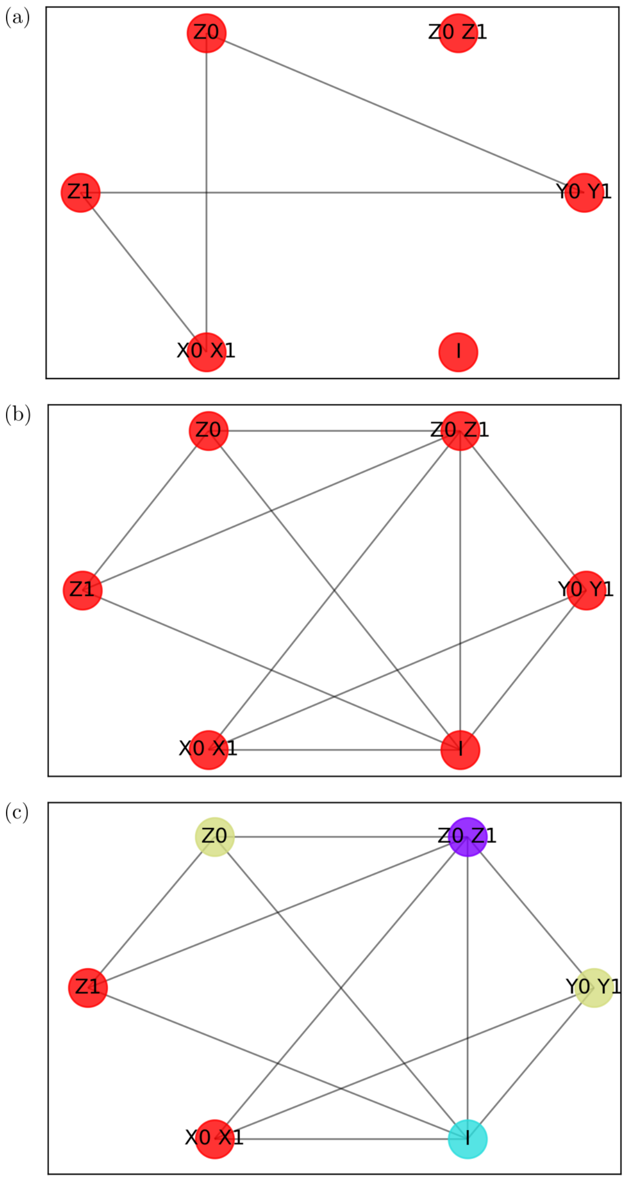


FIG. 5. Illustration of graph coloring approach to finding anticommuting sets of a given Hamiltonian. The graph in (a) shows the qubit Hamiltonian of H_2, where edges connect anticommuting P_i nodes. The complementary graph of (a) is given in (b). A graph coloring of (b) gives the anticommuting clique cover, which is shown in (c).

Partitioning into anticommuting sets H_{S_l} was performed using NETWORKX [38]. First, a graph of the qubit Hamiltonian was built, where each node is a term in the Hamiltonian. Next, edges are put between nodes on the graph that anticommute. Finally, a graph coloring of the complement graph was performed. This searches for the minimum number of colors required to color the graph, where no neighbors of a node can have the same color as the node itself. The "largest first" coloring strategy in NETWORKX was used [38,39]. Each unique color represents an anticommuting clique. Figure 5 shows

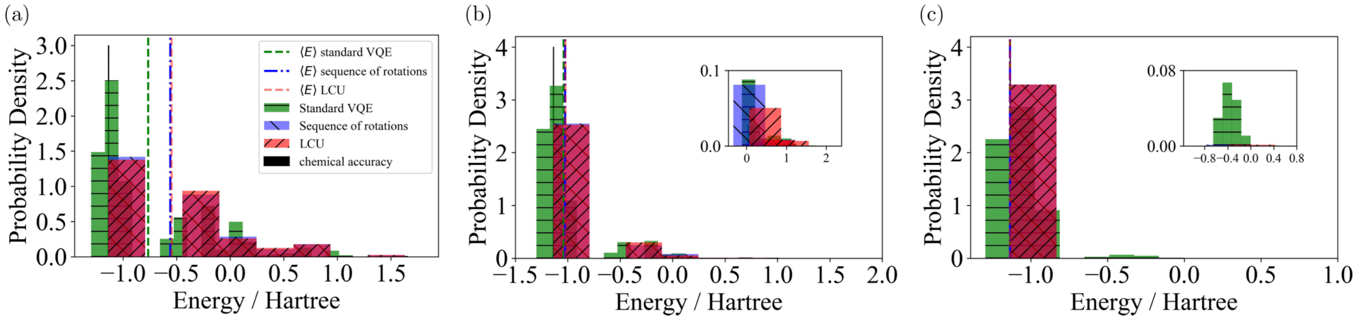


FIG. 6. Probability density functions of single-shot VQE estimates of the ground-state energy e_j of H_2. A bin is given to every possible energy outcome. Note that $E_{\mathrm{FCI}} = -1.137\,28$ Ha, $t_o = 5$, $t_p = 3$, and $M = 1.2663 \times 10^6$. The raw results from ibmqx2 are given in (a), (b) shows these results with measurement error mitigation applied, and (c) gives results from simulation on a noise-free QPU.

the method applied to H_2. This approach is the minimum clique cover problem mapped to a graph coloring problem. Further numerical details for each Hamiltonian can be found in Appendix A.

The input state $|\psi(\vec{\theta})\rangle$ for all calculations was the exact full configuration interaction (FCI) ground state, found by diagonalizing the Hamiltonian. As our aim was to investigate different implementations of unitary partitioning, this meant that the ansatz optimization step in VQE was not required.

For the simulations performed on IBM's ibmqx2 quantum processing unit (QPU), a measurement error mitigation strategy available in QISKIT was utilized and is a simple inversion procedure [40]. The quantum circuits required were generated by the qiskit.ignis *complete_meas_cal* method and executed alongside each separate ibmqx2 experiment, with the maximum number of shots (8192). This sampling cost was not included in the number of calls to the quantum device. The *CompleteMeasFitter* method in qiskit.ignis [41] was used to generate the calibration matrix required for measurement error mitigation [40,41].

We denote the number of terms in the original Hamiltonian by t_o and t_p for the unitary partitioned Hamiltonian. For each implementation, we fix M, which is the total number of calls to the QPU. This can be thought of as a measurement budget. The total number of e_j samples—single-shot estimates of all n-fold Pauli operators in the Hamiltonian—is $N_o = M/t_o$ for the original Hamiltonian and $N_p = M/t_p$ for the partitioned Hamiltonian. Clearly, because unitary partitioning reduces the number of terms in H_q, more energy samples are obtained for a fixed M.

B. Results

For a given preparation of the true ground state of H_2, we compare both implementations of measurement reduction by unitary partitioning against a standard VQE calculation on IBM's open access quantum device [ibmq 5 Yorktown (ibmqx2)] and QISKIT's qasm simulator [41]. The quantum circuits required are given in Appendix A. Figure 6 shows the distribution of single-shot energy estimates of all three techniques applied to molecular hydrogen. The average energy is given by $\langle E\rangle = \frac{1}{N}\sum_{j=0}^{N-1} e_j$. To compare each method, the measurement budget was fixed to $M = 1.2663 \times 10^6$. A calibration matrix method available in QISKIT was used to mitigate measurement errors and was used to amend the raw outputs from ibmqx2.

The qubit Hamiltonian for H_2 has five terms, which is reduced to three terms by unitary partitioning, not including the identity term. The number of energy estimates e_j obtained was 253 260 for standard VQE and 5/3 of this for unitary partitioning by the sequence-of-rotations method. This is because the smaller number of terms allowed a correspondingly larger number of samples to be taken for a fixed M. The total number of e_j samples from ibmqx2 for these techniques was reduced to 253 074 and 421 951 after measurement error mitigation was applied. The LCU approach to unitary partitioning is probabilistic and requires postselection on the all-zero state of the ancilla register. After postselection, our simulation of unitary partitioning as a LCU on ibmqx2 gave 333 407 raw e_j samples and 332 763 e_j samples after measurement error mitigation was applied to the raw output. Our emulation of this method on a noise-free QPU gave 336 390 e_j samples after postselection. The theoretical maximum possible number of samples for LCU would be the same as the sequence-of-rotations method if all samples obtained were successful.

The reason a normal distribution is not obtained is due to the number of terms in the qubit Hamiltonian for H_2. At most only 32 distinct values of e_j are possible for standard VQE and 8 under unitary partitioning, and so we do not expect the central limit argument to apply here.

To investigate the distribution of energies obtained from each method in more detail, we simulated the larger problem of LiH using QISKIT's *statevector_simulator* [41]. Further details are given in Appendix A 2. Figure 7 summarizes the results. Again, each data point is an energy estimate found from the weighted measurement outcomes of a single-shot VQE run. The standard qubit Hamiltonian for this problem is made up of 630 terms; after applying unitary partitioning, the Hamiltonian is made up of 102 terms, not including the identity term. The measurement budget was fixed at $M = 1.018\,521 \times 10^9$. The total number of energy estimates e_j for standard VQE, the sequence-of-rotations, and the LCU methods after postselection were 1 616 700, 9 985 500, and 1 447 349, respectively.

We performed the Kolmogorov-Smirnov [42] and Shapiro-Wilk tests [43] on the data in Fig. 7 to check for normality. The P values obtained in all cases were smaller than 0.05, and thus we could not assume a normal distribution. This

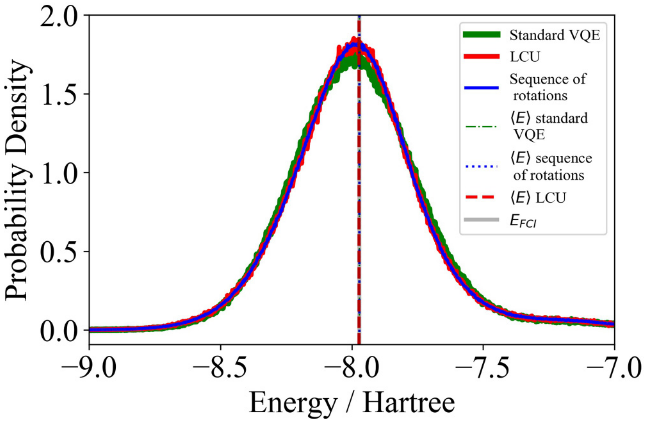


FIG. 7. Probability density function of single-shot VQE estimates of the ground-state energy e_j of LiH, from a noise-free QPU simulation. The number of bins was set to 2500, and the center of each was plotted. Note $E_{\rm FCI} = -7.971\,18$ Ha, $t_o = 630$, $t_p = 102$, and $M = 1.018\,521 \times 10^9$. Individual plots are given in Appendix A 3.

may be caused by insufficient samples allowing convergence to the central limit or the problem size still being too small. To estimate the statistics of the true distribution, we thus employed bootstrap resampling [44].

C. Discussion

In our results, for a fixed measurement budget M we obtain a set of independent identically distributed random energy samples $\{e_1, e_2, \dots, e_N\}$. The standard deviation of this sample σ_e converges to the true standard deviation as the number of samples increases. As the number of samples increases, the error on the sample mean decreases. The standard error of the mean (SEM) is defined as

$$\text{SEM} = \sigma_e/\sqrt{N} \tag{41}$$

for N energy samples. Because we are considering a fixed measurement budget M, the effect of unitary partitioning will be to increase the number of energy samples N and hence reduce the SEM.

To benchmark each method, we compare σ_e and the SEM of the ground-state energy samples. The 95% confidence intervals (CIs) were calculated using bootstrapping with 10 000 resamples with replacement. The full statistical analysis is given in Table I.

Qualitatively, the noise-free LiH simulation results in Fig. 7 show that VQE with unitary partitioning applied as either a LCU or a sequence of rotations give a similar distribution of energies compared with standard VQE. This is expected, as unitary partitioning leaves a given molecular Hamiltonian H_q unchanged [Eq. (9)]; thus both the standard deviation $\sigma = \sqrt{\langle {H_q}^2 \rangle - \langle H_q \rangle^2}$ and full configuration interaction ground-state energy $E_{\rm FCI}$ will remain unchanged.

Quantitatively, the σ_e of ground-state energy estimates of LiH for each method were very similar, with the largest difference being 4.8 mHa. Note that σ of the full population is independent of the number of samples N taken. Thus even though the distributions in Fig. 7 look very similar, the number of data points in each curve is significantly different and therefore the SEM is significantly different for each implementation.

