RESEARCH ARTICLE | DECEMBER 09 2025

Measurement-efficient ADAPT-VQE with the SOAP parameter optimizer

Xinyan Jiang ; Zirui Sheng ; Cunxi Gong ; Weitang Li ; Zhigang Shuai

J. Chem. Phys. 163, 224118 (2025)
https://doi.org/10.1063/5.0296397

The Journal of Chemical Physics ARTICLE pubs.aip.org/aip/jcp

Measurement-efficient ADAPT-VQE with the SOAP parameter optimizer

Cite as: J. Chem. Phys. 163, 224118 (2025); doi: 10.1063/5.0296397
Submitted: 13 August 2025 • Accepted: 12 November 2025 •
Published Online: 9 December 2025

Xinyan Jiang,[1] Zirui Sheng,[1] Cunxi Gong,[1] Weitang Li,[1,a)] and Zhigang Shuai[1,2]

AFFILIATIONS

[1] Guangdong Basic Research Center of Excellence for Aggregate Science, School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Guangdong 518172, China
[2] MOE Key Laboratory for Organic OptoElectronics and Molecular Engineering, Department of Chemistry, Tsinghua University, Beijing 100084, China

[a)] **Author to whom correspondence should be addressed:** liwt31@gmail.com

ABSTRACT

Quantum computing on near-term noisy intermediate-scale quantum devices holds significant promise for simulating complex chemical systems. Among various variational quantum algorithms, the adaptive derivative-assembled pseudo-Trotter ansatz variational quantum eigensolver (ADAPT-VQE) is widely used for generating molecule-specific adaptive ansätze for different molecules, yet its measurement requirement is extensive, calling for suitable optimizers. In this study, we utilize the ADAPT-VQE algorithm enhanced by a powerful optimizer termed sequential optimization with an approximate parabola (SOAP) to calculate molecular energies. These computations are carried out through classical simulations using the TenCirChem software. Our results demonstrate the efficiency and robustness of the SOAP optimizer for ADAPT-VQE. Furthermore, we show that SOAP performs effectively across different ADAPT-VQE ansatz element pools. This work presents a strategy to mitigate the substantial measurement requirements associated with ADAPT-VQE.

I. INTRODUCTION

Quantum computing on near-term noisy intermediate-scale quantum (NISQ) devices presents promising opportunities for simulating complex chemical systems.[1–5] However, current quantum computers face significant challenges, including limited qubit connectivity,[6] constrained physical resources,[7] and susceptibility to quantum noise.[8–10] Before the dawn of the era of fully Fault-Tolerant Quantum Computing (FTQC),[11] to better utilize the capabilities of existing equipment, joint research efforts of hardware and software for quantum computing have been undertaken. Quantum computer hardware has advanced rapidly in recent years, transitioning from laboratory prototypes to practical quantum chips capable of surpassing state-of-the-art supercomputers in certain tasks.[12] Specifically, superconducting systems,[13–15] trapped-ion systems,[16–19] and photonic systems[20–22] represent some of the leading development trends for quantum computing hardware. The development of quantum computing software is equally crucial.[23] In particular, the software ecosystem encompasses quantum computer simulators and quantum algorithms, both are essential to leverage the capabilities of modern quantum hardware. Quantum computer simulators allow for the emulation of quantum environments on classical computing platforms. Currently, numerous practical software packages are available, such as Qiskit,[24] PennyLane,[25] and Q2 Chemistry,[26] which facilitate the research, design, and validation of quantum chemistry algorithms. Among these software packages, we developed TenCirChem, an efficient, open-source Python library designed to simulate quantum chemistry algorithms, offering low-cost, highly flexible, and scalable solutions for testing quantum algorithms.[27]

The search for robust and efficient quantum algorithms has been a major driving force behind advances in quantum computing.[28–34] The phase estimation algorithm (PEA)[35] was the first quantum algorithm proposed to simulate electronic structure problems.[36–38] Although PEA achieves high computational accuracy, its practical deployment is limited by the lengthy and coherent gate sequences required for precise operation.[39] In contrast, the variational quantum eigensolver (VQE)[40] addresses this limitation by adopting a hybrid quantum–classical approach. By combining a reconfigurable quantum processing unit (QPU) with a classical optimization algorithm executed on a classical processing unit (CPU),

VQE significantly reduces the coherence time requirements for quantum state evolution,[40] making it more compatible with NISQ devices.[41,42] However, VQE relies on a preselected ansatz, which often leads to approximate wave functions and energies, especially for strongly correlated systems. This approach sacrifices accuracy for shorter circuit depths, making it less effective for systems requiring high precision.[38]

The adaptive derivative-assembled pseudo-Trotter ansatz variational quantum eigensolver (ADAPT-VQE) algorithm is a quantum–classical hybrid algorithm that adaptively constructs a compact ansatz by iteratively adding fermionic operators. It improves on VQE by eliminating the need for a preselected ansatz, which often fails for strongly correlated systems. Moreover, by adaptively building the ansatz through adding operators based on energy gradients, ADAPT-VQE ensures higher accuracy and compactness while maintaining manageable circuit depth.[38] However, ADAPT-VQE incurs an additional measurement cost due to the multiple rounds of parameter optimization. The ADAPT-VQE algorithm has many variants. For example, Overlap-ADAPT-VQE[43] has several advantages over ADAPT-VQE for strongly correlated systems because its wave function expansion is achieved by iteratively maximizing overlap with intermediate reference states containing partial correlation information, which saves circuit depth and avoids the overparameterized ansatz caused by the sensitivity of ADAPT-VQE to local energy minima. Pruned-ADAPT-VQE[44] uses an automated and cost-effective refinement technique that eliminates redundant operators from the ansatz while maintaining convergence properties, resolving poor operator selection, operator reordering, and fading operator problems in ADAPT-VQE and improving algorithm performance. Qubit-ADAPT-VQE[45] reduces circuit depth while maintaining the accuracy of ADAPT-VQE, making it a promising choice for building hardware-efficient ansatz (HEA) on quantum computers. TETRIS-ADAPT-VQE[46] focuses on reducing the coherence time of ADAPT-VQE by incrementally assembling the variational ansatz, incorporating a few operators per cycle in a manner tailored to the structure of the problem. As the system scales up, its improvement in circuit depth relative to the original algorithm becomes more pronounced, bringing the practical quantum advantage for complex systems closer to realization. In short, the ADAPT-VQE algorithm represents a significant advancement in quantum computational chemistry, evolving into a family of methods that can accommodate multiple use cases.

In variational quantum algorithms, the parameter optimizer is responsible for iteratively adjusting the parameters of a variational quantum circuit to minimize the value of a cost function.[47] An efficient optimizer, especially a gradient-free one, is important for ADAPT-VQE because it can mitigate the excessive number of measurements required for high accuracy and better resist interference from quantum noise.[48] Emphasis on a gradient-free optimizer is motivated by the difficulty in finding the UCC gradients and by the performance limitations of NISQ devices. The constrained optimization by linear approximation (COBYLA)[49] algorithm, the simultaneous perturbation gradient approximation (SPSA) algorithm,[50] the Powell's method,[51] and the Nelder–Mead algorithm[52] are some of the popular gradient-free optimizers in quantum computing research.[48,53–56] Recently, a new optimizer termed the sequential optimization with approximate parabola (SOAP) optimizer was reported.[48] It accelerates quantum chemical simulations by sequentially approximating energy landscapes with parabolic fits and adaptively updating search directions, enabling efficient parameter correlation modeling and noise suppression with minimal quantum measurements. The SOAP optimizer has a wide range

TABLE I. Energy evaluations required for convergence with various optimizers.

		Bond length d											
		0.5 Å		1.0 Å		1.5 Å		2.0 Å		2.5 Å		3.0 Å	
Molecule	Method	F[a]	Q[b]	F	Q	F	Q	F	Q	F	Q	F	Q
LiH	SOAP	1015	1015	905	905	743	743	824	824	752	752	203	852
	Nelder-Mead	11 733	11 447	7425	8360	5024	4752	6902	6049	7143	6701	8826	8959
	COBYLA	5251	5030	3299	4521	3275	4323	3376	3321	4436	2887	4932	3142
	Powell	2720	2753	2733	2779	2464	2485	2495	2438	2493	2717	2731	2699
	BFGS	219 752	203 432	189 016	177 544	108 384	103 336	112 592	130 720	117 624	118 008	121 104	142 416
BeH_2	SOAP	1425	1425	1419	1419	1137	1046	1240	1339	937	1180	2119	6873
	Nelder-Mead	16 144	14 129	10 855	11 240	9999	10 007	148 750	13 604	165 595	173 949	103 923	283 377
	COBYLA	5621	10 411	5222	7897	6194	5337	4866	7516	17 165	10 069	23 842	28 905
	Powell	4047	4049	4229	4154	4287	4292	4604	3927	14 037	12 095	6729	25 119
	BFGS	492 264	443 656	368 184	339 976	334 384	294 600	328 960	285 448	1 911 904	1 784 536	794 312	2 743 648
H_6	SOAP	2531	2531	2691	2681	2948	4746	2316	34 736	13 104	16 199	1315	766
	Nelder-Mead	28 669	29 430	40 716	39 304	747 569	1 237 291	83 675	1 164 132	765 467	28 298	147 727	48 780
	COBYLA	8974	9096	10 201	10 346	16 917	40 217	59 885	77 590	21 674	51 508	42 180	14 196
	Powell	8633	8342	11 018	10 662	13 854	25 136	58 855	77 131	54 844	48 201	57 122	9433
	BFGS	863 112	882 200	641 016	611 448	6 039 360	28 632 392	6 266 832	29 738 776	4 636 112	7 707 752	765 856	468 912

[a]Fermionic ansatz element pool.
[b]Qubit ansatz element pool.

of advantages, including efficiency, noise resilience, rapid convergence, scalability, and ease of implementation. However, despite its promising potential, the SOAP optimizer has not yet been integrated with ADAPT-VQE, leaving room for further research and validation.

