

# qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor

Ho Lun Tang,<sup>1</sup> V. O. Shkolnikov,<sup>1</sup> George S. Barron,<sup>1</sup> Harper R. Grimsley,<sup>2</sup> Nicholas J. Mayhall,<sup>2</sup> Edwin Barnes,<sup>1</sup> and Sophia E. Economou<sup>1</sup>

<sup>1</sup>*Department of Physics, Virginia Tech, Blacksburg, VA 24061, USA*

<sup>2</sup>*Department of Chemistry, Virginia Tech, Blacksburg, VA 24061, USA*

(Dated: June 18, 2020)

Quantum simulation, one of the most promising applications of a quantum computer, is currently being explored intensely using the variational quantum eigensolver. The feasibility and performance of this algorithm depend critically on the form of the wavefunction ansatz. Recently in *Nat. Commun.* 10, 3007 (2019), an algorithm termed ADAPT-VQE was introduced to build system-adapted ansätze with substantially fewer variational parameters compared to other approaches. This algorithm relies heavily on a predefined operator pool with which it builds the ansatz. However, *Nat. Commun.* 10, 3007 (2019) did not provide a prescription for how to select the pool, how many operators it must contain, or whether the resulting ansatz will succeed in converging to the ground state. In addition, the pool used in that work leads to state preparation circuits that are too deep for a practical application on near-term devices. Here, we address all these key outstanding issues of the algorithm. We present a hardware-efficient variant of ADAPT-VQE that drastically reduces circuit depths using an operator pool that is guaranteed to contain the operators necessary to construct exact ansätze. Moreover, we show that the minimal pool size that achieves this scales linearly with the number of qubits. Through numerical simulations on  $H_4$ ,  $LiH$  and  $H_6$ , we show that our algorithm (“qubit-ADAPT”) reduces the circuit depth by an order of magnitude while maintaining the same accuracy as the original ADAPT-VQE. A central result of our approach is that the additional measurement overhead of qubit-ADAPT compared to fixed-ansatz variational algorithms scales only linearly with the number of qubits. Our work provides a crucial step forward in running algorithms on near-term quantum devices.

## I. INTRODUCTION

Finding the ground state of a many-body interacting electronic Hamiltonian is one of the most important problems in modern quantum chemistry and physics. As the dimension of the Hamiltonian scales exponentially with the number of particles, accurate classical simulations can only be performed for systems with few electrons. Although many classical computational techniques have been developed to approximate the ground electronic state, no classical method is available which can perform accurately for arbitrary systems with polynomial effort. While density functional theory (DFT) [1, 2] has been tremendously useful for unraveling the microscopic details of weakly correlated molecules and materials, the most accurate form (Kohn-Sham DFT [1]) relies on a single Slater determinant wavefunction, which fails to describe strong correlation with current functionals. Unlike DFT, whose accuracy relies on a density functional which is not able to be systematically improved, wavefunction-based methods such as configuration interaction (CI) or coupled-cluster (CC) can be used to describe many-body systems with clear paths to arbitrary accuracy. However, in the presence of strong correlation such methods incur an exponential computational cost due to the large number of Slater determinants involved. Alternatively, methods based on tensor network states [3–7] (most notably density matrix renormalization group [8–10]) can exploit rank sparsity in low-energy states of one-dimensional systems to simulate strongly correlated systems, with poly-

nomial cost. This polynomial scaling is lost in higher dimensions, and the computational cost again grows exponentially. A radically different approach is Feynman’s proposal to study quantum systems using quantum computers [11]. Recent reviews provide a comprehensive background and discuss new developments in quantum simulation with quantum computers [12–14].

The long-term method for simulating chemical systems with quantum computers is the quantum phase estimation algorithm (PEA) [15, 16], which shows an exponential speedup over classical algorithms [17, 18]. Since the number of gates, i.e. unitary operations, involved in this algorithm is very large, it requires a long coherent evolution which can only be realized in fault-tolerant quantum computers [19]. These scalable, error-correcting devices may take decades to realize experimentally. In the meantime, the community is exploring algorithms that can be applied to existing and near-term processors, namely noisy intermediate-scale quantum (NISQ) devices [19].

A promising algorithm for NISQ hardware is the variational quantum eigensolver (VQE) [20, 21]. VQE is a hybrid method that combines classical computational power with a quantum processor. Based on the variational principle in quantum physics, the VQE algorithm constructs a trial wavefunction by applying gates on a quantum device and estimates the average energy by measuring the Hamiltonian on that device. This measured energy is then minimized by tuning the quantum circuit. The circuit parameter optimization is performed by a classical computer, and quantum resources are only

used for the classically intractable parts (state preparation and energy evaluation) of the calculation. Compared to PEA, VQE has much more modest requirements on the coherence times of the quantum processor, and it has already been realized on NISQ devices, such as superconducting qubits [22, 23], photons [20], and trapped ions [24, 25]. The accuracy of VQE is highly dependent on the explicit form of the wavefunction ansatz and can only obtain the exact ground state energy if the ansatz is capable of representing any state in the subspace that contains the ground state. For example, if symmetries of the ground state are known, then one can construct an ansatz that represents any state in the corresponding symmetry subspace [26], guaranteeing that the exact ground state can be expressed in terms of the ansatz.

One of the commonly used ansätze for VQE is unitary coupled cluster singles and doubles (UCCSD) [27, 28]. Stemming from the coupled cluster theory used in chemistry, this unitary version is more suitable for quantum circuit implementation. The ‘singles and doubles’ in UCCSD means only single and double excitation operators are included in the ansatz, each carrying its own variational parameter. One drawback of UCCSD is that including all the singles and doubles operators leads to a (potentially unnecessarily) deep circuit and a large number of parameters to optimize. To reduce this complexity, several alternatives have been proposed which attempt to keep only the most important operators [29–36]. A second drawback is that UCCSD is also generally not exact and suffers from ambiguities in operator ordering upon factorization into a product of exponentiated operators (Trotterization), which is a necessary step in converting the ansatz to a state preparation circuit [37, 38]. Another approach to building the ansatz is to use the most accessible gates in the quantum device, alternating single-qubit gates and two-qubit gates layer by layer. This is referred to as a hardware-efficient ansatz [23]. Although this approach lessens the demands on the quantum processor, the resulting wavefunction ansatz can lead to difficulties in parameter optimization [39]. This problem was addressed by constructing particle-conserving entangling gates instead of ordinary two-qubit gates [40, 41] and symmetry-preserving circuits [26]. While these fixed-ansatz approaches can be applied to any problem and can reduce the number of variational parameters and circuit depths, further improvement may still be possible by tailoring the ansatz to a given simulation problem.

Recently, a new algorithm that provides a systematic method to build an ansatz dynamically was introduced. This algorithm, termed Adaptive Derivative Assembled Pseudo-Trotter (ADAPT) VQE [29], employs a predetermined pool of operators from which the ansatz is dynamically constructed. The ansatz is grown iteratively, such that at each step, the operator that affects the energy the most is added to the ansatz. Using fermionic operators as a pool, Ref. [29] demonstrated that ADAPT-VQE substantially outperforms UCCSD, in terms of both number of variational parameters and accuracy. This re-

sult demonstrates the promise of the ADAPT algorithm. However, due to the gate overhead of the fermion-to-spin mapping, the operators considered in Ref. [29] translate to a fairly large number of quantum gates, and, therefore, while the number of parameters is very low, the circuit depth (which is significantly reduced compared to UCCSD) may still be impractically large, limiting the applicability of ADAPT-VQE to NISQ devices. Even more importantly, it is not clear (i) how the operator pool should be chosen in general, (ii) how many operators it should contain, and (iii) what guarantees that the pool is complete, i.e., that it enables convergence to the ground state.

In this paper, we address these issues by introducing a hardware-efficient variant of ADAPT-VQE that substantially reduces both the number of measurements and the circuit depths needed to achieve convergence. We term this algorithm qubit-Adaptive Derivative Assembled Problem-Tailored (qubit-ADAPT) VQE, in contrast with the implementation in Ref. [29], which we refer to as fermionic-ADAPT in this paper. Through classical simulations of several different molecules, we demonstrate that compared to fermionic-ADAPT, qubit-ADAPT reduces the circuit depth by an order of magnitude while maintaining the same accuracy. Moreover, we introduce a pool completeness criterion that determines whether a given pool will generate an exact ADAPT ansatz. We prove that the minimal number of pool operators that satisfy this condition grows linearly with the number of qubits. This is much smaller than the quartic scaling originally assumed in fermionic-ADAPT, and it demonstrates that the additional measurement overhead of ADAPT-VQE remains modest for larger systems (increasing only linearly over conventional, fixed-ansatz VQEs). Our results pave the way toward both practical and accurate VQE algorithms on NISQ devices.

This paper is organized as follows. First, we briefly review fermionic-ADAPT and estimate the corresponding circuit depth in Sec. II. In Sec. III, we introduce qubit-ADAPT and provide a detailed description of the operator pool. In Sec. III A, we compare the performance of qubit-ADAPT and fermionic-ADAPT through numerical simulations of  $H_4$ ,  $LiH$  and  $H_6$  molecules. We show that the minimal size of the operator pool for qubit-ADAPT scales linearly in the number of qubits in Sec. III B. We conclude in Sec. IV. Details of the estimate for the circuit depth of fermionic-ADAPT and a constructive proof of the linear scaling of minimal complete pools are included in appendices.

## II. CIRCUIT DEPTH ESTIMATE FOR FERMIONIC-ADAPT

The ADAPT ansatz is grown by one operator  $\hat{\tau}_i = -\hat{\tau}_i^\dagger$  at each iteration, and after the  $n$ -th iteration is given by

$$|\psi^{ADAPT}(\vec{\theta})\rangle = e^{\theta_k \hat{\tau}_k} \dots e^{\theta_2 \hat{\tau}_2} e^{\theta_1 \hat{\tau}_1} |\psi^{HF}\rangle, \quad (1)$$

where  $|\psi^{HF}\rangle$  is the HF state. These operators are selected from an operator pool defined upfront. At each iteration, the operator which induces the maximum change to the energy is selected. This energy response is represented by the gradient of the energy with respect to the corresponding parameter, i.e.

$$\frac{\partial}{\partial \theta_i} \langle E \rangle = \langle \psi | [\hat{H}, \hat{\tau}_i] | \psi \rangle, \quad (2)$$

which can be measured on the quantum device. The ansatz keeps growing until the norm of the gradient vector,

$$\|\vec{g}\| = \sqrt{\sum_i \left( \frac{\partial}{\partial \theta_i} \langle E \rangle \right)^2}, \quad (3)$$

is zero or, in practice, smaller than a chosen threshold,  $\epsilon$ . Compared to ordinary VQE, ADAPT-VQE requires additional measurements to obtain the gradient at each iteration; the number of these measurements is roughly equal to the size of the pool times the number of terms in the Hamiltonian [42], although the number of measurements needed to compute the mean energy can be decreased using recently developed techniques [43].