In contrast, for the noise-free simulation of H_2 [Fig. 6(c)], the sample standard deviations of e_j from VQE with unitary partitioning applied were an order of magnitude lower than those from standard VQE. We expect that this is due to the small number of distinct e_j outcomes for this specific problem under unitary partitioning.

As unitary partitioning is designed to require fewer terms to be measured, for a fixed measured budget M, the total number of energy estimates will be larger. The sequence-of-rotations construction of unitary partitioning is deterministic and will always give more e_j samples than conventional VQE. Hence the SEMs of both H_2 and LiH noiseless simulations using the sequence-of rotations method were an order of magnitude lower than those of standard VQE. On the other hand, the LCU realization of unitary partitioning is probabilistic. Even though fewer terms need measurement, postselection requires some samples to be discarded. We see this in the

TABLE I. The mean, standard deviation, and standard error of the mean for each method calculating the ground-state energies of H_2 and LiH using single-shot VQE. The simulator back end represents a noise-free QPU emulator, and ibmqx2 represents a real quantum device. ibmqx2-raw are the raw experimental results from the QPU, and ibmqx2-mit are the results with measurement error mitigation applied. 95% confidence intervals (CIs) were calculated using bootstrap resampling [44]. SeqRot, sequence of rotations. Here, 3.1661e-01, 3.1661 × 10^{-1}.

Molecule	Method	Back end	N	$\langle E \rangle$ (Ha)	$\langle E \rangle$ 95% CI (Ha)	σ_e (Ha)	σ_e 95% CI (Ha)	SEM (Ha)	SEM 95% CI (Ha)
H_2	LCU	ibmqx2 - mit	332763	−1.0212	[−1.0222, −1.0201]	3.1661e-01	[3.1504e-01, 3.1816e-01]	5.4886e-04	[5.4617e-04, 5.5158e-04]
H_2	SeqRot	ibmqx2 - mit	421951	−1.0246	[−1.0256, −1.0237]	3.1984e-01	[3.1838e-01, 3.2132e-01]	4.9239e-04	[4.9017e-04, 4.9459e-04]
H_2	standard	ibmqx2 - mit	253074	−1.0381	[−1.0394, −1.0367]	3.4240e-01	[3.4053e-01, 3.4424e-01]	6.8063e-04	[6.7697e-04, 6.8432e-04]
H_2	LCU	ibmqx2 - raw	333407	−0.54537	[−0.54757, −0.54315]	6.5287e-01	[6.5129e-01, 6.5442e-01]	1.1307e-03	[1.1279e-03, 1.1334e-03]
H_2	SeqRot	ibmqx2 - raw	422100	−0.55863	[−0.56063, −0.55666]	6.5025e-01	[6.4885e-01, 6.5162e-01]	1.0009e-03	[9.9873e-04, 1.0030e-03]
H_2	standard	ibmqx2 - raw	253260	−0.76468	[−0.76685, −0.76247]	5.6625e-01	[5.6441e-01, 5.6810e-01]	1.1252e-03	[1.1215e-03, 1.1287e-03]
H_2	LCU	simulator	336390	−1.1373	[−1.1374, −1.1373]	1.5655e-02	[1.4307e-02, 1.7082e-02]	2.6992e-05	[2.4692e-05, 2.9392e-05]
H_2	SeqRot	simulator	422100	−1.1373	[−1.1373, −1.1372]	1.6286e-02	[1.5084e-02, 1.7533e-02]	2.5068e-05	[2.3147e-05, 2.6968e-05]
H_2	standard	simulator	253260	−1.1370	[−1.1377, −1.1363]	1.7752e-01	[1.7634e-01, 1.7872e-01]	3.5276e-04	[3.5040e-04, 3.5512e-04]
LiH	LCU	simulator	1447349	−7.9719	[−7.9723, −7.9714]	2.7268e-01	[2.7213e-01, 2.7322e-01]	2.2665e-04	[2.2620e-04, 2.2711e-04]
LiH	SeqRot	simulator	9985500	−7.9712	[−7.9714, −7.9710]	2.7292e-01	[2.7271e-01, 2.7312e-01]	8.6367e-05	[8.6303e-05, 8.6432e-05]
LiH	standard	simulator	1616700	−7.9716	[−7.9720, −7.9712]	2.6817e-01	[2.6773e-01, 2.6860e-01]	2.1091e-04	[2.1056e-04, 2.1125e-04]

simulation for LiH, where the LCU approach actually has the fewest e_j samples at 1 447 349 compared with 1 616 700 for standard VQE. As the σ_e of all three approaches are similar, the LCU implementation has the highest SEM in this case. The advantage over standard VQE is thus dependent on the success probability, which for each circuit is given by the inverse l_1 norm squared of the operator to be implemented as a LCU. Importantly, postselection is only performed on the ancilla register. The success probability is inversely proportional to the dimension of the ancilla Hilbert space. The number of ancilla qubits scales logarithmically with the number of terms in each anticommuting set ($n_{\text{ancilla}} = \lceil \log_2(|H_{S_l}| - 1) \rceil$); the dimension of the ancilla Hilbert space, and hence success probability, is inversely proportional to the size of the anticommuting sets.

The experimental results for H_2 on ibmqx2 show that applying the unitary partitioning technique does not appreciably change the performance of VQE, when combined with error mitigation techniques. We suspect that this is due to the extra coherent resources required to perform R_l causing an increase in errors, which offsets the improvement of the SEM given by the technique. We expect this to be mitigated as gate fidelities increase.

The experimental execution of R_l by LCU on ibmqx2 performed comparably to the sequence-of-rotations realization. Ignoring postselection issues, the LCU algorithm is more complex and requires more qubits to implement. We believe that this motivates further examination of the use of more advanced quantum algorithms on NISQ devices.

A particular feature of our results from ibmqx2 (Fig. 6) is that the mean ground-state energy obtained is overestimated by a seemingly constant amount. We suspect that this could be due to two effects. Firstly, our ansatz circuit prepares the ground state. Any coherent errors in this circuit will increase the energy of the state prepared by virtue of the variational principle [45]. Secondly, inspecting the qubit Hamiltonian for H_2, most coefficients are positive. As our results overestimate the energy, this implies that measurement outcomes of each n-fold Pauli operator are more frequently $+1$ causing each estimate of e_j to be larger. This could be an indication of a higher $|0\rangle$ count on each qubit or $P(0|1) > P(1|0)$.

The single-qubit gate error rates of IBM QPUs are in the range of 0.1–0.3%, and 2-qubit gate error rates are in the range of 2–5% [46]. The most error-prone operation is measurement, and ibmqx2 on average has a measurement error rate of 4%, but this can be much higher (13%) [46]. This large measurement error is apparent when comparing the raw and measurement-error-mitigated results from the QPU simulation of H_2. In future experiments, it would be interesting to improve measurement fidelity, for example, by using invert-and-measure designs [46] as well as flipping the qubit encoding ($|0\rangle \mapsto |1\rangle$ and $|1\rangle \mapsto |0\rangle$) as in Ref. [47], or by other mitigation schemes [48].

In the original work in which these techniques were proposed, it was shown that the variance of the different methods should be similar [25]. This is observed on both the QPU emulator and the quantum device.

Crucially, when partitioning the qubit Hamiltonian into anticommuting cliques [Eq. (9)], the greatest measurement reduction is obtained if the minimum clique cover is found. This cover has the fewest H_{S_l} sets possible. However, nonoptimal clique covers still give a measurement reduction. As the size of the quantum circuit for R_l is proportional to the number of terms in H_{S_l}, we propose that for practical applications a nonoptimal clique cover is beneficial. By splitting the problem Hamiltonian into pairs of anticommuting operators ($|H_{S_l}| = 2 \ \forall \{l\}_{l=0,1,\ldots,m_c-1}$), the extra coherent resources required to perform R_l are experimentally realistic for current and near-term devices. This offers a constant factor improvement to the number of measurements required. A detailed circuit analysis of different implementations of unitary partitioning is given in the next section.

IV. CIRCUIT ANALYSIS

In order to investigate the circuit depth of each technique, we consider circuits made up of arbitrary single-qubit and CNOT gates. However, often when analyzing fault-tolerant protocols it is common to consider the universal gate set of Clifford and T gates. The LCU method only requires arbitrary rotations in the operator G, whereas the sequence-of-rotations method requires a rotation for every operator in an anticommuting set, apart from for $P_w^{(l)}$. The relative depth of LCU vs sequence-of-rotations circuits would be interesting to explore in this setting.

A. Sequence-of-rotations circuit analysis

In Sec. II A, it was shown that R_{S_l} could be constructed by a sequence of $R_{wk}^{(l)}$ rotations [Eq. (14)]. Writing the rotation operators in their exponentiated form,

$$R_{S_l} = \prod_{\substack{k=0 \\ \forall k \neq w}}^{|H_{S_l}|-1} R_{wk}^{(l)}(\theta_{wk}) = \prod_{\substack{k=0 \\ \forall k \neq w}}^{|H_{S_l}|-1} e^{\left(-i\frac{\theta_{wk}}{2}\mathcal{X}_{wk}^{(l)}\right)}. \tag{42}$$

We need to consider the cost to perform each $R_{wk}^{(l)}$ rotation. Whitfield *et al.* [49] show how to build the required quantum circuits for these types of operators, and an example is illustrated in Fig. 8.

Every $R_{wk}^{(l)}$ circuit will require $\mathcal{O}(2(N_s - 1))$ CNOT gates, 1 $R_z(\theta)$ gate, and $\mathcal{O}(2N_s)$ change-of-basis gates $\{H, R_x(\theta)\}$. Here, N_s is the number of system qubits. A single $R_{wk}^{(l)}$ is needed for each term in the set S_l, apart from for $P_w^{(l)}$.

The total number of rotations that make up the full sequence-of-rotations operator R_{S_l} is therefore $|H_{S_l}| - 1$. $|H_{S_l}|$ is the size of the anticommuting set. The overall gate count scales as $\mathcal{O}((2N_s + 1)(|H_{S_l}| - 1))$ single-qubit gates and $\mathcal{O}(2(N_s - 1)(|H_{S_l}| - 1))$ CNOT gates.

We note that there is a choice in the ordering of $R_{wk}^{(l)}$ when constructing R_{S_l}. By choosing an ordering that maximizes the common substring between Pauli strings defining $R_{wk}^{(l)}$ (lexicographical order), it is possible to cancel the common change-of-basis gates between subsequent $R_{wk}^{(l)}$ rotations. We refer the reader to Ref. [50], which gives further possible gate cancellations—including CNOT cancellations. This can significantly reduce the circuit depth when constructing R_{S_l}.

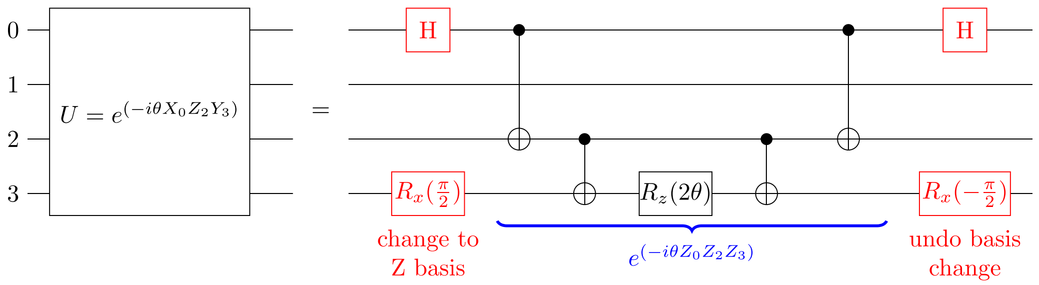


FIG. 8. Quantum circuit to perform a unitary operator given as an exponentiated n-fold tensor product of Pauli operators [49].

B. LCU implementation

In the linear-combination-of-unitaries approach to unitary partitioning, R_l is written as a linear combination of n-fold Pauli operators [Eqs. (37a) and (37b)]—up to a complex sign. Figure 2 shows how to implement such operators.