In this paper, we combine the ADAPT-VQE algorithm and the SOAP optimizer for quantum computation of molecular energies. We test three molecular systems with increasing complexity: LiH, BeH_2, and H_6. We compare SOAP with three other classical gradient-free optimization algorithms: Nelder–Mead, COBYLA, and Powell. We show the advantages of the SOAP optimizer over other optimizers through a series of experiments. Our work contributes to the development of the ADAPT-VQE family of algorithms by using the SOAP optimizer, which provides an average 18-fold speed-up compared to other gradient-free optimizers (Table I).

II. METHODS

A. ADAPT-VQE algorithm

The ADAPT-VQE algorithm is a greedy algorithm that iteratively grows a problem-tailored ansatz by adding one operator at a time. To begin, an ansatz element pool with a group of anti-Hermitian operators is defined $\{\hat{P}_m\}$, where $\hat{P}_m^\dagger = -\hat{P}_m$. Different choices are available for this pool, as demonstrated in previous research.[45,57] Then, a reference state, which is usually the Hartree–Fock (HF) ground state, is chosen as the starting point of the iterations:

$$\left|\Psi^{\mathrm{ref}}\right\rangle = \left|\Psi^{\mathrm{HF}}\right\rangle. \tag{1}$$

For every iteration, the measurement of the expectation value of the commutator of the Hamiltonian and each ansatz element is performed. The program will terminate if the pool gradient norm is smaller than a threshold. Otherwise, the ansatz element with the largest energy gradient will be added to the ansatz. Based on the new ansatz, a reoptimization for the parameters via VQE will be performed before the algorithm goes to the next iteration. This process finds the direction at which the energy decreases the fastest until the energy decrease of pending steps can be ignored.

The proposed algorithm proceeds through the following steps:

1. **Hamiltonian preparation**: On a classical processor, compute the one- and two-body integrals and transform the fermionic Hamiltonian into a qubit representation using an appropriate mapping (e.g., Bravyi–Kitaev transformation).
2. **Ansatz element pool initialization**: Define a fixed set of ansatz elements (operator pool) that will remain unchanged throughout the algorithm execution.
3. **Gradient computation**: Initialize the quantum computation using the HF state as the reference state $|\Psi^{\mathrm{ref}}\rangle$ and the identity operator as the initial ansatz. For each iteration k, compute the energy gradient for every ansatz element $\hat{P}_m$ in the pool:

$$\begin{aligned}\left.\frac{\partial E}{\partial \theta_m}\right|_{\theta_m=0} &= \left[\frac{\partial}{\partial \theta_m}\left\langle \Psi^{(k-1)}\left|e^{-\theta_m \hat{P}_m}\hat{H}e^{\theta_m \hat{P}_m}\right|\Psi^{(k-1)}\right\rangle\right]_{\theta_m=0} \\ &= \langle \Psi^{(k-1)}|[\hat{H},\hat{P}_m]|\Psi^{(k-1)}\rangle. \end{aligned} \tag{2}$$

4. **Termination check and operator selection**: Terminate if the gradient norm falls below a predetermined threshold. Otherwise, select the ansatz element $\hat{P}_m$ with the largest gradient magnitude, initialize its parameter $\theta_m = 0$, and append it to the current ansatz (without modifying the original operator pool):

$$|\widetilde{\psi}^m\rangle = \hat{P}_m(\theta_m)\left|\psi^{m-1}\right\rangle = \prod_{i=1}^{m}\hat{P}_i(\theta_i)|\Psi^{\mathrm{ref}}\rangle. \tag{3}$$

5. **Parameter optimization**: Reoptimize all variational parameters $\boldsymbol{\theta} = (\theta_1,\ldots,\theta_m)$ using a hybrid quantum–classical approach:

$$\boldsymbol{\theta}' = \underset{\boldsymbol{\theta}}{\mathrm{argmin}}\ \langle\Phi(\boldsymbol{\theta})|\hat{H}|\Phi(\boldsymbol{\theta})\rangle, \tag{4}$$

where $|\Phi(\boldsymbol{\theta})\rangle$ represents the parameterized quantum state.

6. **Ansatz update**: Construct the new ansatz state with optimized parameters:

$$|\psi^m\rangle := \prod_{i=1}^{m}\hat{P}_i(\theta_i')|\Psi^{\mathrm{ref}}\rangle. \tag{5}$$

7. **Iteration**: Return to Step 3 and repeat until convergence criteria are met.

B. SOAP algorithm

The SOAP optimizer is used due to its efficiency, robustness, flexibility, and gradient-free characteristics, which make it particularly suitable for quantum computing applications.[48] This section systematically presents the SOAP algorithm. We first establish the theoretical foundations by discussing second-order Møller–Plesset perturbation theory (MP2) and the line search procedure using the approximate parabola (LSAP) algorithm. Building upon these concepts, we then detail the operational workflow of the algorithm. This structured approach provides the reader with a comprehensive understanding of the SOAP methodology.

The SOAP algorithm is rooted in the intuition that the initial guess for the UCC ansatz is already well positioned near a highly favorable local minimum, or even the global minimum. In a typical UCC calculation, MP2 determines the initial value of parameter θ for the wavefunction $|\phi(\vec{\theta})\rangle$ through the equation,

$$\theta_{ij}^{ab} = \frac{h_{ijba} - h_{ijab}}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b}, \tag{6}$$

in which θ_{ij}^{ab} is linked to the excitation from ij orbitals to ab orbitals, h_{pqrs} denotes the two-electron integral, and ε_p is the ground-state energy of the HF state. The optimization performance of an optimizer is determined not only by the optimization techniques it employs but also by the choice of its initial state. Moreover, an effective initial guess must be readily accessible to minimize computational overhead. The effectiveness of MP2 in optimization comes from two key factors: its ability to efficiently generate high-quality

initial parameter estimates through computationally inexpensive calculations and its robust performance validation across experimental data. Notably, empirical studies have consistently demonstrated its efficacy for both weakly and strongly correlated systems. Although generating initial parameters $\vec{\theta}$ through MP2 theory provides excellent starting points, alternative initialization schemes can be employed in practice. In our previous work, we have shown that SOAP performs well with the initial guess $\vec{\theta} = 0$, which conveniently reduces the parameterized state $|\phi(\vec{\theta})\rangle$ to the HF state.

The LSAP algorithm serves as the core component of the SOAP algorithm. As mentioned above, the SOAP algorithm is based on the assumption that the vector parameter θ is very close to the minimum, where the energy can be approximated through

$$E(\vec{\theta} + x\vec{v}) \approx ax^2 + bx + c, \tag{7}$$

in which the left-hand side is the energy approximation for $\vec{\theta}$ stepping on an arbitrary direction $\vec{v}$. The right-hand side represents a parabolic approximation with coefficients a, b, and c. The coefficients can be solved via three energy evaluations, followed by a shift of $\left(-\frac{b}{2a}\right)\vec{v}$ added to $\vec{\theta}$. Through continuously implementing this procedure, $\vec{\theta}$ moves closer and closer to the minimum until a certain condition is satisfied, then the parabola fit is accomplished.

In the above introduction, there are two key quantities, $\vec{v}$ and x. The direction vector $\vec{v}$ is within a list of directional vectors $\mathcal{V} = [\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_N]$. The direction vector list is variational and changes as the program progresses, which gives the algorithm good flexibility. The small scalar x_m is defined by

$$x_m = mu, \quad m \in \mathbb{Z}, \tag{8}$$

where u is a constant. Based on this definition, we define y_m as

$$y_m = E(\vec{\theta}_{i-1} + x_m\vec{v}_i). \tag{9}$$

Here, $\vec{\theta}_{i-1}$ is the parameter from the previous optimization, and $x_m\vec{v}_i$ represents a shift for $\vec{\theta}_{i-1}$ in the direction $\vec{v}_i$ with amplitude x_m. This energy measurement step will be performed by a quantum computer in a real application. Then, based on (7), in which three energy evaluations are required for the approximation, the parabola fit will be made. Because the vector parameter $\vec{\theta}$ is very close to the minimum, here we choose $m = -1, 0, 1$ and define the data set for the parabola fit as

$$\begin{aligned}\mathcal{S} &= \{(x_i, y_i) \mid i = -1, 0, 1\} = \{(x_{-1}, y_{-1}), (x_0, y_0), (x_1, y_1)\} \\ &= \{(-u, y_{-1}), (0, y_0), (u, y_1)\}.\end{aligned} \tag{10}$$

Two energy measurements, together with an energy input $E(\vec{\theta}_{i-1})$ from the previous optimization, are combined to form this data set.

However, certain edge cases emerge when the initial guess x_0 lies far from the true minimum, resulting in failed energy convergence. This challenge is particularly pronounced for strongly correlated systems. To improve algorithmic universality, we introduce a specialized modification scheme to handle these exceptional cases. It adds (x_4, y_4) to the data set S and fits the parabola through the least-squares method:

$$\min \sum_{(x_i, y_i) \in \mathcal{F}} (ax_i^2 + bx_i + c - y_i)^2. \tag{11}$$

This modification not only resolves the problem of corner cases but also avoids adding undue complexity to the algorithm.

In addition, the algorithm can be further simplified when x_0 is sufficiently close to the minimum. To avoid extensive measurements, we can calculate the energy through

$$E(\vec{\theta}_i) = c - \frac{b^2}{4a}. \tag{12}$$

This simplification maintains the accuracy of the algorithm and improves its computational efficiency.