In order to attain a compact circuit for the ADAPT ansatz, we want to minimize both the number of parameters and the circuit depth. While the first requirement can be satisfied by the general structure of the ADAPT-VQE algorithm, the second one is not guaranteed and depends on the chosen pool. Using fermionic operators, a *parameter-efficient* operator pool can be constructed from spin-adapted single excitation operators:

$$\hat{\tau}_1 \propto |\uparrow\rangle_a \langle \uparrow|_b + |\downarrow\rangle_a \langle \downarrow|_b - h.c., \quad (4)$$

and double excitation operators:

$$\begin{aligned} \hat{\tau}_{2,T} &\propto |T, 1\rangle_{pq} \langle T, 1|_{rs} + |T, -1\rangle_{pq} \langle T, -1|_{rs} \\ &\quad + |T, 0\rangle_{pq} \langle T, 0|_{rs} - h.c. \\ \hat{\tau}_{2,S} &\propto |S, 0\rangle_{pq} \langle S, 0|_{rs} - h.c., \end{aligned} \quad (5)$$

where  $a, b, p, q, r, s$  are spatial orbitals and  $T, S$  refer to triplets and singlets formed by  $p, q$  or  $r, s$ . To implement ADAPT-VQE on qubits, we have to map these fermionic operators to Pauli operators, which has the consequence that a *parameter-efficient* pool is most likely not *gate-efficient*. In this paper, we employ the Jordan-Wigner (JW) mapping, i.e.,  $\hat{a}_i \rightarrow \prod_{j=0}^{i-1} Z_j (X_i - iY_i)$ . Adapting the methods we develop here to other mappings [44–46] will be the subject of future work.

In the JW mapping, a double excitation operator may contain more than one fermionic operator  $a_p^\dagger a_q^\dagger a_r a_s - h.c.$ , which is transformed into at most 8 Pauli strings. (A product of four fermionic operators gives 16 terms in total, but each symmetric term is cancelled by its Hermitian conjugate [35].) These excitation operators conserve  $S^2$ ,  $S_z$  and particle number by summing a number of

Pauli strings together, which results in a high gate count per excitation operator.

In order to make a rough estimate of the number of CNOT gates involved in one operator from the fermionic pool, we consider the generalized singles-doubles excitation fermionic pool. ‘Generalized’ indicates that the excitations are not restricted to occur only from occupied to virtual orbitals. Rather, all combinations of excitations are included. To obtain the gate count, we perform first-order Trotterization of each unitary, i.e.,  $e^{\theta_i \hat{\tau}_i} \approx \prod_j e^{\theta_j \hat{P}_j}$ , where  $P_j$  are the Pauli strings appearing in  $\hat{\tau}_i$  after the JW mapping. For simplicity, we assume only double excitation operators are picked by the algorithm, which makes this a conservative estimate as any single excitation included would result in a smaller CNOT-to-parameter count ratio. The number of CNOTs needed for a single Pauli string  $\hat{P}_i$  with length  $q$  is  $2(q-1)$  [48]. Here, the length  $q$  is the number of non-identity Pauli operators in the string. The average number of CNOTs involved in a spin-adapted double excitation operator is approximately  $\bar{N}_{Pauli}(6 + 2\bar{N}_Z)\bar{N}_{spin}$ , where  $\bar{N}_{Pauli}$  is the average number of Pauli strings in a doubles operator  $a_p^\dagger a_q^\dagger a_r a_s - h.c.$ ,  $\bar{N}_Z$  is the average number of Pauli  $Z$ ’s in a Pauli string due to the anticommutation relation of fermionic operators, and  $\bar{N}_{spin}$  is the average number of doubles operators summed in a spin-adapted operator (Eq.(5)). By Appendix. A, these quantities are given by

$$\bar{N}_{Pauli} = \frac{8m(5m-7)}{m(5m-1)-8}, \quad (6)$$

$$\bar{N}_Z = \frac{12m^3 - 30m^2 + 34m - 20}{3(5m^2 - m - 8)}, \quad (7)$$

$$\bar{N}_{spin} = \frac{5m^2 - m - 8}{m^2 + m}, \quad (8)$$

where  $m$  is the number of spatial orbitals. For large  $m$ , the number of CNOTs in a spin-adapted doubles operator is approximately  $64m$ .

### III. QUBIT-ADAPT

Because the spin-adapted fermionic operators each introduce a large number of CNOTs into the state preparation circuit, we are motivated to construct a new pool consisting of operators that involve fewer CNOTs. One way is to break down the spin-adapted fermionic operators after the JW mapping and choose the individual Pauli strings as the operator pool  $\hat{\tau} = \hat{P} = i \prod_i p_i$ ,  $p_i \in \{X, Y, Z\}$ . This more hardware-efficient choice effectively reduces  $\bar{N}_{spin}$  and  $\bar{N}_{Pauli}$  to 1, while it contains the same basic elements as the spin-adapted fermionic pool. An important property of this qubit pool is that it only contains Pauli strings with odd numbers of  $Y$ ’s because the fermionic operators are real, hence  $\hat{P} = i \prod_i p_i$  has to be real. We refer to these as ‘odd’ Pauli strings. The

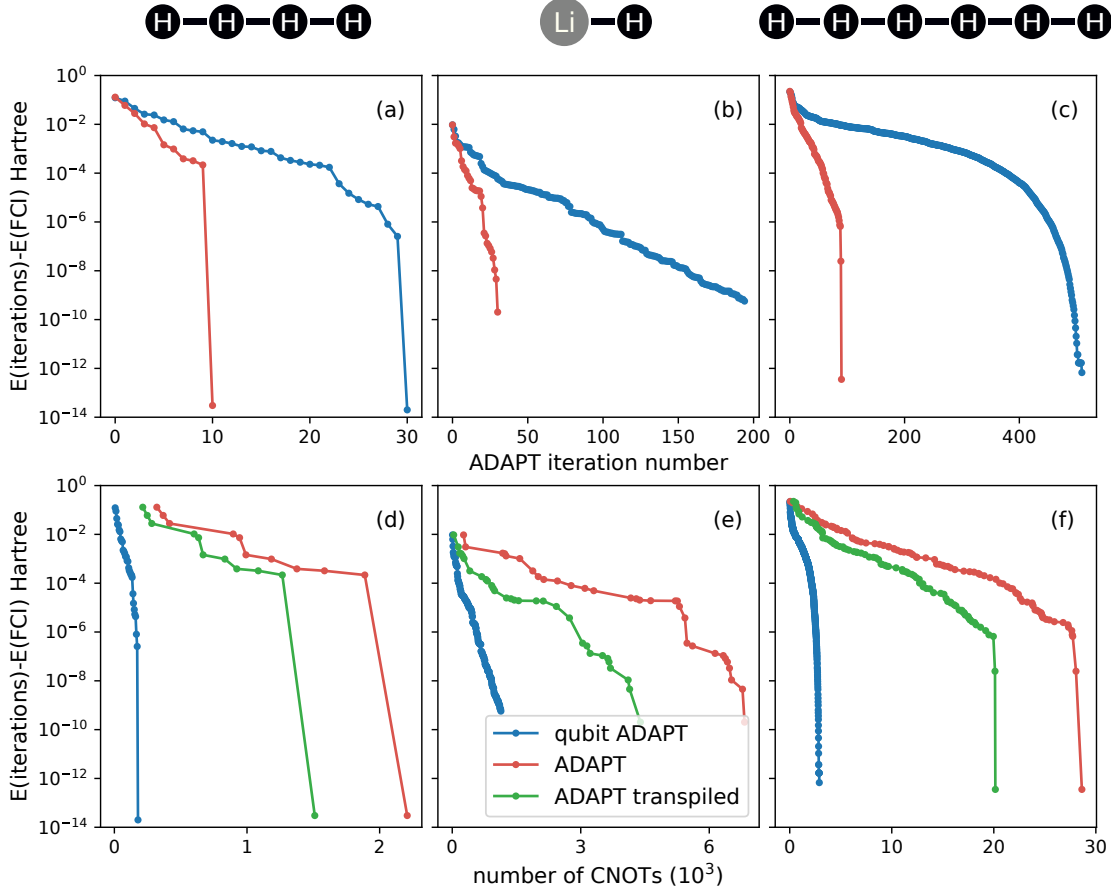


FIG. 1. Ground state energy difference between ADAPT and the exact (FCI) result for (a,d) H<sub>4</sub> at bond distance 1.5 Å, (b,e) LiH at bond distance 2 Å, and (c,f) H<sub>6</sub> at bond distance 1.5 Å. The bond distances are chosen to ensure that correlation effects are substantial. All the calculations are performed using an STO-3G basis and start from restricted Hartree-Fock orbitals without using a frozen-core approximation such that we have 8 spin-orbitals for H<sub>4</sub>, 12 spin-orbitals for LiH, and 12 spin-orbitals for H<sub>6</sub>. Results are shown for qubit-ADAPT (blue), fermionic-ADAPT (orange) and transpiled fermionic-ADAPT (green). Panels (a)-(c) show the energy difference as a function of ADAPT iterations (which is equal to the number of variational parameters), while (d)-(f) show it as a function of the number of CNOTs in the corresponding state preparation circuit. All CNOT gate counts are obtained using the qiskit command `count_ops`. The transpiled counts were obtained using `qiskit.transpile` [47] with heavy optimization, which includes canceling back-to-back CNOTs and “commutative cancellation”. Although qubit-ADAPT requires more variational parameters, it entails significantly fewer CNOTs compared to fermionic-ADAPT.

remaining “even” Pauli strings have no effect on the energy. This is because  $\hat{H}$  is symmetric (time-reversal symmetry is preserved), and so the expectation value of the commutator in Eq. (2) will vanish for real  $|\psi\rangle$  if  $\hat{\tau}_i$  is symmetric (i.e., even). We can therefore restrict  $\hat{\tau}_i$  to the odd Pauli strings, which are antisymmetric. Using such operators in the pool also ensures that the ansatz remains real throughout the qubit-ADAPT algorithm, which should be the case when  $\hat{H}$  is time-reversal symmetric. The length of these strings ranges from 2 to  $n$  due to the Pauli  $Z$  chain responsible for the fermionic anticommutation relation. To further reduce gate depth, we remove these Pauli  $Z$  chains from the operators. Numerically, we find that these two pools (with and without

$Z$  chains) perform similarly. The pool without  $Z$  chains gives Pauli strings with maximum length 4. The size of this pool is much smaller than the full set of Pauli strings with maximum length 4, as we only pick operators that already appear in the fermionic pool, which are capable of transforming  $|\psi^{HF}\rangle$  to the ground state. We refer to this reduced pool as the “qubit pool”. Below, we demonstrate that this pool produces significantly shallower state preparation circuits compared to fermionic-ADAPT. We also provide evidence that the size of the qubit pool can be reduced dramatically down to a size that scales only linearly in the number of qubits, which substantially cuts down on the number of measurements needed to run qubit-ADAPT.



### A. Numerical simulations

We compare the performance of the qubit pool to the fermionic pool in terms of the number of parameters and the number of CNOTs for different molecules,  $H_4$ , LiH and  $H_6$  (Fig. 1). In each case, we choose a bond distance such that correlation effects are significant:  $r = 1.5\text{\AA}$  for  $H_4$  and  $H_6$ , and  $r = 2\text{\AA}$  for LiH. All the calculations are performed using an STO-3G basis and start from restricted Hartree-Fock orbitals without using a frozen-core approximation such that we have 8 spin-orbitals for  $H_4$  and 12 spin-orbitals for each of the other two molecules. All CNOT gate counts are obtained using the qiskit command `count_ops`, where we consider the case of all-to-all qubit connectivity for concreteness. Although these counts will increase as the connectivity is reduced, we do not expect the relative performance of the two algorithms to change significantly. In the case of fermionic-ADAPT, we show gate counts before and after transpilation is used. The counts before transpilation agree well with the estimates obtained in Sec. A.

