



Layering and subpool exploration for adaptive variational quantum eigensolvers: Reducing circuit depth, runtime, and susceptibility to noise

Christopher K. Long ^{1,2} Kieran Dalton ^{1,2} Crispin H. W. Barnes,²

David R. M. Arvidsson-Shukur ¹ and Normann Mertig ¹

¹*Hitachi Cambridge Laboratory, J. J. Thomson Avenue, Cambridge CB3 0HE, United Kingdom*

²*Cavendish Laboratory, Department of Physics, University of Cambridge, Cambridge CB3 0HE, United Kingdom*



(Received 13 September 2023; accepted 23 January 2024; published 12 April 2024)

Adaptive variational quantum eigensolvers (ADAPT-VQEs) are promising candidates for simulations of strongly correlated systems on near-term quantum hardware. To further improve the noise resilience of these algorithms, recent efforts have been directed towards compactifying, or *layering*, their *Ansatz* circuits. Here, we broaden the understanding of the algorithmic layering process in three ways. First, we investigate the noncommutation relations between the different elements that are used to build ADAPT-VQE *Ansätze*. In doing so, we develop a framework for studying and developing layering algorithms, which produce shallower circuits. Second, based on this framework, we develop a new subroutine that can reduce the number of quantum-processor calls by optimizing the selection procedure with which a variational quantum algorithm appends *Ansatz* elements. Third, we provide a thorough numerical investigation of the noise-resilience improvement available via layering the circuits of ADAPT-VQE algorithms. We find that layering leads to an improved noise resilience with respect to amplitude-damping and dephasing noise, which, in general, affect idling and nonidling qubits alike. With respect to depolarizing noise, which tends to affect only actively manipulated qubits, we observe no advantage of layering.

DOI: [10.1103/PhysRevA.109.042413](https://doi.org/10.1103/PhysRevA.109.042413)

I. INTRODUCTION

Quantum chemistry simulations of strongly correlated systems are challenging for classical computers [1]. While approximate methods often lack accuracy [1–5], exact methods become infeasible when the system sizes exceed more than 34 spin orbitals—the largest system for which a full configuration interaction (FCI) calculation has been conducted [5]. For this reason, simulations of many advanced chemical systems, such as enzyme active sites and surface catalysts, rely on knowledge-intense, domain-specific approximations [6]. Therefore, developing general chemistry simulation methods for quantum computers could prove valuable.

Variational quantum eigensolvers (VQEs) [1,7–19] are a class of quantum-classical methods intended to perform chemistry simulations on near-term quantum hardware [20,21]. More specifically, VQEs calculate upper bounds to the ground-state energy E_0 of a molecular Hamiltonian H using the Rayleigh-Ritz variational principle

$$E_0 \leq E(\vec{\theta}) \equiv \text{Tr}(H\Lambda(\vec{\theta})[\rho_0]). \quad (1)$$

A quantum processor is used to apply a parametrized quantum circuit to an initial state. In the presence of noise, the quantum circuit can, in general, be represented by the parametrized completely positive trace-preserving (CPTP) map $\Lambda(\vec{\theta})$, and

the initial state can be represented by the density matrix ρ_0 . We will use square brackets to enclose a state acted upon by a CPTP map. This generates a parametrized trial state $\rho(\vec{\theta}) \equiv \Lambda(\vec{\theta})[\rho_0]$ that is hard to represent on classical computers. The energy expectation value $E(\vec{\theta})$ of $\rho(\vec{\theta})$ gives a bound on E_0 , which can be accurately sampled using polynomially few measurements [1,8]. A classical computer then varies $\vec{\theta}$ to minimize $E(\vec{\theta})$ iteratively. Provided that the *Ansatz* circuit is sufficiently expressive, $E(\vec{\theta})$ converges to E_0 and returns the ground-state energy. Initial implementations of VQEs on near-term hardware have been reported in [7,22–26]. Despite these encouraging results, several refinements are needed to alleviate trainability issues [27–33] and to make VQEs feasible for molecular simulations with larger numbers of orbitals. Moreover, recent results indicate that the noise resilience of VQE algorithms must be improved to enable useful simulations [14,30,34].

Adaptive VQEs (ADAPT-VQEs) [10] are promising VQE algorithms, which partially address the issues of trainability and noise resilience. They operate by improving the *Ansatz* circuits in t_{\max} consecutive steps

$$\Lambda_t(\theta_1, \dots, \theta_t) = A_t(\theta_t) \circ \Lambda_{t-1}(\theta_{t-1}, \dots, \theta_1), \quad (2)$$

starting from the identity map $\Lambda_0 = \text{id}$. Here, $t = 1, \dots, t_{\max}$ indexes the step and \circ denotes functional composition of the CPTP maps. An *Ansatz* element $A_t(\theta_t)$ is added to the *Ansatz* circuit in each step. The *Ansatz* element $A_t(\theta_t)$ is chosen from an *Ansatz*-element pool \mathcal{P} by computing the energy gradient for each *Ansatz* element and picking the *Ansatz* element with the steepest gradient. Numerical evidence suggests

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

that such ADAPT-VQEs are readily trainable and can minimize the energy landscape [32]. In the original proposal of ADAPT-VQE, the *Ansatz*-element pool was physically motivated, comprising single and double fermionic excitations. Since then, different types of *Ansatz*-element pools have been proposed to minimize the number of CNOT gates in the *Ansatz* circuit and thus improve the noise resilience of ADAPT-VQE [11–13,35].

ADAPT-VQEs still face issues. Compared with other VQE algorithms, ADAPT-VQEs make more calls to quantum processors. This is because in every iteration, finding the *Ansatz* element with the steepest energy gradient requires at least $O(|\mathcal{P}|)$ quantum processor calls. This makes more efficient pool-exploration strategies desirable. Moreover, noise poses serious restrictions on the maximum depth of useful VQE *Ansatz* circuits [14]. This makes shallower *Ansatz* circuits desirable. A recent algorithm called TETRIS-ADAPT-VQE compresses VQE *Ansatz* circuits into compact layers of *Ansatz* elements [15]. This yields shallower *Ansatz* circuits. However, it has not yet been demonstrated that shallower *Ansatz* circuits lead to improved noise resilience. It is, therefore, important to evaluate whether such shallow *Ansatz* circuits boost the noise resilience of ADAPT-VQEs.

In this paper, we broaden the understanding of TETRIS-like layering algorithms. First, we show how noncommuting *Ansatz* elements can be used to define a topology on the *Ansatz*-element pool. Based on this topology, we present Subpool Exploration: a pool-exploration strategy to reduce the number of quantum-processor calls when searching for *Ansatz* elements with large energy gradients. We then investigate several flavors of algorithms to layer and shorten *Ansatz* circuits. Benchmarking these algorithms, we find that alternative layering strategies can yield equally shallow *Ansatz* circuits as TETRIS-ADAPT-VQE. Finally, we investigate whether shallow VQE circuits are more noise-resilient. We do this by benchmarking both standard and layered ADAPT-VQEs in the presence of noise. For amplitude damping and dephasing noise, which globally affect idling and nonidling qubits alike, we observe an increased noise resilience due to shallower *Ansatz* circuits. On the other hand, we find that layering is unable to mitigate depolarizing noise, which acts locally on actively manipulated qubits.

The remainder of this paper is structured as follows: In Sec. II, we introduce notation and the ADAPT-VQE algorithm. In Secs. III and IV, subpool exploration and layering for ADAPT-VQE are described and benchmarked, respectively. We study the runtime advantage of layering in Sec. V. In Sec. VI, we investigate the effect of noise on layered VQE algorithms. Finally, we conclude in Sec. VII.

II. PRELIMINARIES: NOTATION AND THE ADAPT-VQE

In what follows, we consider second-quantized Hamiltonians on a finite set of N spin orbitals:

$$H = \sum_{p,q=1}^N h_{pq} a_p^\dagger a_q + \sum_{p,q,r,s=1}^N h_{pqrs} a_p^\dagger a_q^\dagger a_r a_s. \quad (3)$$

a_p^\dagger and a_p denote fermionic creation and annihilation operators of the p th spin-orbital, respectively. The coefficients h_{pq} and

h_{pqrs} can be efficiently computed classically—we use the Psi4 package [36].

In this manuscript, we represent creation and annihilation operators using the Jordan-Wigner representation [1]. While other representations, such as Bravyi-Kitaev, should also work, we make this choice to maintain compatibility with previous work on ADAPT-VQE [10–13,17,18,32,35]. This results in creation and annihilation operators given by

$$a_p^\dagger \mapsto Q_p^\dagger Z_p \quad a_p \mapsto Q_p Z_p, \quad (4)$$

respectively. Here,

$$Q_p^\dagger := \frac{1}{2}(X_p - iY_p), \quad Q_p := \frac{1}{2}(X_p + iY_p) \quad (5)$$

are the qubit creation and annihilation operators, and X_p, Y_p, Z_p denote Pauli operators acting on qubit p . The fermionic phase is represented by

$$Z_p := \bigotimes_{q < p} Z_q. \quad (6)$$

Anti-Hermitian operators T generate *Ansatz elements* that form Stone's-encoded unitaries parametrized by one real parameter θ :

$$A(\theta)[\rho] := \exp(\theta T) \rho \exp(-\theta T). \quad (7)$$

Different ADAPT-VQE algorithms choose T from different types of operator pools. There are three common types of operator pools. The fermionic pool $\mathcal{P}^{\text{Fermi}}$ [10] contains fermionic single and double excitations generated by anti-Hermitian operators:

$$T_q^p := a_p^\dagger a_q - a_q^\dagger a_p, \quad (8)$$

$$T_{rs}^{pq} := a_p^\dagger a_q^\dagger a_r a_s - a_s^\dagger a_r^\dagger a_q a_p, \quad (9)$$

where $p, q, r, s = 1, \dots, N$. The qubit-excitation-based (QEB) pool \mathcal{P}^{QEB} [11] contains single and double qubit excitations generated by anti-Hermitian operators:

$$T_q^p := Q_p^\dagger Q_q - Q_q^\dagger Q_p, \quad (10)$$

$$T_{rs}^{pq} := Q_p^\dagger Q_q^\dagger Q_r Q_s - Q_s^\dagger Q_r^\dagger Q_q Q_p. \quad (11)$$

The qubit pool $\mathcal{P}^{\text{qubit}}$ [13] contains parametrized unitaries generated by strings of Pauli-operators $\sigma_p \in \{X_p, Y_p, Z_p\}$:

$$T_{pq} := i\sigma_p \sigma_q, \quad (12)$$

$$T_{pqrs} := i\sigma_p \sigma_q \sigma_r \sigma_s. \quad (13)$$

Further definitions and discussions of all three pools are given in Appendix E. It is worth noting that all *Ansatz elements* have quantum-circuit representations composed of multiple standard single- and two-qubit gates [35]. Since all pools consist of operators acting on up to four qubits, all pools contain $O(N^4)$ elements. Note that this scaling remains valid when restricting pools to spin-conserving *Ansatz elements*.

ADAPT-VQEs optimize several objective functions. At iteration step t , the energy landscape is defined by

$$E_t(\theta_1, \dots, \theta_1) \equiv \text{Tr}[H A_t(\theta_1) \circ \dots \circ A_1(\theta_1)[\rho_0]]. \quad (14)$$

A global optimizer may repeatedly evaluate E_t and its partial derivatives at the end of the t th iteration to return a set of

ALGORITHM 1. ADAPT-VQE [10].

```

1: Initialize state  $\rho_0 \leftarrow \rho_{\text{HF}}$ , circuit  $\Lambda_0 \leftarrow \text{id}$ , and pool  $\mathcal{P}$ .
2: Initialize energy bound  $\mathcal{E}_0 \leftarrow \infty$  and accuracy  $\varepsilon$ .
3: for  $t = 1, \dots, t_{\text{max}}$  do
4:   Select Ansatz element:  $A_t \leftarrow \arg \min_{A \in \mathcal{P}} L_t(A)$ 
5:   Set circuit:  $\Lambda_t(\theta_1, \dots, \theta_1) \leftarrow A_t(\theta_1) \circ \Lambda_{t-1}(\theta_{t-1}, \dots, \theta_1)$ 
6:   Optimize circuit:  $(\theta_t^*, \dots, \theta_t^*) \leftarrow \arg \min E_t(\theta_1, \dots, \theta_1)$ 
7:   Update energy bound:  $\mathcal{E}_t \leftarrow E_t(\theta_t^*, \dots, \theta_t^*)$ 
8:   Update state:  $\rho_t \leftarrow \Lambda_t(\theta_t^*, \dots, \theta_t^*)[\rho_0]$ 
9:   if  $\mathcal{E}_{t-1} - \mathcal{E}_t < \varepsilon$  then
10:    return energy bound  $\mathcal{E}_t$ 
11: return energy bound  $\mathcal{E}_{t_{\text{max}}}$ 
    
```

optimal parameters:

$$(\theta_t^*, \dots, \theta_1^*) = \arg \min_{(\theta_1, \dots, \theta_1) \in \mathbb{R}^r} E_t(\theta_1, \dots, \theta_1). \quad (15)$$

These parameters set the upper energy bound of the t th iteration:

$$\mathcal{E}_t = E_t(\theta_t^*, \dots, \theta_1^*). \quad (16)$$

A loss function $L_t : \mathcal{P} \rightarrow \mathbb{R}$ is used to pick an *Ansatz* element from the operator pool \mathcal{P} at each iteration t :

$$A_t = \arg \min_{A \in \mathcal{P}} L_t(A). \quad (17)$$

Throughout this paper, we use the standard gradient loss of ADAPT-VQEs, defined in Eq. (20). We denote the state after $t - 1$ iterations with optimized parameters $\theta_{t-1}^*, \dots, \theta_1^*$ by

$$\rho_{t-1} = \Lambda_{t-1}(\theta_{t-1}^*, \dots, \theta_1^*)[\rho_0]. \quad (18)$$

Further, we define the energy expectation after adding the *Ansatz* element $A \in \mathcal{P}$ as

$$E_{t,A}(\theta) = \text{Tr}(HA(\theta)[\rho_{t-1}]). \quad (19)$$

Then, the loss is defined by

$$L_t(A) = - \left| \frac{\partial E_{t,A}(\theta)}{\partial \theta} \right|_{\theta=0} = -|\text{Tr}([H, T]\rho_{t-1})|. \quad (20)$$

We consider alternative loss functions in Appendix C.

The ADAPT-VQE starts by initializing a state ρ_0 . Often, ρ_0 is the Hartree-Fock state ρ_{HF} . The algorithm then builds the *Ansatz* circuit Λ_t by first adding *Ansatz* elements $A_t \in \mathcal{P}$ of minimal loss L_t , according to Eq. (17). Then, the algorithm optimizes the *Ansatz* circuit parameters according to Eq. (15). This generates a series of upper bounds,

$$\mathcal{E}_0 > \mathcal{E}_1 > \dots > \mathcal{E}_t, \quad (21)$$

until the improvement of consecutive bounds drops below a threshold ε such that $\mathcal{E}_{t-1} - \mathcal{E}_t < \varepsilon$, or the maximum iteration number t_{max} is reached. The final bound (\mathcal{E}_t or $\mathcal{E}_{t_{\text{max}}}$) is then returned to approximate E_0 . A pseudocode of the ADAPT-VQE is given in Algorithm 1.

III. SUBPOOL EXPLORATION AND LAYERING FOR ADAPT-VQES

In this section, we present two subroutines to improve ADAPT-VQEs. The first subroutine optimally layers *Ansatz*

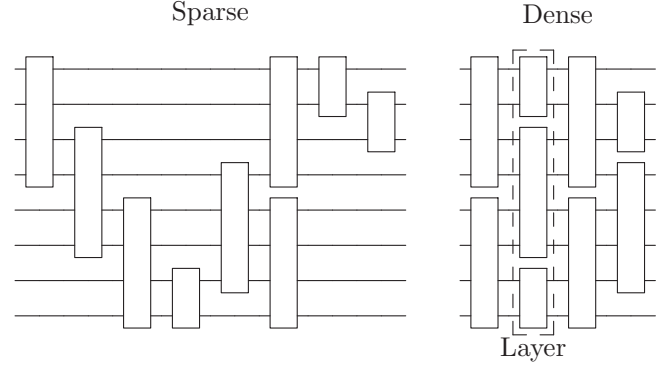


FIG. 1. Layering: A sparse *Ansatz* circuit (left), as produced by standard ADAPT-VQEs, can be compressed to a dense structure (right) by layering. Boxes denote *Ansatz* elements. Each line represents a single qubit. Note that *Ansatz* circuit elements entangle two or four qubits.

elements, as depicted in Fig. 1. We call the process of producing dense (right-hand side) *Ansatz* circuits instead of sparse (left-hand side) *Ansatz* circuits “layering.” This subroutine can be used to construct shallower *Ansatz* circuits, which may make ADAPT-VQEs more resilient to noise. The second subroutine is subpool exploration. It searches *Ansatz*-element pools in successions of noncommuting *Ansatz* elements. Subpool exploration is essential for layering and can reduce the number of calls an ADAPT-VQE makes to a quantum processor. Combining both subroutines results in algorithms similar to TETRIS-ADAPT-VQE [15]. (A detailed comparison of our algorithms to TETRIS-ADAPT-VQE will be given after the proof of Property 4).

A. Commutativity and support

Commutativity of *Ansatz* elements is a central notion underlying our subroutines. Informally, our definitions will be as follows. Two *Ansatz* elements operator-commute if they commute for all parameter values. Two *Ansatz* elements support-commute if they do not act on the same qubits. Generalized commutativity will refer to operator or support commutativity, or the extended definition in Appendix I. The generalized noncommuting set of an element A is the set of *Ansatz* elements that do not commute in the generalized sense with A . More formally:

Definition 1 (Operator commutativity). Two *Ansatz* elements $A, B \in \mathcal{P}$ are said to “operator-commute” iff $A(\theta)$ and $B(\phi)$ commute for all θ and ϕ :

$$\{A, B\}_0 = 0 \iff \forall \theta, \phi \in \mathbb{R} : [A(\theta), B(\phi)] = 0. \quad (22)$$

Conversely, two *Ansatz* elements $A, B \in \mathcal{P}$ do not operator-commute iff there exist parameters for which the corresponding operators do not commute:

$$\{A, B\}_0 \neq 0 \iff \exists \theta, \phi \in \mathbb{R} : [A(\theta), B(\phi)] \neq 0. \quad (23)$$

Definition 2 (Operator noncommuting set). Given an *Ansatz*-element pool \mathcal{P} and an *Ansatz* element $A \in \mathcal{P}$, we define its operator noncommuting set as follows:

$$\mathcal{N}_0(\mathcal{P}, A) := \{B \in \mathcal{P} : \{A, B\}_0 \neq 0\}. \quad (24)$$

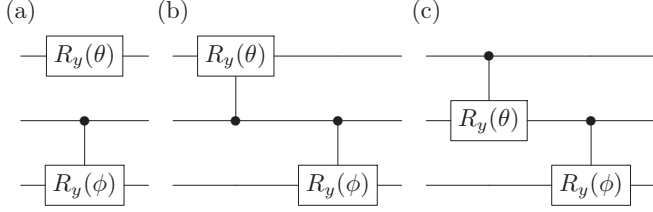


FIG. 2. A diagram comparing support and operator commutativity. (a) The elements both support- and operator-commute. Note that $R_y(\theta)$ only has qubit support on the top qubit line. (b) The elements operator-commute but do not support-commute. (c) The elements neither support- nor operator-commute.

Operator commutativity is central to layering. The operator noncommuting set is central to subpool exploration. Structurally similar and more intuitive notions can be defined using qubit support:

Definition 3 (Qubit support). Let $\mathcal{B}(\mathcal{H})$ denote the set of superoperators on a Hilbert space \mathcal{H} . Let $\mathcal{H}_{\mathcal{Q}} := \bigotimes_{q \in \mathcal{Q}} \mathcal{H}_q$ denote the Hilbert space of a set of all qubits $\mathcal{Q} \equiv \{1, \dots, N\}$ where \mathcal{H}_q is the Hilbert space corresponding to the q th qubit. Consider a superoperator $A \in \mathcal{B}(\mathcal{H}_{\mathcal{Q}})$. First, we define the superoperator subset that acts on a qubit subset $\mathcal{W} \subseteq \mathcal{Q}$ as

$$\mathcal{B}_{\mathcal{W}} := \{B \otimes \text{id} : B \in \mathcal{B}(\mathcal{H}_{\mathcal{W}})\} \subseteq \mathcal{B}(\mathcal{H}_{\mathcal{Q}}). \quad (25)$$

Then, we define the qubit support of a superoperator A as its minimal qubit subset \mathcal{W} :

$$\text{Supp}(A) := \bigcap_{\mathcal{W} \subseteq \mathcal{Q} : A \in \mathcal{B}_{\mathcal{W}}} \mathcal{W}. \quad (26)$$

The notion of support extends to parametrized *Ansatz* elements:

$$\text{Supp}(B) := \bigcup_{\theta} \text{Supp}(B(\theta)), \quad (27)$$

where B is a parametrized *Ansatz* element.

Intuitively, the qubit support of an *Ansatz* element A is the set of all qubits the operator A acts on nontrivially (see Fig. 2). The concept of qubit support allows one to define support commutativity of *Ansatz* elements as follows:

Definition 4 (Support commutativity). Two *Ansatz* elements $A, B \in \mathcal{P}$ are said to “support-commute” iff their qubit support is disjoint,

$$\{A, B\}_S = 0 \iff \text{Supp}(A) \cap \text{Supp}(B) = \emptyset. \quad (28)$$

Conversely, two *Ansatz* elements $A, B \in \mathcal{P}$ do not support-commute iff their supports overlap,

$$\{A, B\}_S \neq 0 \iff \text{Supp}(A) \cap \text{Supp}(B) \neq \emptyset. \quad (29)$$

Definition 5 (Support noncommuting set). Given an *Ansatz*-element pool \mathcal{P} and an *Ansatz* element $A \in \mathcal{P}$, we define the set of *Ansatz* elements with overlapping support as

$$\mathcal{N}_S(\mathcal{P}, A) := \{B \in \mathcal{P} : \{B, A\}_S \neq 0\}. \quad (30)$$

Operator commutativity and support commutativity are not equivalent—see Fig. 2. However, the following properties hold. Elements supported on disjoint qubit sets

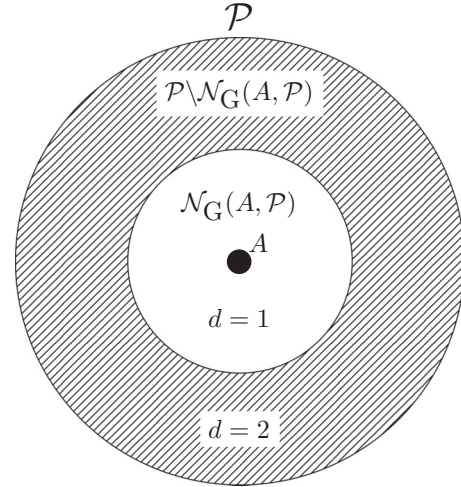


FIG. 3. A diagram of the distance d from an element $A \in \mathcal{P}$ under the pool metric, Definition 6. The *Ansatz* element A (black dot) is surrounded by *Ansatz* elements of the noncommuting set $\mathcal{N}_G(\mathcal{P}, A)$ of distance 1 (white circle). All other elements $\mathcal{P} \setminus \mathcal{N}_G(\mathcal{P}, A)$ have distance 2 (gray shaded region).

operator-commute:

$$\forall A, B \in \mathcal{P} \quad \{A, B\}_S = 0 \Rightarrow \{A, B\}_O = 0. \quad (31)$$

Conversely, operator-noncommuting *Ansatz* elements act on at least one common qubit, which implies they are support-noncommuting:

$$\forall A, B \in \mathcal{P} \quad \{A, B\}_O \neq 0 \Rightarrow \{A, B\}_S \neq 0. \quad (32)$$

The last relation also implies that the operator-noncommuting set of A is contained in its support-noncommuting set,

$$\mathcal{N}_O(\mathcal{P}, A) \subseteq \mathcal{N}_S(\mathcal{P}, A). \quad (33)$$

We further generalize the notions of operator and support commutativity in Appendix I. Henceforth, we will use generalized commutativity to denote either operator or support commutativity or any other type of commutativity specified in Appendix I. Further, \mathcal{N}_G and $\{\bullet, \bullet\}_G$ will be used to denote the generalized noncommuting set and the generalized commutator, respectively.

For later reference, we note that generalized noncommuting sets induce a topology on \mathcal{P} via the following discrete metric.

Definition 6 (Pool metric). Let $d : \mathcal{P} \times \mathcal{P} \rightarrow \{0, 1, 2\}$ define a discrete metric such that (i) $\forall A \in \mathcal{P}$, set $d(A, A) = 0$. (ii) $\forall A, B \in \mathcal{P}$ with $A \neq B$ and $\{A, B\}_G \neq 0$, set $d(A, B) = 1$. (iii) $\forall A, B \in \mathcal{P}$ with $A \neq B$ and $\{A, B\}_G = 0$, set $d(A, B) = 2$.

With this metric, the generalized noncommuting elements $\mathcal{N}_G(\mathcal{P}, A)$ form a ball of distance one around each *Ansatz* element $A \in \mathcal{P}$. The metric is represented diagrammatically in Fig. 3. This allows us to identify an element $A \in \mathcal{P}$ as a local minimum if there is no element with lower loss within A ’s generalized noncommuting set.

Property 1 (Local minimum). Let \mathcal{P} be an *Ansatz*-element pool with the pool metric of Definition 6, and let $L : \mathcal{P} \rightarrow \mathbb{R}$

denote a loss function. Then, any element $A \in \mathcal{P}$ for which

$$L(A) \leq \min_{B \in \mathcal{N}_G(\mathcal{P}, A)} L(B) \quad (34)$$

is a local minimum in \mathcal{P} with respect to L .

This property is important as we will later show that subpool exploration always returns local minima.