The algorithm proceeds through the following steps:

1. **Initialization**: Receive the previous optimization parameters $\vec{\theta}_{i-1}$, energy E_{i-1}, and current direction vector $\vec{v}_i$ as inputs.
2. **Formulate** $\mathcal{S}$: Perform energy evaluations along the search direction:

$$\mathcal{S} = \{(x_{-1}, y_{-1}), (x_0, y_0), (x_1, y_1)\}, \quad \text{where} \begin{cases} y_{-1} = E(\vec{\theta}_{i-1} + x_{-1}\vec{v}_i), \\ y_1 = E(\vec{\theta}_{i-1} + x_1\vec{v}_i), \\ y_0 = E_{i-1}. \end{cases} \tag{13}$$

3. **Corner case handling**: Identify the minimum energy point $y_m = \min(y_{-1}, y_0, y_1)$:
   - If $y_m = y_0$, proceed to Step 4.
   - If $y_m = y_{-1}$, set $k = -4$.
   - If $y_m = y_1$, set $k = 4$.
     Evaluate $y_k = E(\vec{\theta}_{i-1} + x_k\vec{v}_i)$:
   - If $y_m < y_k$, expand $\mathcal{S}$ with (x_k, y_k).
   - Else, update $\vec{\theta}_i \leftarrow \vec{\theta}_{i-1} + x_k\vec{v}_i, E_i \leftarrow y_k$ and jump to Step 7.
4. **Parabolic fitting**: Determine coefficients (a, b, c) for model $y = ax^2 + bx + c$ using $\mathcal{S}$.
5. **Parameter update**: Compute the new parameters:

$$\vec{\theta}_i = \vec{\theta}_{i-1} - \frac{b}{2a}\vec{v}_i. \tag{14}$$

6. **Energy evaluation**:

$$E_i = \begin{cases} c - \dfrac{b^2}{4a} & \text{if } \mathcal{S} \text{ has 3 elements}, \\ E(\vec{\theta}_i) & \text{if } \mathcal{S} \text{ has 4 elements}. \end{cases} \tag{15}$$

7. **Output**: Return the optimized parameters $\vec{\theta}_i$ and energy E_i.

Following our detailed analysis of MP2 theory and the LSAP algorithm, particularly their ecological niches within the SOAP framework, we now provide the procedures of the SOAP algorithm.

The SOAP algorithm initializes with MP2 amplitudes as starting parameters $\vec{\theta}_0$, computing the initial energy $E_0 = E(\vec{\theta}_0)$. A direction set $\mathcal{V}$ is initialized to provide candidate search directions for optimization. During each iteration, for each candidate direction $\vec{v}_i \in \mathcal{V}$, SOAP uses LSAP to identify a local energy minimum in the

subspace spanned by $\vec{v}_i$, yielding updated parameters $\vec{\theta}_i$, and energy $E_i = E(\vec{\theta}_i)$. Then the energy reduction $\Delta_i = E_{i-1} - E_i$ quantifies the optimization progress along each direction. The algorithm tracks the largest Δ_i observed, identifying the most promising direction for subsequent refinement.

A key feature of SOAP is its adaptive direction set update mechanism based on Powell's method, a classical unconstrained optimization technique.[51] Before we begin the discussion, let us start with some definitions to better understand this section. $\vec{\theta}_0$ and $\vec{\theta}_N$ are defined as the set of parameters before and after iteration over $\mathcal{V}$. E_0 and E_N are the measured energies for $\vec{\theta}_0$ and E_N, respectively. After evaluating the extrapolated energy $E_{\text{ext}} = E(2\vec{\theta}_N - \vec{\theta}_0)$ to probe the energy landscape beyond the current parameters, conditional checks determine whether to retain or replace candidates in $\mathcal{V}$: if $E_{ext} \geq E_0$ (indicating non-improving extrapolation) or the condition

$$2(E_0 - 2E_N + E_{\text{ext}})[(E_0 - E_N) - \Delta] \geq (E_0 - E_{\text{ext}})^2 \Delta \qquad (16)$$

is satisfied, the iteration skips further updates. Otherwise, the least effective direction (v_j, associated with the highest recorded Δ) is removed from $\mathcal{V}$, and a new direction based on the normalized displacement $\frac{\vec{\theta}_N - \vec{\theta}_0}{|\vec{\theta}_N - \vec{\theta}_0|}$ is prioritized in front of $\mathcal{V}$. This strategy biases the search toward regions of significant energy reduction while discarding inefficient directions, combining local subspace approximations (via LSAP) for fine-grained minimization with global direction updates (via Powell's algorithm) to escape local minima.

The process repeats until convergence, eventually producing optimized circuit parameters $\vec{\theta}_{\text{opt}}$ at the minimum energy and the corresponding minimum energy $E(\vec{\theta}_{\text{opt}})$. The efficiency of SOAP lies in its hybrid approach: by integrating local energy minimization with global adaptive directionality, it accelerates convergence in quantum circuit optimization tasks, particularly when navigating complex energy landscapes where efficient search directions are critical for performance.

The total framework of our algorithm, which combines ADAPT-VQE with SOAP, is shown in Fig. 1. In subsequent sections, we refer to the iterative process within the overall ADAPT-VQE framework as the "outer iterations," while designating the optimizer-level iterations within the SOAP module as "inner iterations."

III. RESULTS AND DISCUSSION

The SOAP optimizer represents a specialized gradient-free optimization method specifically developed for quantum computing applications. To systematically evaluate its performance within the ADAPT-VQE framework, we conduct comparative studies with three established gradient-free optimizers: Nelder–Mead, COBYLA, and Powell. Our testbed comprises molecular systems of increasing complexity, including LiH, BeH_2, and H_6. All numerical simulations are performed classically using TenCirChem[27] to emulate quantum computing environments, as illustrated in Fig. 1. This controlled experimental setup enables rigorous

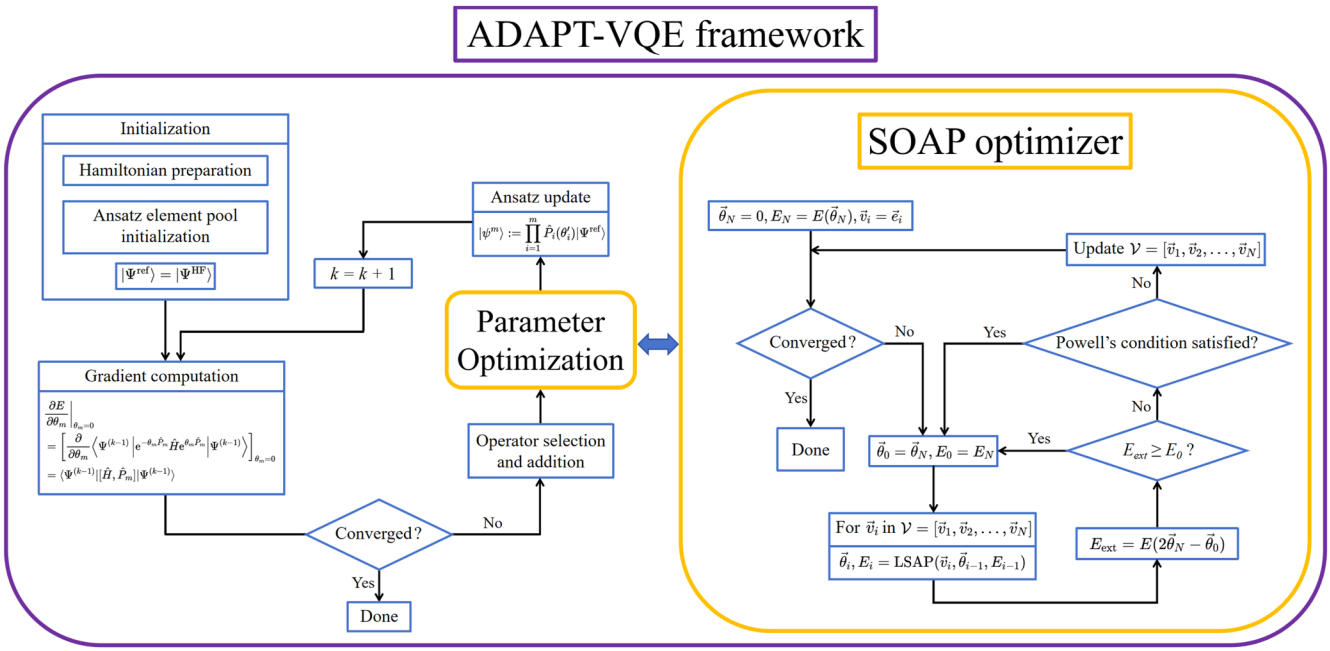

FIG. 1. Algorithm framework of the ADAPT-VQE method with the SOAP optimizer. The purple container (main structure) encompasses the components of the ADAPT-VQE algorithm, while the embedded yellow section exclusively handles the optimization subroutine execution via the SOAP optimizer. The "Operator unchanged" box in this context specifically represents "If the operators chosen in two consecutive iterations are the same" in the algorithmic process. The "Powell's condition" refers to the core decision criterion (16) in Powell's algorithm.

benchmarking of SOAP's performance against conventional optimization approaches.

Notably, in our implementation, we initialize all parameters $\vec{\theta} = 0$, which directly yields the HF state $|\phi(\vec{0})\rangle = |\mathrm{HF}\rangle$. This choice simplifies the initialization procedure while remaining consistent with the overall algorithmic framework. Although we adopt this straightforward initialization scheme in the present work, alternative approaches—including but not limited to MP2-based initialization—warrant investigation in future studies to further assess the algorithm's performance.

This study systematically evaluates optimizer performance through two complementary approaches: (1) ideal molecular simulations without noise and (2) realistic simulations incorporating Gaussian noise. In both regimes, we examine molecular systems across various bond lengths to assess robustness under different electronic configurations. To ensure comprehensive evaluation, we employ two distinct ansatz element pools (operator pools) fundamental to the ADAPT-VQE protocol: the fermionic pool[38] (consisting of fermionic excitation operators) and the qubit pool[45] (composed of Pauli string operators). These pools represent fundamentally different approaches to ansatz construction in variational quantum algorithms. Our findings demonstrate that the SOAP optimizer maintains consistent performance across both operator pools, highlighting its versatility under different ansatz construction paradigms.