It is evident from Fig. 1 that in the case of the qubit pool, more parameters are used compared to the fermionic pool. On the other hand, the number of CNOTs is reduced significantly, by about an order of magnitude in the case of  $H_6$ . Switching from the fermionic pool to the qubit pool increases the number of parameters in the ansatz, which is the price for compressing the circuit depth. However, the increase in parameter number only increases the required classical computational power during the classical optimization, while the decrease in circuit depth reduces the demands on the quantum processor. For NISQ devices, the number of CNOTs that can be implemented within the coherence time is very limited, so the ability of qubit-ADAPT to divert more of the computational cost away from the quantum processor and onto the classical optimizer should be advantageous overall.

To evaluate the effectiveness of using the gradient as a means to grow the ADAPT ansatz, we compare the performance of qubit-ADAPT with random operator orderings drawn from the same qubit pool. In Fig. 2, we see that qubit-ADAPT always converges much faster than the random orderings for both  $H_4$  and LiH. When random orderings are used for  $H_4$ , convergence to the ground state requires 46-78 parameters, compared to only 30 parameters for qubit-ADAPT. In the case of LiH, the random orderings require more than three times as many parameters as qubit-ADAPT to converge. We can only provide a lower bound on the number of parameters needed to converge the random ordering results in Fig. 2(b) due to the long computational times needed in this case. These findings suggest that the role of the gradient selection in qubit-ADAPT is crucial for larger problems.

Fig. 3 shows that the performance of qubit-ADAPT remains essentially the same across different bond distances. Results are shown for LiH, where the bond distance is varied from  $1\text{\AA}$  to  $3\text{\AA}$ . The fact that the curves

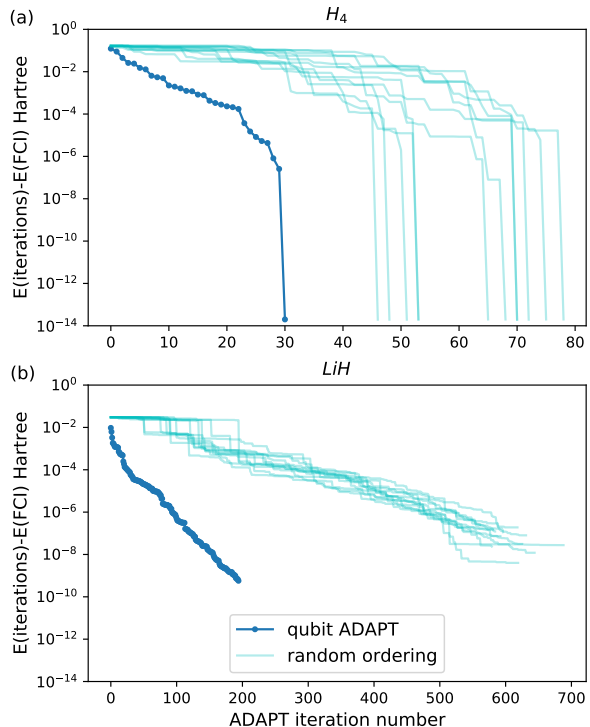


FIG. 2. The energy error of qubit-ADAPT versus ansätze with random operator orderings as a function of the number of iterations for (a)  $H_4$  and (b) LiH. The qubit pool is used in all cases, and the orbital bases and bond distances are chosen as in Fig. 1.

corresponding to different bond lengths largely overlap one another shows that the rate of convergence does not change significantly. Because the amount of correlation in the ground state depends on the bond distance (more correlations tend to arise as the bonds are stretched), this suggests that the convergence of qubit-ADAPT is not sensitive to the strength of correlations. This finding indicates that qubit-ADAPT is a promising approach to studying strongly correlated systems.

### B. Operator pool reduction

So far, a drawback of the qubit pool is its large size compared to the fermionic pool, which in turn leads to proportionally more measurements at each iteration. Even though the qubit pool is defined from the fermionic pool, which is a small subset of the full Pauli group, the pool size grows quickly with the number of orbitals. However, many of these operators are redundant in the sense that eliminating them has no effect on the convergence of the algorithm. For example, there are pairs of operators that are related by a global rotation, e.g.  $X_0Y_1Y_2Y_3$  and  $Y_0X_1X_2X_3$ , so we only need to keep one of them in the pool; discarding the other does not have any significant effect on the performance of qubit-ADAPT.

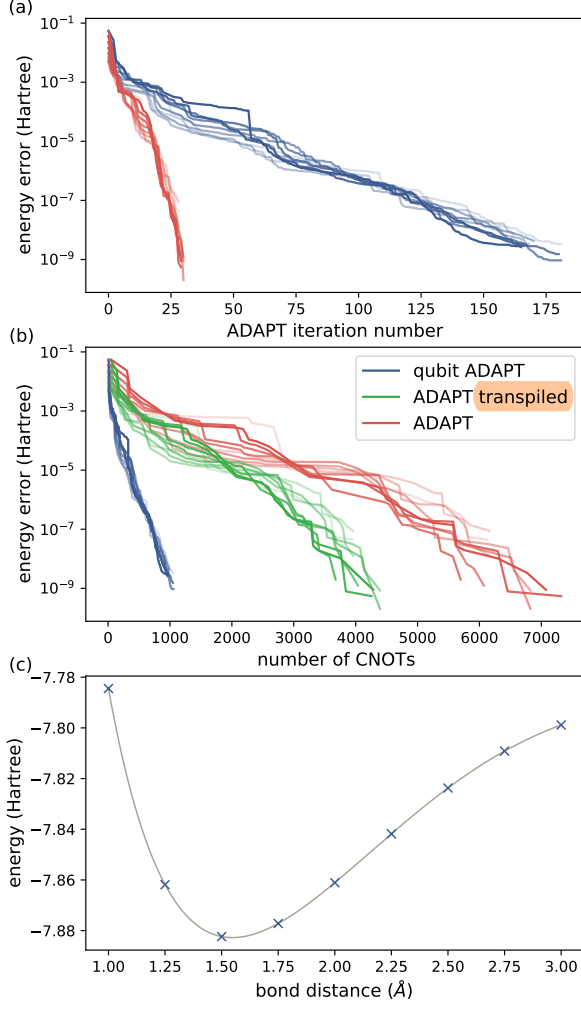


FIG. 3. qubit-ADAPT and fermionic-ADAPT energy error of LiH for various bond distances. The bond distance varies from 1 Å to 3 Å with darker curves in (a), (b) corresponding to larger bond distance. The energy error is plotted against (a) the number of parameters and (b) the number of CNOTs. The performance remains similar across different bond lengths and hence across different amounts of correlation in the ground state. The orbital basis used is the same as in Fig. 1. (c) The final energies obtained by qubit-ADAPT (points marked by  $\times$ ) for each bond distance considered are plotted along with the exact ground state energies (solid curve).

The redundancy in the pool can be illustrated by removing randomly selected operators from the pool and monitoring the impact on convergence. First we randomly remove 3/4 of the operators in the pool. As shown in Fig. 4, despite this large reduction of the pool size, the performance of the algorithm is similar to that with the original pool. However, as is also evident in Fig. 4, if we further remove more operators, the pool is sometimes incomplete, and the energy may not converge to the ground state energy. We can understand the tolerance of the al-

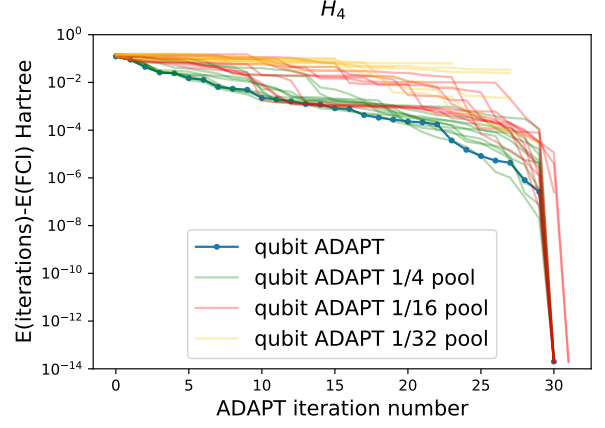


FIG. 4. qubit-ADAPT energy error of  $H_4$  using the full qubit pool, and randomly chosen 1/4, 1/16 and 1/32 pools are plotted against the number of iterations. For the 1/4 and 1/16 pools, the algorithm can converge for almost every run. For the 1/32 pool, most of the runs can only reach an energy error of  $10^{-3}$ . The orbital basis and bond distance are chosen as in Fig. 1.

gorithm to the drastic reduction of the pool by studying the Hilbert space spanned by the pool operators. Starting from the expression for  $|\psi^{ADAPT}(\vec{\theta})\rangle$  in Eq. 1 and using the Baker-Campbell-Hausdorff formula repeatedly, we obtain

$$|\psi^{ADAPT}(\vec{\theta})\rangle = e^{\sum_i \phi_i A_i} |\psi^{HF}\rangle, \quad (9)$$

where the  $A_i$  include all the pool operators and their commutators, and the  $\phi_i$  are functions of  $\vec{\theta}$ . Therefore, the Hilbert space spanned by the pool is determined by the set  $\{A_i\}$ . Note that if the original pool is comprised of odd Pauli strings, then so is  $\{A_i\}$ . **If the operators in  $\{A_i\}$  can transform the reference state to any real state in the  $n$ -qubit Hilbert space, then the qubit-ADAPT ansatz is guaranteed to be exact**, and it is capable of converging to the ground state. (Here, the only symmetry we impose is time-reversal.) Note that if  $\{A_i\}$  includes all the odd Pauli strings (of which there are  $2^{n-1}(2^n - 1)$ , which scales exponentially with the number of qubits), then we could create an arbitrary orthogonal transformation in Eq. (9). However, spanning the Hilbert space requires only a subset of the odd Pauli strings, because only  $2^n - 1$  real parameters are required to create an arbitrary real state. In particular, we need  $\{A_i\}$  to be such that for an arbitrary state  $|\psi\rangle$ , the states  $A_i |\psi\rangle$  form a complete basis. In this case, we refer to  $\{A_i\}$  as a complete basis of operators.

The problem then is to determine the minimal pools that produce a complete basis of operators. For a given pool, we define the overlap matrix as  $M_{ij} = \langle \psi | A_i^\dagger A_j | \psi \rangle$  where  $|\psi\rangle$  is an arbitrary real state. If the rank of  $M$  satisfies  $r(M) \geq 2^n - 1$ , this pool is called complete. To determine the smallest complete pools, we randomly generate many different pools of increasing size and compute

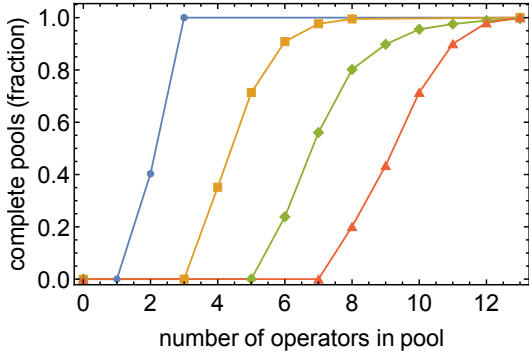


FIG. 5. The fraction of pools that are complete as a function of pool size for two (blue circles), three (yellow squares), four (green diamonds), and five (red triangles) qubits.