Such an operator can be implemented using the LCU technique. To achieve this, the gates required to realize $G^{(l)}$ [Eq. (38)] and $U_{\text{LCU}}^{(l)}$ [Eq. (39)] are required. We will not explicitly consider the construction of G, as it heavily depends on the ancilla state required and many different approaches are possible. In the worst case, without introducing any additional qubits, $\mathcal{O}(N_c \log_2(N_c)^2)$ standard 1- and 2-bit gate operations are required [51]. Here, the number of control qubits N_c is given by the number of operators that $U_{\text{LCU}}^{(l)}$ is constructed from. In this case, $N_c = \lceil \log_2(|H_{S_l}| - 1) \rceil$.

On the other hand, the quantum circuit to construct U_{LCU} is well defined. Overall, $(|H_{S_l}| - 1)$ N_c-bit controlled P_q gates are required. To efficiently construct each control P_q gate, $N_w = (N_c - 1)$ work qubits are employed. The control states of the ancilla qubits are stored on these work qubits using Toffoli gates [52]. An example is shown in Fig. 9. For every control P_i a cascade of $\mathcal{O}(2(N_c - 1))$ Toffoli gates are required.

With no circuit simplifications on the ancilla and work qubit registers, the number of Toffoli gates required scales as $\mathcal{O}(2^{N_c}(2N_c - 2))$. However, significant simplifications can be made. Here, we assume that all 2^{N_c} control states are required.

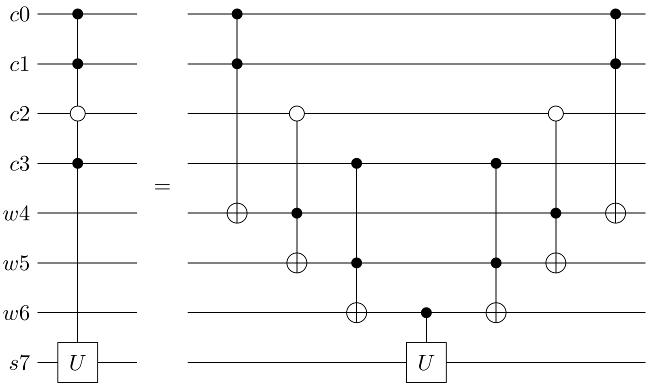
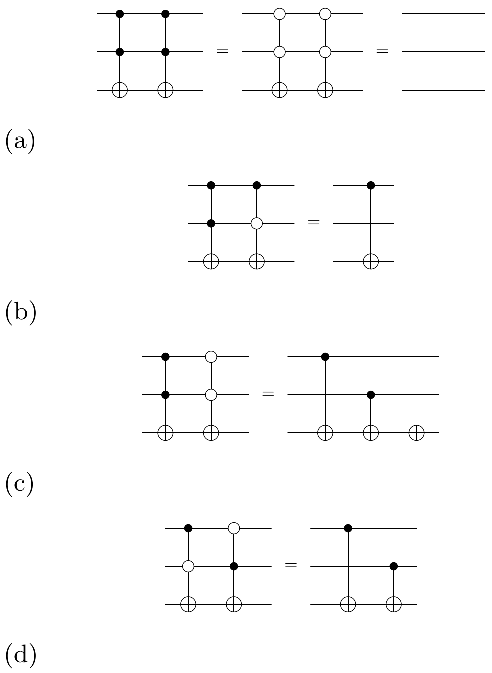


FIG. 9. Example quantum circuit required to perform 4-bit controlled U gate [52]. Note that c, w, and s denote control, work, and system qubits, respectively.

Importantly, if we arrange the sequence of control gates optimally, we can reuse some of the work qubits when the control states overlap. Figures 10 and 11 show different possible circuit templates that can be utilized to simplify the quantum circuits.

To maximize the circuit simplifications on the ancilla and work qubit registers, we show how a Gray encoding of U_{LCU} should be used. (Figure 26 in Appendix B shows the control states for $N_c = 5$ in a Gray and binary encoding.) Importantly, in a Gray code, adjacent bit strings only differ by 1 bit [53]. In other words, the Hamming distance between adjacent control states is always 1.

Consider x leading bits in common between adjacent control bit strings, where $2 \leqslant x \leqslant (N_c - 1)$. In the circuit picture, these are the cases when the top x controls between two adjacent control unitaries are in common. For these cases, we get $2(x - 1)$ trivial Toffoli reductions [Fig. 10(a)] on the Toffoli gates between the control unitaries. The number of times each x occurs is given by 2^x, and therefore the total trivial Toffoli



(a)

(b)

(c)

(d)

FIG. 10. (a)–(d) Toffoli-Toffoli circuit templates. The identities in (b) and (c) are given in Refs. [54] and [55], respectively.

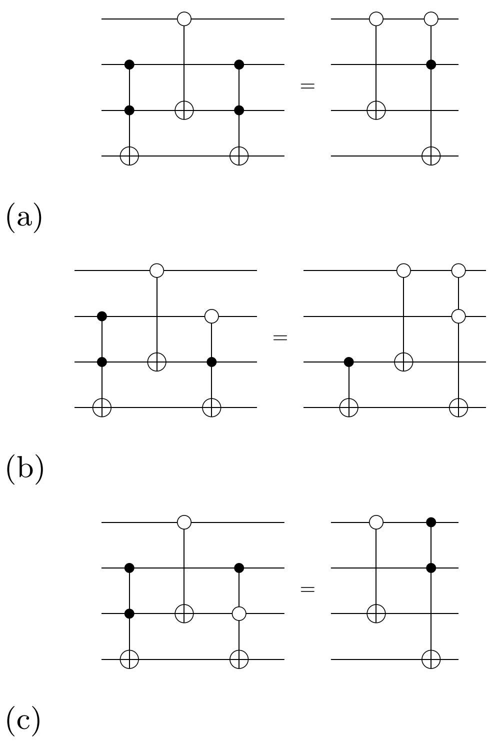


FIG. 11. Toffoli-CNOT-Toffoli circuit templates.

reduction is given by

$$\sum_{x=2}^{k} 2^x\; 2(x-1) = 2^{(k+2)}(k-2)+8$$
$$= 2^{N_c+1}(N_c-3)+8, \quad \text{where } k = N_c-1. \tag{43}$$

Next, consider the case of $x = N_c - 1$. After the trivial Toffoli gate cancellations, there must be two Toffoli gates that must differ by 1 bit in a Gray code. These will cancel to a single CNOT—illustrated in Fig. 10(b). This occurs 2^x times generating an additional $2^x = 2^{N_c-1}$ CNOT gates and removing a further $2^x(2) = 2^{x+1} = 2^{N_c}$ Toffoli gates. No further reductions are possible. A full example of this is given in Fig. 12.

For $0 \leqslant x < (N_c - 1)$, after the trivial Toffoli simplifications, it will always be possible to convert the two Toffoli gates into a CNOT gate. Again they must differ by 1 bit in a Gray code, and the template in Fig. 10(b) can be applied. The remaining circuit will have Toffoli-CNOT-Toffoli. These can be further reduced by applying the template shown in Fig. 11. In a Gray encoding, the template in Fig. 11(a) can always be applied, giving an optimal reduction. Figure 13(a) shows an example case with a Gray encoding. Figure 13(b) shows a less optimal reduction, which occurs when a binary encoding is used. Figure 14 shows another example, where a binary encoding again results in a less optimal reduction compared with a Gray code.

In a Gray encoding, three Toffoli gates will be canceled at each step, and one CNOT gate will be generated. The number of Toffoli gates removed in this process is given by

$$\sum_{x=0}^{k} 2^x(3) = 3(2^{k+1}-1)$$
$$= 3(2^{N_c-1}-1), \quad \text{where } k = N_c-2. \tag{44}$$

The increase in CNOT count is given in Eq. (45).

$$\sum_{x=0}^{k} 2^x = 2^{k+1}-1$$
$$= 2^{N_c-1}-1, \quad \text{where } k = N_c-2. \tag{45}$$

The total number of CNOT gates is given by Eq. (45). In a Gray encoding, the optimized number of Toffoli gates is given by Eq. (46). Here, T is short for Toffoli.

$$\text{Toffoli count} = \underbrace{(2^{N_c}(2N_c-2))}_{\text{no reductions}} - \underbrace{(2^{N_c+1}(N_c-3)+8)}_{\text{trivial TT}}$$
$$- \underbrace{(2^{N_c})}_{\text{TT to CNOT for } x=N_c-1} - \underbrace{(3(2^{N_c-1}-1))}_{\substack{0\leqslant x\leqslant N_c-2\\ \text{TT to CNOT and}\\ \text{T-CNOT-T to CNOT-T}}}$$
$$= 3(2^{N_c-1})-5. \tag{46}$$

When U_{LCU} is encoded using a Gray code and all 2^{N_c} control states are used, there will be $3(2^{N_c-1})-5$ Toffoli gates

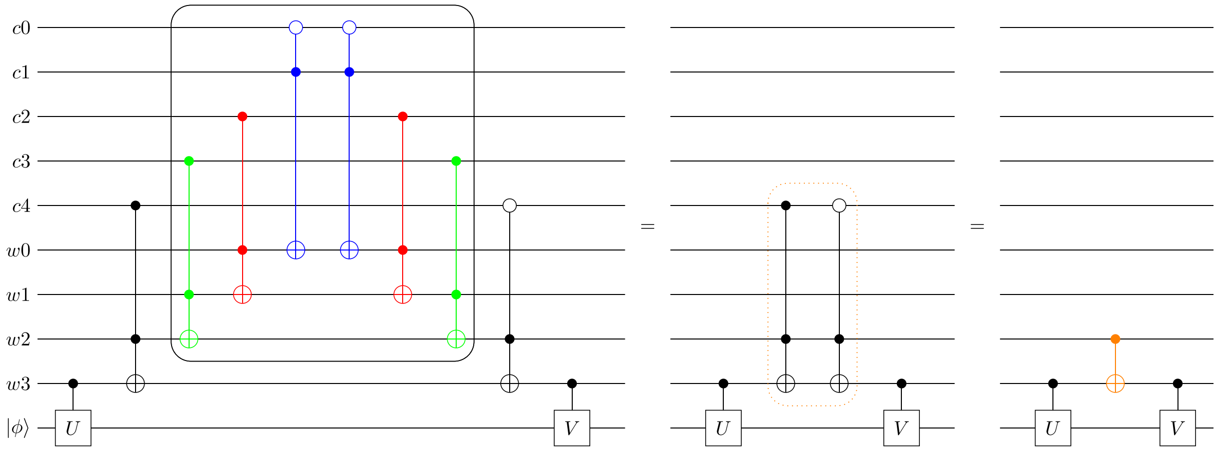


FIG. 12. Example of optimal circuit reduction, when pairs of adjacent control bit strings have $N_c - 1$ leading bits in common. This will occur for both the Gray and binary encodings.

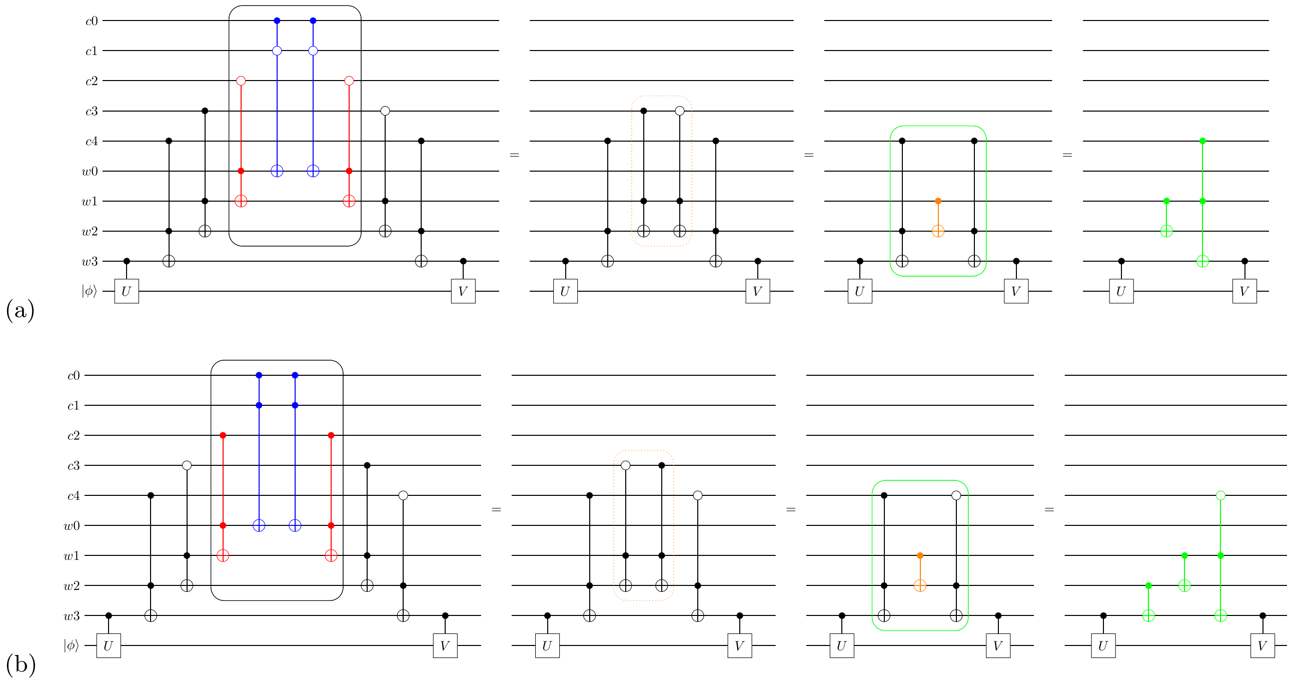


FIG. 13. Example partial reductions when control strings between adjacent unitaries differ by x leading bits. (a) gives an example simplification for 10011-U and 10001-V (Gray encoding). (b) gives an example reduction for 11101-U and 11110-V (binary encoding).

and $2^{N_c-1} - 1$ additional CNOT gates. Each Toffoli gate can be decomposed into 9 single-qubit gates and 6 CNOT gates [56]; therefore the reduced gate count requires $27(2^{N_c-1}) - 45$ single-qubit gates and $19(2^{N_c-1}) - 31$ CNOT gates.