A. Molecular simulation without noise

The computational efficiency of variational quantum algorithms can be quantified by the number of quantum circuit executions required to achieve a target precision in the measured observable. Here we conduct a preliminary comparison of quantum algorithm performance under different optimizers by counting the number of algorithm calls to the energy evaluation function without noise (Table I). We choose five optimizers, including four gradient-free optimizers (SOAP, Nelder–Mead, COBYLA, and Powell) and a gradient-based optimizer (BFGS). In this phase, we impose no constraints on either the maximum number of outer iterations or required precision, instead relying on the algorithm's self-terminating conditions as the stopping criterion. Each table cell is divided into two parts: the left side shows the result for the "fermionic ansatz element pool," and the right side shows the result for the "qubit ansatz element pool."

Our systematic evaluation of convergence efficiency across four gradient-free optimizers demonstrates the superior performance of SOAP in quantum chemistry simulations within the ADAPT-VQE framework (Table I), showing an average of 18-fold speed-up compared to other gradient-free optimizers. For LiH, the simplest test case, SOAP consistently converged with 200–1015 energy evaluations across all bond lengths (0.5–3.0 Å), representing efficiency improvements of 3.8× over Powell (2464–2779 evaluations), 4.6× over COBYLA (2887–5251), and 11.6× over Nelder–Mead (4752–11733). This performance advantage becomes more pronounced in complex systems: for BeH_2 at 2.5 Å using the fermionic ansatz pool, SOAP required only 937 evaluations compared to 17 165 (COBYLA), 14 037 (Powell), and 165 595 (Nelder–Mead), corresponding to efficiency gains of 18.3×, 15.0×, and 176.8×, respectively.

In challenging H_6 optimizations, the overall data exhibit a distinct volcano-shaped distribution for the optimizers, where the evaluations of both short and long bond lengths are relatively low, while the evaluations of intermediate bond lengths are significantly higher. Remarkably, SOAP's maximum evaluation count for H_6 (34 736 at 2.0 Å qubit ansatz) remained 2.2× lower than COBYLA's worst H_6 performance (77 590 at 2.0 Å qubit ansatz) and represented merely 2.8% of Nelder–Mead's worst-case requirement (1 237 291 at 1.5 Å qubit ansatz).

Also, SOAP maintained its benchmark-leading efficiency independent of ansatz selection, achieving identical fermionic/qubit evaluation counts in 58% of LiH tests. This contrasts sharply with COBYLA, which exhibited >100% variance in critical cases (e.g., BeH_2 at 0.5 Å: 5621 evaluations for fermionic vs10 411 for qubit ansatz). These results establish SOAP's unique capability to reduce variational optimization costs while guaranteeing stability in broad quantum chemical regimes.

Comparison between gradient-free optimizers and the gradient-based optimizer BFGS shows that BFGS is substantially more computationally expensive, requiring one to two orders of magnitude more evaluations than gradient-free optimizers. This is because BFGS relies on the parameter-shift rule, which requires measuring the gradient with respect to each parameter individually, leading to a large number of quantum measurements, especially when the number of parameters is large. In contrast, SOAP determines gradients by measuring only the overall energy of the circuit. As a result, SOAP requires significantly fewer measurements and demonstrates better performance compared to BFGS. Here we only conduct a preliminary comparison and will not include it in the following discussion.

In particular, our experimental results reveal unexpected behavior of the SOAP optimizer in LiH systems. While it is frequently assumed that different ansatz element pools should lead to distinct ansatz growth patterns under the ADAPT-VQE framework and consequently different convergence characteristics, SOAP exhibits remarkable consistency in certain cases. For traditional optimizers (Nelder–Mead, COBYLA, and Powell), the number of energy evaluations indeed varies significantly between fermionic and qubit pools, as theoretically expected. However, SOAP demonstrates identical convergence efficiency across both pool types in five LiH test cases (bond lengths 0.5–2.5 Å), despite producing fundamentally different quantum circuits. This is because SOAP by default sets 0.001 hartree as the convergence criterion, which is smaller than the energy difference of the fermionic and qubit pools.

We additionally analyzed the ansatz structures generated by SOAP on LiH systems (Table II). The results show that different ansatz element pools produce different ansatz. This is demonstrated by the different gate numbers, different CNOT gate numbers, different circuit depths, etc. At the same time, we find that the qubit ansatz element pool yields fewer gates, fewer CNOT gates, and shallower circuit depth than the fermionic ansatz element pool, offering lateral support for prior research.[45]

While Table I provides a useful comparison, its conclusions are limited because the experimental setup did not adequately control for computational conditions (i.e., result accuracy may have varied between runs). To address this limitation, we apply uniform criteria in subsequent analyses: a fixed number of ADAPT-VQE outer iterations and comparable solution accuracy. Since the LiH molecule

TABLE II. Basic information for ansatz for ADAPT-VQE on LiH with fermionic and qubit ansatz element pools at convergence.

	Bond length d									
	0.5 Å		1.0 Å		1.5 Å		2.0 Å		2.5 Å	
Parameter	F[a]	Q[b]	F	Q	F	Q	F	Q	F	Q
Energy evaluations	1015	1015	905	905	743	743	824	824	752	752
Variational parameters	21	21	20	20	18	18	19	19	18	18
Gate count	454	210	424	196	356	168	398	182	364	168
CNOT gates	354	172	330	160	278	136	314	148	286	136
Qubits	12	12	12	12	12	12	12	12	12	12
Circuit depth	257	123	247	100	201	87	227	105	210	92
Multicontrol gates	26	26	26	24	20	20	22	22	20	20

[a]Fermionic ansatz element pool.
[b]Qubit ansatz element pool.

is relatively simple, we select it for comparison and plotted a series of graphs showing how optimization accuracy varied with the number of ADAPT-VQE outer iterations (see Fig. 2).

In Fig. 2, we can see that the four gradient-free optimizers share similar dissociation curves for LiH at different bond lengths. That is because they all work under the ADAPT-VQE framework without Gaussian noise. Figure 2 reveals that for the fermionic ansatz element pool, the energy calculations for the LiH molecule achieve satisfactory accuracy ($<1.6 \times 10^{-3}$ hartree) with an ADAPT-VQE outer iteration of 15, across all tested bond lengths. In brief, setting the outer iteration of the ADAPT-VQE algorithm to 15 is feasible.

Thus, we evaluate the performance of the four algorithms when setting the outer iterations to 15, which corresponds to the same accuracy for different optimizers (Table III). Analysis of total energy evaluations (summing fermionic and qubit pools) demonstrates that the SOAP optimizer significantly accelerates convergence compared to all other tested gradient-free methods (Nelder–Mead, COBYLA, Powell) across LiH bond lengths (0.5–3.0 Å) for ADAPT-VQE (15 iterations). Specifically, SOAP achieves the most dramatic acceleration vs Nelder–Mead, where reductions in evaluations range from 77% (at 3.0 Å) to 90% (at 0.5 Å), consistently exceeding a 77% reduction overall and reflecting an order-of-magnitude efficiency gain.

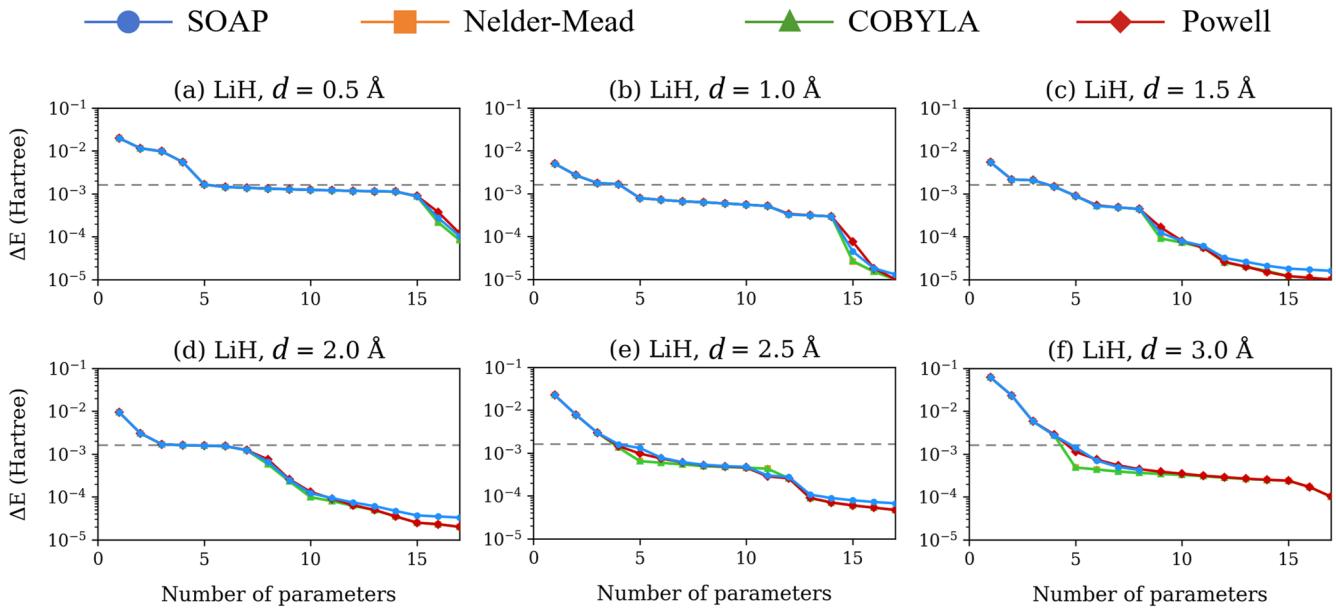

FIG. 2. Dissociation curves of LiH at different bond lengths for the fermionic ansatz element pool. The horizontal axis represents the outer iterations of the ADAPT-VQE algorithm, and the vertical axis represents the difference between the calculated energy and the exact energy, where $\Delta E = E_{\text{approx}} - E_{\text{FCI}}$. The dashed lines show the chemical accuracy threshold (1.6×10^{-3} hartree).