To gain intuition about the previously defined notions, we consider the *Ansatz* elements of the QEB and the Pauli pools, Eqs. (10)–(13). The *Ansatz* elements of these pools have qubit support on either two or four qubits, as is illustrated in Fig. 1. Commuting *Ansatz* elements with disjoint support can be packed into an *Ansatz*-element layer, which can be executed on the quantum processor in parallel. This is the core idea of layering, which helps to reduce the depths of *Ansatz* circuits. Moreover, as generalized noncommuting *Ansatz* elements must share at least one qubit, we conclude that the generalized noncommuting set $\mathcal{N}_G(\mathcal{P}, A)$ has at most $O(N^3)$ *Ansatz* elements. This is a core component of subpool exploration. Analytic expressions for the cardinalities of the generalized noncommuting sets are given in Appendix G. In Appendix F, we prove that two different fermionic excitations operator-commute iff they act on disjoint or equivalent sets of orbitals. The same is true for qubit excitations. Pauli excitations operator-commute iff the generating Pauli strings differ in an even number of places within their mutual support.

B. Subpool exploration

In this section, we introduce *subpool exploration*, a strategy to explore *Ansatz*-element pools with fewer quantum-processor calls. Subpool exploration differs from the standard ADAPT-VQE as follows. Standard ADAPT-VQEs evaluate the loss of every *Ansatz* element in the *Ansatz*-element pool \mathcal{P} in every iteration of ADAPT-VQE (Algorithm 1, line 4). This leads to $O(|\mathcal{P}|)$ quantum-processor calls to identify the *Ansatz* element of minimal loss. Instead, subpool exploration evaluates the loss of a reduced number of *Ansatz* elements by exploring a sequence of generalized noncommuting *Ansatz*-element subpools. This can lead to a reduced number of quantum-processor calls and returns an *Ansatz* element which is a local minimum of the pool \mathcal{P} . The details of subpool exploration are as follows.

Algorithm: Let \mathcal{P} denote a given pool and L a given loss function. Instead of naively computing the loss of every *Ansatz* element in \mathcal{P} , our algorithm explores \mathcal{P} iteratively by considering subpools, $\mathcal{S}_m \subsetneq \mathcal{P}$, in consecutive steps. During this process, the algorithm successively determines the *Ansatz* element with minimal loss within subpool \mathcal{S}_m as

$$A_m = \arg \min_{A \in \mathcal{S}_m} L(A). \quad (35)$$

Meanwhile, the corresponding loss value is stored:

$$\mathcal{L}_{m+1} = L(A_m). \quad (36)$$

Iterations are halted when loss values stop decreasing. The key point of subpool exploration is to update the subpools \mathcal{S}_m using the generalized noncommuting set generated by A_m :

$$\mathcal{S}_{m+1} = \mathcal{N}_G(\mathcal{P} \setminus \mathcal{S}_{\leq m}, A_m) \subseteq \mathcal{N}_G(\mathcal{P}, A_m), \quad \forall m \geq 0, \quad (37)$$

ALGORITHM 2. Subpool exploration.

```

1: Input: Pool  $\mathcal{P}$ , initial subpool  $\mathcal{S}_0$ , and loss function  $L$ .
2: Initialize loss value  $\mathcal{L}_0 \leftarrow \infty$ .
3: for  $m = 0, \dots$  do
4:   Select Ansatz element  $A_m \leftarrow \arg \min_{A \in \mathcal{S}_m} L(A)$ .
5:   Update loss value  $\mathcal{L}_{m+1} \leftarrow L(A_m)$ .
6:   if  $\mathcal{L}_{m+1} < \mathcal{L}_m$  then
7:     Update subpool  $\mathcal{S}_{m+1} = \mathcal{N}_G(\mathcal{P} \setminus \mathcal{S}_{\leq m}, A_m)$ .
8:   else
9:     return  $A_m, \mathcal{S}_m$ .
```

where $\mathcal{S}_{\leq m} := \cup_{l=0}^m \mathcal{S}_l$. A pseudocode summary of subpool exploration is given in Algorithm 2, a visual summary is displayed in Fig. 4, and a toy example on four qubits is included in Fig. 5. We now discuss aspects of subpool exploration.

Efficiency: Let m_s denote the index of the final iteration, and define the set of searched *Ansatz* elements as

$$\mathcal{S} := \cup_{m=0}^{m_s} \mathcal{S}_m. \quad (38)$$

As loss values of *Ansatz* elements that have been explored are stored in a list, it follows that subpool exploration requires only $|\mathcal{S}|$ loss function calls. On the other hand, exploring the whole pool in ADAPT-VQE requires $|\mathcal{P}|$ loss-function calls. Since \mathcal{S} is a subset of \mathcal{P} , subpool exploration may reduce the number of quantum-processor calls:

$$\mathcal{S} \subseteq \mathcal{P} \Rightarrow |\mathcal{S}| \leq |\mathcal{P}|. \quad (39)$$

To give a specific example, consider the QEB and qubit pools. Those pools contain $O(N^4)$ *Ansatz* elements. On the other hand, generalized noncommuting sets have $O(N^3)$ *Ansatz* elements. Thus, by choosing an appropriate initial subpool, we can ensure that $|\mathcal{S}_m| = O(N^3)$ for all subpools. Especially if the number of searched subpools is $m_s = O(1)$, subpool exploration can return *Ansatz* elements of low loss while exploring only $O(N^3)$ *Ansatz* elements.

We note that this pool-exploration strategy ignores certain *Ansatz* elements. In particular, it may miss the optimal *Ansatz* element with minimal loss. Nevertheless, as explained in the following paragraphs, it will always return *Ansatz* elements which are *locally optimal*. This ensures that the globally optimal *Ansatz* element can always be added to the *Ansatz* circuit later in the algorithm.

Optimality: As the set of explored *Ansatz* elements \mathcal{S} is a subset of \mathcal{P} , the *Ansatz* element returned by subpool exploration,

$$A_{\mathcal{S}}^* := \arg \min_{A \in \mathcal{S}} L(A), \quad (40)$$

may be suboptimal to the *Ansatz* element returned by exploring the whole pool,

$$A_{\mathcal{P}}^* := \arg \min_{A \in \mathcal{P}} L(A). \quad (41)$$

That is,

$$\mathcal{S} \subset \mathcal{P} \Rightarrow \arg \min_{A \in \mathcal{P}} L(A) \leq \arg \min_{A \in \mathcal{S}} L(A). \quad (42)$$

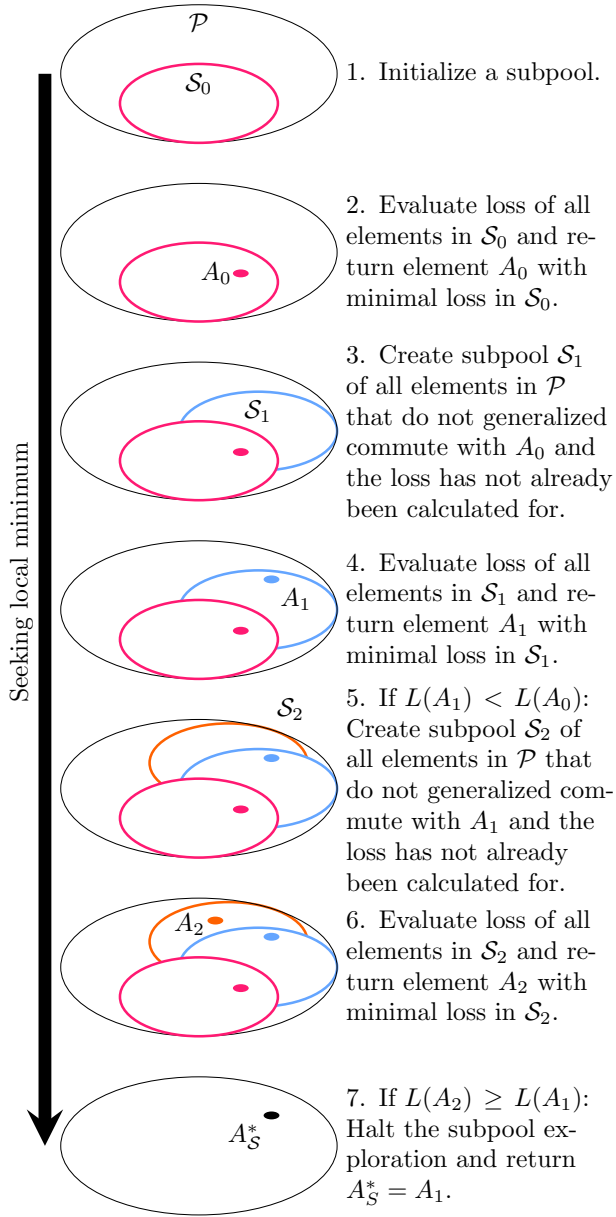


FIG. 4. Visualization of our strategy for subpool exploration. The pool \mathcal{P} is successively explored in subpools \mathcal{S}_m , with *Ansatz* elements A_m of minimal loss generating future subpools through their generalized noncommuting set.

Yet, there are a couple of useful properties that pertain to the output A_S^* of subpool exploration. At first, the outputs of subpool exploration are local minima.

Property 2 (Local optimality). Any *Ansatz* element A_S^* returned by subpool exploration is a local minimum.

The proof of this property is immediate. Moreover, as subpool exploration constructs subpools from generalized noncommuting sets, the only *Ansatz* elements $B \in \mathcal{P}$ with $L(B) < L(A_S^*)$ must necessarily generalized commute with $A_S^* \in \mathcal{P}$.

Property 3 (Better *Ansatz* elements generalized commute). Let \mathcal{P} denote a pool and L denote a loss function. Let $A_S^* \in \mathcal{P}$

denote the final output of subpool exploration. Then,

$$\forall B \in \mathcal{P} \text{ with } L(B) < L(A_S^*) \Rightarrow \{A_S^*, B\}_G = 0 \quad (43)$$

$$\Rightarrow \{A_S^*, B\}_O = 0. \quad (44)$$

Proof. We prove this property by contradiction. Assume that there is an *Ansatz* element $L(B) < L(A_S^*)$ such that $\{A_S^*, B\}_G \neq 0$. This implies that B is in the generalized noncommuting set $\mathcal{N}_G(\mathcal{P}, A_S^*)$ and exploring the corresponding subpool would have produced $L(B) < L(A_S^*)$ leading to the exploration of $\mathcal{N}_G(\mathcal{P}, B)$. This, in turn, can only return an *Ansatz* element with a loss $L(B)$ or smaller. This would contradict A_S^* having been the final output of the algorithm. Finally, we use Eq. (31) to show Eq. (44). ■

Property 3 is useful as it ensures that subpool exploration can find better *Ansatz* elements, which first were missed, in subsequent iterations. To see this, suppose a first run of subpool exploration returns a local minimum $A \in \mathcal{P}$. Further, suppose there is another local minimum $B \in \mathcal{P}$ such that $L(B) < L(A)$. Property 3 ensures that A and B generalized commute. Hence, by running subpool exploration repeatedly on the remaining pool, we are certain to discover the better local minimum eventually. Ultimately, this will allow for restoring the global minimum.

Initial subpool: So far, we have not specified any strategy for choosing the initial set \mathcal{S}_0 . This can be done, for example, by taking the subpool of a single random *Ansatz* element $A_0 \in \mathcal{P}$. Alternatively, one can compose \mathcal{S}_0 of random *Ansatz* elements enforcing an appropriate pool size, e.g., $|\mathcal{S}_0| = O(N^3)$ for QEB and qubit pools.

We will refer to the ADAPT-VQE with subpool exploration as the Explore-ADAPT-VQE. This algorithm is realized by replacing line 4 in Algorithm 1 with subpool exploration, Algorithm 2, with $L \rightarrow L_t$.

C. Layering

Below, we describe two methods for arranging generalized noncommuting *Ansatz* elements into *Ansatz*-element layers. Figure 6 diagrammatically represents both methods.

Definition 7 (Ansatz-element layer). Let \mathcal{A} be a subset of \mathcal{P} . We say that \mathcal{A} is an *Ansatz*-element layer iff

$$\{A, B\}_G = 0 \quad \forall A, B \in \mathcal{A} \text{ such that } A \neq B. \quad (45)$$

We denote the operator corresponding to the action of the mutually generalized-commuting *Ansatz* elements of an *Ansatz*-element layer \mathcal{A} with

$$\mathcal{A}^o(\vec{\theta}) = \prod_{A \in \mathcal{A}} A(\theta_A). \quad (46)$$

Here, $\vec{\theta}$ is the parameter vector for the layer:

$$\vec{\theta} \equiv \{\theta_A : A \in \mathcal{A}\}. \quad (47)$$

We note that for support commutativity, the product can be replaced by the tensor product.

Since *Ansatz*-element layers depend on parameter vectors, the update rule is

$$\Lambda_t(\vec{\vartheta}_t) = \mathcal{A}^o(\vec{\theta}_t) \circ \Lambda_{t-1}(\vec{\vartheta}_{t-1}), \quad \vec{\vartheta}_t = (\vec{\theta}_t, \vec{\vartheta}_{t-1}). \quad (48)$$

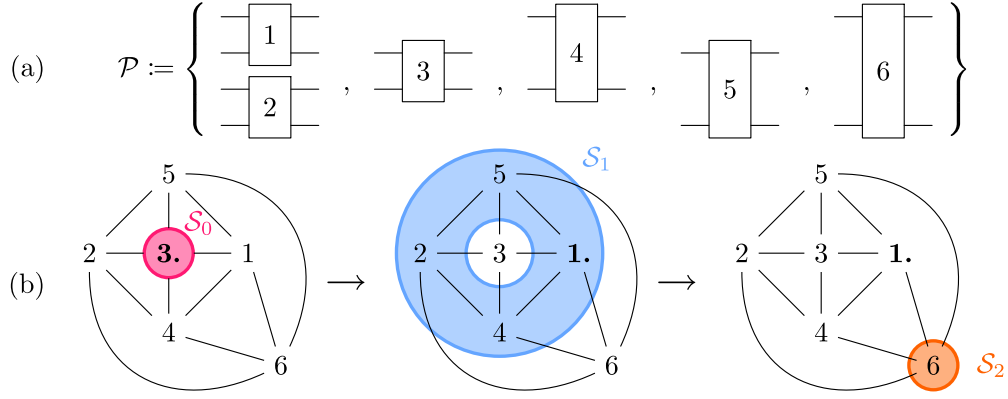


FIG. 5. A toy example of subpool exploration. (a) The Ansatz element pool \mathcal{P} comprises of six two-qubit operators acting on four qubits. Each Ansatz element is labeled 1, ..., 6. (b) Graphs visualize Ansatz elements as vertices. A vertex pair is connected by an edge if the corresponding Ansatz elements do not support-commute. The dynamics of subpool exploration are visualized as follows. Left: The initial pool is $\mathcal{S}_0 := \{3\}$, and $A_0 := 3$ is found to be the Ansatz element of minimal loss. Center: Next, we search the noncommuting set of $A_0 := 3$ given by $\mathcal{S}_1 := \{1, 2, 4, 5\}$. Within \mathcal{S}_1 we determine the loss of all elements. Assume the element of minimal loss is found to be element $A_1 := 1$. Right: Next, we search the noncommuting set of $A_1 := 1$ given by $\{3, 4, 5, 6\}$. We have already evaluated the losses of $\{3, 4, 5\}$, so we set the subpool to $\mathcal{S}_2 := \{6\}$ and determine its minimum loss. Assuming the loss of $A_2 = 6$ is larger than the loss of $A_1 = 1$, we return $A_S^* = 1$ as the local minimum. In this toy example, the whole pool is searched, and the local minimum is the global minimum.

As before, the algorithm is initialized with $\Lambda_0 = \text{id}$ and $\vec{\vartheta} = ()$. To make the dependence on the Ansatz circuit explicit, we denote the energy landscape as

$$E_\Lambda(\vec{\vartheta}) := \text{tr}[H \Lambda(\vec{\vartheta})[\rho_0]]. \quad (49)$$

The energy landscape of the t th iteration is denoted as

$$E_t(\vec{\vartheta}_t) \equiv E_{\Lambda_t}(\vec{\vartheta}_t), \quad (50)$$

and its optimal parameters are

$$\vec{\vartheta}_t^* = \arg \min_{\vec{\vartheta}} E_t(\vec{\vartheta}). \quad (51)$$

Further, the gradient loss [cf. Eq. (20)] is

$$L_{\Lambda(\vec{\vartheta})}(A) \equiv -|\text{tr}[[H, T_A] \Lambda(\vec{\vartheta})[\rho_0]]|, \quad (52)$$

where the definitions in Eqs. (20) and (52) satisfy the following relation:

$$L_t(A) = L_{\Lambda_{t-1}(\vec{\vartheta}_{t-1}^*)}(A). \quad (53)$$

With this notation in place, we proceed to describe two methods to construct Ansatz-element layers.

1. Static layering

Our algorithm starts by initializing an empty Ansatz-element layer and the remaining pool \mathcal{P}' to be the entire pool \mathcal{P} . Further, the loss is set such that $L \leftarrow L_{\Lambda_{t-1}}$ for the t th layer. The algorithm proceeds to fill the Ansatz-element layer by successively running subpool exploration to pick an Ansatz element A_n in $n = 0, \dots, n_{\max}$ iterations. This naturally induces an ordering on the layer. At every step of the iteration, the corresponding generalized noncommuting set $\mathcal{N}_G(\mathcal{P}', A_n)$ is removed from the remaining pool \mathcal{P}' . If the loss of the selected Ansatz element A_n is smaller than a predefined threshold $L(A) < \ell$, it is added to the Ansatz-element layer \mathcal{A} . The layer is completed once the pool is exhausted ($\mathcal{P}' = \emptyset$) or the maximal iteration count n_{\max} is reached. A pseudocode summary of static layering is given in Algorithm 3.

In Static-ADAPT-VQE, static layering is used to grow an Ansatz circuit iteratively. In each iteration, the layer is appended to the Ansatz circuit, and the Ansatz-circuit parameters are reoptimized. Iterations halt once the decrease in energy falls below ε , the energy accuracy per Ansatz element. A summary of Static-ADAPT-VQE is given in Algorithm 4.

We establish the close relationship between static layering and TETRIS-ADAPT-VQE in the following property.

Property 4 Assume that all Ansatz elements $A, B \in \mathcal{P}$ have distinct loss $L(A) \neq L(B)$. Using support commutativity and provided that $\ell = 0$ and n_{\max} are sufficiently large to

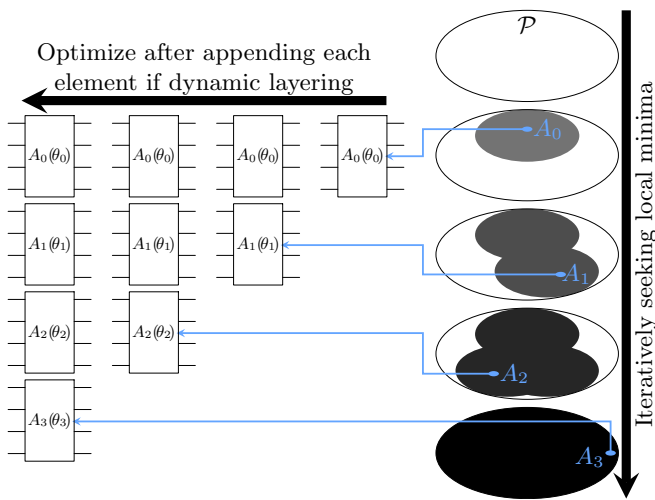


FIG. 6. Visualization of layer construction and successive pool reduction. Gray areas indicate the removal of generalized noncommuting sets corresponding to Ansatz elements A_n added to the layer \mathcal{A} . Parameters can either be optimized once the whole layer is fixed (static layering) or after adding each Ansatz element (dynamic layering).

ALGORITHM 4. Static-ADAPT-VQE.

```

1: Initialize state  $\rho_0 \leftarrow \rho_{\text{HF}}$ , Ansatz circuit  $\Lambda_0 \leftarrow \text{id}$ , pool  $\mathcal{P}$ .
2: Initialize energy bound  $\mathcal{E}_0 \leftarrow \infty$  and accuracy  $\varepsilon$ .
3: Initialize maximal loss  $\ell$  and iteration count  $n_{\text{max}}$ .
4: for  $t = 1, \dots, t_{\text{max}}$  do
5:   Get layer:  $\mathcal{A}_t \leftarrow \text{BuildStaticLayer}(\mathcal{P}, L_{\Lambda_{t-1}}, \ell, n_{\text{max}})$ 
6:   Set Ansatz circuit:  $\Lambda_t(\vec{\vartheta}_t) \leftarrow \mathcal{A}_t^\circ(\vec{\theta}_t) \circ \Lambda_{t-1}(\vec{\vartheta}_{t-1})$ 
7:   Optimize Ansatz circuit:  $\vec{\vartheta}_t^* \leftarrow \arg \min E_t(\vec{\vartheta}_t)$ 
8:   Set Ansatz circuit:  $\Lambda_t \leftarrow \Lambda_t(\vec{\vartheta}_t^*)$ 
9:   Update energy bound:  $\mathcal{E}_t \leftarrow E_t(\theta_t^*, \dots, \theta_1^*)$ 
10:  if  $\mathcal{E}_{t-1} - \mathcal{E}_t < \varepsilon |\mathcal{A}_t|$  then
11:    return energy bound  $\mathcal{E}_t$ 
12: return energy bound  $\mathcal{E}_{t_{\text{max}}}$ 

```

ensure that the whole layer is filled, Static-ADAPT-VQE and TETRIS-ADAPT-VQE will produce identical *Ansatz*-element layers.

Proof. This property is proven by induction. Assume that the previous iterations of ADAPT-VQE have yielded a specific *Ansatz* circuit $\Lambda_{t-1}(\vec{\vartheta}^*)$. The next layer of *Ansatz* elements \mathcal{A}_t can be constructed either by Static-ADAPT-VQE or TETRIS-ADAPT-VQE. For both algorithms, the equivalence of $\Lambda_{t-1}(\vec{\vartheta}^*)$ implies that the loss function, Eq. (53), of any *Ansatz* element is identical throughout the construction of the layer \mathcal{A}_t . First, by setting the maximal loss $\ell = 0$, we ensure that Static-ADAPT-VQE only accepts *Ansatz* elements with a nonzero gradient. For TETRIS-ADAPT-VQE, this is always the case by design. Next, we note that if an *Ansatz* element is placed on a qubit by Static-ADAPT-VQE, then by Property 3 there exists no *Ansatz* element that acts actively on this qubit and generates a lower loss. Moreover, there exists no *Ansatz* element with identical loss that acts nontrivially on this qubit, as we assume that all *Ansatz* elements have a distinct loss. Similarly, TETRIS-ADAPT-VQE places *Ansatz* elements from lowest to highest loss and ensures no two *Ansatz* elements have mutual support. Thus, if an *Ansatz* element is placed on a qubit by TETRIS-ADAPT-VQE, there exists no *Ansatz* element with a lower loss that acts nontrivially on this qubit. Again, there also exists no *Ansatz* element with identical loss supported by this qubit, as we assume that all *Ansatz* elements have a distinct loss. Combining these arguments, both Static- and TETRIS-ADAPT-VQE will fill the *Ansatz*-element layer \mathcal{A}_t with equivalent *Ansatz* elements. The *Ansatz* elements may be chosen in a different order. By

ALGORITHM 3. Build static layer.

```

1: Input: Pool  $\mathcal{P}$ , loss  $L$ , max. loss  $\ell$ , max. iteration  $n_{\text{max}}$ 
2: Initialize remaining pool  $\mathcal{P}' \leftarrow \mathcal{P}$ 
3: Initialize Ansatz layer  $\mathcal{A} \leftarrow \emptyset$ 
4: for  $n = 0, \dots, n_{\text{max}}$  do
5:   Set  $A \leftarrow \text{SubpoolExploration}(\mathcal{P}', L)$ 
6:   if  $L(A) < \ell$  then
7:     Update layer  $\mathcal{A} \leftarrow \mathcal{A} \cup \{A\}$ .
8:   Reduce pool  $\mathcal{P}' \leftarrow \mathcal{P}' \setminus \mathcal{N}_G(\mathcal{P}', A)$ 
9:   if  $\mathcal{P}' = \emptyset$  then break
10: return  $\mathcal{A}$ 

```

induction, the equivalence of Λ_{t-1} and \mathcal{A}_t implies the equivalence of the *Ansatz* circuit Λ_t . ■

Static versus TETRIS: We now compare Static- and TETRIS-ADAPT-VQE. First, provided Static-ADAPT-VQE uses support commutativity, completes full *Ansatz*-element layer (by setting $n_{\text{max}} = N$), and uses a maximal loss of $\ell = 0$, Property 4 implies that Static-ADAPT-VQE and TETRIS-ADAPT-VQE will produce identical *Ansatz* circuits layer by layer. Second, it is worth noting that TETRIS-ADAPT-VQE will construct each *Ansatz*-element layer from the *Ansatz* element with the maximal to the lowest loss. On the other hand, Static-ADAPT-VQE will construct each *Ansatz*-element layer in random order. Thus, Static- and TETRIS-ADAPT-VQE have identical *Ansatz* circuits only if an *Ansatz*-element layer is completed. Finally, if Static-ADAPT-VQE does not use support commutativity, or is not allowed to complete a full layer (by setting $n_{\text{max}} < N$), or does not operate at a maximally allowed loss $\ell < 0$, there is no reason for TETRIS- and Static-ADAPT-VQE to produce equivalent *Ansatz* circuits.

2. Dynamic layering

In static layering, *Ansatz*-circuit parameters are optimized after appending a whole layer with several *Ansatz* elements. In dynamic layering, on the other hand, *Ansatz*-circuit parameters are reoptimized every time an *Ansatz* element is appended to a layer. The motivation for doing so is to simplify the optimization process. The price is having to run the global optimization more times. We now describe how to perform dynamic layering.