So far these counts do not include any gate that acts on the system register, as different approaches are possible. In the next two sections we analyze two different possibilities—a cascade approach and a direct approach. We consider each case with a Gray encoding of the control unitaries.

1. LCU cascade

In the cascade approach, each control-P_i operator is performed using different changes of basis, a cascade of CNOT

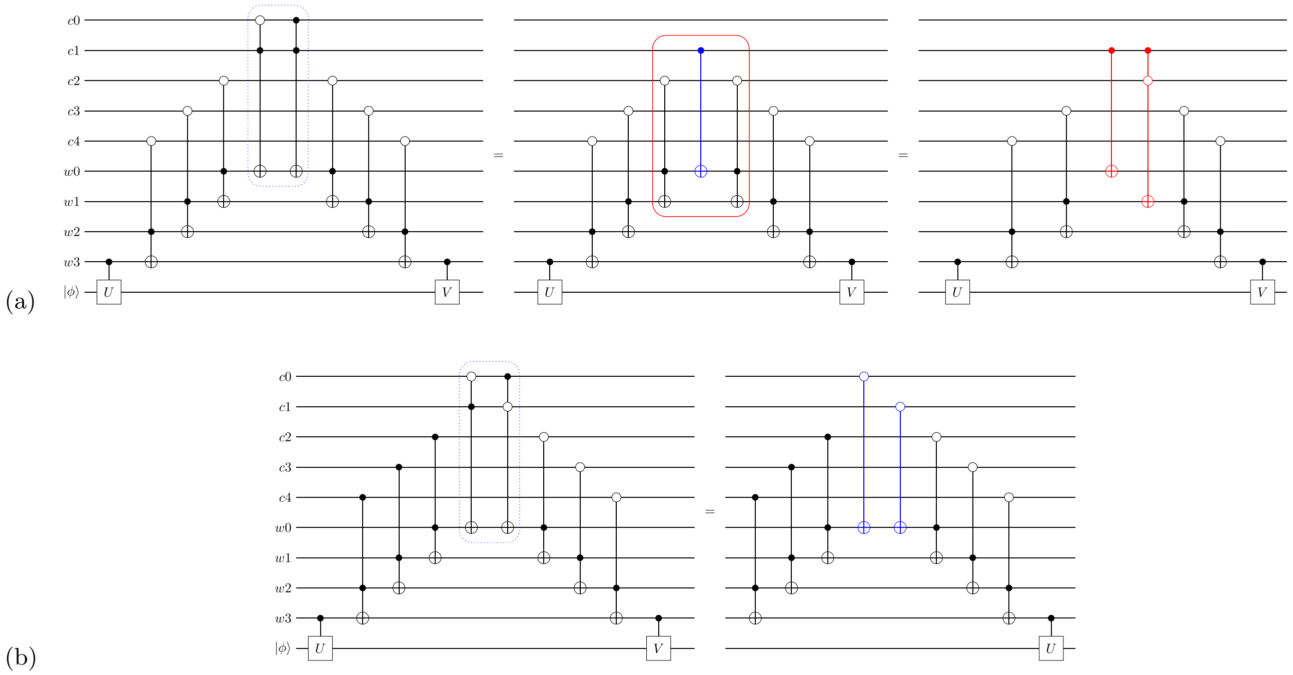


FIG. 14. Singular case when the leading bits between adjacent control unitaries differ ($x = 0$). (a) gives the Gray encoded case, and (b) gives the binary encoded case.

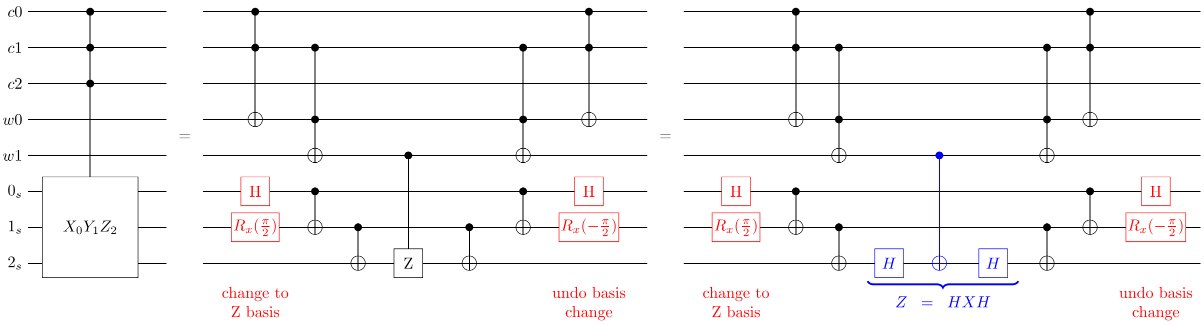


FIG. 15. Example quantum circuit to perform a multicontrol P_i gate via a cascade approach.

gates and a CNOT controlled by the work qubits. This approach requires a change of basis for every X and Y of each P_i gate of U_{LCU} [Eq. (39)]. In general requiring $\mathcal{O}(2N_s)$ gates per P_i. The resulting operator can then be implemented using a cascade of $\mathcal{O}(2(N_s-1))$ CNOT gates and a control Z gate. Two Hardamard gates can convert the control Z gate into a CNOT gate. An example is shown in Fig. 15.

The additional gate count on the system register will scale as $\mathcal{O}(2^{N_c}(2N_s+2))$ single-qubit gates and $\mathcal{O}(2^{N_c}(2N_s-1))$ CNOT gates.

The full gate count in a Gray encoding, including the ancilla register, scales as $\mathcal{O}(2^{N_c-1}(4N_s+31)-45)$ single-qubit gates and $\mathcal{O}(2^{N_c-1}(4N_s+17)-31)$ CNOT gates. Importantly, $N_c = \lceil \log_2(|H_{S_l}|-1)\rceil$ and $N_w = N_c - 1$.

2. LCU direct

Compared with the cascade approach, the direct approach implements each control n-fold Pauli operator of U_{LCU} [Eq. (39)] on the system register directly. $\mathcal{O}(2N_s)$ change-of-basis gates and $\mathcal{O}(N_s)$ CNOT gates are required per control gate. The number of single-qubit and CNOT gates scales as $\mathcal{O}(2N_s(2^{N_c}))$ and $\mathcal{O}(N_s(2^{N_c}))$, respectively. Figure 16 illustrates the approach.

The overall gate count over the system and ancilla registers scales as $\mathcal{O}(2^{N_c-1}(4N_s+27)-45)$ single-qubit gates and $\mathcal{O}(2^{N_c-1}(2N_s+19)-31)$ CNOT gates.

3. LCU constant factor ($|H_{S_l}| \leqslant 5$)

The scaling is different when $N_c \leqslant 2$ and no work qubits are required. For $N_c = 1$, all gates in U_{LCU} are controlled by 1 ancilla qubit. The circuit for G [Eq. (38)] is defined by a single R_y rotation. The circuit for U_{LCU} is implemented by $\mathcal{O}(4(N_s))$ change-of-basis gates and $\mathcal{O}(2N_s)$ CNOT gates.

For $N_c = 2$, the direct and cascade approaches can be used to construct U_{LCU}. The direct approach requires $\mathcal{O}(4N_s)$ Toffoli and $\mathcal{O}(8N_s)$ change-of-basis gates. The cascade approach requires $\mathcal{O}(8N_s)$ change-of-basis, $\mathcal{O}(8N_s)$ CNOT, and $\mathcal{O}(4)$ Toffoli gates.

By limiting the size of each anticommuting clique to $|H_{S_l}| \leqslant 5\ \forall \{l\}_{l=0,1,\ldots,m_c-1}$, no work qubits will be required to implement R_l.

The quantum circuits required to implement unitary partitioning under these conditions are realistic for implementation on current and near-term devices. This offers a constant factor improvement on the number of measurements required.

If anticommuting cliques $|H_{S_l}| > 5$ are present, they can be partitioned into separate subsets, each of size less than 5. The produced sets will still be valid anticommuting cliques. A renormalization on all subsets must also be performed.

4. Further LCU simplifications

An additional circuit simplification consists of the possible cancellations in the gates making up U_{LCU}. We did not explicitly consider this in this paper but note its clear application. The ordering of the control unitaries in U_{LCU} is arbitrary, and common qubit-wise Pauli strings can be canceled. The optimal reduction is obtained if Pauli operators on common qubits are maximized: This is known as a lexicographical ordering. An example reduction is given in Fig. 17.

We did not employ this process for our H_2 simulation, as it offered no improvement. The LiH problem would have benefited from this reduction; however, as we only simulated

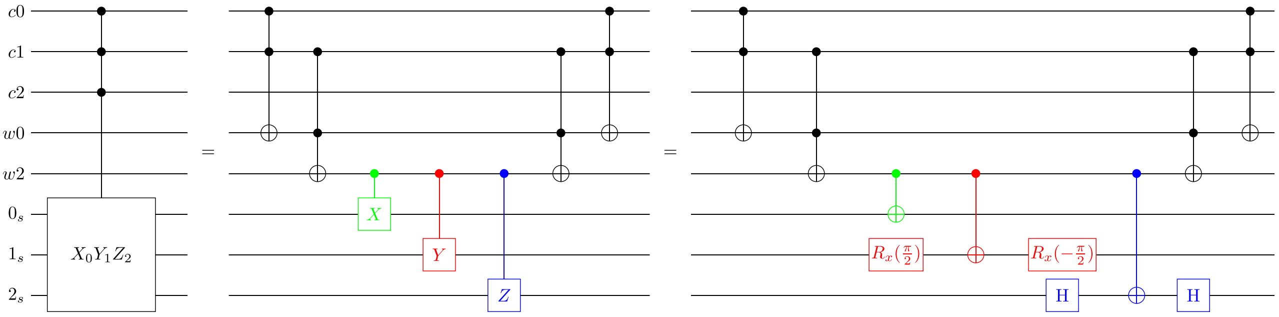


FIG. 16. Example quantum circuit to perform a multicontrol P_i gate via a direct approach.