TABLE III. Number of energy evaluations required for convergence for SOAP and other gradient-free optimizers on LiH with ADAPT-VQE (15 outer iterations).

	Bond length d												
	0.5 Å		1.0 Å		1.5 Å		2.0 Å		2.5 Å		3.0 Å		
Method	F[a]	Q[b]	F	Q	F	Q	F	Q	F	Q	F	Q	$\bar{e}$[c]
SOAP	551	551	530	530	530	530	532	532	539	539	203	560	2.36×10^{-4}
Nelder-Mead	5215	6022	5645	6845	4098	3798	5691	4961	6007	5117	4824	4507	2.09×10^{-4}
COBYLA	2150	2080	1736	1911	1623	1718	1876	1834	1765	1889	1606	1638	2.09×10^{-4}
Powell	1563	1556	1593	1558	1377	1404	1455	1409	1410	1427	1466	1470	2.20×10^{-4}

[a]Fermionic ansatz element pool.
[b]Qubit ansatz element pool.
[c]Mean error across all experimental conditions for the corresponding optimizer.

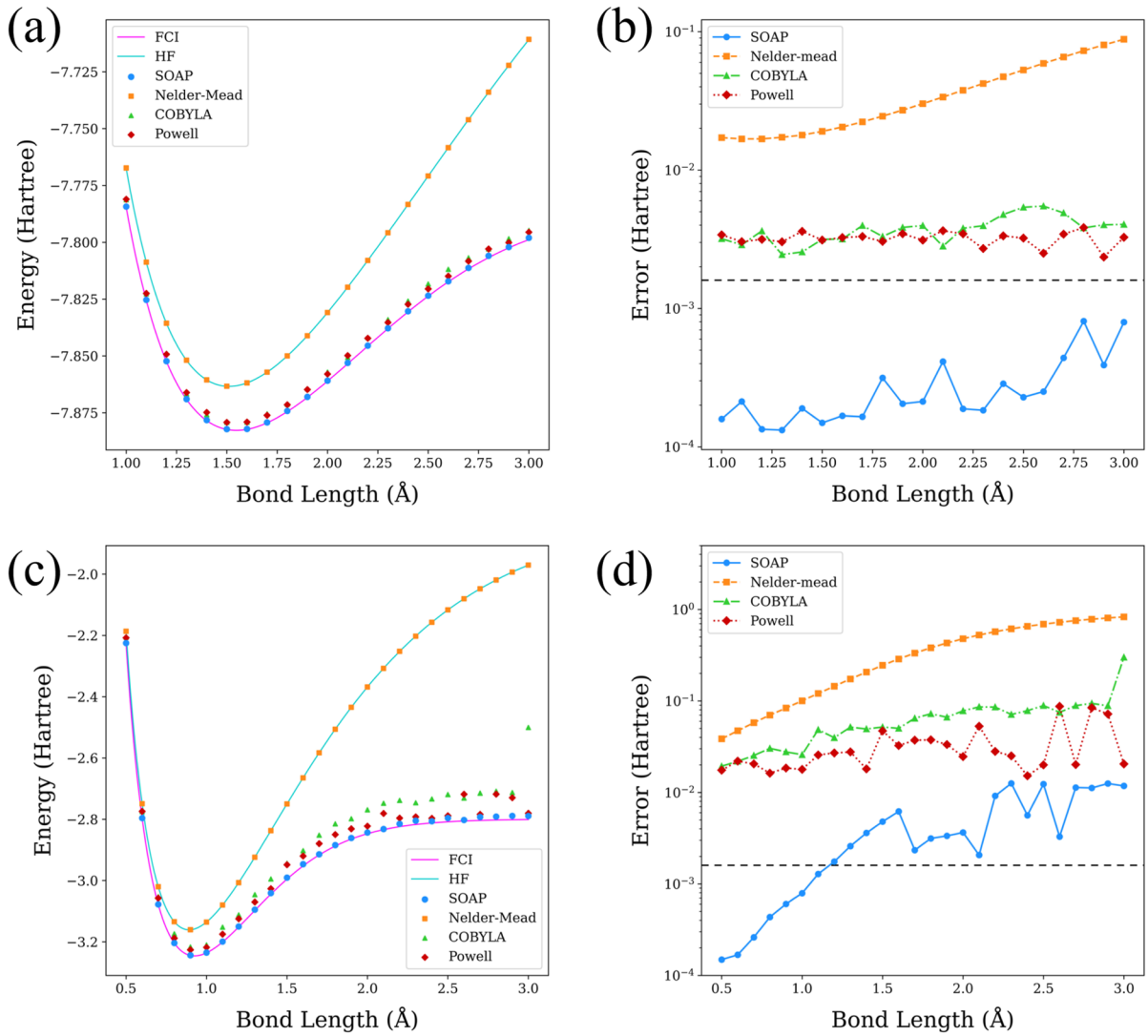

FIG. 3. (a, c) Potential energy surface and (b, d) error for the gradient-free optimizers under the ADAPT-VQE framework for the fermionic ansatz element pool. The molecular systems for (a) and (b) are LiH systems with different bond lengths, while the molecular systems for (c) and (d) are H_6 systems with different bond lengths. The Gaussian noise added is 0.001 hartree. The reference values are the FCI results. The dashed line in the right panel shows the chemical accuracy threshold (1.6×10^{-3} hartree).

Similarly, SOAP maintains a substantial advantage over COBYLA, achieving significant acceleration exceeding 50% at nearly every bond length, with values spanning from 51% (1.5 Å) to 59% (3.0 Å), resulting in consistent computational savings exceeding half the cost. Even when compared to the relatively more efficient Powell method, SOAP still delivers meaningful efficiency improvements; specifically, reductions range from 40% (1.0, 1.5 Å) to 50% (2.0, 3.0 Å), confirming SOAP's superiority despite Powell's baseline performance. Therefore, while a slight increase in qubit evaluations of SOAP at 3.0 Å warrants observation, the overwhelming, consistent

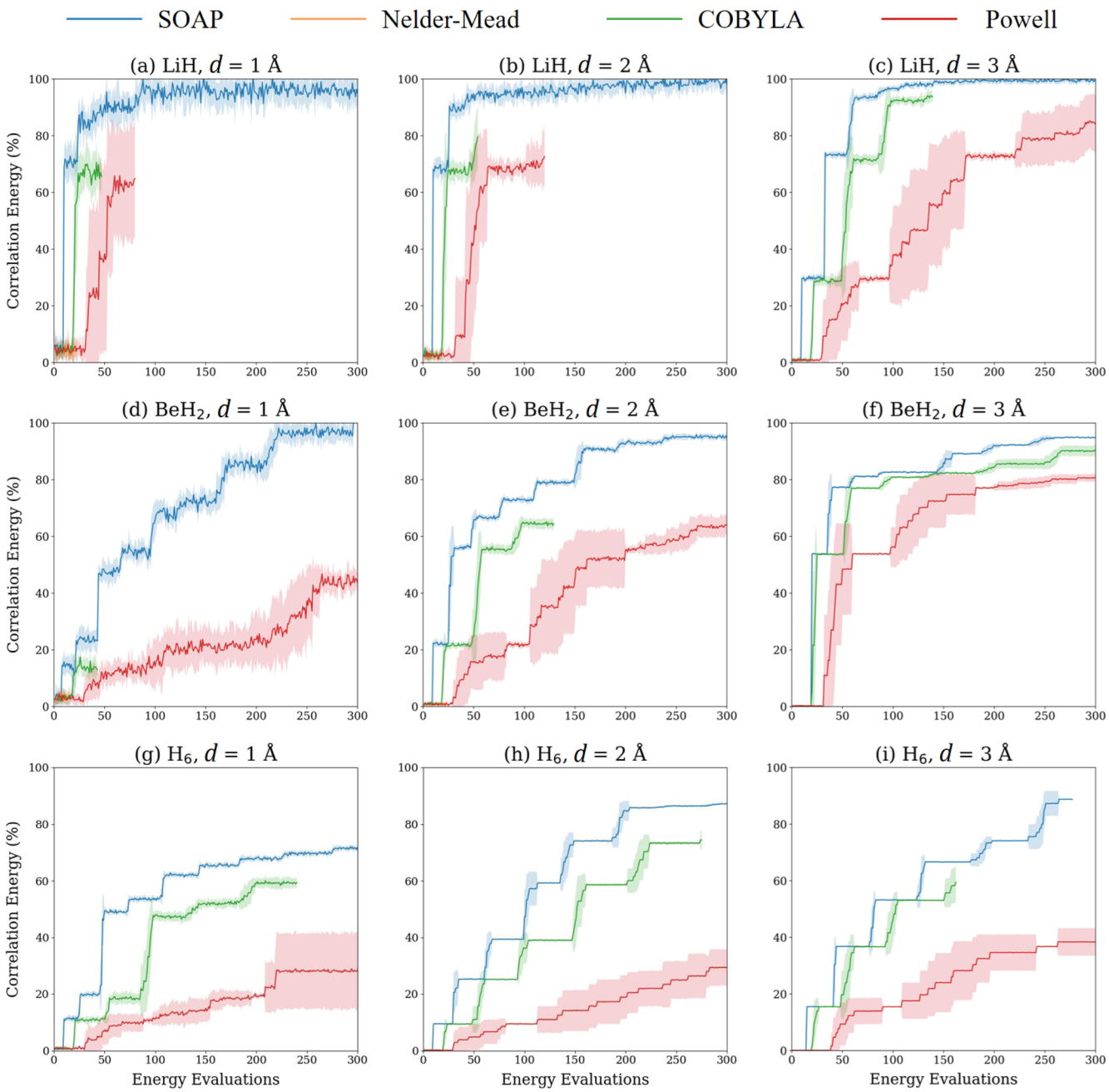

FIG. 4. Convergence curve for gradient-free optimizers including SOAP based on noisy simulation for the fermionic ansatz element pool. Gaussian noise with a standard deviation of 0.001 hartree is added to the simulation process. For each optimization technique, the standard deviation from ten independent simulations is used to form the semi-transparent regions. The [(a)–(c)], [(d)–(f)], and [(g)–(i)] panels are for the LiH, BeH_2, and H_6 molecules, respectively. From left to right, the bond length ranges from d = 1.0 Å to d = 3.0 Å.

trend across the dataset clearly establishes SOAP as the most computationally efficient gradient-free optimizer for this task, achieving profound resource savings (up to 90%) over common alternatives and significant savings (40%–59%) even against the best-performing comparator. In short, SOAP outperforms all other gradient-free optimizers when setting the outer iterations to 15.