The starting point is a given *Ansatz* circuit Λ , a set of optimal parameters $\vec{\vartheta}^*$ [Eq. (51)] and their corresponding energy bound $\mathcal{E} \equiv E_\Lambda(\vec{\vartheta}^*)$. The remaining pool \mathcal{P}' is initiated to be the entire pool \mathcal{P} . Starting from an empty layer \mathcal{A}' and a temporary *Ansatz* circuit $\Lambda' = \Lambda$, a layer is constructed dynamically by iteratively adding *Ansatz* elements A to \mathcal{A}' and Λ' while simultaneously reoptimizing the *Ansatz*-circuit parameters $\vec{\vartheta}^*$. Based on the loss L' induced by the currently optimal *Ansatz* circuit $\Lambda'(\vec{\vartheta}^*)$, subpool exploration is used to select *Ansatz* elements A . Simultaneously, the pool of remaining *Ansatz* elements \mathcal{P}' is shrunk by the successive removal of the generalized noncommuting sets $\mathcal{N}_G(\mathcal{P}', A)$. Finally, *Ansatz* elements are only added to the layer \mathcal{A} if their loss is below a threshold ℓ and the updated energy bound \mathcal{E}' exceeds a gain threshold of ε . A pseudocode summary is given in Algorithm 5.

Dynamic-ADAPT-VQE iteratively builds dynamic layers \mathcal{A}_t and appends those to the *Ansatz* circuit Λ_{t-1} . The procedure is repeated until an empty layer is returned; that is, no *Ansatz* element is found that reduces the energy by more than ε . Alternatively, the algorithm halts when the (user-specified) maximal iteration count t_{max} is reached. A pseudocode summary is given in Algorithm 6.

IV. BENCHMARKING NOISELESS PERFORMANCE

In this section, we benchmark various aspects of subpool exploration and layering in noiseless settings. To this end, we use numerical state-vector simulations to study a wide variety of molecules summarized in Table I. While BeH_2 and H_2O

ALGORITHM 5. Build dynamic layer.

```

1: Input: Ansatz  $\Lambda$ , bound  $\mathcal{E}$ , opt. params  $\vec{\vartheta}^*$ 
2: Get: Pool  $\mathcal{P}$ , accuracy  $\varepsilon$ , max. loss  $\ell$ ,  $n_{\max}$ 
3: Initialize pool  $\mathcal{P}' \leftarrow \mathcal{P}$ , layer  $\mathcal{A}' \leftarrow \emptyset$ , Ansatz circuit  $\Lambda' \leftarrow \Lambda$ 
4: for  $n = 0, \dots, n_{\max}$  do
5:   Update loss  $L' \leftarrow L_{\Lambda'(\vec{\vartheta}^*)}$ 
6:   Set  $A \leftarrow \text{SubpoolExploration}(\mathcal{P}', L')$ 
7:   if  $L'(A) < \ell$  then
8:     Minimize  $(\theta^*, \vec{\vartheta}^*) \leftarrow \arg \min_{(\theta, \vec{\vartheta})} E_{A \circ \Lambda'}(\theta, \vec{\vartheta})$ 
9:     Set bound  $\mathcal{E}' \leftarrow E_{A \circ \Lambda'}(\theta^*, \vec{\vartheta}^*)$ 
10:    if  $\mathcal{E} - \mathcal{E}' \geq \varepsilon$  then
11:      Update layer  $\mathcal{A} \leftarrow \mathcal{A} \cup \{A\}$ 
12:      Update Ansatz circuit  $\Lambda' \leftarrow A \circ \Lambda'$ 
13:      Update opt. params  $\vec{\vartheta}^* \leftarrow (\theta^*, \vec{\vartheta}^*)$ 
14:      Reduce pool  $\mathcal{P}' \leftarrow \mathcal{P}' \setminus \mathcal{N}_G(\mathcal{P}', A)$ 
15:    if  $\mathcal{P}' = \emptyset$  then break
16: return Layer  $\mathcal{A}'$ , optimal params  $\vec{\vartheta}^*$ , bound  $\mathcal{E}$ .

```

are among the larger molecules to be benchmarked, H_4 and H_6 are prototypical examples of strongly correlated systems [10,13,32]. Our simulations demonstrate the utility of subpool exploration in reducing quantum-processor calls. Further, we show that when compared to standard ADAPT-VQE, both Static- and Dynamic-ADAPT-VQE reduce the Ansatz circuit depths to similar extents. All simulations use the QEB pool because it gives a higher resilience to noise than the fermionic pool and performs similarly to the qubit pool [14]. Moreover, unless stated otherwise, we use support commutativity to ensure that Static-ADAPT-VQE produces Ansatz circuits equivalent to TETRIS-ADAPT-VQE.

A. Efficiency of subpool exploration

We begin by illustrating the ability of subpool exploration to reduce the number of loss function calls when searching for a suitable Ansatz element A to append to an Ansatz circuit. To this end, we present Explore-ADAPT-VQE (ADAPT-VQE with subpool exploration) using the QEB pool, Eq. (10), and operator commutativity. We set the initial subpool, \mathcal{S}_0 , such that it consists of a single Ansatz element selected uniformly at random from the pool. To provide evidence of a reduction in the number of loss-function calls, we track the number of subpools searched, m_s , to find a local minimum. The results are depicted in Fig. 7. There is a tendency to terminate subpool

ALGORITHM 6. Dynamic-ADAPT-VQE.

```

1: Initialize state  $\rho_0 \leftarrow \rho_{\text{HF}}$ , Ansatz circuit  $\Lambda_0 \leftarrow \text{id}$ , pool  $\mathcal{P}$ .
2: Initialize accuracy  $\varepsilon$  and maximal loss  $\ell$ .
3: Initialize iteration counts  $t_{\max}, n_{\max}$ .
4: Initialize energy bound  $\mathcal{E}_0 \leftarrow \infty$ .
5: Initialize optimal params  $\vartheta_0^*$  as empty vector.
6: for  $t = 1, \dots, t_{\max}$  do
7:    $\mathcal{A}_t, \vec{\vartheta}_t^*, \mathcal{E}_t \leftarrow \text{BuildDynamicLayer}(L_{\Lambda_{t-1}}, \vec{\vartheta}_{t-1}^*, \mathcal{E}_{t-1})$ 
8:   if  $\mathcal{A}_t = \emptyset$  then
9:     return energy bound  $\mathcal{E}_t$ 
10:   Set Ansatz circuit:  $\Lambda_t \leftarrow \mathcal{A}_t^o \circ \Lambda_{t-1}$ 
11: return energy bound  $\mathcal{E}_{t_{\max}}$ 

```

exploration after visiting two or three subpools. This should be compared with the maximum possible QEB-pool values of m_s : $N - 2 = 6, 10, 10, 12, 12$ for $\text{H}_4, \text{LiH}, \text{H}_6, \text{BeH}_2$, and H_2O , respectively. Thus, Fig. 7 shows that subpool exploration reduces the number of loss-function calls in the cases tested.

B. Reducing Ansatz-circuit depth

Next, we compare the ability of Static-(TETRIS)- and Dynamic-ADAPT-VQE to reduce the depth of the Ansatz circuits as compared to standard and Explore-ADAPT-VQE. The data are depicted in Fig. 8. Here, we depict the energy error,

$$\Delta_t = \mathcal{E}_t - E_{\text{FCI}}, \quad (54)$$

given as the distance of the VQE predictions \mathcal{E}_t from the FCI ground-state energy E_{FCI} as a function of (left) the Ansatz-circuit depths and (right) the number of Ansatz-circuit parameters. The left column shows that layered ADAPT-VQEs achieve lower energy errors with shallower Ansatz circuits. Meanwhile, the right column demonstrates that all ADAPT-VQEs achieve similar energy accuracy with respect to the number of Ansatz-circuit parameters.

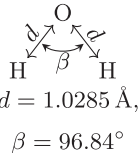
C. Reducing runtime

In this section, we provide numerical evidence that subpool exploration and layering reduce the runtime of ADAPT-VQE. A mathematical analysis of asymptotic runtimes will follow in Sec. V. To provide evidence of a runtime reduction in numerical simulations, we show that layered ADAPT-VQEs require fewer expectation value evaluations (and thus shots and quantum processor runtime) to reach a given accuracy. Our numerical results are depicted in Figs. 9 and 10 for expectation-value evaluations related to calculating losses and parameter optimizations, respectively. We now discuss our results.

To convert data accessible in numerical simulations (such as loss function and optimizer calls) into runtime data (such as expectation values and shots), we proceed as follows. For our numerical data, we evaluate runtime in terms of the number of expectation value evaluations rather than processor calls or shots. This is justified as the number of shots (or processor calls) is directly proportional to the number of expectation values in our simulations [1,8]. Next, we evaluate the runtime requirements associated with loss-function evaluations by tracking the number of times a loss function is called. The evaluation of the loss function over a subpool \mathcal{S} is recorded as $|\mathcal{S}| + 1$ expectation-value evaluations, assuming the use of a finite-difference rule. Thus we produce the data presented in Fig. 9. Finally, we evaluate the runtime requirements of the optimizer by tracking the number of energy expectation values or gradients it requests. The gradient of P variables is then recorded as $P + 1$ energy expectation value evaluations, assuming the use of a finite-difference rule. This gives the data in Fig. 10.

In Fig. 9, we show that layered ADAPT-VQEs require fewer loss-related expectation-value evaluations to reach a given energy accuracy. We attribute this advantage to subpools gradually shrinking during layer construction. They thus

TABLE I. Table of molecular conformations and the corresponding number of STO-3G spin-orbitals N used in numerical simulations.

Name	H ₄	LiH	H ₆	BeH ₂	H ₂ O
Orbitals N	8	12	12	14	14
Structure	H \xleftrightarrow{d} H \xleftrightarrow{d} H \xleftrightarrow{d} H $d = 3 \text{ \AA}$	Li \xleftrightarrow{d} H $d = 1.546 \text{ \AA}$	H \xleftrightarrow{d} H \xleftrightarrow{d} H \xleftrightarrow{d} H \xleftrightarrow{d} H \xleftrightarrow{d} H $d = 0.735 \text{ \AA}$	H \xleftrightarrow{d} Be \xleftrightarrow{d} H $d = 1.316 \text{ \AA}$	 $d = 1.0285 \text{ \AA}$, $\beta = 96.84^\circ$

require fewer loss function evaluations per *Ansatz* element added to the *Ansatz*-element circuit. We further notice that Explore-ADAPT-VQE does not reduce the loss-related expectation values required for standard ADAPT-VQE. We attribute this result to our examples' small pool sizes, with only 8–14 qubits. As qubit sizes increase, we expect a more noticeable advantage for Explore-ADAPT-VQE, as discussed in Sec. V.

In Fig. 10 (left), we show that Static-ADAPT-VQE reduces the number of optimizer calls needed to reach a given accuracy. As expected, the left column shows that Static-ADAPT-VQE calls the optimizer $O(N)$ times less than any other algorithm. This is expected, as standard, Explore-, and Dynamic-ADAPT-VQE calls the optimizer each time a new *Ansatz* element is added to the *Ansatz*-element circuit. Meanwhile, Static-ADAPT-VQE calls the optimizer only after adding a whole layer of $O(N)$ *Ansatz* elements to the *Ansatz*-element circuit. In Fig. 10 (right), we analyze how the reduced number of optimizer calls translates to the number of optimizer-related expectation values required to reach a given accuracy. The data were obtained using a BFGS optimizer with a gradient norm tolerance of 10^{-12} Ha and a relative step tolerance of zero. Compared to the optimizer calls on the left of the figure,

we notice two trends. Dynamic-ADAPT-VQE, while being on par with standard and Explore-ADAPT-VQE for optimizer calls, tends to use a higher number of expectation value evaluations. Similarly, Static-ADAPT-VQE, while having a clear advantage over standard and Explore-ADAPT-VQE for

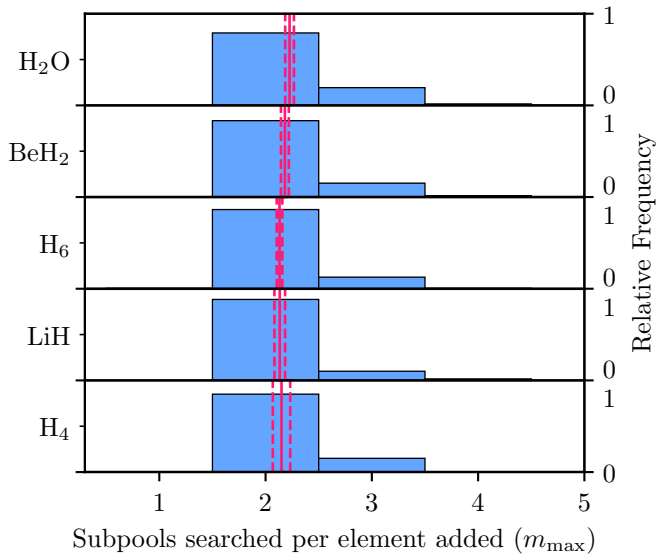


FIG. 7. Histograms of the relative frequencies of the number of subpools searched for identifying a suitable *Ansatz* element m_s , with Explore-ADAPT-VQE. The mean and uncertainty in the mean are indicated by solid and dashed lines, respectively.

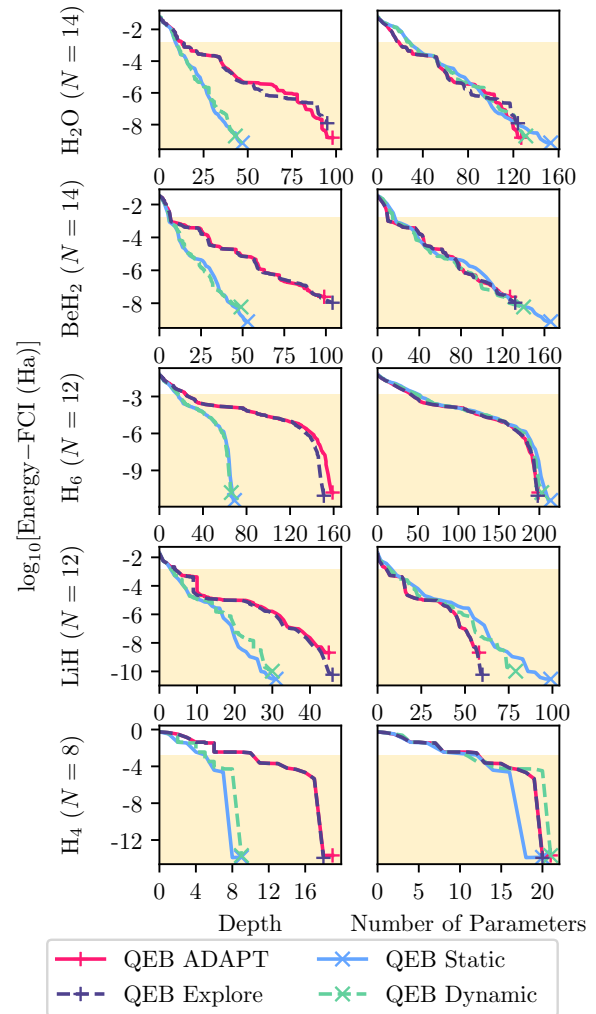


FIG. 8. The energy accuracy is plotted as a function of *Ansatz*-circuit depth (left) and the number of *Ansatz*-circuit parameters (*Ansatz* elements, right), for QEB standard, Explore-, Static- (TETRIS)-, and Dynamic-ADAPT-VQE. These simulations use support commutation. Each row shows data for a specific molecule. The region of chemical accuracy is shaded in cream.

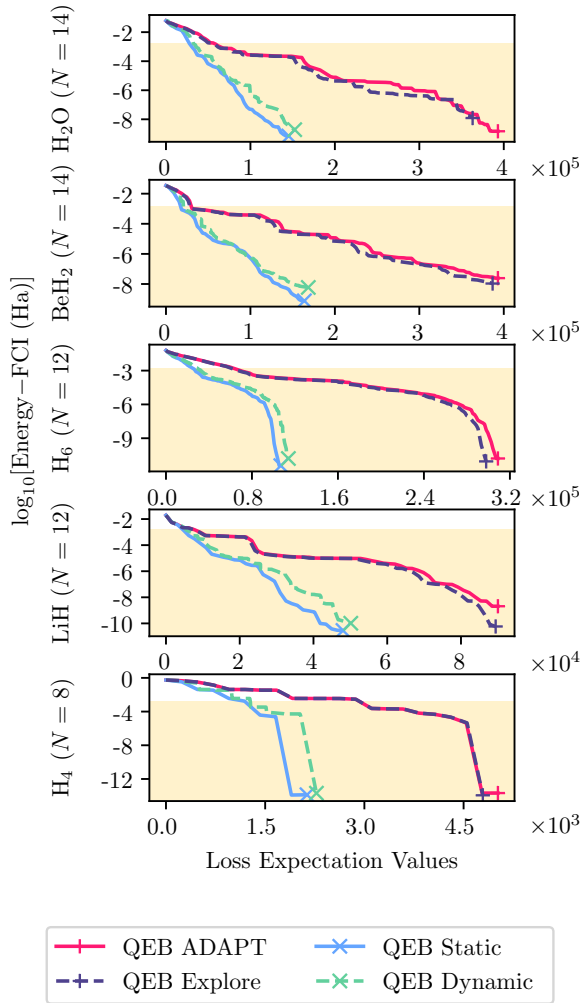


FIG. 9. Energy accuracy against the number of loss function calls. Using the QEB pool and support commutation, we compare standard-, Explore-, Static-(TETRIS)-, and Dynamic-ADAPT-VQE. Each row shows data for a specific molecule, with the number of orbitals increasing up the page. Energy accuracies better than chemical accuracy are shaded in cream.

optimizer calls, tends to have a reduced advantage (and for LiH, even a disadvantage) when it comes to optimizer-related expectation value evaluations. These observations hint towards an increased optimization difficulty for layered ADAPT-VQEs. These observations may be highly optimizer-dependent and should be further investigated in the future.

D. Additional benchmarks

We close this section by referring the reader to additional benchmarking data presented in the Appendixes.

In Appendix B, we compare support to operator commutativity for the qubit pool. In Appendix C, we compare the steepest-gradient loss to the largest-energy-reduction loss. We also compare the QEB pool to the qubit pool in Appendix C.

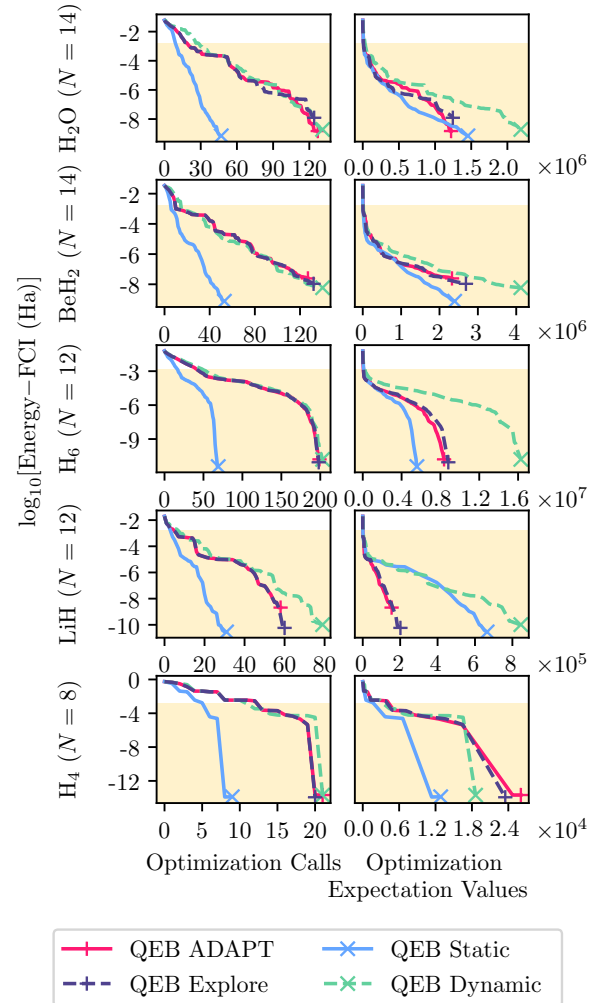


FIG. 10. Energy accuracy against the number of times the *Ansatz* is optimized (left column), and the number of expectation values calculated during optimizer calls (right column). Using the QEB pool and support commutation, we compare standard-, Explore-, Static-(TETRIS)-, and Dynamic-ADAPT-VQE. Each row shows data for a specific molecule, with the number of orbitals increasing up the page. Energy accuracies better than chemical accuracy are shaded in cream.

V. RUNTIME ANALYSIS

In this section, we analyze the asymptotic runtimes of standard, Explore-, Dynamic-, and Static-ADAPT-VQE. We find that under reasonable assumptions, Static-ADAPT-VQE can run up to $O(N^2)$ faster than standard ADAPT-VQE. In what follows, we quantify asymptotic runtimes using $O(x)$, $\Omega(x)$, or $\Theta(x)$ to state that a quantity scales at most, at least, or exactly with x , respectively. For definitions, see Appendix A. We begin our runtime analysis by listing some observations, assumptions, and approximations.

(a) Each algorithm operates on N qubits to find the ground state of a Hamiltonian H .

(b) *Ansatz* circuits are improved by successively adding *Ansatz* elements with a single parameter to the *Ansatz* circuit. This results in iterations $p = 1, \dots, P$, where the p th *Ansatz* circuit has p parameters.

(c) In each iteration p , the algorithm spends runtime on evaluating $N_L(p)$ loss functions.

(d) In each iteration p , the algorithm spends runtime on optimizing p circuit parameters.

(e) Using the finite-difference method, we approximate each of the $N_L(p)$ loss functions in (c) by using two energy-expectation values. This results in evaluating at most $2N_L(p)$ energy expectation values on the quantum computer in the p th iteration.

(f) We assume that the optimizer in (d) performs a heuristic optimization of *Ansatz* circuits with p parameters in polynomial time. Thus, in the p th iteration, a quantum computer must conduct $N_O(p) = \Theta(p^\alpha)$ evaluations of the energy landscape and $N_O(p) = \Theta(p^\alpha)$ evaluations of energy expectation values.

(g) For each energy expectation value in (e) and (f), we assume that a constant number of shots $N_S(H)$ is needed to reach a given accuracy. Note that $N_S(H)$ may depend on N through the Hamiltonian, but it does not depend on p . This is a standard assumption in VQE [1,8].

(h) For each shot in (g), one must execute an *Ansatz* circuit with p *Ansatz* elements. Here, we assume that the runtime $C(p)$ of an *Ansatz* circuit with p *Ansatz* elements is proportional to its depth $d(p)$, i.e., $C(p) = \Theta(d(p))$.

Combining (e),(g),(h) and (f),(g),(h), we can estimate the runtime each algorithm spends on evaluating losses and performing the optimization, respectively:

$$R_L = \sum_{p=1}^P 2N_L(p)N_S(H)C(p) = N_S(H) \sum_{p=1}^P N_L(p)\Theta(d(p)), \quad (55a)$$

$$R_O = \sum_{p=1}^P N_O(p)N_S(H)C(p) = N_S(H) \sum_{p=1}^P \Theta(p^\alpha)\Theta(d(p)). \quad (55b)$$

Below we further analyze these runtime estimates for standard, Explore-, Dynamic-, and Static-ADAPT-VQE.

In standard ADAPT-VQE, we reevaluate the loss of each *Ansatz* element in every iteration. Thus, $N_L(p) = |\mathcal{P}|$. Moreover, the circuit depth $d(p)$ is upper bounded by $d(p) = O(p)$. In the best-case scenario, ADAPT-VQE may arrange *Ansatz* elements into layers accidentally (an effect more likely for large N). This can compress the circuit depths down to $d = \Omega(p/N)$. We summarize this range of possible circuit depths using the compact expression $d(p) = \Theta(pN^{-\gamma})$, with $\gamma \in [0, 1]$. In numerical simulations, we typically observe that $\gamma \approx 0$, i.e., the depth of an *Ansatz* circuit is proportional to the number of *Ansatz* elements. These expressions for $N_L(p)$ and $d(p)$ allow us to estimate the runtime of standard ADAPT-VQE algorithms:

$$R_L^A = |\mathcal{P}|N_S(H)\Theta(P^2N^{-\gamma}), \quad (56a)$$

$$R_O^A = N_S(H)\Theta(P^{2+\alpha_A}N^{-\gamma}). \quad (56b)$$

Explore-ADAPT-VQE results in circuits of the same depths as ADAPT-VQE, i.e., $d = \Theta(pN^{-\gamma})$. However, the use of subpool exploration in Explore-ADAPT-VQE may reduce the number of loss-function evaluations $N_L(p)$. As discussed in Sec. III B (paragraph on *Efficiency*), in the best-case scenario, the number of loss function evaluations per iteration is lower

bounded by $N_L(p) = \Omega(|\mathcal{P}|/N)$. In the worst-case scenario, subpool exploration may explore the whole pool of *Ansatz* elements, such that $N_L(p) = O(|\mathcal{P}|)$. Based on these relations, we can estimate the runtime of Explore-ADAPT-VQE:

$$R_L^E = \begin{cases} |\mathcal{P}|N_S(H)\Omega(P^2N^{-(1+\gamma)}), \\ |\mathcal{P}|N_S(H)O(P^2N^{-\gamma}), \end{cases} \quad (57a)$$

$$R_O^E = N_S(H)\Theta(P^{2+\alpha_E}N^{-\gamma}). \quad (57b)$$

Dynamic-ADAPT-VQE has the same scaling of the number of loss function evaluations per iteration, $N_L(p)$, as Explore-ADAPT-VQE. Thus, $N_L(p) = \Omega(|\mathcal{P}|/N)$ in the best case and $N_L(p) = O(|\mathcal{P}|)$ in the worst case. The circuit depth of Dynamic-ADAPT-VQE scales as $d(p) = \Theta(p/N)$. One can observe a clear benefit from layering. The upper bound, $d(p) = O(p)$, in standard and Explore-ADAPT-VQE becomes $d(p) = O(p/N)$ in Dynamic-ADAPT-VQE. Using these relations for $N_L(p)$ and $d(p)$, we can estimate the runtime of Dynamic-ADAPT-VQE:

$$R_L^D = \begin{cases} |\mathcal{P}|N_S(H)\Omega(P^2N^{-2}), \\ |\mathcal{P}|N_S(H)O(P^2N^{-1}), \end{cases} \quad (58a)$$

$$R_O^D = N_S(H)\Theta(P^{2+\alpha_D}N^{-1}). \quad (58b)$$

The analysis of Static-ADAPT-VQE's runtime is more straightforward with respect to the layer count t than to the parameter count p . Therefore, we revisit and modify our previous observations, assumptions, and approximations.