$r(M)$  in each case to test for completeness. We did this for up to 7 qubits. Our numerical investigations reveal that, surprisingly, the minimal pool size required for the overlap matrix to have the required rank of  $2^n - 1$  is only  $2n - 2$ . This is evident in Fig. 5, which shows the fraction of pools that are complete for various pool sizes and numbers of qubits. In each case, complete pools are found for pool sizes that contain at least  $2n - 2$  operators. Note that not only is this much smaller than the Hilbert space, it is also much smaller than the size of the fermionic pool, which scales like  $n^4$ .

Randomly selecting pools of size  $2n - 2$  and testing for completeness by computing  $r(M)$  is a numerically intensive process that quickly becomes infeasible as the number of qubits increases. This is further exacerbated by the fact that the fraction of complete pools of size  $2n - 2$  becomes smaller as  $n$  increases, as is evident in Fig. 5, which raises the question of whether complete pools of size  $2n - 2$  even exist for large values of  $n$ . In Appendix B, we prove analytically using induction that complete pools of size  $2n - 2$  exist for any  $n$ . In fact, the proof is constructive: We present two families of minimal pools that are provably complete for any number of qubits. One of these pools, which we call  $\{V_j\}_n$  where  $j = 1, \dots, 2n - 2$ , is defined recursively as  $\{V_j\}_n = \{Z_n\{V_k\}_{n-1}, iY_n, iY_{n-1}\}$ , starting from the pool for  $n = 2$  qubits:  $\{V_j\}_2 = \{iZ_2Y_1, iY_2\}$ . This pool is comprised of generators for single-qubit  $Y$  rotations and conditional  $Y$  rotations, and it contains operators that act on up to all  $n$  qubits. In Appendix B, we prove that these operators are sufficient to rotate any real state to any other real state in the Hilbert space. As shown in Appendix C,  $\{V_j\}_n$  can be mapped to a second family of minimal complete pools, which we call  $\{G_j\}_n$ , that has a very local structure. This pool contains all two-qubit operators of the form  $iZ_{k+1}Y_k$  that act on two neighboring qubits labeled by  $k$  and  $k + 1$ . There are  $n - 1$  such operators.  $\{G_j\}_n$  also contains all single-qubit Pauli  $iY$  operators except on the first qubit. Therefore, this pool contains  $2n - 2$  operators. In Appendix C, we show that the  $G_j$  can be obtained from commutators of the  $V_j$  for

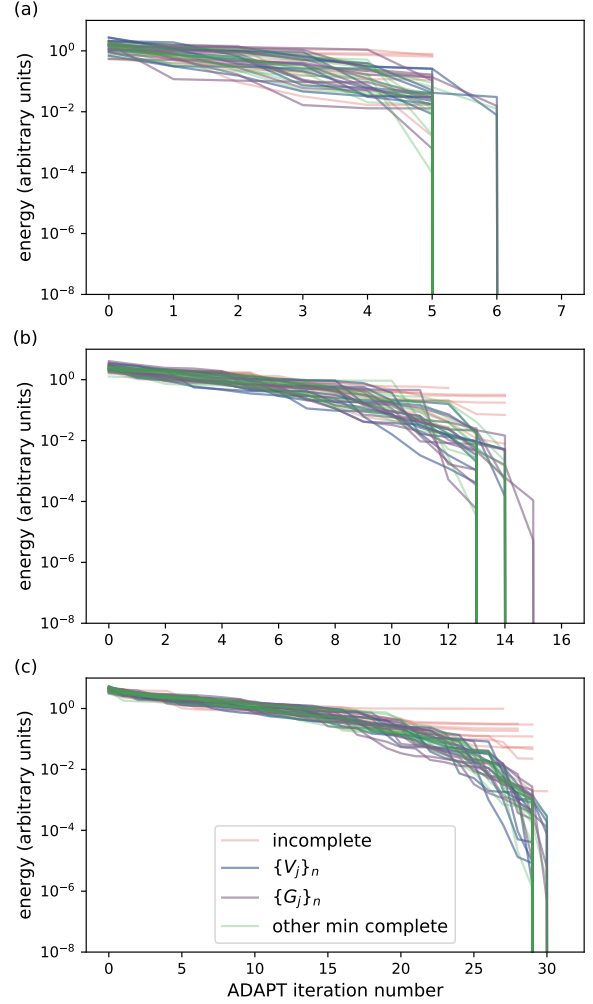


FIG. 6. Energy error curves for three different kinds of minimal complete pools versus incomplete pools. Results for 3, 4 and 5-qubit random Hamiltonians are shown in panels (a), (b) and (c) respectively.

any  $n$ , and so the completeness of the former follows from that of the latter.

To investigate the importance of the pool being complete, we run qubit-ADAPT for random real Hamiltonians of 3, 4 and 5 qubits with random initial states and for pools consisting of  $2n - 2$  operators randomly chosen from the set of odd Pauli strings. We also run simulations using the minimal complete pools  $\{V_j\}_n$  and  $\{G_j\}_n$ . The operator coefficients in each Hamiltonian (which is taken to be real and symmetric) are obtained by sampling uniformly in the range  $[-2, 2]$  with 10 samples for each of these 3 cases. The corresponding energy error curves are illustrated in Fig. 6. Each curve is the result for a different random Hamiltonian. All of the curves that fail to converge correspond to incomplete pools. For these cases, even though the gradient goes to zero the ground state is

not reached because important operators are never generated. On the other hand, the runs with complete pools always converge, highlighting the importance of this criterion. For the cases considered, we find that 20-40% of pools containing  $2n - 2$  operators are complete. These findings shed light on the question of what constitutes a good operator pool for ADAPT-VQE. These points were not appreciated in the original paper [29], and moreover, they suggest that the fermionic pool used in that work is overcomplete. Furthermore, the fact that the minimal pool size is linearly proportional to  $n$  means that the number of additional measurements needed for each step of qubit-ADAPT is also linear in  $n$ . Thus, the extra measurement overhead of qubit-ADAPT remains modest as the problem size increases.

#### IV. CONCLUSIONS

In conclusion, we introduced a more efficient and NISQ-compatible version of the ADAPT-VQE algorithm, called qubit-ADAPT. The basic idea of qubit-ADAPT is to use a pool consisting of Pauli strings (rather than fermionic operators) so that the number of CNOT gates associated with each pool operator is reduced. We established a completeness condition that guarantees that a pool will generate an exact ADAPT ansatz, and we proved that the smallest possible pool that obeys this criterion scales linearly with the number of qubits. We also provide a constructive approach to generating minimal complete pools for an arbitrary number of qubits. These results remove the ad-hoc elements of the ADAPT algorithm and lead to a substantial reduction in the depths of state preparation circuits and in the number of measurements needed to run ADAPT-VQE on realistic hardware.

#### ACKNOWLEDGMENTS

This research was supported by the National Science Foundation (Award No. 1839136) and the US Department of Energy (Award No. de-sc0019199).

#### Appendix A: CNOT estimation

From the structure of the pool operators, we can try estimating the number of CNOT gates as a function of the number of variational parameters. These estimates are described below.

##### Qubit ADAPT

In order to get a conservative estimate on the CNOT count, we assume that all operators picked are four-qubit pauli strings, since exponentiating an  $n$ -qubit Pauli string

requires  $2(n-1)$  CNOT gates, the number of CNOT gates is

$$\bar{N}_{CNOT} = 6 \quad (A1)$$

##### Fermionic ADAPT

In the case of fermionic-ADAPT, the number of CNOTs used can be estimated as

$$\bar{N}_{CNOT} = (6 + 2 \times \bar{N}_Z) \times \bar{N}_{Pauli} \times \bar{N}_{spin} \quad (A2)$$

where  $\bar{N}_Z$  is the average number of Pauli-Z used for fixing the antisymmetry of fermionic operators,  $\bar{N}_{spin}$  is the average number of fermionic operators in each spin-adapted group and  $\bar{N}_{Pauli}$  is the average number of Pauli strings from each fermionic operators.

$$\begin{aligned} \bar{N}_{spin} &= \frac{\text{Total \# of fermionic terms } (a_p^\dagger a_q^\dagger a_r a_s)}{\text{Total spin adapted double ops in pool}} \\ &: \# \text{ of fermionic terms in a spin adapted op} \\ \bar{N}_{Pauli} &= \frac{\text{Total number of Pauli strings after JW}}{\text{Total fermionic terms}} \\ &: \# \text{ of Pauli strings from each fermionic term} \\ \bar{N}_Z &= \frac{\text{Total length of Pauli Z from all fermionic}}{\text{Total \# of fermionic terms}} \\ &: \text{increases the length of Pauli strings} \end{aligned} \quad (A3)$$

To compute these total counts, we can first classify the combination of spatial orbitals in 5 groups:

1.  $p \neq q \neq r \neq s$
2. One of  $p, q =$  One of  $r, s$
3.  $p = q \neq r \neq s$
4.  $p = q = r \neq s$
5.  $p = q \neq r = s$ .

Since some of them can have both triplet and singlet while the others can only have singlet. And the number of fermionic terms in spin adapted grouping also depends on the combination of spatial orbitals.