The above experiments and analysis reveal that the SOAP optimizer outperforms other gradient-free optimizers within the ADAPT-VQE framework when not considering Gaussian noise. Whether in the case of allowing natural termination of the ADAPT-VQE algorithm or in the case of setting the outer iteration of the ADAPT-VQE algorithm to 15 to get similar and satisfactory accuracy, SOAP outperforms all other optimizers with far fewer energy evaluations and good stability when changing the ansatz element pool.

B. Molecular simulation with noise

Previous sections established the superior performance of the SOAP optimizer compared to other gradient-free optimizers in noiseless ADAPT-VQE simulations. However, real quantum computing systems inevitably contain noise. While our initial computational experiments demonstrated the method's efficiency, they did not fully assess its robustness under practical conditions. To bridge this gap, we examine SOAP's performance in noisy environments by incorporating Gaussian-distributed random noise (sampled during each energy evaluation) to simulate realistic quantum hardware conditions. The subsequent analysis primarily focuses on the fermionic ansatz element pool.

We first evaluate optimization accuracy at convergence (Fig. 3) for LiH and H_6 systems across various bond lengths, using a Gaussian noise level of 0.001 hartree. Two reference benchmarks are included: the HF energy (optimization starting point) and the full configuration interaction (FCI) energy (exact theoretical target). Figure 3 shows that SOAP consistently achieves the highest accuracy for both LiH and H_6 systems, with results closest to the FCI energy. In contrast, Nelder–Mead shows problematic convergence behavior, stalling near the initial HF energy. COBYLA and Powell display intermediate performance, with neither consistently outperforming the other across different test cases. For the H_6 system, the accuracy of SOAP is more pronounced for shorter bond lengths, while it deteriorates as the bond length increases.

Then, we explore the behavior of different optimizers for different molecular systems for the fermionic ansatz element pool (Fig. 4). Gaussian noise with a standard deviation of 0.001 hartree is added to the simulation process. To better present our data, the energy values are rescaled by the correlation energy $E_{corr} = E_{HF} - E_{FCI}$. Our ADAPT-VQE optimization procedure uses the HF state as the initial state and optimizes toward the exact energy of the molecular systems, which can be calculated via the FCI approach. When the value on the vertical axis approaches 1, it indicates closer proximity to the FCI energy, while values approaching 0 signify closer proximity to the HF energy. For each optimizer, we perform ten energy evaluations to determine the mean value and the standard deviation of the correlation energy. For all experiments associated with the subfigures, the SOAP optimizer shows much better performance than other optimizers. Conversely, the Nelder–Mead optimizer exhibits suboptimal performance for all the experiments, as it is unable to resist the interference of quantum noise and terminates at low accuracy. In LiH systems, SOAP reaches high accuracy (over 90% correlation energy) at around 100 energy evaluations, while most of the other optimizers converge incorrectly under low-precision conditions. Even for the better cases (COBYLA, $d = 3.0$ Å; Powell, $d = 3.0$ Å), the optimizers are still less efficient and less precise than SOAP. In BeH_2 systems, things get more complicated. In some cases ($d = 1.0$ Å, $d = 2.0$ Å), SOAP shows much better robustness

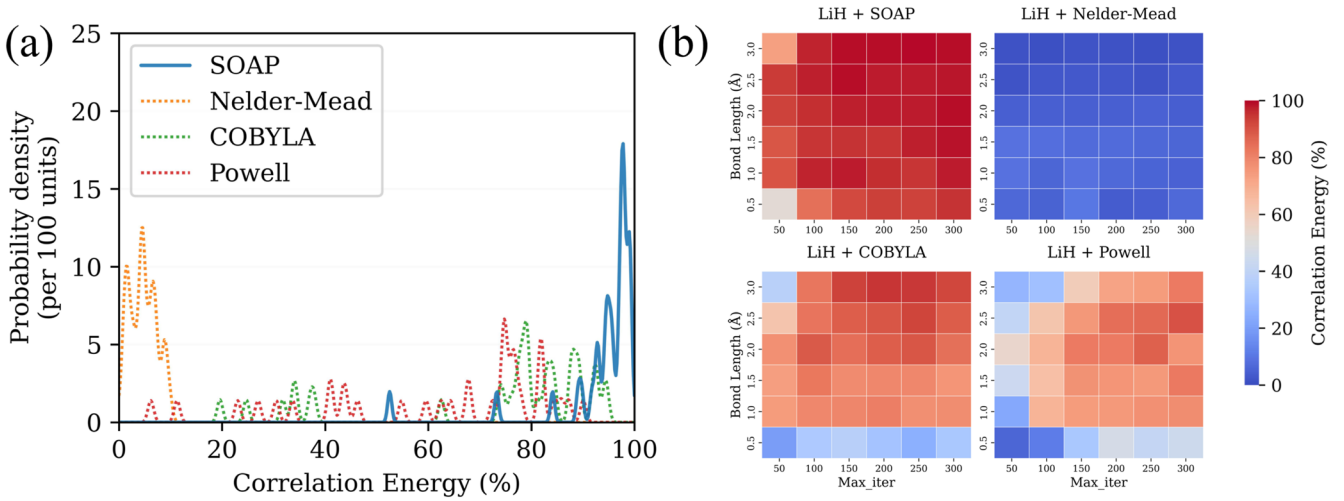

FIG. 5. (a) Probability density distribution and (b) Heatmap of correlation energy for gradient-free optimizers including SOAP based on noisy simulation for the fermionic ansatz element pool. The molecular system of interest is LiH. Gaussian noise with a standard deviation of 0.001 hartree is added to the simulation process. The horizontal axis represents the selected maximum number of calls for the optimizers, which are 50, 100, 150, 200, 250, and 300 times, respectively. The vertical axis represents the bond length of the molecule.

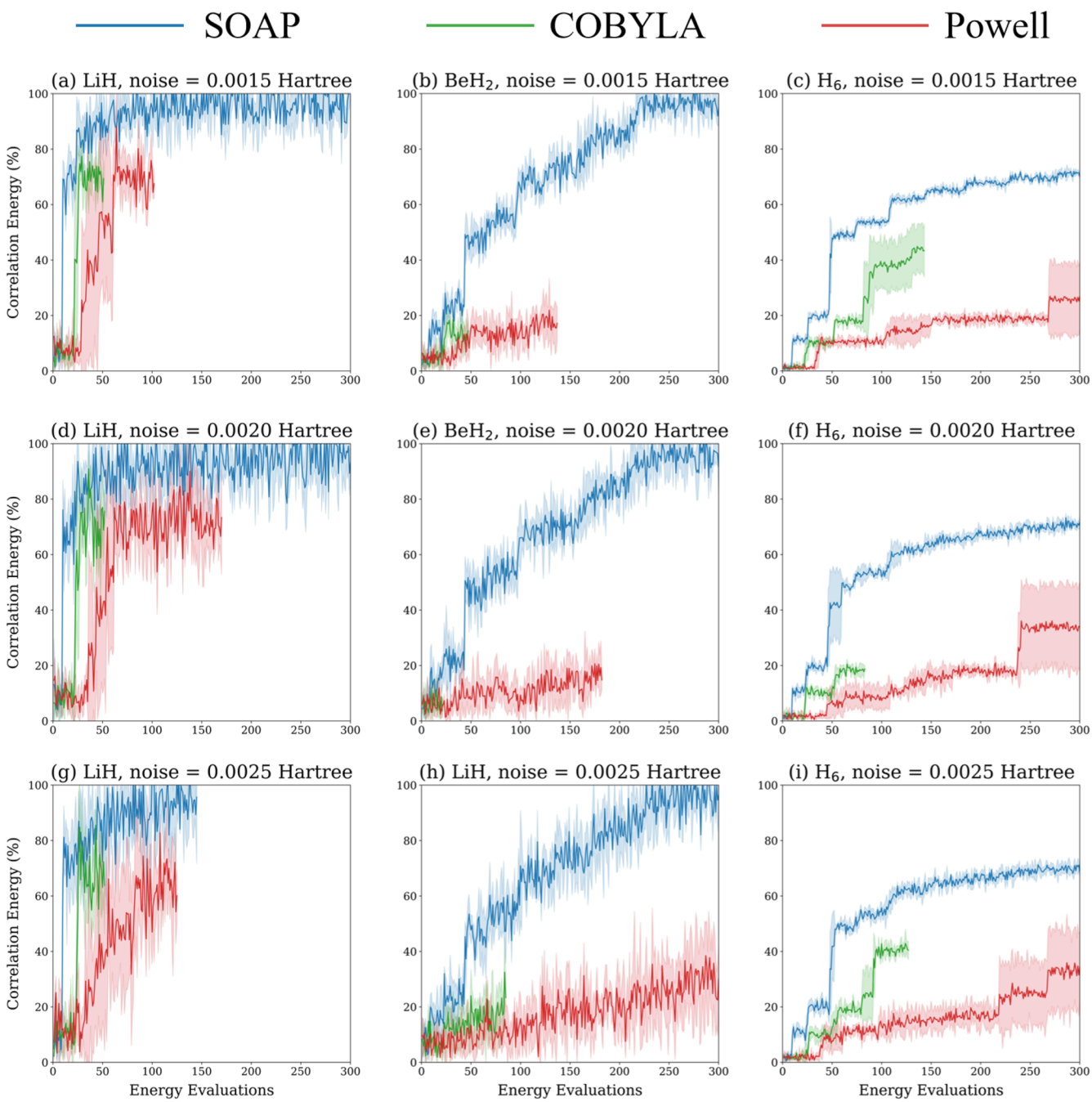

FIG. 6. Convergence curve for SOAP, COBYLA, and Powell based on noisy simulation for the fermionic ansatz element pool. Gaussian noise with standard deviations of 0.0015, 0.0020, and 0.0025 hartree is added to the simulation process. For each optimization technique, the standard deviation from four independent simulations is used to form the semi-transparent regions. A uniform bond length of 1 Å was used for all molecular systems.