(a) Static-ADAPT-VQE operates on N qubits to find the ground state of a Hamiltonian H .

(b) Static-ADAPT-VQE builds *Ansatz* circuits in layers indexed by $t = 1, \dots, t_{\max}$. The t th layer contains $n_{\text{tot}}(t) = \Theta(N)$ *Ansatz* elements. Since each *Ansatz* element depends on a single parameter, a layer contains $n_{\text{tot}}(t) = \Theta(N)$ circuit parameters. Summing the parameters in each layer gives the total number of parameters in the circuit: $P = \sum_{t=1}^{t_{\max}} n_{\text{tot}}(t)$.

(c) For each layer t , Static-ADAPT-VQE spends runtime on evaluating the loss of $N_L(t) = |\mathcal{P}|$ *Ansatz* elements.

(d) For each layer t , Static-ADAPT-VQE spends runtime on optimizing $p(t) = \sum_{t'=1}^t n_{\text{tot}}(t') = \Theta(N)t$ circuit parameters.

(e) Using the finite-difference method, we approximate each of the $N_L(t)$ loss functions in (c) by using two energy expectation values. This results in evaluating at most $2N_L(t) = 2|\mathcal{P}|$ energy expectation values on the quantum computer in the t th iteration.

(f) Again, we assume that the optimizer in (d) performs a heuristic optimization of *Ansatz* circuits with $p(t)$ parameters in polynomial time. Thus, in the t th layer a quantum computer must conduct $N_O(t) = \Theta(p(t)^{\alpha_S})$ evaluations of the energy landscape and $N_O(t) = \Theta(p(t)^{\alpha_S})$ evaluations of energy expectation values. Using $p(t) = \Theta(N)t$ from (d), this implies that $N_O(t) = \Theta(N^{\alpha_S}t^{\alpha_S})$.

(g) As before, for each energy expectation value in (e) and (f), we assume that a constant number of shots $N_S(H)$ is needed to reach a given accuracy. Note that $N_S(H)$ may depend on N through the Hamiltonian, but it does not depend on p .

TABLE II. The ratio of the runtimes of the listed algorithms to the runtime of standard ADAPT-VQE. N is the qubit number, and P is the number of parameters in the final *Ansatz* circuit. See text for further explanation.

Algorithm	R_L/R_L^{ADAPT}	R_O/R_O^{ADAPT}
Explore	$\Omega(N^{-1}), O(1)$	$\Theta(1)$
Dynamic	$\Omega(N^{-2+\gamma}), O(N^{-1+\gamma})$	$\Theta(N^{-1+\gamma})$
Static	$\Theta(N^{-2+\gamma})$	$\Theta(N^{-2+\gamma})$

(h) For each shot in (g), one must execute an *Ansatz* circuit with $p(t)$ *Ansatz* elements. Again, we assume that the runtime $C(t)$ of an *Ansatz* circuit with $p(t)$ *Ansatz* elements is proportional to its depth $d(p(t))$, i.e., $C(t) = \Theta(d(p(t)))$. Due to layering, the circuit depth of Static-ADAPT-VQE scales as $d(p) = \Theta(p/N)$. (This scaling is identical for Dynamic-ADAPT-VQE.) This results in $C(t) = \Theta(p(t)/N)$. Further, using $p(t) = \Theta(N)t$ from (d), we find that each shot in (g) requires a circuit runtime of $C(t) = \Theta(t)$.

Combining the updated (e),(g),(h) and (f),(g),(h), we find the loss- and optimization-related runtimes of Static-ADAPT-VQE, respectively:

$$R_L^S = \sum_{t=1}^{t_{\max}} 2N_L(t)N_S(H)C(t) = |\mathcal{P}|N_S(H)\Theta(t_{\max}^2), \quad (59a)$$

$$R_O^S = \sum_{t=1}^{t_{\max}} N_O(t)N_S(H)C(t) = N_S(H)\Theta(N^{\alpha_S}t_{\max}^{\alpha_S+2}). \quad (59b)$$

Since $P = \Theta(N)t_{\max}$ implies $t_{\max} = P\Theta(N^{-1})$, we can simplify these runtime estimates:

$$R_L^S = |\mathcal{P}|N_S(H)\Theta(P^2N^{-2}), \quad (60a)$$

$$R_O^S = N_S(H)\Theta(P^{2+\alpha_S}N^{-2}). \quad (60b)$$

We summarize this section by listing the ratios of asymptotic runtimes for Explore-, Dynamic-, and Static-ADAPT-VQE divided by the asymptotic runtime of standard ADAPT-VQE in Table II. Here, we assume equal polynomial scaling ($\alpha_A = \alpha_E = \alpha_D = \alpha_S$) of the optimization runtime for standard, Explore, Dynamic-, and Static-ADAPT-VQE. As expected from our numerical runtime analysis in Sec. IV C, for typical ADAPT-VQE circuit depth (where $\gamma = 0$), Static-ADAPT-VQE can provide the largest runtime reduction. This reduction is quadratic in the number of qubits: $\Theta(N^{-2})$. Further improvements to bounding the number of losses in Explore- and Dynamic-ADAPT-VQE are discussed in Appendix H.

VI. NOISE

In this section, we explore the benefits of reducing ADAPT-VQEs' *Ansatz*-circuit depths with respect to noise. Our main finding is that the use of layering to reduce *Ansatz*-circuit depths mitigates global amplitude-damping and global dephasing noise, where idling and nonidling qubits are affected alike. However, reduced *Ansatz*-circuit depths do not mitigate the effect of local depolarizing noise, which exclusively affects qubits operated on by noisy (two-qubit, CNOT)

gates. The explanation for this, we show, is that the *Ansatz*-circuit depth is a good predictor for the effect of global amplitude-damping and dephasing noise. On the other hand, we show that the errors induced by local depolarizing noise are approximately proportional, not to the depth, but to the number of (CNOT) gates. For this reason, a shallower *Ansatz* circuit with the same number of noisy two-qubit gates will not reduce the sensitivity to depolarizing noise.

A. Noise models

Our noise models focus on superconducting architectures, and we tune our analysis towards the IBM Quito (IBM Quantum Falcon r4T) processor. For this processor, the quoted two-qubit gate times refer to CNOT gates. Thus, we will take CNOT gates to be our native two-qubit gate. Further, our simulations use one- and two-qubit gate-execution times of 35.5 and 295.1 ns, respectively.¹ Further, we assume all-to-all connectivity. Similar native gates and execution times apply to silicon quantum processors [37].

In our simulations, we model *amplitude damping* of a single qubit by the standard amplitude-damping channel. (For detailed expressions of the amplitude-damping channel and the other noise channels we use, see Appendix J I.) Its decay constant is determined by the inverse T_1 time: $\omega_1 = 1/T_1$. Similarly, we model *dephasing* of a single qubit by the standard dephasing channel. The T_1 and T_2^* times determine its phase-flip probability via the decay constant $\omega_z = 2/T_2^* - 1/T_1$. Finally, we model *depolarization* of a single qubit by a symmetric depolarizing channel with depolarization strength $p \in [0, 1]$, where $p = 0$ leaves a pure qubit pure and $p = 1$ brings it to the maximally mixed state.

In our simulations, we model the effects of amplitude damping, dephasing, and depolarizing noise on the *Ansatz* circuits Λ_t in a layer-by-layer approach. This is illustrated in Fig. 11. We decompose the *Ansatz* circuit Λ_t into $l = 1, \dots, \ell_t$ layers of support-commuting *Ansatz*-element layers $\{\mathcal{A}_l\}$:

$$\Lambda_t = \mathcal{A}_{\ell_t}^{\circ} \circ \dots \circ \mathcal{A}_l^{\circ} \circ \dots \circ \mathcal{A}_1^{\circ}. \quad (61)$$

For amplitude damping and dephasing noise, each *Ansatz*-element layer \mathcal{A}_l° is transpiled into columns of native gates that can be implemented in parallel (see Ref. [35] for more details). The native gate with the longest execution time of each native-gate column sets the column execution time. The sum of the column execution times then gives the execution time τ_l of the *Ansatz*-element layer \mathcal{A}_l° . After each *Ansatz*-element layer \mathcal{A}_l° , amplitude damping is implemented by applying an amplitude-damping channel to every qubit $r = 1, \dots, N$ in an amplitude-damping layer. This results in an amplitude-damped *Ansatz* circuit $\Lambda_t(\omega_1)$. Similarly, after each *Ansatz*-element layer \mathcal{A}_l° , dephasing is implemented by applying a dephasing channel to every qubit $r = 1, \dots, N$ in a dephasing layer. This results in a dephased *Ansatz* circuit $\Lambda_t(\omega_z)$. Finally, for depolarizing noise, we apply the whole *Ansatz*-element layer and then a depolarizing channel to each

¹These values were taken from the IBM Quantum services.

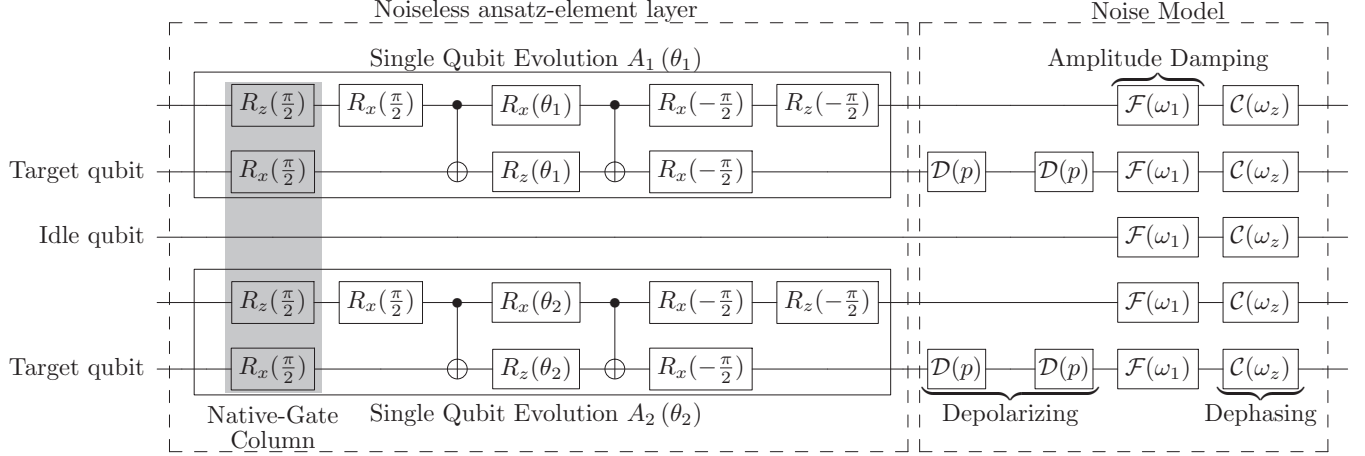


FIG. 11. Circuit diagram visualizing the layer-by-layer noise model: on the left, a noiseless *Ansatz*-element layer with two support-commuting *Ansatz* elements is decomposed into columns of native gates. On the right, noise is added to the *Ansatz*-element layer. For global amplitude damping (or dephasing) noise, the channel \mathcal{F} (or \mathcal{C}) is applied to each qubit. For local depolarizing noise, the channels \mathcal{D} are applied to the target of the noisy (two-qubit, CNOT) gate.

qubit. The strength of a qubit's depolarizing channel is determined by the exact number of times that the qubit was the target in a CNOT gate in the preceding layer. This results in a depolarized *Ansatz* circuit $\Lambda_t(p)$. For a visualization of our layer-by-layer-based noise model, see Fig. 11. For detailed mathematical expressions, see Appendix J2.

We note that applying the noise channels after each *Ansatz*-element layer could be refined by applying the noise channels after each gate in the *Ansatz*-element layer. However, as shown in Ref. [14], such a gate-by-gate noise model, as opposed to our layer-by-layer based noise model, would increase computational costs and has limited effect on the results. In what follows, we collectively refer to amplitude-damped *Ansatz* circuits $\Lambda_t(\omega_1)$, dephased *Ansatz* circuits $\Lambda_t(\omega_z)$, and depolarized *Ansatz* circuits $\Lambda_t(p)$ as $\Lambda_t(\alpha)$. Here, α refers to the key noise parameters ω_1 , ω_z , or p of each respective noise model.

B. Energy error and noise susceptibility

Going forward, we analyze the effect of noise on the energy error [cf. Eq. (54)]

$$\Delta_t(\alpha) = \mathcal{E}_t(\alpha) - E_{\text{FCI}}. \quad (62)$$

$\Delta_t(\alpha)$ now depends not only on the iteration step t , but also on the noise parameter α , via the noise-dependent expectation value

$$\mathcal{E}_t(\alpha) = \text{Tr}[H \Lambda_t(\alpha) [\rho_0]]. \quad (63)$$

To analyze the energy error, we expand the methodology of Ref. [14]. More specifically, we decompose the energy error into two contributions:

$$\Delta_t(\alpha) = \Delta_t(0) + [\Delta_t(\alpha) - \Delta_t(0)]. \quad (64)$$

The first term, $\Delta_t(0)$, is the energy error of the noiseless *Ansatz* circuit. The second term, $\Delta_t(\alpha) - \Delta_t(0)$, is the energy error due to noise. Subsequently, we Taylor-expand the energy error due to noise to first order:

$$[\Delta_t(\alpha) - \Delta_t(0)] = \chi_t \alpha + O(\alpha^2). \quad (65)$$

As depicted in Fig. 12, in the regime of small noise parameters α (where energy errors are below chemical accuracy), the linear approximation is an excellent predictor for the energy error. Conveniently, this allows us to summarize the effect of noise on the energy error through the noise susceptibility χ_t , defined as

$$\chi_t := \left. \frac{\partial \mathcal{E}_t(\alpha)}{\partial \alpha} \right|_{\alpha=0} = \text{Tr} \left[H \left. \frac{\partial \Lambda_t(\alpha)}{\partial \alpha} \right|_{\alpha=0} [\rho_0] \right]. \quad (66)$$

In Appendix K, we calculate the noise susceptibility χ_t of amplitude damping \mathcal{F} , dephasing \mathcal{C} , and depolarizing \mathcal{D} noise:

$$\chi_t^{\mathcal{F}} := \left. \frac{\partial \mathcal{E}_t(\omega_1)}{\partial \omega_1} \right|_{\omega_1=0} = \ell_t N \times d \mathcal{E}(\Lambda_t, \mathcal{F}), \quad (67a)$$

$$\chi_t^{\mathcal{C}} := \left. \frac{\partial \mathcal{E}_t(\omega_z)}{\partial \omega_z} \right|_{\omega_z=0} = \ell_t N \times d \mathcal{E}(\Lambda_t, \mathcal{C}), \quad (67b)$$

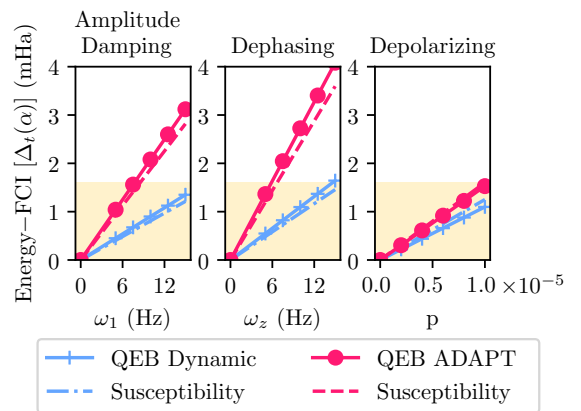


FIG. 12. Energy error $\Delta_t(\alpha)$ for an *Ansatz* circuit Λ_t of H_4 as a function of noise strength $\alpha = \omega_1, \omega_z, p$ going from the left to right, respectively. Connected dots and crosses are calculated using full density-matrix simulations. Dashed lines show the corresponding extrapolation using noise susceptibility.

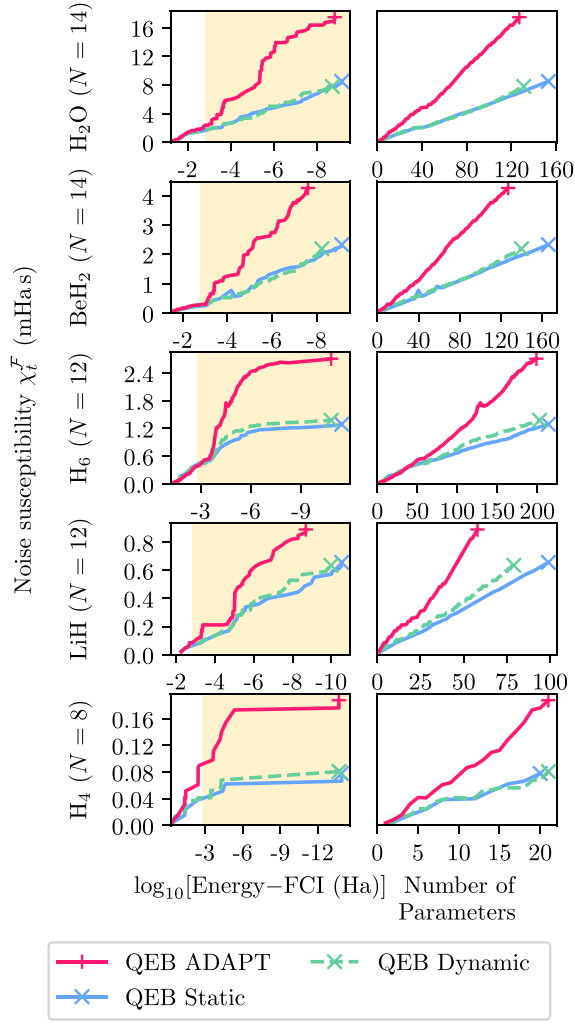


FIG. 13. Amplitude-damping noise susceptibility as a function of (left) energy accuracy and (right) the number of parameters in *Ansatz* circuits Λ_t . QEB-ADAPT is compared to support-based Dynamic-ADAPT-VQE. Each row shows data for a specific molecule, with the number of orbitals increasing up the page. Energy accuracies better than chemical accuracy are shaded in cream.

$$\chi_t^{\mathcal{D}} := \left. \frac{\partial \mathcal{E}_t(p)}{\partial p} \right|_{p=0} = N_{II} \times d\mathcal{E}(\Lambda_t, \mathcal{D}), \quad (67c)$$

respectively. Here, N denotes the number of qubits; ℓ_t is the number of *Ansatz*-element layers in the *Ansatz* circuit Λ_t ; N_{II} is the number of noisy (two-qubit, CNOT) gates in the *Ansatz* circuit Λ_t ; and the $d\mathcal{E}$'s denote the average energy fluctuations, defined in Eqs. (K9) of Appendix K. As discussed further in Appendix K, the average energy fluctuations can be calculated from noiseless expectation values. This allows us to compute the noise susceptibility with faster state-vector simulations rather than computationally demanding density-matrix simulations.

C. Benchmarking layered circuits with noise

In this section, we compare the noise susceptibility of standard, Static- (TETRIS-), and Dynamic-ADAPT-VQE in the presence of noise. As before, we showcase these algorithms

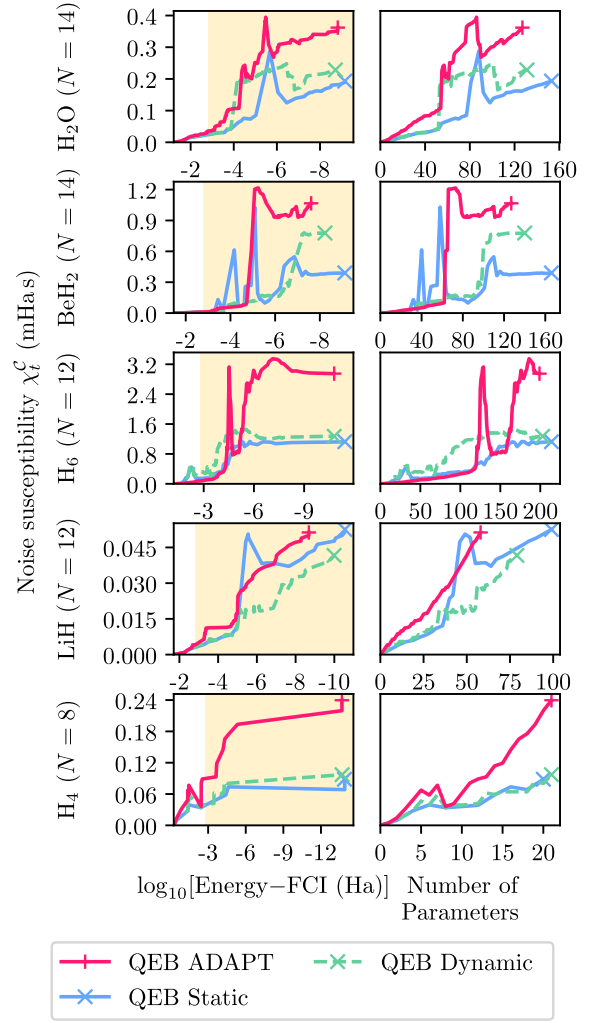


FIG. 14. Same as Fig. 13, but for dephasing noise.

on a range of molecules (summarized in Table I) using the QEB pool with support commutativity. When performing our comparison, we grow the *Ansatz* circuits Λ_t and optimize its parameters in noiseless settings, as previously discussed in Ref. [14]. We then compute the noise susceptibility of Λ_t as described in the previous section. The results for amplitude damping, dephasing, and depolarizing noise are depicted in Figs. 13–15, respectively. In all three figures, we plot the noise susceptibility as a function of (left) the noiseless energy accuracy $\Delta_t(0)$ or (right) the number of parameters. The rows of each plot depict different molecules in order of increasing spin orbitals from bottom to top: H₄, H₆, LiH, BeH₂, and H₂O. To interpret the graphs in Figs. 13–15, we remind the reader that larger values of noise susceptibility correspond to lower noise-tolerance. Thus, larger values of noise susceptibility in Figs. 13–15 can be interpreted as worse performances.

Layering benefits: From Fig. 13, it is evident that layering is successful in mitigating the effect of amplitude-damping noise. Here, we observe that the noise susceptibility of Static- and Dynamic-ADAPT-VQE is approximately half that of standard ADAPT-VQE. This is a clear indication that layering can reduce the effect of noise. In Fig. 14, we observe that layering also tends to reduce the noise susceptibility in the

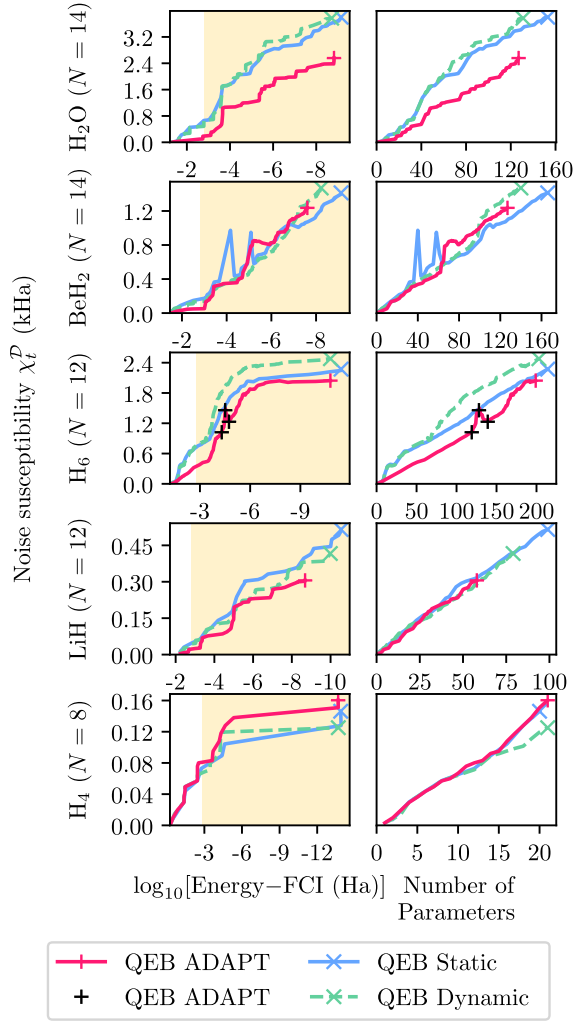


FIG. 15. Same as Fig. 13, but for depolarizing noise. In addition, black crosses correspond to density matrix simulations corroborating noise susceptibility via finite differences. The black crosses are discussed further in Appendix D.

presence of dephasing noise. However, in this scenario, the advantage is less consistent across different *Ansatz* circuits and molecules. Finally, in Fig. 15, we observe that for depolarizing noise, all algorithms tend to produce similar noise susceptibilities. Sometimes, one shows an advantage over the other, and *vice versa*, depending on the *Ansatz* circuit and molecule. Our simulations indicate no clear disadvantage of using layering in the presence of depolarizing noise. In summary, our numerical simulations suggest that layering is useful for mitigating global amplitude damping and dephasing noise. Moreover, layering seems to have neither a beneficial nor a detrimental effect in the presence of local depolarizing noise. To explain these findings, we further investigate the dependence of noise susceptibility on several circuit parameters in Sec. VID.

Gate-fidelity requirements: We now use the noise susceptibility data in Figs. 13–15 to estimate the fidelity requirements for operating ADAPT-VQEs. For this estimation, recall that quantum chemistry simulations of energy eigenvalues target an accuracy of 1.6 mHa. To achieve this chemical accuracy, we require the energy error due to noise to be smaller than

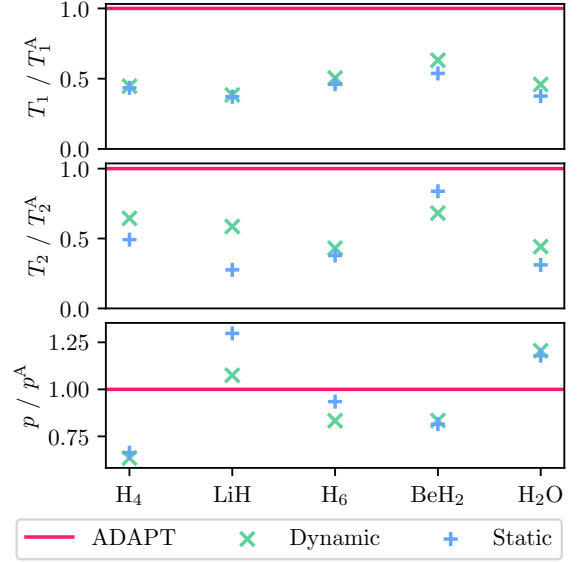


FIG. 16. The ratio of the minimal T_1 and T_2 times and maximal depolarizing probability p for Dynamic- and Static-ADAPT-VQE to standard ADAPT-VQE required to reach an accuracy of $\Delta = 10^{-7}$ mHa for each molecule.