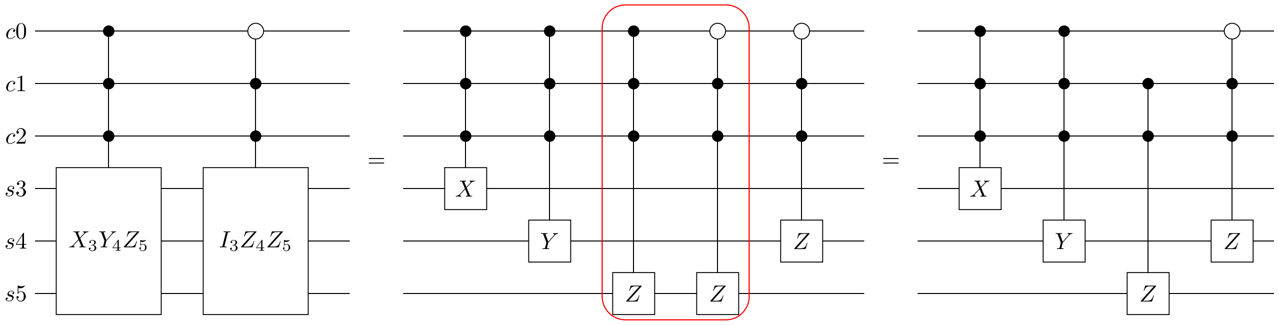


FIG. 17. Example lexicographical circuit simplification.

this problem on a QPU emulator, multicontrol gates could be performed directly. We therefore did not decompose these operations into their single- and 2-qubit gates, and we simulated the control P_i gates directly.

C. LCU unary implementation

Another approach to implementing the LCU is having each control unitary in Eq. (39) controlled by its own qubit. Hence the number of control qubits required is $N_c = |H_{S_l}| - 1$. This is known as a unary or one-hot encoding [57]. An example encoding for eight states is given in Table II.

Under a unary encoding, the quantum circuit for G [Eq. (38)] is made up of a single R_y rotation on each ancillary qubit, where the amplitude $\sqrt{\frac{\alpha_q}{\|\vec{\alpha}_q\|_1}}$ is encoded on either the $|0\rangle_a$ or $|1\rangle_a$ state. This is due to each α_q being real and positive. The number of single-qubit gates on the ancilla register scales as $\mathcal{O}(2(|H_{S_l}| - 1))$. No work qubits are required. The cascade or direct approach can then be utilized to implement the gates which act on the system register.

Figure 18 illustrates this approach with each amplitude encoded on the $|0\rangle_a$ state. This implementation uses an exponentially small subspace of the ancilla qubits' Hilbert space [57]. This is not an efficient use of quantum memory, but reduces the circuit depth of LCU significantly.

D. Discussion

Table III summarizes the different resources required to implement R_l via different approaches.

In summary, the sequence-of-rotations implementation of unitary partitioning gives the lowest gate count. For the different LCU implementations, the direct unary approach provides the lowest depth quantum circuits at the cost of requiring an ancilla qubit for each unitary in $U_{\text{LCU}}^{(l)}$.

Recently, the largest implementation of VQE to date was only able to make use of 12 qubits out of 53 available [6], and these unused qubits could be utilized for unary encoding. When this overhead becomes prohibitively large for sizable H_{S_l} sets, the Gray encoding schemes can be used.

V. CONCLUSION

Our work shows that the unitary partitioning technique for measurement reduction can significantly improve the precision of variational calculations. For a fixed measurement

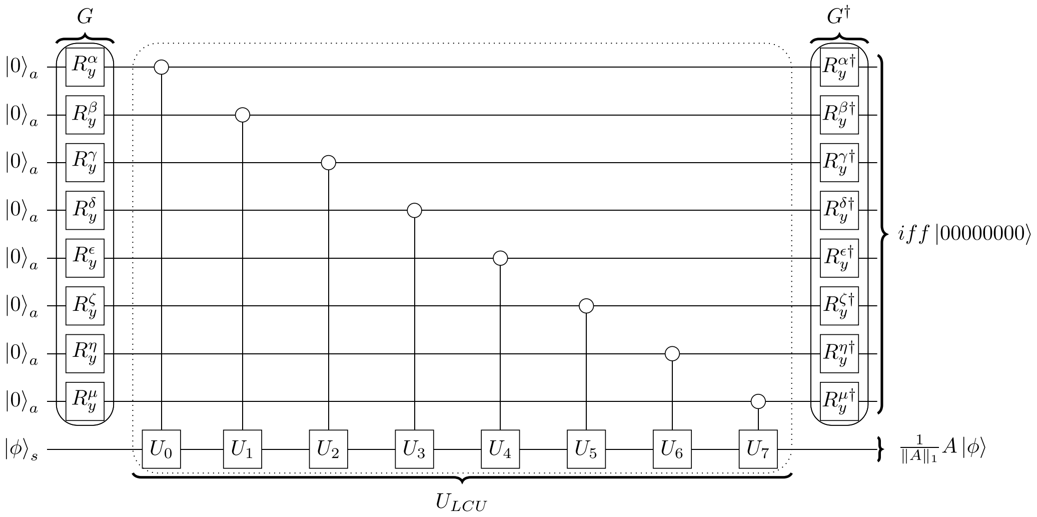


FIG. 18. Example unary encoded LCU circuit where $\langle 00\cdots 0|_a\, G^\dagger U_{\text{LCU}} G\, |00\cdots 0\rangle_a = \frac{A}{\|A\|_1}$.

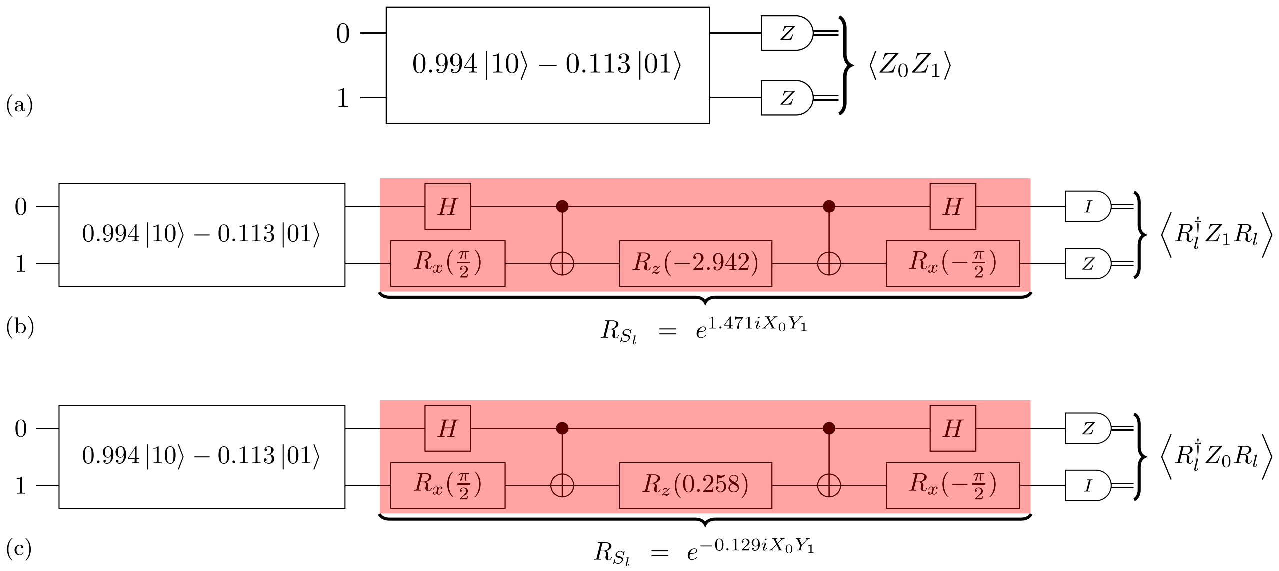


FIG. 19. Quantum circuits required to find the ground state of H_2 using VQE with unitary partitioning applied as a sequence of rotations. Tables IV and V define all operators required. Note that in (a) $l = 0$, in (b) $l = 2$, and in (c) $l = 3$.

budget M, fewer terms need to be estimated, and thus the total number of separate energy estimates is increased. As the sample standard deviation of energies is similar for the different approaches, the standard error of the mean will be lower when unitary partitioning is applied. Our results indicate that the deterministic sequence-of-rotations implementation offers the best improvement, which we find in our noiseless simulation of H_2 and LiH. In contrast, the LCU approach is probabilistic, and some measurements must be discarded. The advantage over standard VQE is thus dependent on the success probability. This naive implementation of LCU can be improved by using oblivious and standard amplitude amplification [30–33], which can boost the probability of success. However, further coherent resources are required.

The experimental results obtained using IBM's NISQ device (ibmqx2) show that VQE with unitary partitioning applied performs no worse than conventional VQE for a fixed M, when combined with error mitigation techniques. Even though unitary partitioning requires fewer terms to be combined to give an energy estimate leading to less statistical noise and more energy samples, we suspect that the additional coherent resources required cause an increased error accumulation, which offsets the advantages given by the technique. As quantum devices continue to improve, this effect should be

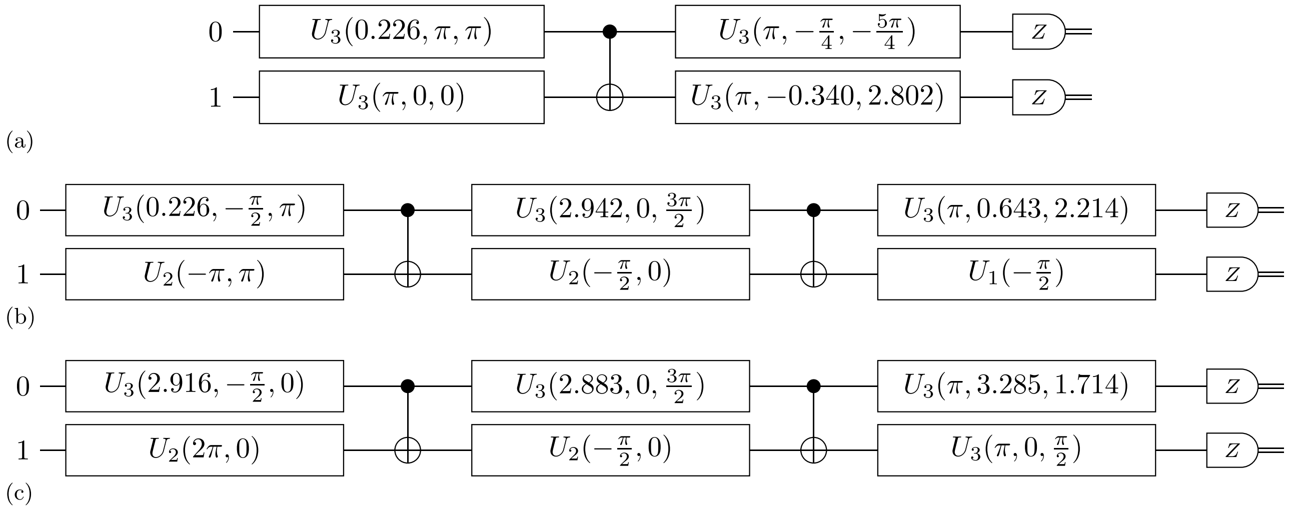


FIG. 20. Quantum circuits required to find the ground state of H_2 using VQE with unitary partitioning applied as a sequence of rotations. Tables IV and V define all operators required. These circuits are the compiled versions of those in Fig. 19 for use on IBM's QPU. Note that in (a) $l = 0$, in (b) $l = 2$, and in (c) $l = 3$.