For group 1, the triplet operator reads as

$$\begin{aligned} \hat{\tau}_{2,T} &\propto |T, 1\rangle_{pq} \langle T, 1|_{rs} + |T, -1\rangle_{pq} \langle T, -1|_{rs} \\ &\quad + |T, 0\rangle_{pq} \langle T, 0|_{rs} - h.c. \\ &= |pq\rangle \langle rs| + \frac{1}{2} (|p\bar{q}\rangle + |\bar{p}q\rangle) (\langle r\bar{s}| + \langle \bar{r}s|) \\ &\quad + |\bar{p}\bar{q}\rangle \langle \bar{r}s| - h.c. \\ &\rightarrow a_p^\dagger a_q^\dagger a_r a_s + \frac{1}{2} \left( a_p^\dagger a_{\bar{q}}^\dagger a_r a_{\bar{s}} + a_p^\dagger a_{\bar{q}}^\dagger a_{\bar{r}} a_s \right. \\ &\quad \left. + a_{\bar{p}}^\dagger a_q^\dagger a_r a_{\bar{s}} + a_{\bar{p}}^\dagger a_q^\dagger a_{\bar{r}} a_s \right) + a_{\bar{p}}^\dagger a_{\bar{q}}^\dagger a_{\bar{r}} a_{\bar{s}} - h.c., \end{aligned}$$



so there are 6 terms in total (12 terms including Hermitian conjugate). For the singlet,

$$\begin{aligned}\hat{\tau}_{2,S} &\propto |S,0\rangle_{pq} \langle S,0|_{rs} - h.c. \\ &= \frac{1}{2} (|p\bar{q}\rangle - |\bar{p}q\rangle) (\langle r\bar{s}| - \langle \bar{r}s|) - h.c. \\ &\rightarrow \frac{1}{2} \left( a_p^\dagger a_{\bar{q}}^\dagger a_r a_{\bar{s}} - a_p^\dagger a_{\bar{q}}^\dagger a_{\bar{r}} a_s \right. \\ &\quad \left. - a_{\bar{p}}^\dagger a_q^\dagger a_r a_{\bar{s}} + a_{\bar{p}}^\dagger a_q^\dagger a_{\bar{r}} a_s \right) - h.c.,\end{aligned}$$

so there are 4 terms in total (8 terms including Hermitian conjugate). For group 2, the triplet operator reads as (for  $q = r$ )

$$\begin{aligned}\hat{\tau}_{2,T} &\rightarrow a_p^\dagger a_q^\dagger a_q a_s + \frac{1}{2} \left( a_p^\dagger a_{\bar{q}}^\dagger a_q a_{\bar{s}} + a_p^\dagger a_{\bar{q}}^\dagger a_{\bar{q}} a_s \right. \\ &\quad \left. + a_{\bar{p}}^\dagger a_q^\dagger a_q a_{\bar{s}} + a_{\bar{p}}^\dagger a_q^\dagger a_{\bar{q}} a_s \right) + a_{\bar{p}}^\dagger a_{\bar{q}}^\dagger a_{\bar{q}} a_{\bar{s}} - h.c.,\end{aligned}$$

while the singlet operator is

$$\begin{aligned}\hat{\tau}_{2,S} &\rightarrow \frac{1}{2} \left( a_p^\dagger a_{\bar{q}}^\dagger a_q a_{\bar{s}} - a_p^\dagger a_{\bar{q}}^\dagger a_{\bar{q}} a_s \right. \\ &\quad \left. - a_{\bar{p}}^\dagger a_q^\dagger a_q a_{\bar{s}} + a_{\bar{p}}^\dagger a_q^\dagger a_{\bar{q}} a_s \right) - h.c.. \end{aligned}$$

For group 3, there is only singlet

$$\begin{aligned}\hat{\tau}_{2,S} &\rightarrow \frac{1}{2} \left( a_p^\dagger a_{\bar{p}}^\dagger a_r a_{\bar{s}} - a_p^\dagger a_{\bar{p}}^\dagger a_{\bar{r}} a_s \right. \\ &\quad \left. - a_{\bar{p}}^\dagger a_p^\dagger a_r a_{\bar{s}} + a_{\bar{p}}^\dagger a_p^\dagger a_{\bar{r}} a_s \right) - h.c. \\ &= a_p^\dagger a_{\bar{p}}^\dagger a_r a_{\bar{s}} + a_p^\dagger a_{\bar{p}}^\dagger a_{\bar{r}} a_s - h.c.. \end{aligned}$$

For group 4, there is only singlet

$$\begin{aligned}\hat{\tau}_{2,S} &\rightarrow \frac{1}{2} \left( a_p^\dagger a_{\bar{p}}^\dagger a_p a_{\bar{s}} - a_p^\dagger a_{\bar{p}}^\dagger a_{\bar{p}} a_s \right. \\ &\quad \left. - a_{\bar{p}}^\dagger a_p^\dagger a_p a_{\bar{s}} + a_{\bar{p}}^\dagger a_p^\dagger a_{\bar{p}} a_s \right) - h.c. \\ &= a_p^\dagger a_{\bar{p}}^\dagger a_p a_{\bar{s}} + a_p^\dagger a_{\bar{p}}^\dagger a_{\bar{p}} a_s - h.c.. \end{aligned}$$

For group 5, there is only singlet

$$\begin{aligned}\hat{\tau}_{2,S} &\rightarrow \frac{1}{2} \left( a_p^\dagger a_{\bar{p}}^\dagger a_r a_{\bar{r}} - a_p^\dagger a_{\bar{p}}^\dagger a_{\bar{r}} a_r \right. \\ &\quad \left. - a_{\bar{p}}^\dagger a_p^\dagger a_r a_{\bar{r}} + a_{\bar{p}}^\dagger a_p^\dagger a_{\bar{r}} a_r \right) - h.c. \\ &= 2a_p^\dagger a_{\bar{p}}^\dagger a_r a_{\bar{r}} - h.c.. \end{aligned}$$

Then we can perform Jordan-Wigner transformation to each of these terms. Each anti-Hermitian pair of fermionic operators having 4 different indices, i.e.  $a_i^\dagger a_j^\dagger a_k a_l - h.c.$ , would result in 8 Pauli strings, i.e.

$$\begin{aligned}a_i^\dagger a_j^\dagger a_k a_l - h.c. &= 2i \prod_n Z_n \left( Y_i X_j X_k X_l + X_i Y_j X_k X_l \right. \\ &\quad \left. - X_i X_j Y_k X_l - X_i X_j X_k Y_l \right. \\ &\quad \left. - X_i Y_j Y_k Y_l - Y_i X_j Y_k Y_l \right. \\ &\quad \left. + Y_i Y_j X_k Y_l + Y_i Y_j Y_k X_l \right).\end{aligned}$$

The other case is 2 of the 4 indices are the same,

$$\begin{aligned}a_i^\dagger a_j^\dagger a_j a_l - h.c. &= \prod_n Z_n (X_i + iY_i) Z_j (X_l - iY_l) - h.c. \\ &= 4i \prod_n Z_n (Y_i Z_j X_l - X_i Z_j Y_l),\end{aligned}$$

only 2 Pauli strings left. From the expression of the 5 cases' fermionic operators, we can then determine the average number of Pauli strings per anti-Hermitian pair for each case.

Group	1	2	3	4	5
# of fermi ops (T)	6	6	NA	NA	NA
# of fermi ops (S)	4	4	2	2	1
Average # of Paulis (T)	8	$(4 \times 2 + 2 \times 8)/6 = 4$	NA	NA	NA
Average # of Paulis (S)	8	$(2 \times 2 + 2 \times 8)/4 = 5$	8	2	8

For the length of Pauli Z, it is equivalent to count the interval between  $i$  and  $j$  and between  $k$  and  $l$  where  $i \leq j \leq k \leq l$ , we calculate them for separate cases, for  $m$  spatial orbitals,

1.  $p \neq q \neq r \neq s$ :

The total number of Pauli-Z is given by the sum

$$\begin{aligned}&\frac{1}{2} \binom{4}{2} \times (6+4) \times \sum_{m_1 < m_2 < m_3 < m_4} \left[ (2m_4 - 2m_3 - 1) \right. \\ &\quad \left. + (2m_2 - 2m_1 - 1) \right] \\ &= \frac{m}{2} (2m^4 - 15m^3 + 40m^2 - 45m + 18).\end{aligned}$$

The prefactor  $\frac{1}{2} \binom{4}{2}$  is counting ways we choose 2 of  $\{p, q, r, s\}$  to be in dagger side, while the prefactor  $(6+4)$  is from the number of fermionic operators for each combination. If we have  $\{i, j, k, l\} = \{1, 2, 3, 4\}$ , which is in group 5, the Jordan-Wigner transformation would bring a Pauli Z chain  $Z_1 Z_2 Z_3$  for the creation/annihilation operator on index 4 and a Pauli Z chain  $Z_1 Z_2$  for operator on index 3, so only  $Z_3$  can survive, but it would combine with the  $X_3 \pm iY_3$  from the operator on index 3, so there is no Pauli Zs effectively. Therefore, we should count the number of integers can be fitted in the intervals  $2m_4 - 2m_3$  and  $2m_2 - 2m_1$ , that explains the terms  $-1$  in the calculation.

In this counting, we treat all the spin orbitals to be spin-up (even number indices) even though each term in  $|T, 0\rangle \langle T, 0|$  and  $|S, 0\rangle \langle S, 0|$  have 2 spin-down and 2 spin-up operators. It is because whenever there are terms with both spin-up and spin-down spin orbitals, there would be another term with the spin totally

Group	1	2	3
# of combinations	$\binom{m}{2}\binom{m-2}{2}/2$	$m\binom{m-1}{2}$	$m\binom{m-1}{2}$
# of spin groups	$\binom{m}{2}\binom{m-2}{2}/2 \times 2$	$m\binom{m-1}{2} \times 2$	$m\binom{m-1}{2}$
# of fermi ops	$\binom{m}{2}\binom{m-2}{2}/2 \times (6+4)$	$m\binom{m-1}{2} \times (6+4)$	$m\binom{m-1}{2} \times 2$
# of Pauli strings	$\binom{m}{2}\binom{m-2}{2}/2 \times (6+4) \times 8$	$m\binom{m-1}{2} \times (6 \times 4 + 4 \times 5)$	$m\binom{m-1}{2} \times 2 \times 8$
# of Pauli Z	$\frac{m}{2}(2m^4 - 15m^3 + 40m^2 - 45m + 18)$	$\frac{10m}{3}(m^3 - 4m^2 + 5m - 2)$	$\frac{2m}{3}(m^3 - 4m^2 + 5m - 2)$

Group	4	5	Total
# of combinations	$2\binom{m}{2}$	$\binom{m}{2}$	$m(m^3 + 2m^2 - m - 2)/8$
# of spin groups	$2\binom{m}{2}$	$\binom{m}{2}$	$m^2(m^2 - 1)/4$
# of fermi ops	$2\binom{m}{2} \times 2$	$\binom{m}{2}$	$m(5m^3 - 6m^2 - 7m + 8)/4$
# of Pauli strings	$2\binom{m}{2} \times 2 \times 2$	$\binom{m}{2} \times 8$	$2m(5m^3 - 15m^2 + 14m - 4)$
# of Pauli Z	$\frac{2m}{3}(2m^2 - 3m + 1)$	0	$m(6m^4 - 21m^3 + 32m^2 - 27m + 10)/6$

TABLE I. For each group, the number of combinations are calculated. For each combination, it has 2 spin group if it has both triplet and singlet operators, only 1 spin group if it only has singlet. For each singlet and triplet operator, the number of fermionic terms are calculated. And then we calculate the number of Paulis in each term. Finally, we counted the length of Pauli Z.

flipped, so the counting of these two terms would cancel each other. For example, consider  $p < q < r < s$ , the  $|T, 0\rangle\langle T, 0|$  terms read as  $\frac{1}{2} \left( a_p^\dagger a_q^\dagger a_r a_s + a_p^\dagger a_q^\dagger a_{\bar{r}} a_s + a_p^\dagger a_q^\dagger a_r a_{\bar{s}} + a_p^\dagger a_q^\dagger a_{\bar{r}} a_{\bar{s}} \right)$  - h.c., the first term and the last term are different by a total spin flip, the number of Pauli Z would be  $2s + 1 - 2r - 1 + 2q + 1 - 2p - 1$  and  $2s - (2r + 1) - 1 + 2q - (2p + 1) - 1$ , so the +1 due to the odd index switches sign by the spin flip. Therefore, the average number of Pauli Z of this pair is the  $2s - 2r - 1 + 2q - 2p - 1$ , which is the number of Pauli Z in the all spin-up term  $a_p^\dagger a_q^\dagger a_r a_s$  (or the all spin-down term).