≈ 1 milli-Hartree: $\chi_t \alpha \lesssim 1$ mHa. Applying this condition to amplitude damping (where $\alpha = 1/T_1$), dephasing (where $\alpha \approx 1/T_2^*$), and depolarizing noise (where $\alpha = p$), we find a set of gate fidelity requirements:

$$T_1 \gtrsim \frac{\chi_t^{\mathcal{F}}}{1 \text{ mHa}}, \quad T_2^* \gtrsim \frac{\chi_t^{\mathcal{C}}}{1 \text{ mHa}}, \quad p \lesssim \frac{1 \text{ mHa}}{\chi_t^{\mathcal{D}}}. \quad (68)$$

The data presented in Figs. 13–15 suggest the following requirements for the gate operations to enable chemically accurate simulations:

$$T_1 \gtrsim 1 \text{ s}, \quad T_2^* \gtrsim 100 \text{ ms}, \quad p \lesssim 10^{-6}. \quad (69)$$

A more detailed breakdown of the maximal p and minimal T_1 and $T_2 := 2/\omega_z$ times for each algorithm and molecule is presented in Fig. 16. These requirements are beyond the current state-of-the-art quantum processors [37,38]. How much these requirements can be improved by error-mitigation techniques [39] remains an open question for future research.

D. Noise-susceptibility scalings

In this section, we investigate the dependence of noise susceptibility on basic circuit parameters, such as the number of qubits N , circuit depth $d \propto \ell_t$, or the number of noisy (two-qubit, CNOT) gates N_{II} . Our analysis will help in understanding why layering can mitigate global amplitude damping and dephasing noise but not local depolarizing noise.

We study numerically how noise susceptibility scales with circuit depth and the number of noisy (two-qubit, CNOT) gates N_{II} . The data are presented in Fig. 17. The top panels show the noise susceptibility in the presence of amplitude damping (left), dephasing (center), and depolarizing noise (right) for various algorithms and molecules. The noise-susceptibility data are presented on a log-log plot as a function of circuit depths d (left and center) as well as N_{II} (right), respectively.

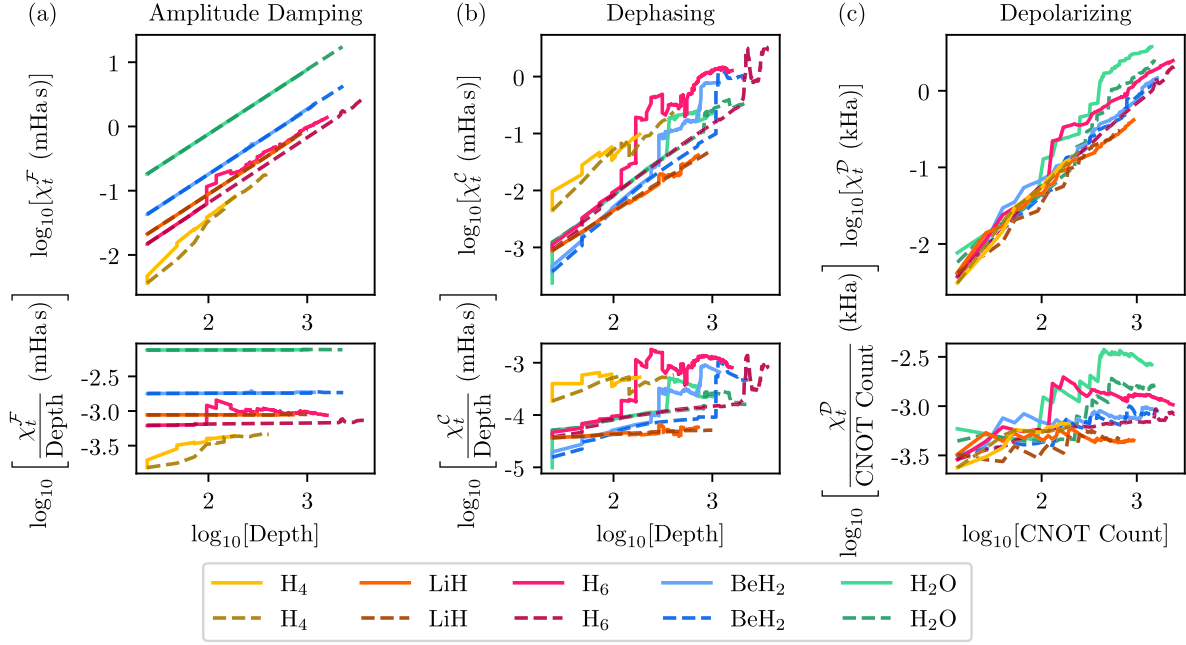


FIG. 17. Noise susceptibility for (a) amplitude-damping, (b) dephasing, and (c) depolarizing noise as a function of (a),(b) *Ansatz*-circuit depths d and (c) the number of noisy CNOT-gates N_{II} . The top panels show noise susceptibility as a function of d (a),(b) and N_{II} (c). The bottom panels of (a) and (b) show noise susceptibility divided by d , as a function d . The bottom panel of (c) shows noise susceptibility divided by N_{II} as a function of N_{II} . The solid lines correspond to the simulation of Dynamic-ADAPT-VQE using support commutation for a range of molecules. The dashed lines represent simulations based on QEB-ADAPT-VQE.

From Fig. 17, we find that the noise susceptibility scales roughly linearly with the plotted parameters. To further analyze this rough linearity, we produce a log-log plot in the bottom panels of χ_t^F/d (left), χ_t^C/d (center), and χ_t^D/N_{II} (right) as a function of d , d , and N_{II} , respectively. Had the scalings of interest been linear, the bottom panels would have depicted constant curves. This is not entirely the case. But, the curves' deviations from constants are sufficiently sublinear to support our claim that the curves in the upper plots are roughly linear.

The scalings observed in Fig. 17 confirm our previous intuition. Based on Eq. (67), and using the assumption that $d\mathcal{E}$ is roughly constant, we would expect that the noise susceptibility in the presence of amplitude damping or dephasing noise is proportional to the circuit depth and the number of qubits:

$$\chi_t^F \propto Nd \quad \text{and} \quad \chi_t^C \propto Nd. \quad (70)$$

This claim is supported by Fig. 17. Moreover, previous studies [14] have found that the noise susceptibility scales linearly with the number of depolarizing two-qubit gates:

$$\chi_t^D \propto N_{II}. \quad (71)$$

Also, this claim is supported by Fig. 17.

Thus, for global (amplitude damping and dephasing) noise, which affects idling and nonidling qubits alike, our analysis indicates that circuit depth is a good predictor of noise susceptibility. On the other hand, for local (depolarizing) noise, which affects only the qubits that are nontrivially operated on, N_{II} is a good predictor of the noise susceptibility. Consequently, we expect that compressing the depth of an *Ansatz*

circuit by layering can mitigate noise in the former, but not the latter, of these settings.

VII. SUMMARY AND CONCLUSION

In this paper, we introduced layering and subpool-exploration strategies for ADAPT-VQEs that reduced circuit depth, runtime, and susceptibility to noise. In noiseless numerical simulations, we demonstrate that layering reduces the depths of an *Ansatz* circuit when compared to standard ADAPT-VQE. We further showed that our layering algorithms achieve circuits that are as shallow as TETRIS-ADAPT-VQE. The reduction in *Ansatz* circuit depth is achieved without increasing the number of *Ansatz* elements, circuit parameters, or CNOT gates in the *Ansatz* circuit. The noiseless numerical simulations further provide evidence that layering and subpool-exploration can reduce the runtime of ADAPT-VQE by up to $O(N^2)$, where N is the number of qubits in the simulation. Finally, we benchmarked the effect of reducing the depth of ADAPT-VQEs on the algorithms' noise susceptibility. For global noise models, which affect idling and nonidling qubits alike (such as our amplitude-damping and dephasing model), we show that the noise susceptibility is approximately proportional to the *Ansatz*-circuit depth. For these noise models, reduced circuit depth due to layering is beneficial in reducing the noise susceptibility of ADAPT-VQEs. For local noise models, where only nonidling qubits are affected by noise (as with our depolarizing noise model), we show that the noise susceptibility is approximately proportional to the number of noisy (two-qubit, CNOT) gates. For these noise models, layering strategies are neither useful nor

harmful, as they hardly change the CNOT count of ADAPT-VQEs. We finish our paper by stating three conclusions from our work.

To layer or not to layer?: Depending on the dominant noise source of a quantum processor, layering may or may not lead to improved noise resilience. For processors where global noise dominates, we recommend layering.

Static or dynamic layering?: Our paper considered static and dynamic layering. Which of the two should be used? Static layering optimizes each layer once, while dynamic layering optimizes the *Ansatz* after adding each *Ansatz* element. Both layering strategies lead to *Ansatz* circuits of similar depths and require a similar number of parameters and CNOT gates to reach a certain energy accuracy. However, static layering calculates significantly fewer energy expectation values on the quantum processor. Therefore, we recommend static layering for the small molecules studied in this work. For larger molecules, dynamic layering could be preferable.

How useful is subpool exploration?: Our paper introduced a pool-exploration strategy that reduces the number of loss-function evaluations and, thereby, the number of calls to the quantum processor. However, in the examples studied in this work, the number of loss-function evaluations was exceeded by the energy-expectation-value calls. Thus, subpool exploration had little impact on the algorithms. Again, this could change when larger molecules are studied.

ACKNOWLEDGMENTS

The authors thank Yordan S. Yordanov for the use of his codebase for VQE protocols, and Wilfred Salmon for insightful discussions. We further thank Sophia E. Economou, Nicholas J. Mayhall, Edwin Barnes, Panagiotis G. Anastasiou, and the Virginia Tech group for fruitful discussions. We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and they do not reflect the official policy or position of IBM or the IBM Quantum team.

APPENDIX A: BIG O, OMEGA, AND THETA NOTATIONS

This Appendix defines big O, big Omega (Knuth definition), and big Theta notations. For our purposes, the notations can be defined, respectively, as

$$f(x) = O(g(x)) \iff \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} < \infty, \quad (\text{A1})$$

$$f(x) = \Omega(g(x)) \iff \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} > 0, \quad (\text{A2})$$

$$f(x) = \Theta(g(x)) \iff 0 < \lim_{x \rightarrow \infty} \frac{f(x)}{g(x)} < \infty. \quad (\text{A3})$$

APPENDIX B: COMMUTATIVITY VERSUS SUPPORT

In this article, we introduced two notions of commutation that can be leveraged in constructing ADAPT-VQE algorithms. They were operator commutation and support commutation, and we will compare them here.

As noted in Appendix F, the operator and support noncommuting sets of the *Ansatz* elements in the QEB pool differ by,

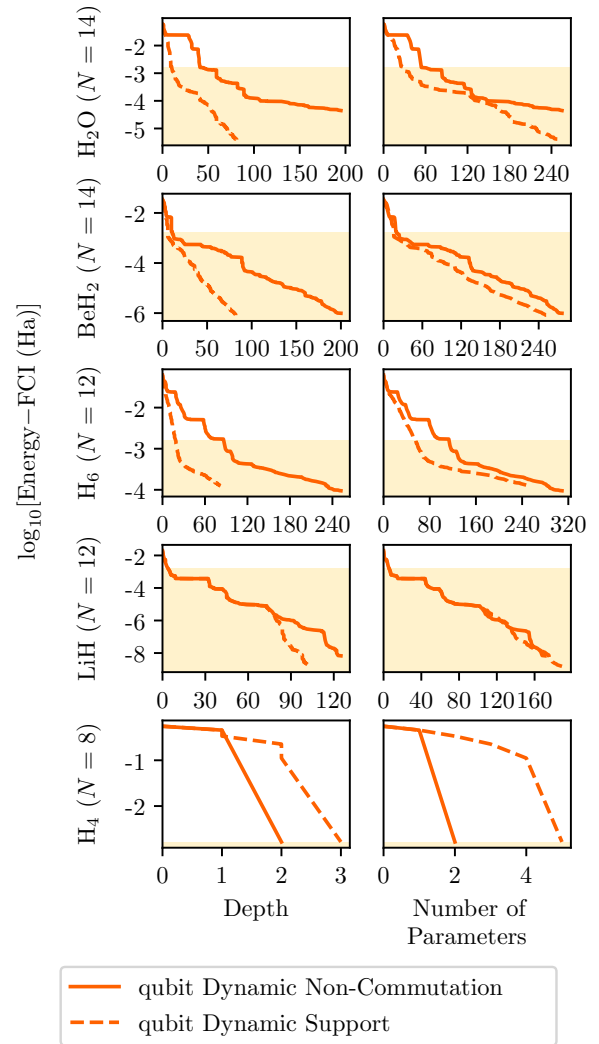


FIG. 18. Energy accuracy against *Ansatz*-circuit depths (left) and the number of *Ansatz*-circuit parameters (*Ansatz* elements, right), for qubit-Dynamic-ADAPT-VQE using support and operator commutation—dashed and solid lines, respectively. Each row shows data for a specific molecule, with the number of orbitals increasing up the page. Energy accuracies better than chemical accuracy are shaded in cream.

at most, two *Ansatz* elements. Additionally, both notions of commutation result in layers of constant depth with respect to the number of qubits. On the other hand, the operator and support noncommuting sets of the qubit pool differ by $\Theta(N^3)$ *Ansatz* elements. Thus, the two types of commutation could construct vastly different *Ansätze*. The *Ansätze* constructed based on support commutation will have layers of constant depth, while those constructed based on operator commutation will have layers of depth $O(N^3)$. Thus, we consider both the support and operator commutation variants of Dynamic-ADAPT-VQE using the qubit pool to highlight the differences between the commutation types. Figure 18 shows the energy error [Eq. (54)] as a function of depth and the number of parameters. For LiH, H₆, BeH₂, and H₂O we observe that support commutation outperforms operator commutation.

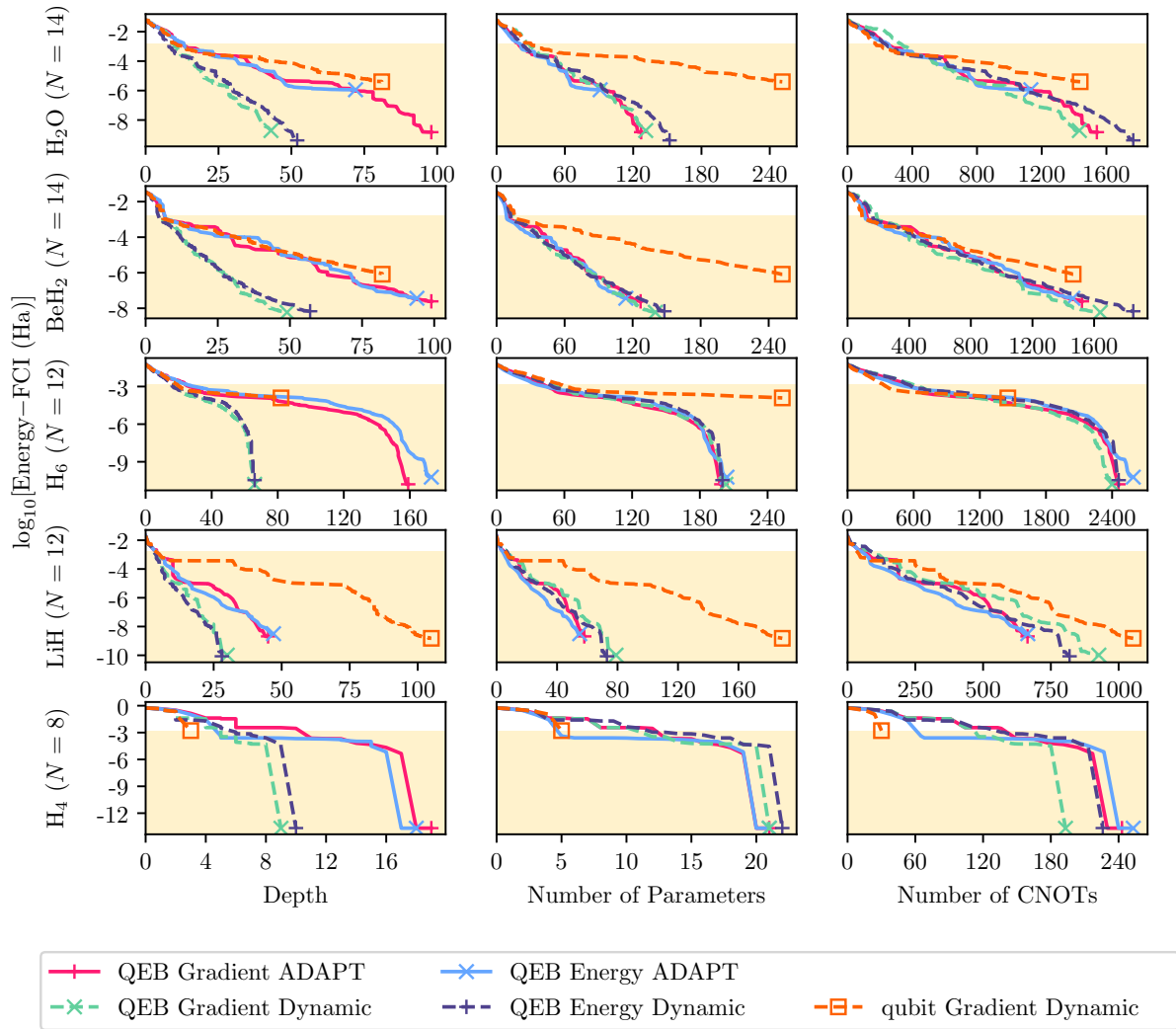


FIG. 19. Energy accuracy against *Ansatz*-circuit depths (left), the number of *Ansatz*-circuit parameters (*Ansatz* elements, center), and the number of CNOT gates in the *Ansatz* circuit (right), for qubit-Dynamic-ADAPT-VQE using support commutation as well as QEB standard and Dynamic-ADAPT-VQE using support commutation with both steepest gradient and largest energy reduction decision rules, are compared. Each row shows data for a specific molecule, with the number of orbitals increasing up the page. Energy accuracies better than chemical accuracy are shaded in cream.

However, for H_4 , we find operator commutation outperforms support commutation.

APPENDIX C: POOL AND DECISION RULE COMPARISONS

This Appendix includes additional comparisons of Dynamic-ADAPT-VQE with two different loss functions and with the QEB and Pauli pools. We refer to the loss function in Eq. (20) as gradient selection. An alternative loss function is

$$L_t(A) = \min_{\theta} E_{t,A}(\theta), \quad (\text{C1})$$

which we will refer to as energy selection. That is, we optimize the *Ansatz* with respect to the last parameter for each *Ansatz* element in the subpool and pick the *Ansatz* element that reduces the energy by the most.

In Fig. 19, we see that both the energy and gradient selection rules perform similarly in energy accuracy for a given depth, number of parameters, and number of CNOT gates. Standard ADAPT-VQE with both gradient and energy selection are included as reference points.

The QEB-Dynamic-ADAPT-VQE algorithms require fewer parameters and shallower *Ansatz* circuits than the qubit-Dynamic-ADAPT-VQE algorithms for a given energy accuracy. However, the energy accuracy of QEB-Dynamic-ADAPT-VQE and qubit-Dynamic-ADAPT-VQE for a given number of CNOT gates is similar, suggesting that the number of two-qubit gates in the *Ansatz* could be a good pool-independent predictor of energy convergence.

Additionally, in Fig. 20, we see that evaluating the energy selection rule is more expensive than the gradient selection rule. However, as the optimization dominates the total number

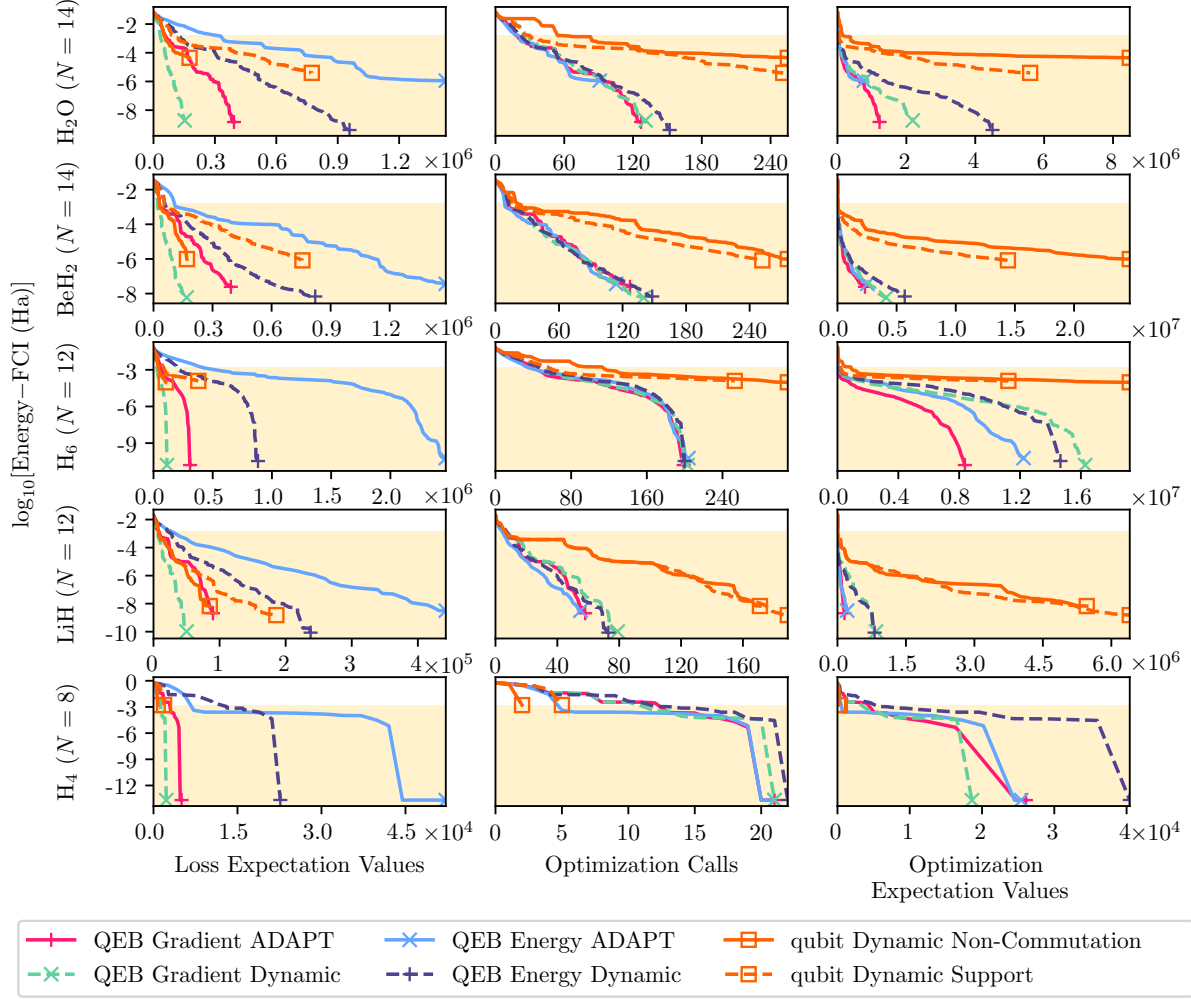


FIG. 20. Energy accuracy against the number of loss function calls (left), the number of times the *Ansatz* is optimized (center), and the number of expectation values calculated during optimizer calls (right), for qubit-Dynamic-ADAPT-VQE using support and operator commutation, as well as QEB standard and Dynamic-ADAPT-VQE using support commutation with both the steepest gradient and largest energy reduction decision rules, are compared. Each row shows data for a specific molecule, with the number of orbitals increasing up the page. Energy accuracies better than chemical accuracy are shaded in cream.

of expectation values we see for Dynamic-ADAPT-VQE, the selection rule makes little difference for LiH, H₆, and BeH₂. That said, the energy selection rule never significantly outperforms the gradient selection rule, which justifies the use of the gradient selection rule throughout this article.

APPENDIX D: NOISE SUSCEPTIBILITY PEAKS

In this Appendix, we comment on the peaks that appear in the noise susceptibility data of Figs. 14 and 15. To verify that these apparent features are not numerical errors, we not only computed the noise susceptibility but we also performed full density-matrix simulations with depolarizing noise for a few points on the H₆ plots. We compute $\mathcal{E}_t(p=10^{-6})$ and

$\mathcal{E}_t(p=0)$ at the values of t located before the peak, at the peak, and after the peak in Fig. 15, respectively. Then, we used the finite-differences method to estimate χ_t^D . The corresponding data points are depicted with black crosses in Fig. 15, row 3, for H₆. These points are in perfect agreement with the noise-susceptibility data.

APPENDIX E: POOL DEFINITIONS

In this Appendix, we make more rigorous definitions of the pools referred to throughout this article. These definitions will prove useful in analyzing the runtime scalings of the algorithms.

Let \mathcal{V} be some set of even integers. In the body of this paper, we take $\mathcal{V} = \{2, 4\}$ for all three pool definitions.

Additionally, let

$$\mathcal{I}(\mathcal{V}) := \{(\vec{k}, \vec{l}) : \vec{k}, \vec{l} \in \mathbb{Z}_n^{q/2} \text{ such that } k_0 \leq k_i, l_i \text{ and all } k_i, l_i \text{ are distinct } \forall i \in [q/2] \quad \forall q \in \mathcal{V}\}. \quad (\text{E1})$$

Definition 8 (Generalized fermionic pool). All the distinct fermionic excitations that act on $q \in \mathcal{V}$ distinct qubits are given by

$$\mathcal{P}^f(N) := \{T_i^{\vec{k}} : (\vec{k}, \vec{l}) \in \mathcal{I}(\mathcal{V})\} \quad \text{where} \quad T_i^{\vec{k}} := \left[\prod_{i \in \vec{k}} a_i^\dagger \right] \left[\prod_{j \in \vec{l}} a_j \right] - \text{H.c.} \quad (\text{E2})$$

This is the fermionic pool over N qubits.