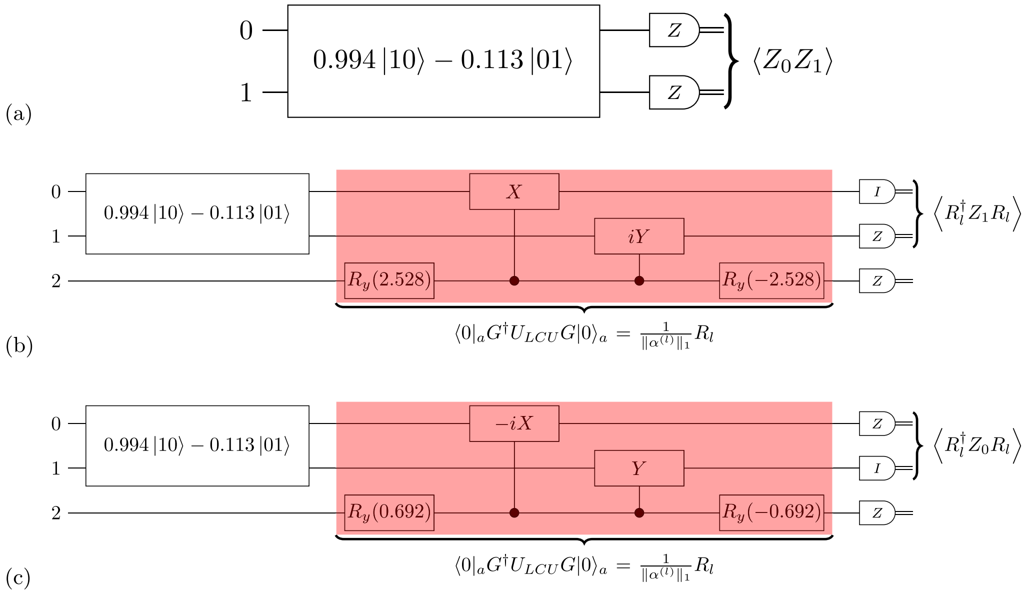

(a)
0
1
$0.994\,|10\rangle - 0.113\,|01\rangle$
Z
Z
$\langle Z_0 Z_1 \rangle$
(b)
0
1
2
$0.994\,|10\rangle - 0.113\,|01\rangle$
X
iY
$R_y(2.528)$
$R_y(-2.528)$
I
Z
Z
$\langle R_l^\dagger Z_1 R_l \rangle$
$\langle 0|_a G^\dagger U_{LCU} G|0\rangle_a = \frac{1}{\|\alpha^{(l)}\|_1} R_l$
(c)
0
1
2
$0.994\,|10\rangle - 0.113\,|01\rangle$
$-iX$
Y
$R_y(0.692)$
$R_y(-0.692)$
Z
I
Z
$\langle R_l^\dagger Z_0 R_l \rangle$
$\langle 0|_a G^\dagger U_{LCU} G|0\rangle_a = \frac{1}{\|\alpha^{(l)}\|_1} R_l$

FIG. 21. Quantum circuits required to find the ground state of H_2 using VQE with unitary partitioning applied as a LCU. Tables IV and VI define all operators and states $|G\rangle$ required. Note that in (a) $l = 0$, in (b) $l = 2$, and in (c) $l = 3$.

reduced, and we expect unitary partitioning will benefit many variational quantum algorithms.

Our work shows how precision can be improved for a fixed number of calls to a QPU; however, an alternate outlook is how this technique may allow larger problems to be studied. For a given precision, applying unitary partitioning requires fewer samples and thus may allow larger-scale simulations to be performed on reasonable timescales.

Future work should investigate how the variance of energies obtained changes if different terms in H_{S_l} are reduced to, as there is flexibility in the unitary partitioning technique. We also note that this work has an interesting application to the recently proposed shot-frugal Random Operator Sampling for Adaptive Learning with Individual Number of Shots (Rosalin) optimizer [58], which uses a weighted random sampling of Hamiltonian terms. Unitary partitioning transforms the Hamiltonian of interest into one with fewer terms of different coefficients; the effect this has on the optimizer's performance is an interesting avenue to explore.

ACKNOWLEDGMENTS

A.R. acknowledges support from the Engineering and Physical Sciences Research Council (EP/L015242/1). P.J.L.

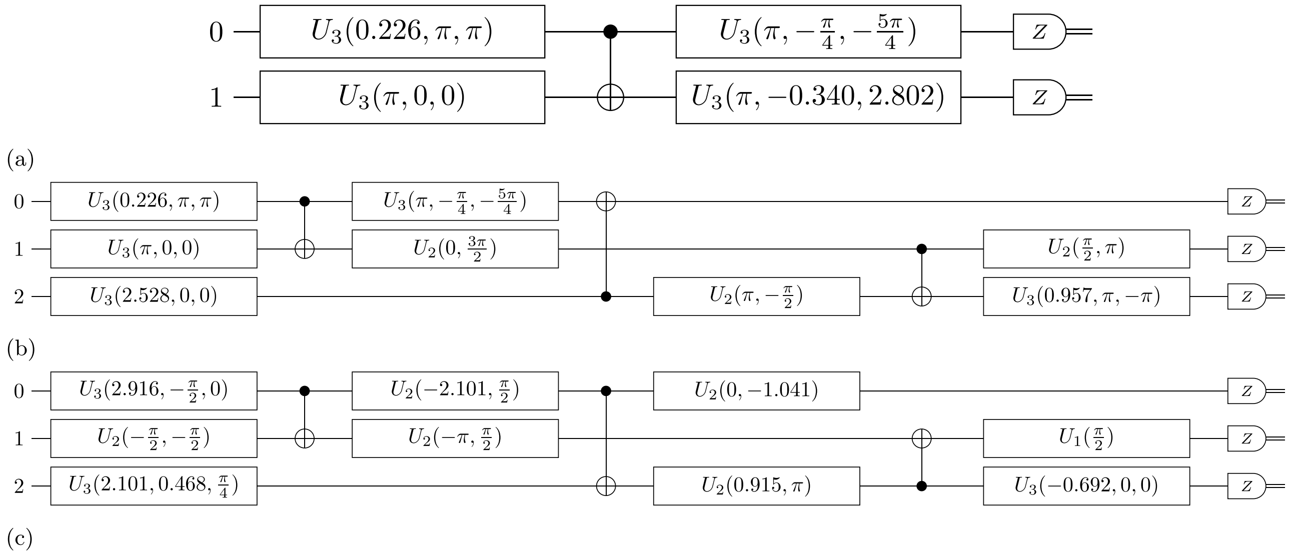

(a)
0
1
$U_3(0.226, \pi, \pi)$
$U_3(\pi, 0, 0)$
$U_3(\pi, -\frac{\pi}{4}, -\frac{5\pi}{4})$
$U_3(\pi, -0.340, 2.802)$
Z
Z
(b)
0
1
2
$U_3(0.226, \pi, \pi)$
$U_3(\pi, 0, 0)$
$U_3(2.528, 0, 0)$
$U_3(\pi, -\frac{\pi}{4}, -\frac{5\pi}{4})$
$U_2(0, \frac{3\pi}{2})$
$U_2(\pi, -\frac{\pi}{2})$
$U_2(\frac{\pi}{2}, \pi)$
$U_3(0.957, \pi, -\pi)$
Z
Z
Z
(c)
0
1
2
$U_3(2.916, -\frac{\pi}{2}, 0)$
$U_2(-\frac{\pi}{2}, -\frac{\pi}{2})$
$U_3(2.101, 0.468, \frac{\pi}{4})$
$U_2(-2.101, \frac{\pi}{2})$
$U_2(-\pi, \frac{\pi}{2})$
$U_2(0, -1.041)$
$U_2(0.915, \pi)$
$U_1(\frac{\pi}{2})$
$U_3(-0.692, 0, 0)$
Z
Z
Z

FIG. 22. Quantum circuits required to find the ground state of H_2 using VQE with unitary partitioning applied as a LCU. Tables IV and VI define all operators and states $|G\rangle$ required. These circuits are the compiled versions of those in Fig. 21 for use on IBM's QPU. Note that in (a) $l = 0$, in (b) $l = 2$, and in (c) $l = 3$.

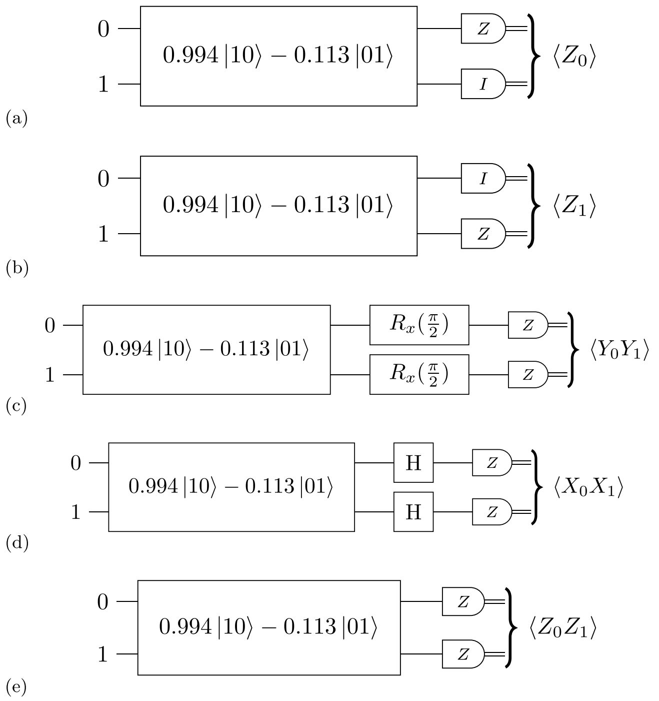


FIG. 23. (a)–(e) Quantum circuits required to find the ground state of H_2 using standard VQE. Each operator in Table IV requires a separate measurement.

and A.T. acknowledge support by the NSF STAQ project (PHY-1818914). P.V.C. acknowledges support from the European Union Horizon 2020 research and innovation programme under Grant Agreement No. 800925 (VECMA [59]). A.R. would also like to thank J. Dborin for useful discussions.

APPENDIX A: NUMERICAL STUDY DETAILS

The ability of the unitary partitioning measurement reduction strategy is dependent on the problem Hamiltonian. To understand the performance of these methods, we investigate Hamiltonians of interest in quantum chemistry. We consider Hamiltonians for H_2 and LiH molecules, each obtained using OPENFERMION-PYSCF [36,37]. These were converted into the qubit Hamiltonian using the Bravyi-Kitaev transformation in OPENFERMION [36]. The following sections give numerical details.

1. Molecular hydrogen

In the minimal STO-3G basis the qubit Hamiltonian for H_2 in the Bravyi-Kitaev (BK) representation is

$$\begin{aligned}H_q^{(H_2)} &= c_0\mathcal{I} + c_1X_0Z_1X_2 + c_2X_0Z_1X_2Z_3 + c_3Y_0Z_1Y_2 \\ &\quad + c_4Y_0Z_1Y_2Z_3 + c_5Z_0 + c_6Z_0Z_1 + c_7Z_0Z_1Z_2 \\ &\quad + c_8Z_0Z_1Z_2Z_3 + c_9Z_0Z_2 + c_{10}Z_0Z_2Z_3 \\ &\quad + c_{11}Z_1 + c_{12}Z_1Z_2Z_3 + c_{13}Z_1Z_3 + c_{14}Z_2. \qquad \text{(A1)}\end{aligned}$$

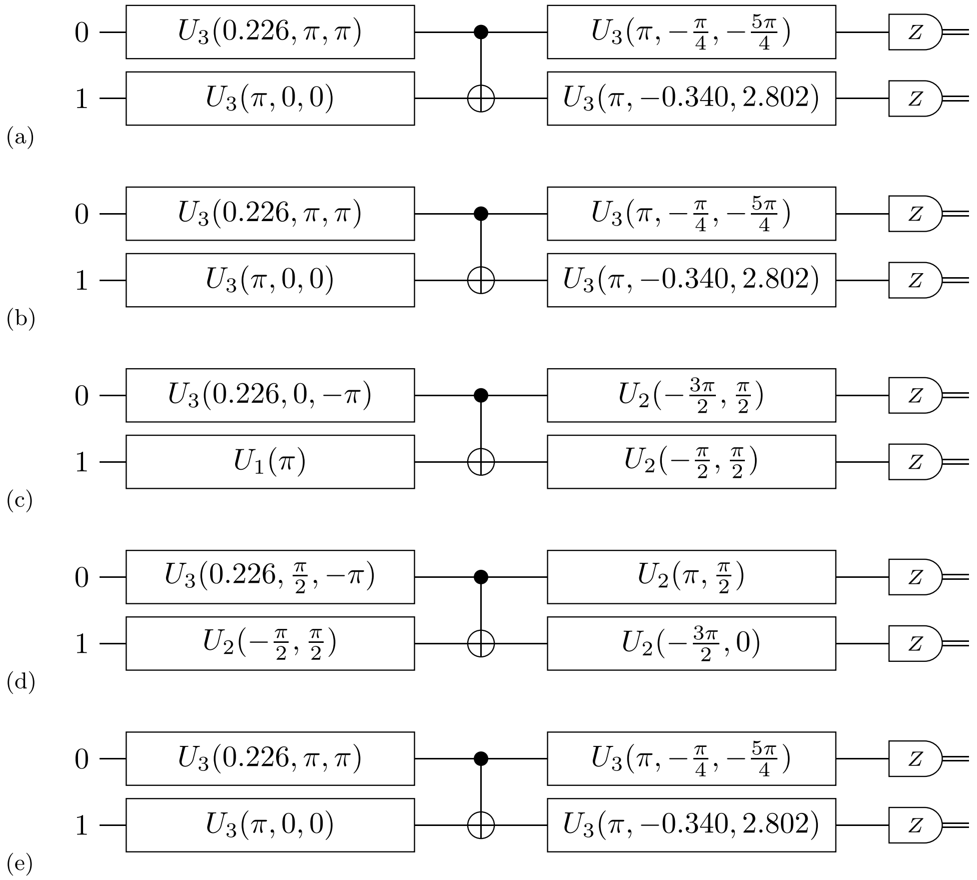


FIG. 24. (a)–(e) Quantum circuits required to find the ground state of H_2 using standard VQE. These circuits are the compiled versions of those in Fig. 23 for use on IBM's QPU.

