IEEE Transactions on Quantum Engineering

Received September 29, 2021; revised December 20, 2021; accepted December 21, 2021; date of publication January 4, 2022; date of current version February 17, 2022.

Digital Object Identifier 10.1109/TQE.2021.3140190

Layer VQE: A Variational Approach for Combinatorial Optimization on Noisy Quantum Computers

XIAOYUAN LIU[1] (Member, IEEE), ANTHONY ANGONE[2], RUSLAN SHAYDULIN[3] (Member, IEEE), ILYA SAFRO[1] (Member, IEEE), YURI ALEXEEV[4] (Senior Member, IEEE), AND LUKASZ CINCIO[5]

[1]Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716 USA
[2]School of Computing, Clemson University, Clemson, SC 29634 USA
[3]Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL 60439 USA
[4]Computational Science Division, Argonne National Laboratory, Lemont, IL 60439 USA
[5]Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545 USA

Corresponding author: Xiaoyuan Liu (e-mail: joeyxliu@udel.edu).

The work of Xiaoyuan Liu, Anthony Angone, Ruslan Shaydulin, Ilya Safro, and Yuri Alexeev was supported in part by the Defense Advanced Research Projects Agency. The work of Ruslan Shaydulin and Yuri Alexeev was also supported by the Laboratory Directed Research and Development through funding from Argonne National Laboratory, provided by the Director, Office of Science, of the U.S. Department of Energy under Contract DE-AC02-06CH11357. The work of Ruslan Shaydulin was also supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research for Quantum Computing program. The work of Lukasz Cincio was supported in part by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under Project 20200056DR and in part by the U.S. DOE, Office of Science, Office of Advanced Scientific Computing Research, under the Accelerated Research in Quantum Computing program.

ABSTRACT Combinatorial optimization on near-term quantum devices is a promising path to demonstrating quantum advantage. However, the capabilities of these devices are constrained by high noise or error rates. In this article, inspired by the variational quantum eigensolver (VQE), we propose an iterative layer VQE (L-VQE) approach. We present a large-scale numerical study, simulating circuits with up to 40 qubits and 352 parameters, that demonstrates the potential of the proposed approach. We evaluate quantum optimization heuristics on the problem of detecting multiple communities in networks, for which we introduce a novel qubit-frugal formulation. We numerically compare L-VQE with the quantum approximate optimization algorithm (QAOA) and demonstrate that QAOA achieves lower approximation ratios while requiring significantly deeper circuits. We show that L-VQE is more robust to finite sampling errors and has a higher chance of finding the solution as compared with standard VQE approaches. Our simulation results show that L-VQE performs well under realistic hardware noise.

INDEX TERMS Combinatorial optimization, hybrid quantum-classical algorithm, quantum optimization.

I. INTRODUCTION

Recent advances in quantum computing hardware open the possibility of demonstrating quantum advantage in practical applications [1], [2]. A promising target application domain is combinatorial optimization with problems becoming classically intractable (in the current state of theory) to solve exactly even for moderately sized instances. This situation suggests that the requirement for the number of qubits needed to tackle certain classically hard combinatorial optimization problems is relatively low, leading to the possibility of noisy intermediate-scale quantum (NISQ) [3] devices becoming competitive with classical state-of-the-art methods for such problems.

Near-term quantum devices are expected to have high noise levels, and only partial error mitigation is currently possible. This situation leads to a constraint on the maximum depth of the quantum circuit that can be reliably executed on NISQ devices. This constraint motivated the development of a number of hybrid quantum-classical algorithms for optimization, most notably the quantum approximate optimization algorithm (QAOA) [4]–[6] and variational quantum algorithms for optimization [7], [8]. These algorithms execute only a short parameterized circuit on the quantum computer and use a classical outer-loop procedure to find "good" parameters [9]. The short parameterized circuit is often referred to as the ansatz. The goal of the outer-loop procedure, in

general, is to find parameters such that the output of the quantum circuit includes high-quality solutions to the combinatorial optimization problem being solved.

The choice of the ansatz is a key problem in hybrid algorithms. One major concern is the expressivity of the chosen ansatz. The ansatz has to be sufficiently expressive, meaning that there should exist parameters with which the ansatz prepares a state suitably close to the solution of the problem (note that we are not referring to the expressibility formally defined in [10]). Another thing to consider is the optimization of the parameters, sufficiently good parameters have to be tractable to find [10].