Definition 9 (Generalized QEB pool). All the distinct qubit excitations that act on $q \in \mathcal{V}$ distinct qubits are given by

$$\mathcal{P}^{\text{QEB}}(N) := \{T_i^{\vec{k}} : (\vec{k}, \vec{l}) \in \mathcal{I}(\mathcal{V})\} \quad \text{where} \quad T_i^{\vec{k}} := \left[\prod_{i \in \vec{k}} Q_i^\dagger \right] \left[\prod_{j \in \vec{l}} Q_j \right] - \text{H.c.} \quad (\text{E3})$$

This is the QEB pool over N qubits.

The cardinality of both the fermionic and QEB pools is

$$|\mathcal{P}^{f, \text{QEB}}(N)| = \sum_{\substack{q \in \mathcal{V} \\ :q \leq N}} \binom{q-1}{\frac{1}{2}q} \binom{N}{q} = \Theta(N^{\max \mathcal{V}}), \quad (\text{E4})$$

where the second choice is the number of sets of q distinct qubits of N , and the first is the number of permutations of these qubits that give distinct excitations.

Definition 10 (Generalized qubit pool). The qubit pool is the set of all the Pauli excitations with an odd number of Y gates in the generator that act on $q \in \mathcal{V}$ distinct qubits.

The cardinality of the qubit pool is

$$|\mathcal{P}^{\text{qubit}}(N)| = \sum_{\substack{q \in \mathcal{V} \\ :q \leq N}} 2^{q-1} \binom{N}{q} = \Theta(N^{\max \mathcal{V}}), \quad (\text{E5})$$

where the combinatorics follow as for the fermionic and QEB pools, but the first choice is replaced with the factor 2^{q-1} .

Lemma 1. For each $\mathcal{P} \in \{\mathcal{P}^{\text{Fermi}}, \mathcal{P}^{\text{QEB}}, \mathcal{P}^{\text{qubit}}\}$, there exists a finite constant V that will depend on the pool definition, such that $|\mathcal{P}(N)|$ is a logarithmically concave function of N for $N \geq V$.

Proof. Consider the polynomial

$$f_q(x) := \prod_{n=0}^{q-1} (x-n), \quad (\text{E6})$$

which has q integer roots: $[0, q-1]$. We note that the k th derivative can be expressed as

$$\frac{d^k f_q}{dx^k} = \sum_{\substack{\vec{\alpha} \in [0, q-1]^k \\ : \alpha_i \neq \alpha_j \forall i, j}} f_{q, \vec{\alpha}}(x), \quad \text{where} \quad f_{q, \vec{\alpha}}(x) \equiv \prod_{\substack{n \in [0, q-1] \\ : n \notin \vec{\alpha}}} (x-n). \quad (\text{E7})$$

Further, let

$$g_{q, \vec{\alpha}}(x) := \frac{f_q^2(x)}{\prod_{l \in \vec{\alpha}} (x-l)}. \quad (\text{E8})$$

Using this notation, we will consider the following products:

$$\frac{d^2 f_q}{dx^2} f_q(x) = \sum_{\substack{\vec{\alpha} \in [0, q-1]^2 \\ : \alpha_1 \neq \alpha_2}} f_{q, \vec{\alpha}}(x) f_q(x) = \sum_{\substack{\vec{\alpha} \in [0, q-1]^2 \\ : \alpha_1 \neq \alpha_2}} g_{q, \vec{\alpha}}(x) \quad (\text{E9})$$

and

$$\begin{aligned} \frac{df_q}{dx} \frac{df_q}{dx} &= \sum_{\vec{\alpha} \in [0, q-1]^2} f_{q, \alpha_1}(x) f_{q, \alpha_2}(x) \\ &= \sum_{\substack{\vec{\alpha} \in [0, q-1]^2 \\ : \alpha_1 \neq \alpha_2}} g_{q, \vec{\alpha}}(x) + \sum_{\alpha_1=0}^{q-1} g_{q, (\alpha_1, \alpha_1)}(x). \end{aligned} \quad (\text{E10})$$

Now note the first terms from each cancel in the difference of these products:

$$\frac{df_q}{dx} \frac{df_q}{dx} - \frac{d^2 f_q}{dx^2} f_q(x) = \sum_{\alpha_1=0}^{q-1} g_{q, (\alpha_1, \alpha_1)}(x), \quad (\text{E11})$$

which is non-negative for $x \geq q$.

Note $f_q(x)$ is logarithmically concave within some convex domain iff $\frac{d^2 f_q}{dx^2} f_q(x) \leq \left[\frac{df_q}{dx} \right]^2$ within the convex domain. Thus, $f_q(x)$ is logarithmically concave for $x \geq q$. As the discrete function $\binom{N}{q} = \frac{1}{q!} f_q(N)$ it must also be logarithmically concave for $N \geq q$. However, $|\mathcal{P}(N)|$ is a linear combination with non-negative coefficients of such functions with $q \in \mathcal{V}$. Therefore, there will be a finite constant depending on these coefficients for which the function $\binom{N}{\max \mathcal{V}}$ dominates sufficiently that $|\mathcal{P}(N)|$ is logarithmically concave. ■

APPENDIX F: GENERALIZED NONCOMMUTATION SETS

In this Appendix, we derive rules to determine whether two *Ansatz* elements from the same pool operator-commute. We consider the qubit pool (Appendix F1), QEB pool (Appendix F2), and fermionic pool (Appendix F3). All these *Ansatz* elements are Stones encoded unitaries, $e^{\theta T}$ for some skew-Hermitian generator T . Thus, two *Ansatz* elements operator-commute iff the corresponding generators commute. Below we derive the conditions under which the generators commute and hence the *Ansatz* elements operator-commute.

1. Pauli excitations

For Pauli excitations, the generators are simply the Pauli strings of length two and four with an odd number of Y operators. Thus, the Pauli strings operator-commute iff the tensor factors in the strings with mutual support differ in an even number of places.

2. Qubit excitations

First, we will consider a generalization of qubit excitation generators which will later prove useful for fermionic excitation generators. Consider the following definitions:

Definition 11 (Singleton matrix). Let $[M_{ij}]_{ab} := \delta_{ai}\delta_{bj}$ be a singleton matrix. Note that these matrices have the following properties:

- (1) $M_{kl}M_{\gamma\delta} \equiv \delta_{l\gamma}M_{k\delta}$,
- (2) $[M_{kl}, M_{\gamma\delta}] \equiv \delta_{l\gamma}M_{k\delta} - \delta_{k\delta}M_{\gamma l}$.

Definition 12 (Set of singleton matrices). Let $\mathcal{M}_l := \{M_{ij} : i, j \in [2^l]\} \subset \mathbb{R}^{2^l \times 2^l}$ for all $l \in \mathbb{N}_0$. Note that this family of sets has the following properties:

- (1) $\mathcal{M}_0 \equiv \{1\}$,
- (2) $\mathcal{M}_{l+m} \equiv \mathcal{M}_l \otimes \mathcal{M}_m \quad \forall l \geq 1$.

Representing the set of operators $\mathcal{Q}_l := \{Q, Q^\dagger\}^{\otimes l}$ in the computational basis is an injection $\mathcal{Q}_l \rightarrow \{M_{ij} \in \mathcal{M}_l : i \neq j\}$ for $l \geq 1$. Thus, we consider the following skew-Hermitian operator $T = M_{ij} + aM_{ji}$, which could represent a qubit excitation generator in the computational basis when $a = -1$.

Theorem 1 (Singleton matrix excitation commutation). Consider the following two operators acting on a tripartite vector space:

$$T_1 = \mathbb{1} \otimes M_{ij} \otimes M_{kl} + a_1 \mathbb{1} \otimes M_{ji} \otimes M_{lk}, \quad (\text{F1})$$

$$T_2 = M_{\alpha\beta} \otimes \mathbb{1} \otimes M_{\gamma\delta} + a_2 M_{\beta\alpha} \otimes \mathbb{1} \otimes M_{\delta\gamma}, \quad (\text{F2})$$

such that $i \neq j, k \neq l, \alpha \neq \beta$, and $\gamma \neq \delta$.

T_1 and T_2 commute if and only if any of the following three conditions hold:

- (1) T_1 and T_2 have disjoint support (i.e. $M_{kl} = M_{\gamma\delta} = 1$),
- (2) $T_1 \propto T_2$,
- (3) T_1 and T_2 have equivalent support and k, l, γ and δ are all distinct.

Proof. Condition (1) follows trivially, as operators with disjoint support always commute. Thus, henceforth, we will assume $M_{kl}, M_{\gamma\delta} \in \mathcal{M}_l$ for $l \geq 1$ (the compliment of Condition (1)). Now consider the product

$$T_1 T_2 = M_{ij} \otimes M_{\alpha\beta} \otimes (M_{kl} M_{\gamma\delta}) \quad (\text{F3})$$

$$+ a_1 M_{ji} \otimes M_{\alpha\beta} \otimes (M_{lk} M_{\gamma\delta}) \quad (\text{F4})$$

$$\begin{cases} k = \delta \text{ and } l = \gamma \text{ and } \alpha_1 \alpha_2 = 1, \\ k = \gamma \text{ and } l = \delta \text{ and } \alpha_1 = \alpha_2 \end{cases} \Rightarrow k = l = \gamma = \delta \text{ and } \alpha_1 = \alpha_2 = \pm 1. \quad (\text{F15})$$

However, we know $k \neq l$ and $\gamma \neq \delta$, so these conditions cannot apply simultaneously. Next, we try the combination

$$\begin{cases} k = \delta \text{ and } l = \gamma \text{ and } \alpha_1 \alpha_2 = 1, \\ k \neq \gamma \text{ and } l \neq \delta \end{cases} \Rightarrow k = \delta \neq l = \gamma \text{ and } \alpha_1 \alpha_2 = 1, \quad (\text{F16})$$

which is possible and corresponds to $T_1 \propto T_2$ (Condition (2)). Similarly,

$$\begin{cases} k \neq \delta \text{ and } l \neq \gamma, \\ k = \gamma \text{ and } l = \delta \text{ and } \alpha_1 = \alpha_2 \end{cases} \Rightarrow k = \gamma \neq l = \delta \text{ and } \alpha_1 = \alpha_2 \quad (\text{F17})$$

is possible and also corresponds to $T_1 \propto T_2$ (Condition (2)). Finally, consider the combination

$$\begin{cases} k \neq \delta \text{ and } l \neq \gamma, \\ k \neq \gamma \text{ and } l \neq \delta \end{cases} \Rightarrow \{k, l, \gamma, \delta\} \text{ are all distinct}, \quad (\text{F18})$$

which is Condition (3), where we have used $k \neq l$ and $\gamma \neq \delta$. ■

$$+ a_2 M_{ij} \otimes M_{\beta\alpha} \otimes (M_{kl} M_{\delta\gamma}) \quad (\text{F5})$$

$$+ a_1 a_2 M_{ji} \otimes M_{\beta\alpha} \otimes (M_{lk} M_{\delta\gamma}). \quad (\text{F6})$$

Thus, we find the commutator using Property 2 of Definition 11:

$$[T_1, T_2] = M_{ij} \otimes M_{\alpha\beta} \otimes (\delta_{l\gamma} M_{k\delta} - \delta_{k\delta} M_{\gamma l}) \quad (\text{F7})$$

$$+ a_1 M_{ji} \otimes M_{\alpha\beta} \otimes (\delta_{k\gamma} M_{l\delta} - \delta_{l\delta} M_{\gamma k}) \quad (\text{F8})$$

$$+ a_2 M_{ij} \otimes M_{\beta\alpha} \otimes (\delta_{l\delta} M_{k\gamma} - \delta_{k\gamma} M_{\delta l}) \quad (\text{F9})$$

$$+ a_1 a_2 M_{ji} \otimes M_{\beta\alpha} \otimes (\delta_{k\delta} M_{l\gamma} - \delta_{l\gamma} M_{\delta k}). \quad (\text{F10})$$

First, suppose the tensor factors in parentheses are nonzero and $M_{ij} \neq 1$. Line (F7) cannot cancel with Line (F8) or (F10) due to the first tensor factor: $M_{ji} \neq M_{ij}$. Further, Line (F7) cannot cancel with Line (F9) due to the bracketed tensor factor as $\gamma \neq \delta$. Thus, T_1 and T_2 do not commute if $M_{ij} \neq 1$ and the first condition is not met. By symmetry, the same argument can be applied if we suppose the tensor factors in parentheses are nonzero and $M_{\alpha\beta} \neq 1$. Therefore, if the first condition is not met, then we require $M_{ij} = M_{\alpha\beta} = 1$ in order for T_1 and T_2 to commute—that is, we require equivalent support.

Suppose now that T_1 and T_2 do have equivalent support. We can simplify the commutator to

$$[T_1, T_2] = \delta_{l\gamma} M_{k\delta} - \delta_{k\delta} M_{\gamma l} \quad (\text{F11})$$

$$+ a_1 (\delta_{k\gamma} M_{l\delta} - \delta_{l\delta} M_{\gamma k}) \quad (\text{F12})$$

$$+ a_2 (\delta_{l\delta} M_{k\gamma} - \delta_{k\gamma} M_{\delta l}) \quad (\text{F13})$$

$$+ a_1 a_2 (\delta_{k\delta} M_{l\gamma} - \delta_{l\gamma} M_{\delta k}). \quad (\text{F14})$$

By noting if $\delta_{ab} = 1$ and $a \neq c$, then $\delta_{cb} = 0$, and that M_{ab} is linearly independent from M_{cd} if $a \neq c$ or $b \neq d$, we can pair the lines as follows: $A := \{(\text{G11}), (\text{G14})\}$ and $B := \{(\text{G12}), (\text{G13})\}$ where no term in A can cancel with a term in B . For the terms in A to cancel, we require $k = \delta$ and $l = \gamma$ and $a_1 a_2 = 1$ as $k \neq l$ and $\gamma \neq \delta$, or $k \neq \delta$ and $l \neq \gamma$. Similarly, for terms in B to cancel, we require $k = \gamma$ and $l = \delta$ and $a_1 = a_2$ as $k \neq l$ and $\gamma \neq \delta$, or $k \neq \gamma$ and $l \neq \delta$.

Now we can try to combine these conditions. First, consider combining the conditions

Corollary 1.1. Now consider qubit excitation generators defined as follows:

$$T = G + aG^\dagger, \quad \text{where } G \in \mathcal{Q}_l. \quad (\text{F19})$$

If two such generators have equivalent support, then either Condition (2) or (3) must hold. Thus, two-qubit excitation generators commute iff they have disjoint or equivalent support.

3. Fermionic excitations

A fermionic excitation generator generalizes to an operator of the following form:

Definition 13 (Generalized fermionic excitation),

$$T = K + bK^\dagger, \quad \text{where } K := \prod_{i \in \vec{i}} a_i^{(x_i)}, \quad a_i^{(x_i)} := \begin{cases} a_i^\dagger, & x_i = 0, \\ a_i, & x_i = 1, \end{cases} \quad (\text{F20})$$

and x is a bit-string and \vec{i} is a tuple of unique orbital indices.

In the Jordan-Wigner encoding, $a_i = Q_i \otimes Z_i$, where $Z_i = \otimes_{j=1}^{i-1} Z_j$ is a Pauli string of Z operators. Now we will define the commutation product and use it to express the generalized fermionic excitation generators as the tensor product of a singleton matrix excitation generator and a Pauli string of Z operators:

Definition 14 (Commutation product). Let $(\bullet, \bullet) : \mathcal{H}^{\times 2} \times \mathcal{H}^{\times 2} \rightarrow \{0, \pm 1\}$ be a mapping from a pair of operators on the Hilbert space \mathcal{H} to $\{0, \pm 1\}$ such that

$$(A, B) := \begin{cases} 1, & [A, B] = 0, \\ -1, & AB + BA = 0, \\ 0 & \text{otherwise,} \end{cases} \quad (\text{F21})$$

which satisfies

- (1) $(A, A) = 1$,
- (2) $(A, B) = (B, A)$,
- (3) $(A, BC) = (A, B)(A, C) = (A, CB)$ if $(A, B)(A, C) \neq 0$,
- (4) $AB = (A, B)BA$ if $(A, B) = \pm 1$.

Using (3), let

$$\left(A, \prod_i B_i \right) \equiv (A, \{B_i\}_i), \quad (\text{F22})$$

if $(A, B_i) \neq 0$ for all i . Note that Q and Q^\dagger anticommute with Z [i.e., $(Q^{(\dagger)}, Z) = -1$].

Lemma 2 (Generalized fermionic excitation tensor factorization). Generalized fermionic excitation generators can be expressed as follows:

$$T = \pm T_s \otimes \tilde{Z}, \quad (\text{F23})$$

where T_s is a singleton matrix excitation generator and $\tilde{Z} \in \{\mathbb{1}, Z\}^{\otimes k}$ for some $k \in \mathbb{N}$.

Proof.

$$K := \prod_{i \in \vec{i}} a_i^{(x_i)} \quad (\text{F24})$$

$$= \prod_{i \in \vec{i}} [Q_i^{(x_i)} Z_i], \quad \text{where } Q_i^{(x_i)} := \begin{cases} Q_i^\dagger, & x_i = 0 \\ Q_i, & x_i = 1 \end{cases} \quad (\text{F25})$$

$$\begin{aligned} &= \left[\bigotimes_{i \in \vec{i}} Q_i^{(x_i)} \right] \cdot \left[\prod_{n=1}^{\dim \vec{i}} (\{Q_{i_m}^{(x_{i_m})}\}_{m=1}^{n-1}, Z_{i_n}) Z_{i_n} \right] \quad (\text{F26}) \\ &= \left[\prod_{n=1}^{\dim \vec{i}} (\{Q_{i_m}^{(x_{i_m})}\}_{m=1}^{n-1}, Z_{i_n}) \right] \cdot \left[\bigotimes_{i \in \vec{i}} Q_i^{(x_i)} \right] \cdot \left[\prod_{n=1}^{\dim \vec{i}} Z_{i_n} \right]. \quad (\text{F27}) \end{aligned}$$

Now note that

$$QZ = -Q; \quad Q^\dagger Z = Q^\dagger. \quad (\text{F28})$$

Thus, the mutual support of $[\bigotimes_{i \in \vec{i}} Q_i^{(x_i)}]$ and $[\prod_{n=1}^{\dim \vec{i}} Z_{i_n}]$ can be removed by replacing the Z operators within the mutual support with $\mathbb{1}$ and obtaining a factor of 1 or -1 . In this process, the factor $[\bigotimes_{i \in \vec{i}} Q_i^{(x_i)}]$ is unchanged and $[\prod_{n=1}^{\dim \vec{i}} Z_{i_n}] \mapsto \tilde{Z}$. Additionally,

$$\left[\prod_{n=1}^{\dim \vec{i}} (\{Q_{i_m}^{(x_{i_m})}\}_{m=1}^{n-1}, Z_{i_n}) \right] \in \{1, -1\}, \quad (\text{F29})$$

which can be combined with the factor of 1 or -1 from earlier to produce a factor ± 1 . Therefore, we find

$$K \equiv \pm \left[\bigotimes_{i \in \vec{i}} Q_i^{(x_i)} \right] \otimes \tilde{Z}; \quad K^\dagger \equiv \pm \left[\bigotimes_{i \in \vec{i}} Q_i^{(x_i)} \right]^\dagger \otimes \tilde{Z}. \quad (\text{F30})$$

Finally, we can substitute this into the form of T :

$$T = \pm \left(\left[\bigotimes_{i \in \vec{i}} Q_i^{(x_i)} \right] + b \left[\bigotimes_{i \in \vec{i}} Q_i^{(x_i)} \right]^\dagger \right) \otimes \tilde{Z}, \quad (\text{F31})$$

and note that the tensor factor in parenthesis is a singleton matrix excitation generator. ■

Moving forward with this form of generalized fermionic excitation generators allows us to use Theorem 1 to show the following commutation relations hold for generalized fermionic excitations:

Theorem 2 (Generalized fermionic excitation commutation). Two generalized fermionic excitations T_1 and T_2 commute iff any of the following conditions are satisfied:

- (1) T_{s1} and T_{s2} have disjoint support and $|\text{Supp}T_{s1} \cap \text{Supp}\tilde{Z}_2| + |\text{Supp}T_{s2} \cap \text{Supp}\tilde{Z}_1|$ is even,
- (2) $T_{s1} \propto T_{s2}$,
- (3) T_{s1} and T_{s2} have equivalent support and can be written as

$$T_{s1} \equiv M_{rs} + b_1 M_{sr}; \quad T_{s2} \equiv M_{tu} + b_2 M_{ut}. \quad (\text{F32})$$

where r, s, t , and u are all distinct.

Proof. Consider the product

$$T_1 T_2 = c_1 T_{s1} \otimes \tilde{Z}_1 \cdot c_2 T_{s2} \otimes \tilde{Z}_2 \quad (\text{F33})$$

$$= c_1 c_2 T_{s1} \otimes \tilde{Z}_1 \cdot T_{s2} \otimes \tilde{Z}_2 \quad (\text{F34})$$

$$= c_1 c_2 (\tilde{Z}_1, T_{s2}) T_{s1} T_{s2} \tilde{Z}_1 \tilde{Z}_2, \quad \text{where tensor products with the identity are implied} \quad (\text{F35})$$

$$= c_1 c_2 (\tilde{Z}_1, T_{s2}) T_{s2} T_{s1} \tilde{Z}_1 \tilde{Z}_2 + c_1 c_2 (\tilde{Z}_1, T_{s2}) [T_{s1}, T_{s2}] \tilde{Z}_1 \tilde{Z}_2 \quad (\text{F36})$$

$$= (\tilde{Z}_1, T_{s2}) (\tilde{Z}_2, T_{s1}) c_2 T_{s2} \otimes \tilde{Z}_2 \cdot c_1 T_{s1} \otimes \tilde{Z}_1 + c_1 c_2 (\tilde{Z}_1, T_{s2}) [T_{s1}, T_{s2}] \tilde{Z}_1 \tilde{Z}_2 \quad (\text{F37})$$

$$= (\tilde{Z}_1, T_{s2}) (\tilde{Z}_2, T_{s1}) T_2 T_1 + c_1 c_2 (\tilde{Z}_1, T_{s2}) [T_{s1}, T_{s2}] \tilde{Z}_1 \tilde{Z}_2, \quad (\text{F38})$$

where $c_1, c_2 \in \{1, -1\}$ and we have used the fact that $(\tilde{Z}_m, T_{sn}) \in \{1, -1\}$ for all $m, n \in \{1, 2\}$.

Therefore, we can write the commutator as

$$[T_1, T_2] = [(\tilde{Z}_1, T_{s2}) (\tilde{Z}_2, T_{s1}) - 1] T_2 T_1 + c_1 c_2 (\tilde{Z}_1, T_{s2}) [T_{s1}, T_{s2}] \tilde{Z}_1 \tilde{Z}_2. \quad (\text{F39})$$

The entries of T_{sn} are drawn from the set $\{0, 1\}$. Thus, if T_{s1} and T_{s2} do not commute, then $T_{s2} T_{s1}$ and $[T_{s1}, T_{s2}]$ are linearly independent, so for $[T_1, T_2] = 0$ we require both both terms to vanish:

$$[(\tilde{Z}_1, T_{s2}) (\tilde{Z}_2, T_{s1}) - 1] T_2 T_1 = 0, \text{ and} \quad (\text{F40})$$

$$c_1 c_2 (\tilde{Z}_1, T_{s2}) [T_{s1}, T_{s2}] \tilde{Z}_1 \tilde{Z}_2 = 0, \quad (\text{F41})$$

which simplifies to

$$[(\tilde{Z}_1, T_{s2}) (\tilde{Z}_2, T_{s1}) - 1] = 0 \quad \text{or} \quad T_2 T_1 = 0, \text{ and} \quad (\text{F42})$$

$$[T_{s1}, T_{s2}] = 0. \quad (\text{F43})$$

Thus, Condition (F43) requires we satisfy at least one of the three conditions from Theorem 1. Therefore, each condition in Theorem 2 corresponds to the condition with the same number in Theorem 1. However, Condition (F42) requires us to strengthen the conditions in Theorem 2.

First, consider when $T_2 T_1 = 0$. The \tilde{Z}_n factors will only produce phase factors and so $T_2 T_1 = 0 \iff T_{s2} T_{s1} = 0$. Using the identities

$$T_{s1} \equiv M_{rs} + b_1 M_{sr}, \quad T_{s2} \equiv M_{tu} + b_2 M_{ut}, \quad (\text{F44})$$

where $r \neq s$ and $t \neq u$, we find

$$0 = T_{s2} T_{s1} = \delta_{ur} M_{ts} + b_2 \delta_{tr} M_{us} + b_1 \delta_{us} M_{tr} + b_2 b_1 \delta_{ts} M_{ur}. \quad (\text{F45})$$

We note M_{ts}, M_{us}, M_{tr} , and M_{ur} are linearly independent as $r \neq s$ and $t \neq u$. Thus, $T_2 T_1 = 0$ iff:

$$u \neq r, \quad (\text{F46})$$

$$t \neq r, \quad (\text{F47})$$

$$u \neq s, \quad (\text{F48})$$

$$t \neq s. \quad (\text{F49})$$

Combining these conditions with $r \neq s$ and $t \neq u$ implies $T_2 T_1 = 0$ iff r, s, t , and u are all distinct. We already obtained this condition from Condition (F43), giving us Condition (3).