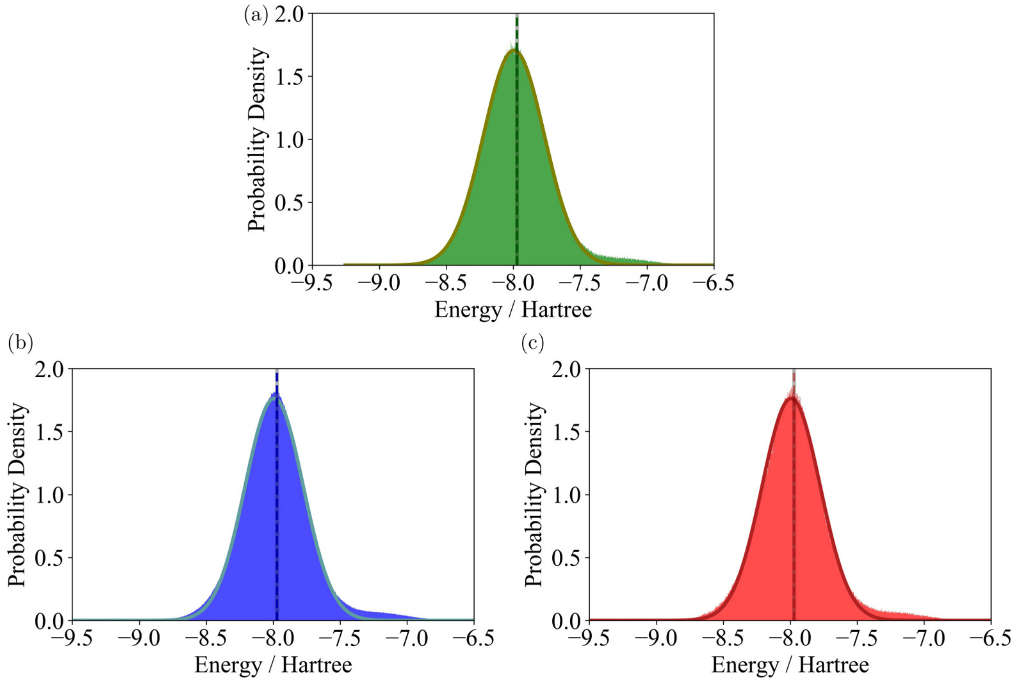


FIG. 25. Probability density function of different single-shot VQE calculations used to estimate the ground-state energy e_j of LiH. All results here are from a noise-free QPU emulation. The green results (a) are data from the standard VQE. The blue (b) and red (c) data are results from the VQE with unitary partitioning applied as a sequence of rotations and a LCU, respectively. The number of bins was set to 2500 for all histograms, and a Gaussian was fitted to each result. The dashed vertical lines show the average energy for each method, and each solid gray vertical line shows the FCI ground-state energy (−7.971 18 Ha).

This Hamiltonian only acts off-diagonally on qubits 0 and 2 [60]; therefore it can be reduced to

$$\begin{aligned} H_q^{(H_2)} &= (c_0 + c_{11} + c_{13})\mathcal{I} + (c_1 + c_2)X_0X_2 + (c_3 + c_4)Y_0Y_2 \\ &+ (c_5 + c_6)Z_0 + (c_7 + c_8 + c_{10})Z_0Z_2 \\ &+ (c_9 + c_{12} + c_{14})Z_2. \end{aligned} \quad \text{(A2)}$$

Overall, 2 qubits are required, and any Pauli operator indexed with a 2 is relabeled with an index of 1. The input state was found by diagonalizing the problem Hamiltonian [Eq. (A2)] - $|\psi_{H_2}^{\text{ground}}\rangle = -0.1125\,|01\rangle + 0.9936\,|10\rangle$. For our calculation, the bond length was set to R(H-H) = 0.74 Å. Note that we index the state from left to right.

To perform unitary partitioning, the qubit Hamiltonian for H_2 needs to be split into anticommuting sets H_{S_l}. As discussed in the main text, the NETWORKX package was utilized to do this [38]. First, a graph of the qubit Hamiltonian was built, where each node is a term in the Hamiltonian. Next, edges were put between nodes on the graph that anticommute. Finally, a graph coloring of the complement graph was performed. This searches for the minimum number of colors required to color the graph, where no neighbors of a node can have the same color as the node itself. The "largest first" coloring strategy in NETWORKX was used. Each unique color represents an anticommuting clique. Figure 5 shows the method applied to H_2, and Table IV gives the resulting anticommuting sets H_{S_l} obtained. Note that the set index l represents a unique color obtained in the graph coloring. This approach is the minimum clique cover problem mapped to a graph coloring problem.

Appendixes A 1 a–A 1 c give the quantum circuits used to estimate the ground state of H_2 by standard VQE and by VQE with unitary partitioning applied.

a. Sequence-of-rotations quantum circuits

Here, we present Table V and Figs. 19 and 20, which describe the sequence-of-rotations quantum circuits.

b. LCU quantum circuits

Here, we present Table VI and Figs. 21 and 22, which describe the LCU quantum circuits.

c. Standard VQE quantum circuits

Here, we present Figs. 23 and 24, which describe the standard VQE quantum circuits.

2. Lithium hydride

In the STO-6G basis the qubit Hamiltonian for LiH contains 631 terms. We considered a bond length of R(Li-H) = 1.45 Å. In total there are 12 spin-orbitals, and the number of qubits required to simulate this system with no reductions is 12. All numerical details are given in an EXCEL spreadsheet (XLS) file, which can be found in the Supplemental Material [61].

The same method given in Sec. A 1 was used to partition the Hamiltonian into anticommuting sets, and 102 cliques were obtained. They have been included in the XLS file. The operators used to implement the sequence-of-rotations and LCU methods have also been included in this file.

TABLE II. Example of unary encoding scheme, compared with decimal, binary, and Gray encodings.

Decimal	Binary	Gray	Unary
0	000	000	00000001
1	001	001	00000010
2	010	011	00000100
3	011	010	00001000
4	100	110	00010000
5	101	111	00100000
6	110	101	01000000
7	111	100	10000000

TABLE III. Upper bound for different resources required to perform R_l for unitary partitioning via different implementations. "LCU Gray" and "LCU unary" represent the cases when U_{LCU} [Eq. (18)] is encoded via a Gray and unary encoding, respectively. N_c and N_w are the number of extra control and work qubits required by a given LCU implementation. Here, reg, register; single, single-qubit gate.

Method	N_c	N_w	G circuit	Ancilla reg (U_{LCU})	System reg (U_{LCU})
LCUGray (cascade)	$\lceil \log_2(\lvert H_{S_l}\rvert - 1)\rceil$	$N_c - 1$	$\mathcal{O}(N_c \log_2(N_c)^2)$ standard 1 − and 2 − bit gate operations [51]	single : $\mathcal{O}(2^{N_c-1}N_s)$ CNOT : $\mathcal{O}(2^{N_c-1}N_s)$	single : $\mathcal{O}(2^{N_c}N_s)$ CNOT : $\mathcal{O}(2^{N_c}N_s)$
LCUGray (direct)					single : $\mathcal{O}(2^{N_c}N_s)$ CNOT : $\mathcal{O}(2^{N_c}N_s)$
LCUunary (cascade)	$\lvert H_{S_l}\rvert - 1$		single: $\mathcal{O}(\lvert H_{S_l}\rvert)$		single : $\mathcal{O}(N_cN_s)$ CNOT : $\mathcal{O}(N_cN_s)$
LCUunary (direct)					single : $\mathcal{O}(N_sN_c)$ CNOT : $\mathcal{O}(N_sN_c)$
SeqRot					single : $\mathcal{O}(N_s\lvert H_{S_l}\rvert)$ CNOT : $\mathcal{O}(N_s\lvert H_{S_l}\rvert)$

TABLE IV. Qubit Hamiltonian for H_2 [Eq. (A2)] at a bond length of 0.74 Å partitioned into anticommuting sets H_{S_l}.

l index	H_{S_l}
0	$(0.5731061703432151 + 0j)Z_0Z_1$
1	$(0.2460355896585992 + 0j)I_0I_1$
2	$\{(-0.4468630738162712 + 0j)I_0Z_1, (0.09060523100759853 + 0j)X_0X_1\}$
3	$\{(0.3428256528955378 + 0j)Z_0I_1, (0.09060523100759853 + 0j)Y_0Y_1\}$

TABLE V. Given the partitioning in Table IV, this table gives the operators and angles required to build R_{S_l} [Eq. (42)] as a sequence of rotations.

l index	γ_l	P_w	$\mathcal{X}_{wk}$	θ_{wk}
2	0.455956044621043	$(1)Z_1$	$(-1)X_0Y_1$	2.941546221798205
3	0.35459658228639496	$(1)Z_0$	$(1)X_0Y_1$	0.25838176362668025

TABLE VI. Given the partitioning in Table IV, this table gives the operators and ancilla states required to build R_l [Eqs. (37a) and (37b)] as a linear combination of unitaries.

l index	γ_l	P_w	$R_l = \sum_q \alpha_q^{(l)} P_q^{(l)}$	$U_{\text{LCU}}^{(l)} = \sum_q \lvert q\rangle_a \langle q\rvert_a \otimes P_q^{(l)}$	$\lvert G\rangle_a = \sum_q \sqrt{\frac{\lvert\alpha_q\rvert}{\lVert\alpha\rVert_1}}\,\lvert q\rangle_a$
2	0.455956044621043	$(1)Z_1$	$(0.09985651653293749)I$ $+(0.9950018472876858i)X_0Y_1$	$\lvert 0\rangle_a \langle 0\rvert_a \otimes I$ $+\lvert 1\rangle_a \langle 1\rvert_a \otimes 1i\, X_0Y_1$	$(0.3020015943219478)\lvert 0\rangle_a$ $+(0.9533074199999713)\lvert 1\rangle_a$
3	0.35459658228639496	$(1)Z_0$	$(0.9916664584717437)I$ $+(-0.12883180951189593i)X_0Y_1$	$\lvert 0\rangle_a \langle 0\rvert_a \otimes I$ $+\lvert 1\rangle_a \langle 1\rvert_a \otimes -1i\, X_0Y_1$	$(0.9407564775082788)\lvert 0\rangle_a$ $+(0.3390829544908076)\lvert 1\rangle_a$

The first sheet of the XLS document contains the LiH Hamiltonian. The first column contains each c_i coefficient, and the second column contains the associated P_i term. In total, there are 631 rows. Note that P_i operators written as [ ] represent the n-fold identity operation.

The second sheet contains the anticommuting sets H_{S_l}. It has the same structure as Table IV and contains 102 sets.

The third sheet gives the operators required to perform unitary partitioning via a sequence of rotations, given the clique cover from the first sheet. It follows the same structure as Table V and contains 98 entries.

The final sheet gives the operators required to perform unitary partitioning via a LCU, given the clique cover from the first sheet. It follows the same structure as Table VI and contains 98 entries.