and precision. For the case where d = 3.0 Å, the advantages of SOAP, though not particularly pronounced, are still present. In H_6 systems, SOAP reaches high accuracy (~90% correlation energy) for some cases (d = 2.0 Å, d = 3.0 Å). For the special case (d = 1.0 Å), the correlation energy of SOAP at the right-hand side is around 0.7 when the computation is still underway. After checking the data, we find SOAP terminates at 91.43% correlation energy after 1096 energy evaluations, reaching high accuracy and outperforming all

its competitors at that stage (COBYLA: 59.43% after 240 evaluations; Powell: 67.99% after 2320 evaluations). An additional aspect of concern in our experiment is the standard deviation of the independent simulations, which is represented by the semi-transparent regions. The semi-transparent regions indicate the robustness of the optimizers, corresponding to their noise resistance and measurement stability. In general, the standard deviations of SOAP and COBYLA are smaller, while the standard deviation of Powell is much larger. However, the COBYLA optimizer, even with a small standard deviation, still suffers severely from false termination problems. Thus, the robustness of SOAP is still the best among the optimizers, with a small standard deviation and resistance to premature termination. In short, the SOAP optimizer outperforms all other gradient-free optimizers under noisy conditions, in terms of efficiency and robustness.

To further demonstrate the power of the SOAP algorithm, we designed another set of experiments. Under conditions when the noise is 0.001 hartree and the maximum number of optimizer calls is limited, we compare the final computational accuracy, which is represented by the size of the correlation energy. The selected maximum numbers of calls for the optimizers are 50, 100, 150, 200, 250, and 300, respectively. Here, we display the probability density distribution as well as the heatmap to provide a more intuitive visualization of the data (Fig. 5). Despite their differing formats, both charts depict identical underlying data. Clearly, the SOAP optimizer is the best among gradient-free optimizers. Except for one very special case (d = 0.5 Å, max iteration = 50), all regions corresponding to SOAP show a red coloration. Within the red regions, a significant proportion is dominated by deep red, which corresponds to the correlation energy of over 0.9. In contrast, the Nelder–Mead optimizer does not perform well for noisy simulation due to its premature termination, as shown in Fig. 4.

All regions corresponding to Nelder–Mead show a blue coloration, providing a clear illustration of what happens when Nelder–Mead fails to optimize. COBYLA and Powell share a common feature. At a small bond length (d = 0.5 Å), even with increased iterations, they can hardly optimize accurately. For SOAP, this phenomenon does not exist, indicating that it can still achieve significant success even under challenging conditions. In this visualization, COBYLA appears to perform a bit better than Powell, as it can reach higher accuracy at a small number of maximum iterations (max iteration = 50, 100). Overall, COBYLA exhibits more red regions than Powell in the heatmap. In the probability density distribution, the region where the correlation energy exceeds 90% is dominant when using SOAP, whereas the region with correlation energy below 90% becomes dominant when using Nelder–Mead. COBYLA outperforms Powell because its correlation energy peaks are overall shifted to higher values, indicating higher accuracy in energy calculations. The convergence of results from these two independent visualization approaches establishes cross-verification.

Still, the energy error of 0.001 hartree is a high requirement for near-term quantum devices. Therefore, we conduct an additional experiment with increasing noise levels to observe the behavior of the optimizers under progressively noisier conditions. The optimizers selected for presentation include SOAP, COBYLA, and Powell. Although the Nelder–Mead optimizer is also studied in previous sections, it exhibits unsatisfactory performance under noisy conditions and is therefore deemed not suitable for inclusion in the discussions here. The molecular systems we choose are LiH (d = 1 Å), BeH_2 (d = 1 Å), and H_6 (d = 1 Å). We investigate the behaviors of the systems when the noise increases from 0.0015 to 0.0025 hartree (Fig. 6). For each optimizer, we perform four energy evaluations to determine the mean value and the standard deviation of the correlation energy. We choose this number because it is easier for the optimizers to prematurely terminate when the noise increases, so we perform fewer energy evaluations to avoid bad truncation. In scenarios where other optimizers converge prematurely with lower accuracy under high-noise conditions, SOAP demonstrates strong noise resistance, ensuring stability, high precision, and a relatively fast convergence rate. This indicates its adaptability to the high energy measurement errors typical in near-term quantum devices.

Based on all the above results, we rank the optimizers under noisy conditions within the ADAPT-VQE framework: SOAP > COBYLA ≈ Powell > Nelder–Mead. SOAP clearly outperforms others in both efficiency and robustness. Nelder–Mead shows limitations in dealing with quantum computing problems. COBYLA converges faster than Powell, while Powell is more robust. Their accuracy is comparable, with performance varying across conditions.

IV. CONCLUSION

In conclusion, this work contributes to the ADAPT-VQE algorithm by applying the SOAP optimizer for the optimization process. We first demonstrate the efficiency of ADAPT-VQE + SOAP through quantum circuit simulation without noise. Table I shows that SOAP offers an average speed-up of 18 times compared to other gradient-free optimizers. Then, we focus on the LiH system and compare the required energy evaluations when setting the outer iteration of ADAPT-VQE to 15 (Table III). The molecular simulation with noise further proves the efficiency of SOAP. Moreover, we move to simulation with the addition of quantum circuit measurement noise, which is modeled through Gaussian noise. We find that SOAP maintains good accuracy after adding Gaussian noise. Study of the optimization process for the optimizers reveals that SOAP stands out for its high accuracy and resistance to measurement noise on quantum computers. The distribution of correlation energies for different optimizers based on the LiH systems shows that, with the addition of Gaussian noise, SOAP recovers most of the correlation energy and outperforms other optimizers. In general, the results of the molecular simulation show that, within the ADAPT-VQE algorithm framework and for different ansatz element pools, the SOAP optimizer outperforms other gradient-free optimizers. The rank of optimizers is SOAP > COBYLA ≈ Powell > Nelder–Mead.

Another important aspect of our work is that we use TenCirChem to simulate the quantum environment on classical computers. This shows the importance of quantum simulation tools for speeding up the development of quantum computing. Although research on quantum computing hardware is irreplaceable, quantum simulation tools offer a more flexible, cheaper, and faster option.

ACKNOWLEDGMENTS

This work is supported by the Guangdong Basic Research Center of Excellence for Aggregate Science, the National Natural Science Foundation of China (Grant Nos. T2350009 and 22433007),