## 2. One of $p, q =$ One of $r, s$

The total number of Pauli-Z is given by the sum

$$\begin{aligned}
& 10 \times \left[ \sum_{m_1=m_2 < m_3 < m_4} (2m_4 - 2m_3 - 1) \right. \\
& + \sum_{m_1 < m_2 = m_3 < m_4} (2m_4 - 2m_3 - 1 + 2m_3 - 2m_1 - 1) \\
& + \left. \sum_{m_1 < m_2 < m_3 = m_4} (2m_2 - 2m_1 - 1) \right] \\
& = \frac{10m}{3} (m^3 - 4m^2 + 5m - 2).
\end{aligned}$$

Here, we don't have the counting prefactor in group 1 as we already decided that one of the two identical indices is in the dagger side.

## 3. $p = q \neq r \neq s$

The total number of Pauli-Z is given by the

$$\begin{aligned}
& \text{sum} \\
& 2 \times \left[ \sum_{m_1=m_2 < m_3 < m_4} (2m_4 - 2m_3 - 1) \right. \\
& + \sum_{m_1 < m_2 = m_3 < m_4} (2m_4 - 2m_3 - 1 + 2m_3 - 2m_1 - 1) \\
& + \left. \sum_{m_1 < m_2 < m_3 = m_4} (2m_2 - 2m_1 - 1) \right] \\
& = \frac{2m}{3} (m^3 - 4m^2 + 5m - 2).
\end{aligned}$$

It is only different from case 2 by the prefactor 2 instead of 10 as it has 2 fermionic operators in total.

## 4. $p = q = r \neq s$

The total number of Pauli-Z is given by the sum

$$\begin{aligned}
& 2 \times \left[ \sum_{m_1=m_2=m_3 < m_4} (2m_4 - 2m_1 - 1) \right. \\
& + \left. \sum_{m_1 < m_2=m_3=m_4} (2m_4 - 2m_1 - 1) \right] \\
& = \frac{2m}{3} (2m^2 - 3m + 1).
\end{aligned}$$

## 5. $p = q \neq r = s$

Since the operators have a form  $a_p^\dagger a_{\bar{p}}^\dagger a_r a_{\bar{r}}$ , they don't have Pauli Z.

Using Eq.(A3) and Table I, the average number can then be calculated as follow

$$\bar{N}_{spin} = \frac{5m^2 - m - 8}{m^2 + m} \quad (\text{A4})$$

$$\bar{N}_{Pauli} = \frac{8(5m^3 - 15m^2 + 14m - 4)}{5m^3 - 6m^2 - 7m + 8} \quad (\text{A5})$$

$$\bar{N}_Z = \frac{12m^3 - 30m^2 + 34m - 20}{3(8 + m - 5m^2)}. \quad (\text{A6})$$

We can estimate the CNOTs required for H4 molecule (m=4) is

$$\begin{aligned} N_{CNOT} &\approx N_{pars} \times N_{Pauli} \times (6 + 2 \times N_Z) \times N_{spin} \\ &= 11 \times N_{Pauli} \times (6 + 2 \times N_Z) \times N_{spin} \\ &= 1928.41 \end{aligned}$$

which is very close to the number we obtained.

	H4	LiH	H6
estimated # of CNOT per param	175	303	303
# of param	11	30	91
total estimated # of CNOT	1928	9078	27536
# of CNOT	2208	6824	28632

This estimation is accurate when all the spin orbitals are evenly explored (H6 and H4), i.e. no ‘core’ orbitals.

## Appendix B: Constructive proof of minimal complete pools

In this appendix, we present a constructive proof that there exist complete pools for qubit-ADAPT containing only  $2n - 2$  operators, which we refer to as minimal complete pools. We first prove that the set of operators  $\{V_i\}_{n=3} = \{iZ_3Z_2Y_1, iZ_3Y_2, iY_3, iY_2\}$  forms a complete pool for  $n = 3$  qubits, i.e., using operators of the form  $\prod_k e^{\theta_k V_k}$ , we can rotate any real state into any other real state. We then prove that the recursively defined pool,  $\{V_i\}_n = \{Z_n\{V_i\}_{n-1}, iY_n, iY_{n-1}\}$ , is a complete pool for  $n$  qubits, such that  $\prod_k e^{\theta_k V_k} |\psi\rangle = |\varphi\rangle$  for any two real states  $|\psi\rangle$  and  $|\varphi\rangle$  in the Hilbert space.

### 1. Complete pool for 3 qubits

Here, we prove that  $\{V_i\}_{n=3} = \{iZ_3Z_2Y_1, iZ_3Y_2, iY_3, iY_2\}$  is a minimal complete pool for three qubits. We will do this by showing that we can map any state  $|\psi\rangle$  to  $|000\rangle$  using only operators of the form  $\prod_k e^{\theta_k V_k}$ .

We begin by decomposing an arbitrary real state  $|\psi\rangle$  as follows:

$$|\psi\rangle = |0\rangle_3 |\psi_0\rangle_{21} + |1\rangle_3 |\psi_1\rangle_{21}. \quad (B1)$$

The subscripts outside the kets indicate which qubit(s) the ket belongs to. In general, the two-qubit states  $|\psi_0\rangle$  and  $|\psi_1\rangle$  do not have the same norm:  $\langle\psi_0|\psi_0\rangle \neq \langle\psi_1|\psi_1\rangle$ . However, we can always perform a rotation  $e^{i\theta Y_3}$  on qubit 3 (the left-most qubit) to transform  $|\psi\rangle$  to the form

$$|\psi'\rangle = e^{i\theta Y_3} |\psi\rangle = |0\rangle_3 |\psi'_0\rangle_{21} + |1\rangle_3 |\psi'_1\rangle_{21}, \quad (B2)$$

where

$$|\psi'_0\rangle = \cos \theta |\psi_0\rangle + \sin \theta |\psi_1\rangle, \quad (B3)$$

$$|\psi'_1\rangle = \cos \theta |\psi_1\rangle - \sin \theta |\psi_0\rangle. \quad (B4)$$

The norms of these states are

$$\begin{aligned} \langle\psi'_0|\psi'_0\rangle &= \cos^2 \theta \langle\psi_0|\psi_0\rangle + \sin^2 \theta \langle\psi_1|\psi_1\rangle \\ &\quad + \sin(2\theta) \langle\psi_0|\psi_1\rangle, \end{aligned} \quad (B5)$$

$$\begin{aligned} \langle\psi'_1|\psi'_1\rangle &= \cos^2 \theta \langle\psi_1|\psi_1\rangle + \sin^2 \theta \langle\psi_0|\psi_0\rangle \\ &\quad - \sin(2\theta) \langle\psi_0|\psi_1\rangle. \end{aligned} \quad (B6)$$

We can make these two norms equal by choosing  $\theta$  such that

$$\tan(2\theta) = \frac{\langle\psi_1|\psi_1\rangle - \langle\psi_0|\psi_0\rangle}{2 \langle\psi_0|\psi_1\rangle}. \quad (B7)$$

Note that this procedure works for arbitrary states  $|\psi_0\rangle$  and  $|\psi_1\rangle$ , not just for two-qubit states.

The heart of the proof amounts to showing that a state of the form in Eq. (B2), where  $\langle\psi'_0|\psi'_0\rangle = \langle\psi'_1|\psi'_1\rangle = 1/2$ , can be transformed into a form where qubit 3 has been factored out:

$$|\psi'\rangle \rightarrow (|0\rangle + |1\rangle)_3 |\chi\rangle_{21}, \quad (B8)$$

where  $|\chi\rangle$  is some two-qubit state. We will do this using the (slightly reduced) set of pool operators  $\{V_i^{\text{red}}\}_3 = \{iZ_3Z_2Y_1, iZ_3Y_2, iY_2\}$ . These are the same as  $\{V_i\}_3$ , except that  $iY_3$  is not included. In the next subsection, we generalize Eq. (B8) to  $n$  qubits and then use it to prove that  $\{V_i\}_n$  is a complete pool.

To prove Eq. (B8), let's begin by expanding  $|\psi'\rangle$  as follows:

$$|\psi'\rangle = |00\rangle |\psi'_{0a}\rangle + |01\rangle |\psi'_{0b}\rangle + |10\rangle |\psi'_{1a}\rangle + |11\rangle |\psi'_{1b}\rangle. \quad (B9)$$

From this point onward, we suppress the subscripts on the kets for notational simplicity. Similarly to how we made the states  $|\psi'_0\rangle$  and  $|\psi'_1\rangle$  have the same norm, here we can apply the conditional operations  $e^{i\theta_0 \frac{1+Z_3}{2} Y_2}$  and  $e^{i\theta_1 \frac{1-Z_3}{2} Y_2}$  to make

$$\langle\psi'_{0a}|\psi'_{0a}\rangle = \langle\psi'_{0b}|\psi'_{0b}\rangle = \langle\psi'_{1a}|\psi'_{1a}\rangle = \langle\psi'_{1b}|\psi'_{1b}\rangle = 1/4. \quad (B10)$$

We assume that this has already been done in Eq. (B9), and that  $|\psi'\rangle$  has been redefined accordingly.

Next, we apply the conditional rotation  $e^{i\phi_1 Z_3 Z_2 Y_1}$  on qubit 1 to bring the state to

$$e^{i\phi_1 Z_3 Z_2 Y_1} |\psi'\rangle = (|00\rangle + |01\rangle) |\chi_0\rangle + |10\rangle |\psi''_{1a}\rangle + |11\rangle |\psi''_{1b}\rangle. \quad (B11)$$

Here, we used the fact that  $|\psi'_{0a}\rangle$  and  $|\psi'_{0b}\rangle$  are real single-qubit states, and thus can be viewed as 2D vectors lying in the same plane. The conditional rotation  $e^{i\phi_1 Z_3 Z_2 Y_1}$  rotates these two vectors in opposite directions because one is conditioned on the state  $|00\rangle$  and the other on  $|01\rangle$ , and eventually the two vectors coincide at  $|\chi_0\rangle$  for a particular value of the rotation angle  $\phi_1$ . This operation also affects the other terms in Eq. (B11) (as indicated with the extra primes), but their form remains the same. This conditional operation will be used repeatedly in what follows.