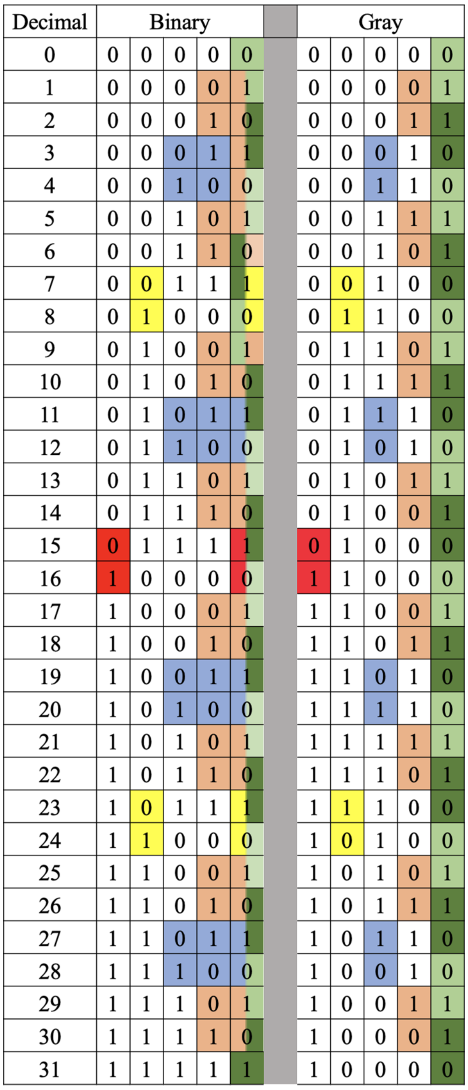
Due to the size of each quantum circuit, they were each simulated once using QISKIT's *statevector_simulator*, and the final state vectors obtained were sampled from using QISKIT's Statevector class *sample_counts* method [41]. This gives qubit measurement outcomes in the computational basis. In our simulation, each control P_i gate required by the LCU method was directly simulated, and neither work qubits nor circuit simplifications were used.

3. LiH histogram results

Here, we present Fig. 25, which provides the LiH histogram results.

APPENDIX B: 5-BIT CONTROL EXAMPLE

In this Appendix, we present Fig. 26, which shows all the control states of a 5-bit code under a binary and Gray encoding.



Decimal	Binary					Gray				
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1
2	0	0	0	1	0	0	0	0	1	1
3	0	0	0	1	1	0	0	0	1	0
4	0	0	1	0	0	0	0	1	1	0
5	0	0	1	0	1	0	0	1	1	1
6	0	0	1	1	0	0	0	1	0	1
7	0	0	1	1	1	0	0	1	0	0
8	0	1	0	0	0	0	1	1	0	0
9	0	1	0	0	1	0	1	1	0	1
10	0	1	0	1	0	0	1	1	1	1
11	0	1	0	1	1	0	1	1	1	0
12	0	1	1	0	0	0	1	0	1	0
13	0	1	1	0	1	0	1	0	1	1
14	0	1	1	1	0	0	1	0	0	1
15	0	1	1	1	1	0	1	0	0	0
16	1	0	0	0	0	1	1	0	0	0
17	1	0	0	0	1	1	1	0	0	1
18	1	0	0	1	0	1	1	0	1	1
19	1	0	0	1	1	1	1	0	1	0
20	1	0	1	0	0	1	1	1	1	0
21	1	0	1	0	1	1	1	1	1	1
22	1	0	1	1	0	1	1	1	0	1
23	1	0	1	1	1	1	1	1	0	0
24	1	1	0	0	0	1	0	1	0	0
25	1	1	0	0	1	1	0	1	0	1
26	1	1	0	1	0	1	0	1	1	1
27	1	1	0	1	1	1	0	1	1	0
28	1	1	1	0	0	1	0	0	1	0
29	1	1	1	0	1	1	0	0	1	1
30	1	1	1	1	0	1	0	0	0	1
31	1	1	1	1	1	1	0	0	0	0

FIG. 26. Five-bit binary and Gray codes. Differences between pairs of adjacent bit strings have been highlighted.

[1] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[2] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

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

[4] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi, Computation of Molecular Spectra on a Quantum Processor with an Error-Resilient Algorithm, Phys. Rev. X **8**, 011021 (2018).

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

[6] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell, B. Burkett, N. Bushnell, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, S. Demura, A. Dunsworth *et al.*, Google AI Quantum and Collaborators, Hartree-Fock on a superconducting qubit quantum computer, Science **369**, 1084 (2020).

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

[8] S. B. Bravyi and A. Y. Kitaev, Fermionic quantum computation, Ann. Phys. (Amsterdam) **298**, 210 (2002).

[9] P. Jordan and E. P. Wigner, Über das paulische äquivalenzverbot, in *The Collected Works of Eugene Paul Wigner* (Springer, New York, 1993), pp. 109–129.

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

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

[12] G. G. Guerreschi and M. Smelyanskiy, Practical optimization for hybrid quantum-classical algorithms, arXiv:1701.01450.

[13] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, Phys. Rev. A **92**, 042303 (2015).

[14] N. C. Rubin, R. Babbush, and J. McClean, Application of fermionic marginal constraints to hybrid quantum algorithms, New J. Phys. **20**, 053020 (2018).

[15] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[16] V. Verteletskyi, T.-C. Yen, and A. F. Izmaylov, Measurement optimization in the variational quantum eigensolver using a minimum clique cover, J. Chem. Phys. **152**, 124114 (2020).

[17] A. F. Izmaylov, T.-C. Yen, and I. G. Ryabinkin, Revising the measurement process in the variational quantum eigensolver: is it possible to reduce the number of separately measured operators? Chem. Sci. **10**, 3746 (2019).

[18] J. Cotler and F. Wilczek, Quantum Overlapping Tomography, Phys. Rev. Lett. **124**, 100401 (2020).

[19] X. Bonet-Monroig, R. Babbush, and T. E. O'Brien, Nearly Optimal Measurement Scheduling for Partial Tomography of Quantum States, Phys. Rev. X **10**, 031064 (2020).

[20] P. Gokhale and F. T. Chong, $O(N^3)$ measurement cost for variational quantum eigensolver on molecular Hamiltonians, arXiv:1908.11857.

[21] A. Jena, S. Genin, and M. Mosca, Pauli partitioning with respect to gate sets, arXiv:1907.07859.

[22] W. J. Huggins, J. R. McClean, N. C. Rubin, Z. Jiang, N. Wiebe, K. B. Whaley, and R. Babbush, Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers, npj Quantum Inf. **7**, 23 (2021).

[23] P. Gokhale, O. Angiuli, Y. Ding, K. Gui, T. Tomesh, M. Suchara, M. Martonosi, and F. T. Chong, Minimizing state preparations in variational quantum eigensolver by partitioning into commuting families, arXiv:1907.13623.

[24] A. F. Izmaylov, T.-C. Yen, R. A. Lang, and V. Verteletskyi, Unitary partitioning approach to the measurement problem in the variational quantum eigensolver method, J. Chem. Theory Comput. **16**, 190 (2020).

[25] A. Zhao, A. Tranter, W. M. Kirby, S. F. Ung, A. Miyake, and P. J. Love, Measurement reduction in variational quantum algorithms, Phys. Rev. A **101**, 062322 (2020).

[26] R. M. Karp, *Complexity of Computer Computations* (Springer, New York, 1972), pp. 85–103.

[27] A. Tranter, P. J. Love, F. Mintert, N. Wiebe, and P. V. Coveney, Ordering of Trotterization: Impact on errors in quantum simulation of electronic structure, Entropy **21**, 1218 (2019).

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

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

[30] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Exponential improvement in precision for simulating sparse Hamiltonians, in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2014), pp. 283–292.

[31] G. G. Guerreschi, Repeat-until-success circuits with fixed-point oblivious amplitude amplification, Phys. Rev. A **99**, 022306 (2019).

[32] L. K. Grover, Quantum Mechanics Helps in Searching for a Needle in a Haystack, Phys. Rev. Lett. **79**, 325 (1997).

[33] M. Boyer, G. Brassard, P. Hoyer, and A. Tapp, Tight bounds on quantum searching, Fortschr. Phys. **46**, 493 (1998).

[34] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, 2019), pp. 193–204.

[35] S. Subramanian, S. Brierley, and R. Jozsa, Implementing smooth functions of a Hermitian matrix on a quantum computer, J. Phys. Commun. **3**, 065002 (2019).

[36] J. R. McClean, N. C. Rubin, K. J. Sung, I. D. Kivlichan, X. Bonet-Monroig, Y. Cao, C. Dai, E. S. Fried, C. Gidney, B. Gimby, P. Gokhale, T. Häner, T. Hardikar, V. Havlíček, O. Higgott, C. Huang, J. Izaac, Z. Jiang, X. Liu, S. McArdle *et al.*, OpenFermion: the electronic structure package for quantum computers, Quantum Sci. Technol. **5**, 034014 (2020).

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

[38] A. A. Hagberg, D. A. Schult, and P. J. Swart, Exploring network structure, dynamics, and function using NetworkX, in *Proceedings of the 7th Python in Science Conference*, edited by G. Varoquaux, T. Vaught, and J. Millman (Association for Computing Machinery, New York, 2008), pp. 11–15.

[39] D. J. Welsh and M. B. Powell, An upper bound for the chromatic number of a graph and its application to timetabling problems, Comput. J. **10**, 85 (1967).

[40] F. Leymann and J. Barzen, The bitter truth about gate-based quantum algorithms in the NISQ era, Quantum Sci. Technol. **5**, 044007 (2020).

[41] H. Abraham, AduOffei, R. Agarwal, I. Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, M. Amy, E. Arbel *et al.*, Qiskit: An open-source framework for quantum computing, 2019.

[42] F. J. Massey Jr., The Kolmogorov-Smirnov test for goodness of fit, J. Am. Stat. Assoc. **46**, 68 (1951).

[43] S. S. Shapiro and M. B. Wilk, An analysis of variance test for normality (complete samples), Biometrika **52**, 591 (1965).

[44] B. Efron, Bootstrap methods: Another look at the jackknife, in *Breakthroughs in Statistics* (Springer, New York, 1992), pp. 569–593.

[45] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic*

Structure Theory, 1st ed. (Dover, New York, 2012).

[46] S. S. Tannu and M. K. Qureshi, Mitigating measurement errors in quantum computers by exploiting state-dependent bias, in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture* (Association for Computing Machinery, New York, 2019), pp. 279–290.

[47] Y. Nam, J.-S. Chen, N. C. Pisenti, K. Wright, C. Delaney, D. Maslov, K. R. Brown, S. Allen, J. M. Amini, J. Apisdorf, K. M. Beck, A. Blinov, V. Chaplin, M. Chmielewski, C. Collins, S. Debnath, K. M. Hudek, A. M. Ducore, M. Keesan, S. M. Kreikemeier *et al.*, Ground-state energy estimation of the water molecule on a trapped-ion quantum computer, npj Quantum Inf. **6**, 33 (2020).

[48] Y. Chen, M. Farahzad, S. Yoo, and T. C. Wei, Detector tomography on IBM quantum computers and mitigation of an imperfect measurement, Phys. Rev. A **100**, 052315 (2019).

[49] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, Simulation of electronic structure Hamiltonians using quantum computers, Mol. Phys. **109**, 735 (2011).

[50] A. Cowtan, S. Dilkes, R. Duncan, W. Simmons, and S. Sivarajah, *Phase gadget synthesis for shallow circuits*, in *Proceedings 16th International Conference on Quantum Physics and Logic (QPL 2019)* Electronic Proceedings in Theoretical Computer Science Vol. 318 (Chapman University, Orange, CA, 2020), pp. 213–228.

[51] G.-L. Long and Y. Sun, Efficient scheme for initializing a quantum register with an arbitrary superposed state, Phys. Rev. A **64**, 014303 (2001).

[52] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. (Cambridge University Press, Cambridge, 2002), Chap. 4, p. 184.

[53] G. Frank, Pulse code communication, U.S. Patent No. 2,632,058 (1953).

[54] M. Z. Rahman and J. E. Rice, Templates for positive and negative control Toffoli networks, in *International Conference on Reversible Computation* (Springer, New York, 2014), pp. 125–136.

[55] 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).

[56] Y. Nam, N. J. Ross, Y. Su, A. M. Childs, and D. Maslov, Automated optimization of large quantum circuits with continuous parameters, npj Quantum Inf. **4**, 23 (2018).

[57] N. P. D. Sawaya, T. Menke, T. H. Kyaw, S. Johri, A. Aspuru-Guzik, and G. G. Guerreschi, Resource-efficient digital quantum simulation of *d*-level systems for photonic, vibrational, and spin-*s* Hamiltonians, npj Quantum Inf. **6**, 49 (2020).

[58] A. Arrasmith, L. Cincio, R. D. Somma, and P. J. Coles, Operator sampling for shot-frugal optimization in variational algorithms, arXiv:2004.06252.

[59] www.vecma.eu.

[60] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus *et al.*, Scalable Quantum Simulation of Molecular Energies, Phys. Rev. X **6**, 031007 (2016).

[61] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevResearch.3.033195 for numerical details of the LiH simulation performed.