and the Shenzhen Science and Technology Program (Grant No. KQTD20240729102028011).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Xinyan Jiang: Data curation (equal); Formal analysis (equal); Investigation (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Zirui Sheng**: Conceptualization (equal); Formal analysis (equal); Project administration (equal); Resources (equal); Writing – review & editing (equal). **Cunxi Gong**: Software (equal). **Weitang Li**: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Investigation (equal); Project administration (equal); Resources (equal); Software (equal); Supervision (equal); Writing – review & editing (equal). **Zhigang Shuai**: Conceptualization (equal); Funding acquisition (equal); Investigation (equal); Methodology (equal); Project administration (equal); Resources (equal); Supervision (equal); Validation (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1]X. Dan, E. Geva, and V. S. Batista, "Simulating non-Markovian quantum dynamics on NISQ computers using the hierarchical equations of motion," J. Chem. Theory Comput. **21**, 1530 (2025).
[2]H. Ma, J. Liu, H. Shang, Y. Fan, Z. Li, and J. Yang, "Multiscale quantum algorithms for quantum chemistry," Chem. Sci. **14**, 3190 (2023).
[3]M. Sarkar, L. Roy, A. Gutal, A. Kumart, and M. Paranjothy, "Quantum simulations of chemical reactions: Achieving accuracy with NISQ devices," arXiv:2503.12084 (2025).
[4]J. D. Weidman, M. Sajjan, C. Mikolas, Z. J. Stewart, J. Pollanen, S. Kais, and A. K. Wilson, "Quantum computing and chemistry," Cell Rep. Phys. Sci. **5**, 102105 (2024).
[5]T. Hoefler, T. Häner, and M. Troyer, "Disentangling hype from practicality: On realistically achieving quantum advantage," Commun. ACM **66**, 82 (2023).
[6]A. Holmes, S. Johri, G. G. Guerreschi, J. S. Clarke, and A. Y. Matsuura, "Impact of qubit connectivity on quantum algorithm performance," Quantum Sci. Technol. **5**, 025009 (2020).
[7]M. Fellous-Asiani, J. H. Chai, R. S. Whitney, A. Auffèves, and H. K. Ng, "Limitations in quantum computing from resource constraints," PRX Quantum **2**, 040335 (2021).
[8]B. Khanal, P. Rivas, A. Sanjel, K. Sooksatra, E. Quevedo, and A. Rodriguez, "Generalization error bound for quantum machine learning in NISQ era—A survey," Quantum Mach. Intell. **6**, 90 (2024).
[9]S. Chen, J. Cotler, H. Y. Huang, and J. Li, "The complexity of NISQ," Nat. Commun. **14**, 6001 (2023).
[10]D. Stilck França and R. García-Patrón, "Limitations of optimization algorithms on noisy quantum devices," Nat. Phys. **17**, 1221 (2021).
[11]A. Katabarwa, K. Gratsea, A. Caesura, and P. D. Johnson, "Early fault-tolerant quantum computing," PRX Quantum **5**, 020101 (2024).
[12]E. G. Rieffel, A. A. Asanjan, M. S. Alam, N. Anand, D. E. Bernal Neira, S. Block, L. T. Brady, and R. Biswas, "Assessing and advancing the potential of quantum computing: A NASA case study," Future Gener. Comput. Syst. **160**, 598 (2024).
[13]M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I. J. Wang, S. Gustavsson, and W. D. Oliver, "Superconducting qubits: Current state of play," Annu. Rev. Condens. Matter Phys. **11**, 369 (2020).
[14]I. Siddiqi, "Engineering high-coherence superconducting qubits," Nat. Rev. Mater. **6**, 875 (2021).
[15]J. Van Damme, S. Massar, R. Acharya, T. Ivanov, D. Perez Lozano, Y. Canvel, M. Demarets, and K. De Greve, "Advanced CMOS manufacturing of superconducting qubits on 300 mm wafers," Nature **634**, 74 (2024).
[16]C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, "Trapped-ion quantum computing: Progress and challenges," Appl. Phys. Rev. **6**, 021314 (2019).
[17]K. R. Brown, J. Chiaverini, J. M. Sage, and H. Häffner, "Materials challenges for trapped-ion quantum computers," Nat. Rev. Mater. **6**, 892 (2021).
[18]D. Schwerdt, L. Peleg, Y. Shapira, N. Priel, Y. Florshaim, A. Gross, A. Zalic, and R. Ozeri, "Scalable architecture for trapped-ion quantum computing using RF traps and dynamic optical potentials," Phys. Rev. X **14**, 041017 (2024).
[19]Y. Zhang, J. Carrasquilla, and Y. B. Kim, "Observation of a non-Hermitian supersonic mode on a trapped-ion quantum computer," Nat. Commun. **16**, 3286 (2025).
[20]A. Dutt, A. Mohanty, A. L. Gaeta, and M. Lipson, "Nonlinear and quantum photonics using integrated optical materials," Nat. Rev. Mater. **9**, 321 (2024).
[21]D. D. K. Wayo, L. Goliatt, and D. Ganji, "Linear optics to scalable photonic quantum computing," arXiv:2501.02513 (2025).
[22]O. Lib and Y. Bromberg, "Resource-efficient photonic quantum computation with high-dimensional cluster states," Nat. Photonics **18**, 1218 (2024).
[23]S. Sepúlveda, A. Cravero, G. Fonseca, and L. Antonelli, "Systematic review on requirements engineering in quantum computing: Insights and future directions," Electronics **13**, 2989 (2024).
[24]A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross *et al.*, "Quantum computing with Qiskit," arXiv:2405.08810 (2024).
[25]V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi *et al.*, "PennyLane: Automatic differentiation of hybrid quantum-classical computations," arXiv:1811.04968 (2018).
[26]Y. Fan, J. Liu, X. Zeng, Z. Xu, H. Shang, Z. Li, and J. Yang, "Q^2Chemistry: A quantum computation platform for quantum chemistry," JUSTC **52**, 2 (2022).
[27]W. Li, J. Allcock, L. Cheng, S.-X. Zhang, Y.-Q. Chen, J. P. Mailoa, Z. Shuai, and S. Zhang, "TenCirChem: An efficient quantum computational chemistry package for the NISQ era," J. Chem. Theory Comput. **19**, 3966 (2023).
[28]Y. Huang, Q. Li, X. Hou, R. Wu, M. H. Yung, A. Bayat, and X. Wang, "Robust resource-efficient quantum variational ansatz through an evolutionary algorithm," Phys. Rev. A **105**, 052414 (2022).
[29]W. Yu, J. Sun, Z. Han, and X. Yuan, "Robust and efficient Hamiltonian learning," Quantum **7**, 1045 (2023).
[30]N. K. Bhasin, S. Kadyan, K. Santosh, R. HP, R. Changala, and B. K. Bala, "Enhancing quantum machine learning algorithms for optimized financial portfolio management," in *2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)* (IEEE, 2024), pp. 1–7.
[31]S. Anagolum, N. Alavisamani, P. Das, M. Qureshi, and Y. Shi, "Élivágar: Efficient quantum circuit search for classification," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (ACM, 2024), Vol. 2, pp. 336–353.
[32]S. M. Ardelean and M. Udrescu, "Hybrid quantum search with genetic algorithm optimization," PeerJ Comput. Sci. **10**, e2210 (2024).
[33]S. M. Venkatesh, A. Macaluso, M. Nuske, M. Klusch, and A. Dengel, "Qubit-efficient variational quantum algorithms for image segmentation," in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2024), Vol. 1, pp. 450–456.
[34]Q. Liang, Y. Zhou, A. Dalal, and P. Johnson, "Modeling the performance of early fault-tolerant quantum algorithms," Phys. Rev. Res. **6**, 023118 (2024).
[35]A. Y. Kitaev, "Quantum measurements and the Abelian Stabilizer Problem," arXiv:quant-ph/9511026 [quant-ph] (1995).

[36]A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, "Simulated quantum computation of molecular energies," Science **309**, 1704 (2005).
[37]S. Lloyd, "Universal quantum simulators," Science **273**, 1073 (1996).
[38]H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," Nat. Commun. **10**, 3007 (2019).
[39]P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan *et al.*, "Scalable quantum simulation of molecular energies," Phys. Rev. X **6**, 031007 (2016).
[40]A. Peruzzo, J. McClean, P. Shadbolt, M. H. Yung, X. Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," Nat. Commun. **5**, 4213 (2014).
[41]Z. T. Li, F. X. Meng, H. Zeng, Z. R. Gong, Z. C. Zhang, and X. T. Yu, "A gradient-cost multiobjective alternate framework for variational quantum eigensolver with variable ansatz," Adv. Quantum Technol. **6**, 2200130 (2023).
[42]Z. T. Li, F. X. Meng, H. Zeng, Z. C. Zhang, and X. T. Yu, "An efficient gradient sensitive alternate framework for VQE with variable ansatz," arXiv:2205.03031 [quant-ph] (2022).
[43]C. Feniou, M. Hassan, D. Traoré, E. Giner, Y. Maday, and J. P. Piquemal, "Overlap-ADAPT-VQE: Practical quantum chemistry on quantum computers via overlap-guided compact Ansätze," Commun. Phys. **6**, 192 (2023).
[44]N. Vaquero-Sabater, A. Carreras, and D. Casanova, "Pruned-ADAPT-VQE: Compacting molecular ansätze by removing irrelevant operators," arXiv:2504.04652 [quant-ph] (2025).
[45]H. L. Tang, V. O. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou, "qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor," PRX Quantum **2**, 020310 (2021).
[46]P. G. Anastasiou, Y. Chen, N. J. Mayhall, E. Barnes, and S. E. Economou, "TETRIS-ADAPT-VQE: An adaptive algorithm that yields shallower, denser circuit ansätze," Phys. Rev. Res. **6**, 013254 (2024).
[47]M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai *et al.*, "Variational quantum algorithms," Nat. Rev. Phys. **3**, 625 (2021).
[48]W. Li, Y. Ge, S. X. Zhang, Y. Q. Chen, and S. Zhang, "Efficient and robust parameter optimization of the unitary coupled-cluster ansatz," J. Chem. Theory Comput. **20**, 3683 (2024).
[49]M. J. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in Optimization and Numerical Analysis*, edited by S. Gomez and J. P. Hennart (Springer Netherlands, Dordrecht, 1994), pp. 51–67.
[50]J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," IEEE Trans. Autom. Control **37**, 332 (1992).
[51]M. J. D. Powell, "An efficient method for finding the minimum of a function of several variables without calculating derivatives," Comput. J. **7**, 155 (1964).
[52]J. A. Nelder and R. Mead, "A simplex method for function minimization," Comput. J. **7**, 308 (1965).
[53]A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, "Error mitigation extends the computational reach of a noisy quantum processor," Nature **567**, 491 (2019).
[54]A. J. McCaskey, Z. P. Parks, J. Jakowski, S. V. Moore, T. D. Morris, T. S. Humble, and R. C. Pooser, "Quantum chemistry as a benchmark for near-term quantum computers," npj Quantum Inf. **5**, 99 (2019).
[55]A. Eddins, M. Motta, T. P. Gujarati, S. Bravyi, A. Mezzacapo, C. Hadfield, and S. Sheldon, "Doubling the size of quantum simulators by entanglement forging," PRX Quantum **3**, 010309 (2022).
[56]X. Bonet-Monroig, H. Wang, D. Vermetten, B. Senjean, C. Moussa, T. Bäck, V. Dunjko, and T. E. O'Brien, "Performance comparison of optimization methods on variational quantum algorithms," Phys. Rev. A **107**, 032407 (2023).
[57]Y. S. Yordanov, V. Armaos, C. H. W. Barnes, and D. R. M. Arvidsson-Shukur, "Qubit-excitation-based adaptive variational quantum eigensolver," Commun. Phys. **4**, 228 (2021).