Finally, as Q and Q^\dagger anticommute with Z , then $[(\tilde{Z}_1, T_{s2}) (\tilde{Z}_2, T_{s1}) - 1] = 0$ occurs iff $|\text{Supp}T_{s1} \cap \text{Supp}\tilde{Z}_2| + |\text{Supp}T_{s2} \cap \text{Supp}\tilde{Z}_1|$ is even. As $T_{s1} \propto T_{s2}$ implies $|\text{Supp}T_{s1} \cap \text{Supp}\tilde{Z}_2| + |\text{Supp}T_{s2} \cap \text{Supp}\tilde{Z}_1|$ is even we need only modify Condition (1). \blacksquare

Corollary 2.1 (Fermionic excitation commutation). Now consider fermionic excitation generators defined as follows:

$$T = K - K^\dagger \quad \text{where } K := \prod_{i \in \vec{i}} a_i^{(x_i)}, \quad a_i^{(x_i)} := \begin{cases} a_i^\dagger, & x_i = 0, \\ a_i, & x_i = 1, \end{cases} \quad (\text{F50})$$

and x is a bit-string and \vec{i} is a tuple of unique orbital indices.

If, for two such generators, the sets of orbitals acted upon are equivalent, then either Condition (2) or (3) must hold. Thus, one can simplify the conditions for commutation to

- (1) The sets of orbitals acted upon are disjoint and $|\text{Supp}T_{s1} \cap \text{Supp}\tilde{Z}_2| + |\text{Supp}T_{s2} \cap \text{Supp}\tilde{Z}_1|$ is even,
- (2) The sets of orbitals acted upon are equivalent.

Corollary 2.2 (Electron conserving fermionic excitation commutation). Fermionic excitations that conserve electron number have even $\dim \vec{i}$, so if the sets of orbitals acted upon by two different fermionic excitations are disjoint, then $|\text{Supp}T_{s1} \cap \text{Supp}\tilde{Z}_2| + |\text{Supp}T_{s2} \cap \text{Supp}\tilde{Z}_1|$ must be even. Hence, we can simplify Condition (1) to the following: the set of orbitals acted upon are disjoint.

APPENDIX G: GENERALIZED NONCOMMUTING SET CARDINALITIES

In this Appendix, we derive the cardinalities of the generalized noncommuting sets for both the QEB pool (Appendix G 1) and the qubit pool (Appendix G 2) using operator and support commutation. Further, we will derive their asymptotic scalings.

To proceed, we will define

$$\mathcal{P}_q(N) := \{A \in \mathcal{P}(N) : |\text{Supp}A = q|\} \quad (\text{G1})$$

as the subpool of elements supported only by q qubits.

1. Generalized QEB pool

The cardinality of the support noncommuting set of $A \in \mathcal{P}_q(N)$ is independent of A and is given by

$$\begin{aligned} C_q^S(N) &\equiv |\mathcal{N}_S[\mathcal{P}(N), A \in \mathcal{P}_q(N)]| \\ &= \sum_{\substack{p \in \mathcal{V} \\ :p \leq N}} \sum_{a=1}^{\min\{p,q\}} \binom{p-1}{\frac{1}{2}p} \binom{q}{a} \binom{N-q}{p-a} - 1 \quad (\text{G2}) \\ &= \Theta(N^{\max \mathcal{V}-1}). \quad (\text{G3}) \end{aligned}$$

Here we sum over the possible numbers of overlapping qubits, a . The third choice is the number of sets of $p-a$ distinct qubits of $N-q$, the second is the number of sets of a distinct qubits of q , and the first is the number of permutations of these qubits that give distinct qubit excitations. The -1 removes the operator inducing the support noncommuting set.

Similarly, we can use the operator commutation rules for qubit excitations to show that

$$\begin{aligned} C_q^O(N) &\equiv |\mathcal{N}_O[\mathcal{P}(N), A \in \mathcal{P}_q(N)]| \\ &= \sum_{\substack{p \in \mathcal{V} \\ :p \leq N}} \sum_{a=1}^{\alpha(p,q)} \binom{p-1}{\frac{1}{2}p} \binom{q}{a} \binom{N-q}{p-a} \quad (\text{G4}) \end{aligned}$$

$$= \Theta(N^{\max \mathcal{V}-1}), \quad (\text{G5})$$

where

$$\alpha(p, q) := \begin{cases} \min\{p, q\}, & p \neq q, \\ q-1, & p = q. \end{cases} \quad (\text{G6})$$

The choices are the same as for the cardinality of the support noncommuting set. However, the summation excludes equivalent support. We note that the operator and support noncommuting sets are almost equivalent:

$$C_q^S(N) - C_q^O(N) = \binom{q-1}{\frac{1}{2}q} - 1 = \Theta(1). \quad (\text{G7})$$

Because $\mathcal{N}_O(\mathcal{P}, A) \subseteq \mathcal{N}_S(\mathcal{P}, A)$, then $C_q^S(N) - C_q^O(N)$ is number of elements by which the sets differ. For single excitations, this difference vanishes, and for double excitations it is two.

Note that because the support and operator noncommuting sets differ by $\Theta(1)$ elements, then an *Ansatz* circuit of mutually operator-commuting *Ansatz* elements will be, at most, exactly $\max_{q \in \mathcal{V}} \binom{q-1}{\frac{1}{2}q} = \Theta(1)$ times deeper than an *Ansatz* circuit of mutually support-commuting *Ansatz* elements.

2. Generalized qubit pool

Similarly, the support noncommuting sets for the qubit pool have cardinalities

$$\begin{aligned} C_q^S(N) &\equiv |\mathcal{N}_S[\mathcal{P}(N), A \in \mathcal{P}_q(N)]| \\ &= \sum_{\substack{p \in \mathcal{V} \\ :p \leq N}} \sum_{a=1}^{\min\{p,q\}} 2^{p-1} \binom{q}{a} \binom{N-q}{p-a} - 1 \quad (\text{G8}) \\ &= \Theta(N^{\max \mathcal{V}-1}). \quad (\text{G9}) \end{aligned}$$

Further, half of the *Ansatz* elements with partial support with $A \in \mathcal{P}_q(N)$ operator-commute with A . Additionally, those with equivalent support to A operator-commute with A . Thus,

$$\begin{aligned} C_q^O(N) &\equiv |\mathcal{N}_O[\mathcal{P}(N), A \in \mathcal{P}_q(N)]| \\ &= \sum_{\substack{p \in \mathcal{V} \\ :p \leq N}} \sum_{a=1}^{\alpha(p,q)} 2^{p-2} \binom{q}{a} \binom{N-q}{p-a} \quad (\text{G10}) \\ &= \Theta(N^{\max \mathcal{V}-1}). \quad (\text{G11}) \end{aligned}$$

Therefore, the support- and operator-noncommuting sets differ asymptotically in size by a perfect of two:

$$C_q^S(N) - C_q^O(N) = C_q^O(N) + 2^{q-1} - 1 = \Theta(N^{\max \mathcal{V}-1}). \quad (\text{G12})$$

This is the cardinality of the set of *Ansatz* elements that do not support-commute but do operator-commute with $A \in \mathcal{P}_q(N)$. Thus, an *Ansatz* circuit of mutually operator-commuting *Ansatz* elements can be, at most, $\Omega(N^{\max \mathcal{V}-1})$ times deeper than an *Ansatz* circuit of mutually support-commuting *Ansatz* elements. But as we can construct any *Ansatz* circuit of mutually operator-commuting *Ansatz* elements by splitting the

pool into $\Theta(N^{\max \nu-1})$ disjoint subsets of mutually support-noncommuting *Ansatz* elements and using each subset to construct a layer of the *Ansatz* circuit, then we can remove all the *Ansatz* elements that do not appear in the *Ansatz* circuit we originally wished to construct—this *Ansatz* circuit is $O(N^{\max \nu-1})$ layers deep. Thus, an *Ansatz* circuit of mutually operator-commuting *Ansatz* elements is at most $\Theta(N^{\max \nu-1})$ times deeper than an *Ansatz* circuit of mutually support-commuting *Ansatz* elements.

APPENDIX H: STATISTICAL ANALYSIS OF SEQUENCES OF NONCOMMUTING SETS

In this Appendix, we derive expressions for the expected cardinality of the subpool searched to find a local minimum by subpool exploration. This will allow us to derive the expected number of loss function evaluations per element appended for Explore- and Dynamic-ADAPT-VQE. First, we will consider how the subpools at each step of subpool exploration vary for support commutation in Appendix H 1. Next, we will extend this result to operator commutation for the QEB pool only in Appendix H 2. Using these results in Appendix H 3 and Appendix H 4, we will derive an upper bound for the expected number of loss function evaluations per element for Explore-ADAPT-VQE in terms of the number of local minima (see Property 1) in the pool

$$M := \left| \left\{ A \in \mathcal{P} : L(A) = \min_{B \in \mathcal{N}_G(\mathcal{P}, A)} L(B) \right\} \right|. \quad (\text{H1})$$

Finally, we will use this upper bound to also upper bound the expected number of loss function evaluations per element for Dynamic-ADAPT-VQE in Appendix H 5.

1. Support-based-commutation sequences

Within this subsection, we do not specify a pool as support commutation allows for a pool agnostic analysis. First, we will consider the case in which \mathcal{S}_0 is a singleton. Next, we will use this to upper bound the general case.

If \mathcal{S}_0 is a singleton, using the recursion relation in Eq. (37), then \mathcal{S}_{m+1} has no operators that support any of $\{A_i\}_{i=0}^{m-1}$. Thus, at step m the remaining pool $\mathcal{P} \setminus \mathcal{S}_{\leq m}$ will only have support

on $N_m := N - \sum_{i=0}^{m-1} \nu_m$ qubits, where

$$\nu_m := \begin{cases} |\text{Supp} A_m \cap (\cup_{i=0}^{m-1} \text{Supp} A_i)| \in [0, q_m - 1], & m \geq 1 \\ q_0, & m = 0 \end{cases} \quad (\text{H2})$$

is the cardinality of the mutual support of the previous *Ansatz* elements and A_m , with $q_m := |\text{Supp} A_m|$. Therefore,

$$|\mathcal{S}_{m+1}| \equiv |\mathcal{N}_S[\mathcal{P} \setminus \mathcal{S}_{\leq m}, A_m \in \mathcal{P}_{q_m}(N)]| \\ = C_{q_m - \nu_m}^S(N_m) + \min\{1, \nu_m\} \quad (\text{H3})$$

$$= \Theta(N_m^{\max \nu-1}). \quad (\text{H4})$$

If $|\mathcal{S}_0| \geq 1$, then we will have preevaluated some *Ansatz* elements in the pool, and so the equivalence is demoted to an inequality:

$$|\mathcal{S}_{m+1}| \equiv |\mathcal{N}_S[\mathcal{P} \setminus \mathcal{S}_{\leq m}, A_m \in \mathcal{P}_{q_m}(N)]| \\ \leq C_{q_m - \nu_m}^S(N_m) + \min\{1, \nu_m\} \quad (\text{H5})$$

$$= O(N_m^{\max \nu-1}). \quad (\text{H6})$$

2. Operator-based-commutation sequences

Here we consider the more complex case of operator commutation. We can no longer remain pool agnostic; we will only consider the QEB pool for ease. First, we note a property of the QEB pool that allows us to establish an approximate equivalence between operator and support commutation sequences for the QEB pool. Using this, we will modify the results for support commutation sequences to QEB operator-commutation sequences.

First, note that the QEB pool has the following property. Consider an element A in the operator noncommuting set of $B \in \mathcal{P}$. All the *Ansatz* elements that do not support-commute but do operator-commute with A form a subset of the operator-noncommuting set of $B \in \mathcal{P}$. That is,

$$A \in \mathcal{N}_O[\mathcal{P}, B] \iff \mathcal{N}_S[\mathcal{P}, A] \setminus \mathcal{N}_O[\mathcal{P}, A] \subset \mathcal{N}_O[\mathcal{P}, B]. \quad (\text{H7})$$

Thus, all *Ansatz* elements that support- and operator-commute with A_m will be in the set $\mathcal{S}_{\leq m}$ for $m \geq 1$ if $A_m \in \mathcal{S}_{\leq m}$. The condition of $A_m \in \mathcal{S}_{\leq m}$ is true in subpool exploration. If, however, the *Ansatz* elements that support- and operator-commute with A_0 are in \mathcal{S}_0 , then this is extended to $m \geq 0$. Thus

$$|\mathcal{S}_{m+1}| \equiv |\mathcal{N}_O[\mathcal{P} \setminus \mathcal{S}_{\leq m}, A_m \in \mathcal{P}_{q_m}(N) \cap (\mathcal{S}_{\leq m})]| \quad (\text{H8})$$

$$\leq \begin{cases} C_{q_m}^O(N_0), & m = 0, \\ C_{q_m - \nu_m}^S(N_1) + \binom{\text{Supp} A_0 - 1}{\frac{1}{2} \text{Supp} A_0} & \text{for } m = 1 \text{ if } \mathcal{N}_S[\mathcal{P}, A_0] \setminus \mathcal{N}_O[\mathcal{P}, A_0] \not\subseteq \mathcal{S}_0, \\ C_{q_m - \nu_m}^S(N_m) + 1 & \text{otherwise,} \end{cases} \quad (\text{H9})$$

with equality if $\mathcal{S}_0 \equiv \{A_0\}$ or $\{A_0\} \cup \mathcal{N}_S[\mathcal{P}, A_0] \setminus \mathcal{N}_O[\mathcal{P}, A_0]$. This means that after the first two steps, subpool exploration is the same, when using the QEB pool, independent of whether support and operator commutativity is used.

3. Statistical analysis of support-based-commutation sequences

This subsection outlines an upper bound for the probability of having terminated during subpool exploration after m steps under a reasonable assumption: subpool exploration is more effective than random sampling. With this framework, we will bound the expected number of steps and elements searched for Explore- and Dynamic-ADAPT-VQE in the subsequent subsections.

Suppose we are seeking a local minimum, as in subpool exploration, but instead, we generate the sequence

$$\mathcal{S}_{m+1} = \mathcal{N}_G(\mathcal{P} \setminus \mathcal{S}_{\leq m}, A_m) \subseteq \mathcal{N}_G(\mathcal{P}, A_m) \quad \forall m \geq 0, \quad (\text{H10})$$

using A_m drawn from a distribution independent of $\{A_l\}_{l=0}^{m-1}$. Suppose further that the set \mathcal{M} of M local minima is distributed randomly in the pool. Let \tilde{m} be the random variable for the number of steps taken such that the minimum *Ansatz* element in $\mathcal{S}_{\leq \tilde{m}-1}$ is a local minimum. Let the probability of $\tilde{m} \leq m$ given a pool \mathcal{P} over N qubits and the sequences \vec{q} and \vec{v} be denoted by $\mathbb{P}(\tilde{m} \leq m | N, \vec{q}, \vec{v})$. As $\mathcal{M} \cap \mathcal{S}_{\leq m-1} = \emptyset \Rightarrow \tilde{m} > m$, then $\mathbb{P}(\tilde{m} > m | N, \vec{q}, \vec{v}) \leq \mathbb{P}(\mathcal{M} \cap \mathcal{S}_{\leq m-1} = \emptyset | N, \vec{q}, \vec{v})$. That is, $\mathbb{P}(\tilde{m} \leq m | N, \vec{q}, \vec{v}) \geq 1 - \mathbb{P}(\mathcal{M} \cap \mathcal{S}_{\leq m-1} = \emptyset | N, \vec{q}, \vec{v})$.

We can calculate $\mathbb{P}(\mathcal{M} \cap \mathcal{S}_{\leq m-1} = \emptyset | N, \vec{q}, \vec{v})$ by fixing the subset $\mathcal{S}_{\leq m-1} \subset \mathcal{P}$ and then sequentially placing the local minima randomly in the pool. The probability that the first local minimum is not placed in $\mathcal{S}_{\leq m-1}$ is $(|\mathcal{P}| - |\mathcal{S}_{\leq m-1}|)/|\mathcal{P}|$. However, note that a local minimum cannot lie within the generalized noncommuting set of another local minimum. Thus, we must remove the *Ansatz* elements of the generalized noncommuting set from both the \mathcal{P} and $\mathcal{S}_{\leq m-1}$ before the next step. Thus, the probability of not placing the i th local minimum in $\mathcal{S}_{\leq m-1}$ conditional on all previous local minima not being placed in $\mathcal{S}_{\leq m-1}$ and the cardinality of the support of the j th local minimum being μ_j is given by

$$\frac{|\mathcal{P}(N_{m-1} - \sum_{j=1}^{i-1} \mu_j)|}{|\mathcal{P}(N - \sum_{j=1}^{i-1} \mu_j)|} \quad (\text{H11})$$

for support commutation. Here we have assumed our initial subpool is a singleton. If the initial subpool is not a singleton, then we will have preevaluated some *Ansatz* elements in future subpools, and so the numerator will decrease. Therefore, this probability constitutes an upper bound. Further, we can bound this probability as follows:

Lemma 3. There exists a finite constant V , such that if $N_{m-1} - \sum_{j=1}^{i-1} \mu_j \geq V$, then

$$\frac{|\mathcal{P}(N_m - \sum_{j=1}^{i-1} \mu_j)|}{|\mathcal{P}(N - \sum_{j=1}^{i-1} \mu_j)|} \leq \frac{|\mathcal{P}(N_m)|}{|\mathcal{P}(N)|}. \quad (\text{H12})$$

Proof. Consider the ratio $r := \frac{f(x-a)}{f(x)}$ where a is non-negative. Now consider when the derivative is non-negative:

$$\frac{dr}{dx} \equiv \frac{f'(x-a)}{f(x)} - \frac{f(x-a)f'(x)}{f^2(x)} \geq 0 \quad (\text{H13})$$

$$\iff \frac{d}{dx} \ln[f(x-a)] \geq \frac{d}{dx} \ln[f(x)]. \quad (\text{H14})$$

This is true iff $f(x)$ is logarithmically concave. Note by Lemma 1 there exists some finite constant V such that $|\mathcal{P}(N)|$ is logarithmically concave. ■

Putting these results together, we can bound $\mathbb{P}(\mathcal{M} \cap \mathcal{S}_{\leq m} = \emptyset | N, \vec{q}, \vec{v}, \vec{\mu})$ for m small enough to satisfy $N_m - \sum_{j=1}^{M-1} \mu_j \geq V$ and then use a looser bound for the remaining terms:

$$\begin{aligned} & \mathbb{P}(\mathcal{M} \cap \mathcal{S}_{\leq m} = \emptyset | N, \vec{q}, \vec{v}, \vec{\mu}) \\ & \leq \prod_{i=1}^M \frac{|\mathcal{P}(N_m - \sum_{j=1}^{i-1} \mu_j)|}{|\mathcal{P}(N - \sum_{j=1}^{i-1} \mu_j)|} \end{aligned} \quad (\text{H15})$$

$$\leq \begin{cases} \prod_{i=1}^M \frac{|\mathcal{P}(N_m)|}{|\mathcal{P}(N)|}, & m \leq m'_{\max}, \\ \prod_{i=1}^M \frac{|\mathcal{P}(N_{m'_{\max}})|}{|\mathcal{P}(N)|}, & m > m'_{\max}, \end{cases} \quad (\text{H16})$$

$$= \begin{cases} \left[\frac{|\mathcal{P}(N_m)|}{|\mathcal{P}(N)|} \right]^M, & m \leq m'_{\max}, \\ \left[\frac{|\mathcal{P}(N_{m'_{\max}})|}{|\mathcal{P}(N)|} \right]^M, & m > m'_{\max}, \end{cases} \quad (\text{H17})$$

where m'_{\max} is the largest m that satisfies $N_m - \sum_{j=1}^{M-1} \mu_j \geq V$. Similarly, m_s is the largest m that satisfies $N_m - \|\vec{\mu}\|_1 \geq 0$. Thus, there will at most be $V - \min \mathcal{V} \geq m_s - m'_{\max}$ terms for which m is too large to satisfy $N_m - \sum_{j=1}^{M-1} \mu_j \geq V$ and we will find these terms will asymptotically yield second-order contributions to our quantities of interest.

As the final upper bound is independent of $\vec{\mu}$ and \vec{q} , then we can marginalize over $\vec{\mu}$ and \vec{q} to get

$$\mathbb{P}(\mathcal{M} \cap \mathcal{S}_{\leq m} = \emptyset | N, \vec{v}) \leq \begin{cases} \left[\frac{|\mathcal{P}(N_m)|}{|\mathcal{P}(N)|} \right]^M, & m \leq m'_{\max}, \\ \left[\frac{|\mathcal{P}(N_{m'_{\max}})|}{|\mathcal{P}(N)|} \right]^M, & m > m'_{\max}, \end{cases} \quad (\text{H18})$$

and finally lower bound the probability of interest:

$$\mathbb{P}(\tilde{m} \leq m | N, \vec{v}) \geq \begin{cases} 1 - \left[\frac{|\mathcal{P}(N_m)|}{|\mathcal{P}(N)|} \right]^M, & m \leq m'_{\max}, \\ 1 - \left[\frac{|\mathcal{P}(N_{m'_{\max}})|}{|\mathcal{P}(N)|} \right]^M, & m > m'_{\max}. \end{cases} \quad (\text{H19})$$

Finally, we note the asymptotic scaling:

$$\left[\frac{|\mathcal{P}(N_m)|}{|\mathcal{P}(N')|} \right]^M \sim \left[\frac{N_m}{N} \right]^{M \max \mathcal{V}}, \quad (\text{H20})$$

which does not necessarily hold for $m > m'_{\max}$, but we no longer have any terms of this form.

However, in our proposed pool exploration strategy, the A_m are drawn from a dependent distribution. We take A_m as the *Ansatz* element with the minimum loss in $\mathcal{S}_{\leq m}$. Assuming this dependent sequence is a better exploration strategy than independent random sampling, then this strategy should drive the sequence to a local minima faster than the independent proposal strategy. Thus, one would expect the cumulative probability distribution of termination to shift to larger

probabilities:

$$\mathbb{P}(\tilde{m} \leq m|N, \vec{v}) \leq \mathbb{P}(\hat{m} \leq m|N, \vec{v}), \quad (\text{H21})$$

where \hat{m} is the random variable for the number of steps taken to find a local minimum using this dependent sequence. That is, \hat{m} is the random variable for the number of steps taken such that the minimum *Ansatz* element in $\mathcal{S}_{\leq \hat{m}-1}$ is a local minimum.

We can write the mean of a monotonically increasing function f with respect to \tilde{m} as

$$\mathbb{E}(f(\tilde{m})|N, \vec{v}) \equiv \sum_{m=0}^{m_s} f(m) \mathbb{P}(\tilde{m} = m|N, \vec{v}) \quad (\text{H22})$$

$$\begin{aligned} &= f(0) + \sum_{m=1}^{m_s} [f(m) - f(m-1)] \\ &\quad \times \mathbb{P}(\tilde{m} \geq m|N, \vec{v}) \end{aligned} \quad (\text{H23})$$

$$\begin{aligned} &= f(0) + \sum_{m=0}^{m_s-1} [f(m+1) - f(m)] \\ &\quad \times [1 - \mathbb{P}(\tilde{m} \leq m|N, \vec{v})] \end{aligned} \quad (\text{H24})$$

$$\begin{aligned} &= f(m_s) - \sum_{m=0}^{m_s-1} [f(m+1) - f(m)] \\ &\quad \times \mathbb{P}(\tilde{m} \leq m|N, \vec{v}), \end{aligned} \quad (\text{H25})$$

and similarly for \hat{m} . Using this, we can lower bound the expected values:

$$\mathbb{E}(\tilde{m}|N, \vec{v}) \geq \mathbb{E}(\hat{m}|N, \vec{v}), \quad (\text{H26})$$

$$\mathbb{E}(|\mathcal{S}_{\leq \tilde{m}}||N, \vec{v}) \geq \mathbb{E}(|\mathcal{S}_{\leq \hat{m}}||N, \vec{v}). \quad (\text{H27})$$

Next, we proceed by considering specifically Explore-ADAPT-VQE.