The next step is to apply a conditional rotation on qubit 2 to bring it to the state  $|0\rangle$  in the first term:

$$e^{i\phi_2 \frac{1+Z_3}{2} Y_2} e^{i\phi_1 Z_3 Z_2 Y_1} |\psi'\rangle = \sqrt{2} |00\rangle |\chi_0\rangle + |10\rangle |\psi''_{1a}\rangle + |11\rangle |\psi''_{1b}\rangle. \quad (\text{B12})$$

We can then apply another conditional operation on qubit 1 to rotate the states  $|\psi''_{1a}\rangle$  and  $|\psi''_{1b}\rangle$  into each other:

$$e^{i\phi_3 Z_3 Z_2 Y_1} e^{i\phi_2 \frac{1+Z_3}{2} Y_2} e^{i\phi_1 Z_3 Z_2 Y_1} |\psi'\rangle = \sqrt{2} |00\rangle |\chi'_0\rangle + (|10\rangle + |11\rangle) |\chi_1\rangle. \quad (\text{B13})$$

This operation brings the two states to some state  $|\chi_1\rangle$ . It also changes  $|\chi_0\rangle$  to  $|\chi'_0\rangle$ . Another conditional rotation on qubit 2 simplifies the second term:

$$e^{i\phi_4 \frac{1-Z_3}{2} Y_2} e^{i\phi_3 Z_3 Z_2 Y_1} e^{i\phi_2 \frac{1+Z_3}{2} Y_2} e^{i\phi_1 Z_3 Z_2 Y_1} |\psi'\rangle = \sqrt{2} |00\rangle |\chi'_0\rangle + \sqrt{2} |10\rangle |\chi_1\rangle. \quad (\text{B14})$$

We need one more conditional operation on qubit 1 to bring this to the desired form:

$$e^{i\phi_5 Z_3 Z_2 Y_1} e^{i\phi_4 \frac{1-Z_3}{2} Y_2} e^{i\phi_3 Z_3 Z_2 Y_1} e^{i\phi_2 \frac{1+Z_3}{2} Y_2} e^{i\phi_1 Z_3 Z_2 Y_1} |\psi'\rangle = \sqrt{2} (|00\rangle + |10\rangle) |\chi\rangle = (|0\rangle + |1\rangle) |0\rangle |\chi\rangle \sqrt{2}. \quad (\text{B15})$$

We have thus proven Eq. (B8). For reasons that will become clear in the next subsection, it is important to note that we did this using only the reduced pool  $\{V_i^{\text{red}}\}_3$ .

All that remains is to show that Eq. (B15) can be mapped to the state  $|000\rangle$ . This can be done using a  $Y_3$  rotation followed by a conditional operation on qubit 1:

$$e^{i\phi_6 Z_3 Z_2 Y_1} e^{i\frac{\pi}{4} Y_3} e^{i\phi_5 Z_3 Z_2 Y_1} e^{i\phi_4 \frac{1-Z_3}{2} Y_2} e^{i\phi_3 Z_3 Z_2 Y_1} e^{i\phi_2 \frac{1+Z_3}{2} Y_2} \times e^{i\phi_1 Z_3 Z_2 Y_1} |\psi'\rangle = |000\rangle. \quad (\text{B16})$$

Since we can map any state to  $|000\rangle$ , it follows that we can map any 3-qubit state to any other 3-qubit state using only the pool operators. We have thus shown that  $\{V_i\}_3 = \{iZ_3 Z_2 Y_1, iZ_3 Y_2, iY_3, iY_2\}$  is a minimal complete pool for three qubits.

## 2. Proof of the complete pool for $n$ qubits

In the previous subsection, we proved that the pool  $\{V_i\}_3$  is a complete pool for three qubits. A key piece of this proof is the statement that we can use the reduced pool  $\{V_i^{\text{red}}\}_3 = \{iZ_3 Z_2 Y_1, iZ_3 Y_2, iY_2\}$  to factor out the 3rd qubit Eq. (B8):

$$|0\rangle |\psi'_0\rangle + |1\rangle |\psi'_1\rangle \rightarrow (|0\rangle + |1\rangle) |\chi\rangle, \quad (\text{B17})$$

where we assume  $\langle\psi'_0|\psi'_0\rangle = \langle\psi'_1|\psi'_1\rangle$ , and  $|\chi\rangle$  is some 2-qubit state. We will prove that a similar statement holds for  $n$  qubits:

**Theorem:** We can factor out the  $n$ th qubit from a generic  $n$ -qubit state:

$$|0\rangle |\psi'_0\rangle + |1\rangle |\psi'_1\rangle \rightarrow (|0\rangle + |1\rangle) |\chi\rangle, \quad (\text{B18})$$

where  $\langle\psi'_0|\psi'_0\rangle = \langle\psi'_1|\psi'_1\rangle$ , and  $|\chi\rangle$  is some  $(n-1)$ -qubit state, using only the reduced  $n$ -qubit pool:

$$\{V_i^{\text{red}}\}_n = \{Z_n \{V_i\}_{n-1}, iY_{n-1}\} = \{Z_n \{V_i^{\text{red}}\}_{n-1}, iZ_n Y_{n-1}, iY_{n-1}\}. \quad (\text{B19})$$

This pool is defined recursively starting from  $\{V_i^{\text{red}}\}_3$ , and it differs from the full pool by only one operator ( $iY_n$ ):

$$\{V_i\}_n = \{Z_n \{V_i\}_{n-1}, iY_n, iY_{n-1}\}. \quad (\text{B20})$$

We will use induction to prove this theorem. We will assume it holds for  $n$  qubits and then show that it also holds for  $n+1$  qubits. This proof will closely follow what we did for three qubits above. Thus, we start with the  $(n+1)$ -qubit state:

$$|0\rangle_{n+1} |\psi'_0\rangle + |1\rangle_{n+1} |\psi'_1\rangle, \quad (\text{B21})$$

where  $|\psi'_0\rangle$  and  $|\psi'_1\rangle$  are  $n$ -qubit states that have the same norm. We want to factor out the left-most  $(n+1)$ th qubit using only the operators  $\{V_i^{\text{red}}\}_{n+1} = \{Z_{n+1} \{V_i^{\text{red}}\}_n, iZ_{n+1} Y_n, iY_n\}$ . We decompose the state further:

$$|00\rangle |\psi'_{0a}\rangle + |01\rangle |\psi'_{0b}\rangle + |10\rangle |\psi'_{1a}\rangle + |11\rangle |\psi'_{1b}\rangle. \quad (\text{B22})$$

As before, we can assume that all four states  $|\psi'_{0a}\rangle, |\psi'_{0b}\rangle, |\psi'_{1a}\rangle, |\psi'_{1b}\rangle$ , have the same norm of  $1/4$ , since this can be achieved by applying two conditional operations  $e^{i\theta_0 \frac{1+Z_{n+1}}{2} Y_n}$  and  $e^{i\theta_1 \frac{1-Z_{n+1}}{2} Y_n}$ .

We next apply a conditional operation built from the operators  $e^{\phi_1 Z_{n+1} \{V_i^{\text{red}}\}_n}$  to map the state to

$$(|00\rangle + |01\rangle) |\chi_0\rangle + |10\rangle |\psi''_{1a}\rangle + |11\rangle |\psi''_{1b}\rangle. \quad (\text{B23})$$

We used the induction hypothesis for  $n$  qubits to obtain this result. Since the left-most qubit is in the state  $|0\rangle$  in the first two terms, the effect of  $e^{\phi_1 Z_{n+1} \{V_i^{\text{red}}\}_n}$  is the same as  $e^{\phi_1 \{V_i^{\text{red}}\}_n}$  on these terms, and the hypothesis for  $n$  qubits then allows us to bring these two terms into a factorized form as above. The last two terms also evolve under this operation, but their general form remains the same since the  $n+1$  qubit remains in the state  $|1\rangle$  under this operation. We follow this with a conditional operation  $e^{i\phi_2 \frac{1+Z_{n+1}}{2} Y_n}$  on the  $n$ th qubit to rotate it to  $|0\rangle$  in the first term:

$$\sqrt{2} |00\rangle |\chi_0\rangle + |10\rangle |\psi''_{1a}\rangle + |11\rangle |\psi''_{1b}\rangle. \quad (\text{B24})$$

Another application of  $e^{\phi_3 Z_{n+1} \{V_i^{\text{red}}\}_n}$  can be used to similarly factorize the last two terms:

$$\sqrt{2} |00\rangle |\chi'_0\rangle + (|10\rangle + |11\rangle) |\chi_1\rangle. \quad (\text{B25})$$



Note that the form of the first term remains the same since this conditional operation does not rotate the  $(n+1)$ th or  $n$ th qubits. This is because all the operators in  $\{V_i^{\text{red}}\}_n$  have either a  $Z$  or a  $I$  on the  $n$ th qubit. Next apply  $e^{i\phi_4 \frac{1-Z_{n+1}}{2} Y_n}$  to rotate the  $n$ th qubit in the second term to  $|0\rangle$ :

$$\sqrt{2}|00\rangle|\chi'_0\rangle + \sqrt{2}|10\rangle|\chi_1\rangle. \quad (\text{B26})$$

Additional applications of  $e^{\phi Z_{n+1}\{V_i^{\text{red}}\}_n}$  bring this to

$$\sqrt{2}(|0\rangle + |1\rangle)|0\rangle|\chi\rangle, \quad (\text{B27})$$

as desired. In order to make this final step, we need to use a slight extension of the theorem in which we factor out the  $(n+1)$ th qubit instead of the  $n$ th qubit. Do we have all the necessary operators to make this extension work? Yes. To see this, recall that in order to factor out the  $n$ th qubit, we need  $\{V_i^{\text{red}}\}_n = \{Z_n\{V_i^{\text{red}}\}_{n-1}, iZ_n Y_{n-1}, iY_{n-1}\}$ . However, here we have

$$Z_{n+1}\{V_i^{\text{red}}\}_n = \{Z_{n+1}Z_n\{V_i^{\text{red}}\}_{n-1}, iZ_{n+1}Z_n Y_{n-1}, iZ_{n+1}Y_{n-1}\}. \quad (\text{B28})$$

The operators  $Z_{n+1}Z_n\{V_i^{\text{red}}\}_{n-1}$  and  $iZ_{n+1}Z_n Y_{n-1}$  produce exactly the same conditional rotations on the state in Eq. (B26) as  $Z_n\{V_i^{\text{red}}\}_{n-1}$  and  $iZ_n Y_{n-1}$  produce on  $\sqrt{2}|0\rangle|\chi'_0\rangle + \sqrt{2}|1\rangle|\chi_1\rangle$ . However, the theorem also requires unconditional  $Y_{n-1}$  rotations on  $|\chi'_0\rangle$  and  $|\chi_1\rangle$ . These can be implemented using the composite operator

$$e^{i\frac{\pi}{4}\frac{1-Z_{n+1}}{2}Y_n} e^{i\phi Z_{n+1}Z_n Y_{n-1}} e^{i\frac{\pi}{4}\frac{1-Z_{n+1}}{2}Y_n}, \quad (\text{B29})$$

which first rotates (B26) into a state in which the  $(n+1)$ th and  $n$ th qubit states in the two terms have the same

parity before performing the  $Z_{n+1}Z_n Y_{n-1}$  rotation, so that  $|\chi'_0\rangle$  and  $|\chi_1\rangle$  undergo the same rotation. The  $(n+1)$ th and  $n$ th qubits are then restored to their original states by the final rotation in Eq. (B29). Given that we have already shown that this procedure works for  $n=3$  qubits, this completes the proof of the theorem.