4. Statistical analysis of Explore-ADAPT-VQE

Now we substitute for the bounding cumulative probability distribution to obtain the asymptotic scaling for the mean:

$$\mathbb{E}(|\mathcal{S}_{\leq \tilde{m}}||N, \vec{v}) = \sum_{m=0}^{m_s} |\mathcal{S}_{\leq m}| \mathbb{P}(\tilde{m} = m|N, \vec{v}) \quad (\text{H28})$$

$$= |\mathcal{S}_0| + \sum_{m=1}^{m_s} |\mathcal{S}_m| \mathbb{P}(\tilde{m} \geq m|N, \vec{v}) \quad (\text{H29})$$

$$= |\mathcal{S}_0| + \sum_{m=0}^{m_s-1} |\mathcal{S}_{m+1}| \mathbb{P}(\tilde{m} > m|N, \vec{v}) \quad (\text{H30})$$

$$\leq |\mathcal{S}_0| + \sum_{m=0}^{m'_{\max}-1} |\mathcal{S}_{m+1}| \left[\frac{|\mathcal{P}(N_m)|}{|\mathcal{P}(N)|} \right]^M + \sum_{m=m'_{\max}}^{m_s-1} |\mathcal{S}_{m+1}| \left[\frac{|\mathcal{P}(N_{m'_{\max}})|}{|\mathcal{P}(N)|} \right]^M \quad (\text{H31})$$

$$= |\mathcal{S}_0| + \Theta \left[\sum_{m=0}^{m'_{\max}-1} N_m^{\max \mathcal{V}-1} \left(\frac{N_m}{N} \right)^{M \max \mathcal{V}} + \sum_{m=m'_{\max}}^{m_s-1} N_m^{\max \mathcal{V}-1} \left(\frac{N_{m'_{\max}}}{N} \right)^{M \max \mathcal{V}} \right], \quad (\text{H32})$$

where Eqs. (H28) and (H29) follow from Eqs. (H22) and (H23), and in Eqs. (H31) and (H32) we have substituted in the inequality given in Eq. (H19) and then the asymptotics in Eqs. (H20) and (E4) or (E5). However, $N_m = O(1)$ for $m \geq m'_{\max}$, so both $\sum_{m=m'_{\max}}^{m_s-1} N_m^{\max \mathcal{V}-1} \left(\frac{N_{m'_{\max}}}{N} \right)^{M \max \mathcal{V}} = O(N^{-M \max \mathcal{V}})$ and $\sum_{m=m'_{\max}}^{m_s-1} N_m^{\max \mathcal{V}-1} \left(\frac{N_m}{N} \right)^{M \max \mathcal{V}} = O(N^{-M \max \mathcal{V}})$. On the other hand, the first term in the first summation of Eq. (H32) is $\Theta(N^{\max \mathcal{V}-1})$. Therefore, assuming $\max \mathcal{V} > 1$, we can neglect the second summation and extend the first summation from m'_{\max} to m_s , to leading order:

$$\mathbb{E}(|\mathcal{S}_{\leq \tilde{m}}||N, \vec{v}) = |\mathcal{S}_0| + O \left[\sum_{m=0}^{m_s-1} N_m^{\max \mathcal{V}-1} \left(\frac{N_m}{N} \right)^{M \max \mathcal{V}} \right]. \quad (\text{H33})$$

Now consider the second term:

$$\sum_{m=0}^{m_s-1} N_m^{\max \mathcal{V}-1} \left(\frac{N_m}{N} \right)^{M \max \mathcal{V}} = N^{\max \mathcal{V}-1} \sum_{m=0}^{m_s-1} \left(1 - \frac{1}{N} \sum_{i=0}^{m-1} \nu_i \right)^{(M+1) \max \mathcal{V}-1}. \quad (\text{H34})$$

Note that we can bound $\frac{1}{N} \sum_{l=0}^{m-1} \nu_l$ as follows:

$$\frac{1}{N} \sum_{l=0}^{m-1} \nu_l \geq \frac{m}{N} = a \frac{m}{m_s}, \quad \text{where } a \equiv \frac{m_s}{N} = \Theta(1). \quad (\text{H35})$$

Note $a \in [0, 1]$, and using this we bound Eq. (H34) with

$$N^{\max \mathcal{V}-1} \sum_{m=0}^{m_s-1} \left(1 - \frac{1}{N} \sum_{j=0}^{m-1} v_j \right)^{(M+1) \max \mathcal{V}-1} \leq N^{\max \mathcal{V}-1} \sum_{m=0}^{m_s-1} \left(1 - a \frac{m}{m_s} \right)^{(M+1) \max \mathcal{V}-1} \quad (\text{H36})$$

$$\sim N^{\max \mathcal{V}-1} m_s \int_0^1 (1 - ax)^{(M+1) \max \mathcal{V}-1} dx \equiv I. \quad (\text{H37})$$

Next, using the substitution $y = 1 - ax$ we can evaluate the integral:

$$I = N^{\max \mathcal{V}-1} \frac{m_s}{a} \int_{1-a}^1 y^{(M+1) \max \mathcal{V}-1} dy \quad (\text{H38})$$

$$= \frac{N^{\max \mathcal{V}-1} m_s}{a(M+1) \max \mathcal{V}} [1 - (1-a)^{(M+1) \max \mathcal{V}}]. \quad (\text{H39})$$

Indeed, M can depend on N , and so it will have an asymptotic scaling $M = \Omega(1), O(N)$. Thus, the asymptotic scaling of the mean is

$$\mathbb{E}(|\mathcal{S}_{\leq \hat{m}}| | N, \vec{v}) = |\mathcal{S}_0| + O\left(\frac{N^{\max \mathcal{V}}}{M}\right) \quad (\text{H40})$$

as $m_s = O(N)$.

Following a similar method, one can show

$$\mathbb{E}(\hat{m} | N, \vec{v}) = O\left(\frac{N}{M}\right). \quad (\text{H41})$$

Further, inequalities (H26) and (H27) yield

$$\mathbb{E}(|\mathcal{S}_{\leq \hat{m}}| | N, \vec{v}) = O\left(|\mathcal{S}_0| + \frac{N^{\max \mathcal{V}}}{M}\right), \quad (\text{H42})$$

$$\mathbb{E}(\hat{m} | N, \vec{v}) = O\left(\frac{N}{M}\right). \quad (\text{H43})$$

As these scalings are independent of \vec{v} , then conditioning our probabilities and means with some prior distribution for \vec{v} will leave the scalings invariant:

$$\mathbb{E}(|\mathcal{S}_{\leq \hat{m}}| | N) = O\left(|\mathcal{S}_0| + \frac{N^{\max \mathcal{V}}}{M}\right), \quad (\text{H44})$$

$$\mathbb{E}(\hat{m} | N) = O\left(\frac{N}{M}\right). \quad (\text{H45})$$

5. Statistical analysis of Dynamic-ADAPT-VQE

Finally, when we additionally apply layering, we have the complication that the pool size reduces throughout the layer. Thus, the upper bound on the means becomes

$$\begin{aligned} \mathbb{E}(|\mathcal{S}_{\leq \hat{m}}| | N) &\leq \sum_{t=1}^{t_{\max}} \sum_{\substack{\vec{q} \in \mathcal{V}^t \\ \|\vec{q}\|_1 \leq N}} \mathbb{E}(|\mathcal{S}_{\leq \hat{m}}| | N - \|\vec{q}\|_1) \mathbb{P}(\vec{q}) \\ &= O\left(|\mathcal{S}_0| + \frac{N^{\max \mathcal{V}}}{M}\right), \end{aligned} \quad (\text{H46})$$

$$\mathbb{E}(\hat{m} | N) \leq \sum_{t=1}^{t_{\max}} \sum_{\substack{\vec{q} \in \mathcal{V}^t \\ \|\vec{q}\|_1 \leq N}} \mathbb{E}(\hat{m} | N - \|\vec{q}\|_1) \mathbb{P}(\vec{q}) = O\left(\frac{N}{M}\right), \quad (\text{H47})$$

where $\mathbb{P}(\vec{q})$ is the frequency at which the algorithm had already placed $\dim \vec{q}$ gates in a layer each acting on q_i qubits prior to a given iteration. $\mathbb{P}(\vec{q})$ will decay with $\dim \vec{q}$ due to early termination or use of larger *Ansatz* elements.

APPENDIX I: GENERALIZED COMMUTATIVITY

We can generalize the notion of commutativity used in the body of this article as follows:

Definition 15 (Generalized commutativity). $\mathcal{N}_G(\mathcal{P}, A)$ and its complement define valid notions of generalized noncommutation and generalized commutation over \mathcal{P} , respectively, if

$$\mathcal{N}_O(\mathcal{P}, A) \subseteq \mathcal{N}_G(\mathcal{P}, A) \quad \forall A \in \mathcal{P}. \quad (\text{I1})$$

In particular, note that Property 3 holds for generalized commutativity. This is because Eq. (I1) ensures that the ordering of *Ansatz* elements found by Algorithm 2 does not matter.

Therefore, if the shallowness of circuits is of less importance, then subsets of the support noncommuting set can be used:

$$\mathcal{N}_O(\mathcal{P}, A) \subseteq \mathcal{N}_G(\mathcal{P}, A) \subseteq \mathcal{N}_S(\mathcal{P}, A) \quad \forall A \in \mathcal{P}. \quad (\text{I2})$$

Alternatively, in physical systems, one can imagine crosstalk from simultaneous qubit operations being problematic, and so a more generous notion of support, including some padding, may be beneficial:

$$\mathcal{N}_O(\mathcal{P}, A) \subseteq \mathcal{N}_S(\mathcal{P}, A) \subseteq \mathcal{N}_G(\mathcal{P}, A) \quad \forall A \in \mathcal{P}. \quad (\text{I3})$$

APPENDIX J: NOISE MODELS

In this Appendix, we document the details of our noise models.

1. Noise channels acting on a single qubit

This subsection describes how noise acts on a single qubit. *Amplitude damping* noise acts on the density matrix of a single-qubit as

$$F(\gamma) \left[\begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix} \right] := \begin{pmatrix} \rho_{00} + \gamma \rho_{11} & \sqrt{1-\gamma} \rho_{01} \\ \sqrt{1-\gamma} \rho_{10} & (1-\gamma) \rho_{11} \end{pmatrix}. \quad (\text{J1})$$

Here, the decay constant γ is given by

$$\gamma(\omega_1, \tau) := 1 - e^{-\omega_1 \tau}, \quad (\text{J2})$$

where τ is the duration for which the qubit is exposed to amplitude damping, and $\omega_1 := T_1^{-1}$ is determined by the T_1 time. Below, we denote amplitude damping of the r th qubit of a density matrix ρ as $\mathcal{F}(\gamma(\omega_1, \tau), r)[\rho]$.

Dephasing noise on a single qubit r of a density matrix ρ is modeled using the following noise channel:

$$\mathcal{C}(p_z, r)[\rho] := (1 - p_z)\rho + p_z \mathcal{Z}(r)[\rho]. \quad (\text{J3})$$

Here, $\mathcal{Z}(r)$ denotes the channel induced by the Pauli-Z gate acting on the r th qubit. The phase-flip probability is given by

$$p_z(\omega_z, \tau) = \frac{1}{2}(1 - e^{-\omega_z \tau}), \quad (\text{J4})$$

where τ is again the duration for which the qubit is dephasing. The decay constant ω_z , on the other hand, is set by the T_1 and T_2^* times:

$$\omega_z = \frac{1}{T_2^*} - \frac{1}{2T_1} = \omega_2^* - \frac{1}{2}\omega_1. \quad (\text{J5})$$

Depolarizing noise on a single qubit r is modeled using the channel

$$\mathcal{D}(p, r)[\rho] := (1 - p)\rho + \frac{p}{3} \sum_{\mathcal{P} \in \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}} \mathcal{P}(r)[\rho]. \quad (\text{J6})$$

Here, $p \in [0, 1]$ is the polarization probability, while $\mathcal{P}(r)$ are the $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ channels induced by the corresponding Pauli gates, acting on qubit r .

2. Noise channels acting on an *Ansatz*-element layer

Next, we explain how the noise channels acting on a single qubit are used in the noisy simulation of an *Ansatz* circuit Λ_t . To begin with, we decompose the *Ansatz* circuit Λ_t into

$l = 1, \dots, \ell_t$ layers of support-commuting *Ansatz*-element layers $\{\mathcal{A}_l\}$ as $\Lambda_t = \mathcal{A}_{\ell_t}^o \circ \dots \circ \mathcal{A}_1^o$, Eq. (61). For amplitude damping and dephasing noise, each *Ansatz*-element layer \mathcal{A}_l^o is transpiled into columns of native gates that can be implemented in parallel. The native gate with the longest execution time of each native-gate column sets the column execution time. The sum of the column execution times then gives the execution time τ_l of the *Ansatz*-element layer \mathcal{A}_l^o . After each *Ansatz*-element layer \mathcal{A}_l^o amplitude damping is implemented by applying an amplitude-damping channel to every qubit $r = 1, \dots, N$ in an amplitude-damping layer:

$$\bar{\mathcal{F}}(\omega_1, \tau_l) = \bigotimes_{r=1}^N \mathcal{F}(\gamma(\omega_1, \tau_l), r). \quad (\text{J7})$$

This results in the amplitude-damped *Ansatz* circuit

$$\Lambda_t(\omega_1) = \bar{\mathcal{F}}(\omega_1, \tau_{\ell_t}) \circ \mathcal{A}_{\ell_t}^o \circ \dots \circ \bar{\mathcal{F}}(\omega_1, \tau_1) \circ \mathcal{A}_1^o. \quad (\text{J8})$$

Similarly, after each *Ansatz*-element layer \mathcal{A}_l^o , dephasing is implemented by applying a dephasing channel to every qubit $r = 1, \dots, N$ in a dephasing layer:

$$\bar{\mathcal{C}}(\omega_z, \tau_l) = \bigotimes_{r=1}^N \mathcal{C}(p_z(\omega_z, \tau_l), r). \quad (\text{J9})$$

This results in the dephased *Ansatz* circuit

$$\Lambda_t(\omega_z) = \bar{\mathcal{C}}(\omega_z, \tau_{\ell_t}) \circ \mathcal{A}_{\ell_t}^o \circ \dots \circ \bar{\mathcal{C}}(\omega_z, \tau_1) \circ \mathcal{A}_1^o. \quad (\text{J10})$$

Finally, for depolarizing noise, we first apply the whole *Ansatz*-element layer. We then apply a depolarizing channel to each qubit, the exact number of times that the qubit was a target qubit of a CNOT gate in the preceding layer. More specifically, defining $M_{l,r}$ to be the number of times the r th qubit was the target qubit of a CNOT gate in the l th layer, we model the effect of depolarization after the layer \mathcal{A}_l^o by

$$\bar{\mathcal{D}}(p, l) = \bigotimes_{r=1}^N \left(\prod_{i=1}^{M_{l,r}} \mathcal{D}(p, r) \right). \quad (\text{J11})$$

This results in the depolarized *Ansatz* circuit

$$\Lambda_t(p) = \bar{\mathcal{D}}(p, \ell_t) \circ \mathcal{A}_{\ell_t}^o \circ \dots \circ \bar{\mathcal{D}}(p, 1) \circ \mathcal{A}_1^o. \quad (\text{J12})$$

APPENDIX K: DERIVATION OF NOISE SUSCEPTIBILITY RELATIONS

In this Appendix, we derive the noise susceptibility expressions of Eqs. (67). We start from the definition of noise susceptibility in Eq. (66) and differentiate the amplitude-damped *Ansatz* circuit $\Lambda_t(\omega_1)$ Eq. (J8), the dephased *Ansatz* circuit $\Lambda_t(\omega_z)$ Eq. (J10), and the depolarized *Ansatz* circuit $\Lambda_t(p)$ Eq. (J12), with respect to the noise parameter ω_1 , ω_z , and p , respectively. This results in the following expressions:

$$\left. \frac{\partial \Lambda_t(\omega_1)}{\partial \omega_1} \right|_{\omega_1=0} = \sum_{l=1}^{\ell_t} \sum_{r=1}^N \mathcal{A}_{\ell_t}^o \circ \dots \circ \left(\left. \frac{\partial \mathcal{F}(\gamma, r)}{\partial \gamma} \right|_{\gamma=0} \times \tau_l \right) \circ \mathcal{A}_l^o \circ \dots \circ \mathcal{A}_1^o, \quad (\text{K1a})$$

$$\left. \frac{\partial \Lambda_t(\omega_z)}{\partial \omega_z} \right|_{\omega_z=0} = \sum_{l=1}^{\ell_t} \sum_{r=1}^N \mathcal{A}_{\ell_t}^o \circ \dots \circ \left(\left. \frac{\partial \mathcal{C}(p_z, r)}{\partial p_z} \right|_{p_z=0} \times \frac{\tau_l}{2} \right) \circ \mathcal{A}_l^o \circ \dots \circ \mathcal{A}_1^o, \quad (\text{K1b})$$

$$\left. \frac{\partial \Lambda_t(p)}{\partial p} \right|_{p=0} = \sum_{l=1}^{\ell_t} \sum_{r=1}^N M_{l,r} \mathcal{A}_{\ell_t}^{\circ} \circ \cdots \circ \left(\left. \frac{\partial \mathcal{D}(p, r)}{\partial p} \right|_{p=0} \right) \circ \mathcal{A}_l^{\circ} \circ \cdots \circ \mathcal{A}_1^{\circ}. \quad (\text{K1c})$$

Next, we compute the derivatives of the individual channels, given as

$$\left. \frac{\partial \mathcal{F}(\gamma, r)}{\partial \gamma} \right|_{\gamma=0} = d\mathcal{F}(r) - \mathbb{1}, \quad (\text{K2a})$$

$$\left. \frac{\partial \mathcal{C}(p_z, r)}{\partial p_z} \right|_{p_z=0} = \mathcal{Z}(r) - \mathbb{1}, \quad (\text{K2b})$$

$$\left. \frac{\partial \mathcal{D}(p, r)}{\partial p} \right|_{p=0} = \frac{1}{3}[\mathcal{X}(r) + \mathcal{Y}(r) + \mathcal{Z}(r)] - \mathbb{1}. \quad (\text{K2c})$$

Here we use the linear, but the no longer positive nor trace-preserving, map

$$d\mathcal{F}(r) = \mathcal{F}\left(\frac{3}{4}, r\right) + \frac{1}{4}\mathcal{R}(r), \quad (\text{K3})$$

with the residual linear map

$$\mathcal{R}\left[\begin{pmatrix} \rho_{00} & \rho_{01} \\ \rho_{10} & \rho_{11} \end{pmatrix}\right] := \begin{pmatrix} \rho_{11} & 0 \\ 0 & -\rho_{11} \end{pmatrix}, \quad (\text{K4})$$

which admits to the following representation:

$$\mathcal{R}[\rho] = K_1 \rho K_1^{\dagger} - K_2 \rho K_2^{\dagger}, \quad \text{with } K_1 := \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}, \quad \text{and } K_2 := \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \quad (\text{K5})$$

Combining the channel derivatives, Eqs. (K2), back into the *Ansatz*-circuit derivatives, Eqs. (K1), and substituting these back into the noise susceptibility definition, Eq. (66), we obtain

$$\chi_t^{\mathcal{F}} = \sum_{l=1}^{\ell_t} \sum_{r=1}^N \tau_l (\text{Tr}[H \mathcal{A}_{\ell_t}^{\circ} \circ \cdots \circ d\mathcal{F}(r) \circ \mathcal{A}_l^{\circ} \circ \cdots \circ \mathcal{A}_1^{\circ}[\rho_0]] - \text{Tr}[H \mathcal{A}_{\ell_t}^{\circ} \circ \cdots \circ \mathbb{1} \circ \mathcal{A}_l^{\circ} \circ \cdots \circ \mathcal{A}_1^{\circ}[\rho_0]]), \quad (\text{K6a})$$

$$\chi_t^{\mathcal{C}} = \sum_{l=1}^{\ell_t} \sum_{r=1}^N \frac{\tau_l}{2} (\text{Tr}[H \mathcal{A}_{\ell_t}^{\circ} \circ \cdots \circ \mathcal{Z}(r) \circ \mathcal{A}_l^{\circ} \circ \cdots \circ \mathcal{A}_1^{\circ}[\rho_0]] - \text{Tr}[H \mathcal{A}_{\ell_t}^{\circ} \circ \cdots \circ \mathbb{1} \circ \mathcal{A}_l^{\circ} \circ \cdots \circ \mathcal{A}_1^{\circ}[\rho_0]]), \quad (\text{K6b})$$

$$\chi_t^{\mathcal{D}} = \sum_{l=1}^{\ell_t} \sum_{r=1}^N M_{l,r} \sum_{\mathcal{P} \in \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}} \frac{1}{3} (\text{Tr}[H \mathcal{A}_{\ell_t}^{\circ} \circ \cdots \circ \mathcal{P}(r) \circ \mathcal{A}_l^{\circ} \circ \cdots \circ \mathcal{A}_1^{\circ}[\rho_0]] - \text{Tr}[H \mathcal{A}_{\ell_t}^{\circ} \circ \cdots \circ \mathbb{1} \circ \mathcal{A}_l^{\circ} \circ \cdots \circ \mathcal{A}_1^{\circ}[\rho_0]]). \quad (\text{K6c})$$

To summarize these expressions in a compact form, we define energy expectation values, where the original *Ansatz* circuit $[\Lambda_t$ in Eq. (61)] is perturbed by a linear map \mathcal{M} acting on qubit r after layer \mathcal{A}_l° as

$$\mathcal{E}(\mathcal{M}, r, l) = \text{tr}[H \mathcal{A}_{\ell_t}^{\circ} \circ \cdots \circ \mathcal{A}_{l+1}^{\circ} \circ \mathcal{M}(r) \circ \mathcal{A}_l^{\circ} \circ \cdots \circ \mathcal{A}_1^{\circ}[\rho_0]]. \quad (\text{K7})$$

As $\mathcal{E}(\mathbb{1}, r, l)$ is identical to the noiseless energy bound $\mathcal{E}_t(0)$, we can further define energy fluctuations as

$$\delta\mathcal{E}(\mathcal{M}, r, l) = \mathcal{E}(\mathcal{M}, r, l) - \mathcal{E}_t(0). \quad (\text{K8})$$

Averaging these fluctuations over the layer and qubit indices, l and r , respectively, further allows us to define an average noise-induced energy fluctuation for amplitude damping, dephasing, and depolarizing noise as

$$d\mathcal{E}(\Lambda_t, \mathcal{F}) = \frac{1}{\ell_t N} \sum_{l=1}^{\ell_t} \sum_{r=1}^N \tau_l \delta\mathcal{E}(d\mathcal{F}, r, l), \quad (\text{K9a})$$

$$d\mathcal{E}(\Lambda_t, \mathcal{C}) = \frac{1}{\ell_t N} \sum_{l=1}^{\ell_t} \sum_{r=1}^N \frac{\tau_l}{2} \delta\mathcal{E}(\mathcal{Z}, r, l), \quad (\text{K9b})$$

$$d\mathcal{E}(\Lambda_t, \mathcal{D}) = \frac{1}{N_H} \sum_{l=1}^{\ell_t} \sum_{r=1}^N \frac{M_{l,r}}{3} \sum_{\mathcal{P} \in \{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}} \delta\mathcal{E}(\mathcal{P}, r, l). \quad (\text{K9c})$$

These expressions allow us to cast the noise susceptibility of amplitude damping, dephasing, and depolarizing noise in a compact form, given by Eqs. (67), as summarized in the body of this article.

- [1] S. McArdle, S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, *Rev. Mod. Phys.* **92**, 015003 (2020).
- [2] D. R. Hartree and W. Hartree, *Proc. Math. Phys. Eng. Sci.* **150**, 9 (1935).
- [3] W. Kohn and L. J. Sham, *Phys. Rev.* **140**, A1133 (1965).
- [4] P. Hohenberg and W. Kohn, *Phys. Rev.* **136**, B864 (1964).
- [5] E. Rossi, G. L. Bendazzoli, S. Evangelisti, and D. Maynau, *Chem. Phys. Lett.* **310**, 530 (1999).
- [6] B. Bauer, S. Bravyi, M. Motta, and G. K.-L. Chan, *Chem. Rev.* **120**, 12685 (2020).
- [7] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, *Nat. Commun.* **5**, 4213 (2014).
- [8] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, *Phys. Rep.* **986**, 1 (2022).
- [9] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, *Quantum Sci. Technol.* **4**, 014008 (2018).
- [10] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, *Nat. Commun.* **10**, 3007 (2019).
- [11] Y. S. Jordanov, V. Armaos, C. H. W. Barnes, and D. R. M. Arvidsson-Shukur, *Commun. Phys.* **4**, 228 (2021).
- [12] Y. S. Jordanov, C. H. W. Barnes, and D. R. M. Arvidsson-Shukur, *Phys. Rev. A* **106**, 032434 (2022).
- [13] H. L. Tang, V. O. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou, *PRX Quantum* **2**, 020310 (2021).
- [14] K. Dalton, C. K. Long, Y. S. Jordanov, C. G. Smith, C. H. W. Barnes, N. Mertig, and D. R. M. Arvidsson-Shukur, *npj Quantum Inf.* **10**, 18 (2024).
- [15] P. G. Anastasiou, Y. Chen, N. J. Mayhall, E. Barnes, and S. E. Economou, *Phys. Rev. Res.* **6**, 013254 (2024).
- [16] X. Liu, A. Angone, R. Shaydulin, I. Safro, Y. Alexeev, and L. Cincio, *IEEE Trans. Quantum Eng.* **3**, 1 (2022).
- [17] C. Feniou, M. Hassan, D. Traoré, E. Giner, Y. Maday, and J.-P. Piquemal, *Commun. Phys.* **6**, 192 (2023).
- [18] S. E. Smart and D. A. Mazziotti, *Phys. Rev. Lett.* **126**, 070504 (2021).
- [19] E. Fontana, M. Cerezo, A. Arrasmith, I. Rungger, and P. J. Coles, *Quantum* **6**, 804 (2022).
- [20] J. Preskill, *Quantum* **2**, 79 (2018).
- [21] Y. Kim, A. Eddins, S. Anand, K. X. Wei, E. van den Berg, S. Rosenblatt, H. Nayfeh, Y. Wu, M. Zaletel, K. Temme, and A. Kandala, *Nature (London)* **618**, 500 (2023).
- [22] 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.*, *Science* **369**, 1084 (2020).
- [23] 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.*, *Phys. Rev. X* **6**, 031007 (2016).
- [24] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Nature (London)* **549**, 242 (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] X. Xue, M. Russ, N. Samkharadze, B. Undseth, A. Sammak, G. Scappucci, and L. M. K. Vandersypen, *Nature (London)* **601**, 343 (2022).
- [27] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, *Nat. Commun.* **9**, 4812 (2018).
- [28] A. Arrasmith, M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles, *Quantum* **5**, 558 (2021).
- [29] M. Cerezo and P. J. Coles, *Quantum Sci. Technol.* **6**, 035006 (2021).
- [30] S. Wang, E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, *Nat. Commun.* **12**, 6961 (2021).
- [31] E. R. Anschuetz and B. T. Kiani, *Nat. Commun.* **13**, 7760 (2022).
- [32] H. R. Grimsley, G. S. Barron, E. Barnes, S. E. Economou, and N. J. Mayhall, *npj Quantum Inf.* **9**, 19 (2023).
- [33] E. Grant, L. Wossnig, M. Ostaszewski, and M. Benedetti, *Quantum* **3**, 214 (2019).
- [34] G. De Palma, M. Marvian, C. Rouzé, and D. S. França, *PRX Quantum* **4**, 010309 (2023).
- [35] Y. S. Jordanov, D. R. M. Arvidsson-Shukur, and C. H. W. Barnes, *Phys. Rev. A* **102**, 062612 (2020).
- [36] J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, N. J. Russ, M. L. Leininger, C. L. Janssen, E. T. Seidl, W. D. Allen, H. F. Schaefer, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, *Wiley Interdiscip. Rev. Comput. Mol.* **2**, 556 (2012).
- [37] P. Stano and D. Loss, *Nat. Rev. Phys.* **4**, 672 (2022).
- [38] L. Ding, M. Hays, Y. Sung, B. Kannan, J. An, A. Di Paolo, A. H. Karamlou, T. M. Hazard, K. Azar, D. K. Kim, B. M. Niedzielski, A. Melville, M. E. Schwartz, J. L. Yoder, T. P. Orlando, S. Gustavsson, J. A. Grover, K. Serniak, and W. D. Oliver, *Phys. Rev. X* **13**, 031035 (2023).
- [39] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, *Rev. Mod. Phys.* **95**, 045005 (2023).