With the theorem proven, it is easy to prove that  $\{V_i\}_{n+1}$  is a complete pool. We already know that a rotation generated by  $Y_{n+1}$  is enough to bring an arbitrary  $(n+1)$ -qubit state to the form needed to apply the theorem. As we just saw, we can then use  $Z_{n+1}\{V_i^{\text{red}}\}_n$ ,  $iZ_{n+1}Y_n$ , and  $iY_n$  to factor out the  $(n+1)$ th qubit and arrive at Eq. (B27). If we then apply another  $Y_{n+1}$  rotation to bring the  $(n+1)$ th qubit to  $|0\rangle$ , we can perform the conditional operation  $e^{i\phi_6 Z_{n+1}\{V_i^{\text{red}}\}_n}$  to rotate  $\sqrt{2}|\chi\rangle$  to  $|0\rangle^{\otimes n-1}$ . Thus, we can rotate an arbitrary  $(n+1)$ -qubit state to  $|0\rangle^{\otimes n+1}$  using only the generators  $\{V_i\}_{n+1} = \{Z_{n+1}\{V_i^{\text{red}}\}_n, iZ_{n+1}Y_n, iY_{n+1}, iY_n\}$ . This in turn implies that we can rotate any  $(n+1)$ -qubit state to any other  $(n+1)$ -qubit state using the same set of operators. Therefore,  $\{V_i\}_{n+1}$  is a complete pool, and it contains  $2n-2$  operators.

### Appendix C: Mapping between minimal complete pools

In this section, we show that the two examples of minimal complete pools discussed in the main text can be mapped into one another. The first example is the pool  $\{V_i\}$  that is constructed iteratively in Appendix B. The second example is the pool comprised of operators that only act on adjacent qubits in a linear array,  $\{G_i\}$ . To facilitate the following analysis, we order the elements in these two pools as follows (where we have ignored the factors of  $i$  in the definition of the pool operators for simplicity):

$$\begin{aligned} V_1 &= ZZ \dots ZY, & V_2 &= ZZ \dots ZYI, & V_3 &= ZZ \dots ZYII, & \dots, & V_{n-1} &= ZYII \dots I, & V_n &= YII \dots I, \\ V_{n+1} &= ZZ \dots ZIYI, & V_{n+2} &= ZZ \dots ZIYII, & \dots, & V_{2n-3} &= ZIYII \dots I, & V_{2n-2} &= IYII \dots I. \end{aligned} \quad (\text{C1})$$

$$\begin{aligned} G_1 &= ZYII \dots I, & G_2 &= IZYII \dots I, & G_3 &= IIZYII \dots I, & \dots, & G_{n-2} &= II \dots IZYI, & G_{n-1} &= II \dots IZY, \\ G_n &= YII \dots I, & G_{n+1} &= IYII \dots I, & G_{n+2} &= I IYII \dots I, & \dots, & G_{2n-3} &= II \dots IYII, & G_{2n-2} &= II \dots IYI. \end{aligned} \quad (\text{C2})$$

We will show that the  $V_i$  can be obtained from commutators of the  $G_i$ . This means that the two pools share the same operator basis, and so completeness of the one pool implies completeness of the other. Since we have already proven that the  $V_i$  are complete in Appendix B, this demonstrates that the  $G_i$  also form a minimal complete pool for any number  $n$  of qubits.

Recall that two Pauli strings do not commute if they

differ by an odd number of Pauli operators, ignoring any qubits with identity operators in at least one of the two strings. In this case, the result of the commutator is proportional to the product of the two Pauli strings. Now

notice that we can write the  $V_i$  as products of the  $G_i$ :

$$\begin{aligned}
 V_{n-k} &\sim G_1 \prod_{j=2}^k G_j G_{j+n-1}, \quad k = 2, \dots, n-1, \\
 V_{n-1} &= G_1, \\
 V_n &= G_n, \\
 V_{2n-k} &\sim G_{k-1} G_k V_{n-k}, \quad k = 3, \dots, n-1, \\
 V_{2n-2} &= G_{n+1}.
 \end{aligned} \tag{C3}$$

The operators in the product in the first line should be ordered such that the operators with the smallest values of  $j$  are on the left and the ones with the largest values

are on the right. Notice that no two adjacent operators in these products commute. Therefore, we can rewrite them as nested commutators of the  $G_i$ . This proves that the  $V_i$  and  $G_i$  produce the same operator basis when all possible commutators of each pool are computed. We already showed in Appendix B that the  $V_i$  are complete. Therefore, the  $G_i$  also constitute a minimal complete pool. This pool is particularly useful for quantum processors containing a linear array of qubits with nearest-neighbor coupling only. Interestingly, the entangling operators in the  $G_i$  pool are similar to the cross-resonance interaction in fixed-frequency superconducting qubits, such as those in the IBM quantum processors [49–51]. This similarity could be exploited to implement qubit-ADAPT with native hardware operations.

- 
- [1] W. Kohn and L. J. Sham, *Phys. Rev.* **140**, A1133 (1965).
  - [2] P. Hohenberg and W. Kohn, *Phys. Rev.* **136**, B864 (1964).
  - [3] A. Kovyrshin and M. Reiher, *The Journal of Chemical Physics* **147**, 214111 (2017), <https://doi.org/10.1063/1.5004693>.
  - [4] K. H. Marti, B. Bauer, M. Reiher, M. Troyer, and F. Verstraete, *New Journal of Physics* **12**, 103008 (2010).
  - [5] V. Murg, F. Verstraete, R. Schneider, P. R. Nagy, and Ö. Legeza, *Journal of Chemical Theory and Computation* **11**, 1027 (2015).
  - [6] S. Szalay, M. Pfeiffer, V. Murg, G. Barcza, F. Verstraete, R. Schneider, and r. Legeza, *International Journal of Quantum Chemistry* **115**, 1342 (2015).
  - [7] K. Choo, A. Mezzacapo, and G. Carleo, *arXiv:1909.12852* [cond-mat, physics:physics, physics:quant-ph] (2019), *arXiv:1909.12852* [cond-mat, physics:physics, physics:quant-ph].
  - [8] S. R. White, *Phys. Rev. Lett.* **69**, 2863 (1992).
  - [9] U. Schollwöck, *Rev. Mod. Phys.* **77**, 259 (2005).
  - [10] G. K.-L. Chan and S. Sharma, *Ann. Rev. Phys. Chem.* **62**, 465 (2011).
  - [11] R. P. Feynman, *International Journal of Theoretical Physics* **21**, 467 (1982).
  - [12] S. McArdle, S. Endo, A. Aspuru-Guzik, S. Benjamin, and X. Yuan, *arXiv:1808.10402* (2018).
  - [13] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferov, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. D. Sawaya, S. Sim, L. Veis, and A. Aspuru-Guzik, *Chemical Reviews* **119**, 10856 (2019).
  - [14] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, *arXiv:2001.03685* (2020).
  - [15] A. Y. Kitaev, *arXiv:quant-ph/9511026* (1995).
  - [16] J. D. Whitfield, J. Biamonte, and A. Aspuru-Guzik, *Molecular Physics* **109**, 735 (2011).
  - [17] D. S. Abrams and S. Lloyd, *Phys. Rev. Lett.* **79**, 2586 (1997).
  - [18] D. S. Abrams and S. Lloyd, *Phys. Rev. Lett.* **83**, 5162 (1999).
  - [19] J. Preskill, *Quantum* **2**, 79 (2018).
  - [20] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O’Brien, *Nature Communications* **5** (2014).
  - [21] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, *New Journal of Physics* **18**, 023023 (2016).
  - [22] 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, *Phys. Rev. X* **8**, 011021 (2018).
  - [23] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Nature* **549**, 242 EP (2017).
  - [24] Y. Shen, X. Zhang, S. Zhang, J.-N. Zhang, M.-H. Yung, and K. Kim, *Phys. Rev. A* **95**, 020501 (2017).
  - [25] 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, *Phys. Rev. X* **8**, 031022 (2018).
  - [26] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, *npj Quantum Information* **6**, 10 (2020).
  - [27] R. J. Bartlett and M. Musiał, *Rev. Mod. Phys.* **79**, 291 (2007).
  - [28] M.-H. Yung, J. Casanova, A. Mezzacapo, J. McClean, L. Lamata, A. Aspuru-Guzik, and E. Solano, *Scientific Reports* **4**, 3589 EP (2014).
  - [29] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, *Nature Communications* **10**, 3007 (2019).
  - [30] I. G. Ryabinkin, T.-C. Yen, S. N. Genin, and A. F. Izmaylov, *J. Chem. Theory Comput.* **14**, 6317 (2018).
  - [31] I. G. Ryabinkin, R. A. Lang, S. N. Genin, and A. F. Izmaylov, *arXiv:1906.11192* (2019).
  - [32] J. Lee, W. J. Huggins, M. Head-Gordon, and K. B. Whaley, *J. Chem. Theory Comput.* **15**, 311 (2019).
  - [33] W. J. Huggins, J. Lee, U. Baek, B. O’Gorman, and K. B. Whaley, *arXiv:1909.09114* (2019).
  - [34] P.-L. Dallaire-Demers, J. Romero, L. Veis, S. Sim, and A. Aspuru-Guzik, *arXiv:1801.01053* (2018).
  - [35] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, *Quantum Science and Technology* **4**, 014008 (2018).
  - [36] Y. Matsuzawa and Y. Kurashige, *arXiv:1909.12410* (2019).
  - [37] F. A. Evangelista, G. Kin-Lic Chan, and G. E. Scuseria, *arXiv:1910.10130* (2019).
  - [38] H. R. Grimsley, D. Claudino, S. E. Economou, E. Barnes, and N. J. Mayhall, *Journal of Chemical Theory and Com-*

- putation **16**, 1 (2020).
- [39] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, *Nature Communications* **9**, 4812 (2018).
  - [40] P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, N. Moll, G. Salis, A. Fuhrer, M. Ganzhorn, D. J. Egger, M. Troyer, A. Mezzacapo, S. Filipp, and I. Tavernelli, *Phys. Rev. A* **98**, 022322 (2018).
  - [41] M. Ganzhorn, D. Egger, P. Barkoutsos, P. Ollitrault, G. Salis, N. Moll, M. Roth, A. Fuhrer, P. Mueller, S. Woerner, I. Tavernelli, and S. Filipp, *Phys. Rev. Applied* **11**, 044092 (2019).
  - [42] Because the gradient depends on the commutator with  $\hat{H}$  such that measurements associated with terms in  $\hat{H}$  which commute with a specific operator in the pool can be skipped, the actual number of measurements needed is difficult to predict, as it will depend on the operator pool chosen. Getting estimates for this number for various pools will be the focus of future work.
  - [43] W. J. Huggins, J. McClean, N. Rubin, Z. Jiang, N. Wiebe, K. B. Whaley, and R. Babbush, arXiv e-prints (2019).
  - [44] S. B. Bravyi and A. Y. Kitaev, *Annals of Physics* **298**, 210 (2002).
  - [45] V. Havlíček, M. Troyer, and J. D. Whitfield, *Phys. Rev. A* **95**, 032332 (2017).
  - [46] K. Setia and J. D. Whitfield, *The Journal of Chemical Physics* **148**, 164104 (2018).
  - [47] H. Abraham *et al.*, “Qiskit: An open-source framework for quantum computing,” (2019).
  - [48] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*, 10th ed. (Cambridge University Press, New York, NY, USA, 2011).
  - [49] C. Rigetti and M. Devoret, *Phys. Rev. B* **81**, 134507 (2010).
  - [50] J. M. Chow, A. D. Córcoles, J. M. Gambetta, C. Rigetti, B. R. Johnson, J. A. Smolin, J. R. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, and M. Steffen, *Phys. Rev. Lett.* **107**, 080502 (2011).
  - [51] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, *Phys. Rev. A* **93**, 060302 (2016).