For combinatorial optimization problems, the solution is classical; in other words, it is a computational basis state. Therefore, the expressivity of the ansatz reduces the ability to prepare a state with a sufficiently large overlap with the computational basis state encoding the solution of the problem. This means that the ansatz can be sufficiently expressive without generating any entanglement or having any quantum properties whatsoever; one layer of single-qubit rotations is sufficient to prepare an arbitrary computational basis state. However, finding good parameters may be challenging for such ansätze. Their structure leads to localized optimization, which is prone to local minima. As we discuss in the following, class of ansätze may be extended to enhance the ability to find good parameters by introducing a correlation between distant parts of the system. A commonly used class of highly expressive ansätze are those with alternating layers of single-qubit and two-qubit gates, where the two-qubit gates are aligned with the connectivity available on the hardware. These ansätze are known as quantum neural networks [11] or hardware-efficient ansätze [12]. An alternative and "natural" approach is the Hamiltonian-evolution ansatz used in QAOA. Such ansätze can be less expressive; however, since the state it has to prepare is a nontrivial entangled state due to the symmetry-preserving properties of the ansatz [13]. This observation has been used by Bravyi *et al.* [14] to show that because of the $\mathbb{Z}_2$ symmetry of the ansatz, QAOA with the constant depth is outperformed by the classical Goemans–Williamson algorithm for MaxCut. As a result, QAOA needs a comparatively large circuit depth to achieve the same (classical) expressivity as compared with hardware-efficient ansatz.

For ansätze with a large number of parameters, the high-quality parameters are typically found by using a classical optimizer. Thus, the second criterion, the ability to find sufficiently good parameters, is typically framed in terms of the cost function landscape that the classical outer-loop routine has to optimize over. Recent results show that highly expressive ansätze, such as hardware-efficient ansätze, suffer from "barren plateaus" in the optimization landscape, making finding high-quality parameters intractable [11], [15]–[24]. At the same time, a series of recent results show that because of the structured nature of the ansatz used in QAOA, one may be able to find high-quality parameters by using machine learning approaches [25]–[27] or by restricting the parameters to a specific physically motivated class [28]–[30].

In this article, we propose a practical approach to combinatorial optimization on near-term quantum computers. We introduce an iterative approach, which we call Layer VQE (L-VQE), inspired by recent advances in hybrid quantum-classical algorithms with an adaptive ansatz [31]–[33]. In L-VQE, we start with one layer of parameterized rotations and increment the size of the ansatz systematically by introducing entangling gates and additional parameterized rotations. To heuristically decrease the likelihood of getting trapped in a local optimum of the parameters, we increment the ansatz before reaching convergence. To guarantee that at each step the quality of the solution does not decrease, we initialize the added ansatz such that it evaluates to identity. We work with qubits aligned in a chain and assume nearest neighbor connectivity, which is a reasonable assumption as the most common hardware topology includes line as a subgraph. Restricting ourselves to this class of problems allows us to benchmark the proposed methods on large problems in simulation by using tensor network techniques. We expect that in practical applications on real quantum hardware, one would organize ansatz layers according to (typically two-dimensional) qubit connectivity to further enhance circuit expressiveness. Quantum circuits on such layouts cannot be, in general, classically efficiently simulated and are, thus, not considered in the present study.

Fig. 1 gives a schematic presentation of L-VQE. We study the algorithm for the problem of detecting k communities in networks, and we propose a novel qubit-frugal formulation with many-body interactions in the Hamiltonian. For a network with n nodes, $n\lceil \log_2 k \rceil$ qubits are required for the circuit. We present a large-scale numerical study of the proposed approach, simulating circuits with up to 40 qubits and 352 rotational gates (i.e., parameters). Our numerical simulation results show that the proposed approach achieves a higher approximation ratio compared with QAOA while requiring significantly lower circuit depth. The proposed approach is more robust to finite sampling error (i.e., if the objective value is not known exactly and is estimated by drawing samples from the quantum state) and performs better than hybrid approaches with a fixed ansatz. Moreover, we show that the proposed approach performs well under realistic hardware noise by using a trapped ion noisy quantum simulator.

The rest of this article is organized as follows. In Section II, we review the relevant background of solving combinatorial optimizations on quantum computers. In Section III, we review related work. Section IV introduces our L-VQE approach, and in Section V, we discuss our novel formulation of the k-community detection problem. Section VI presents our numerical simulation results, and finally in Section VII, we summarize our conclusions.

IEEE Transactions on Quantum Engineering

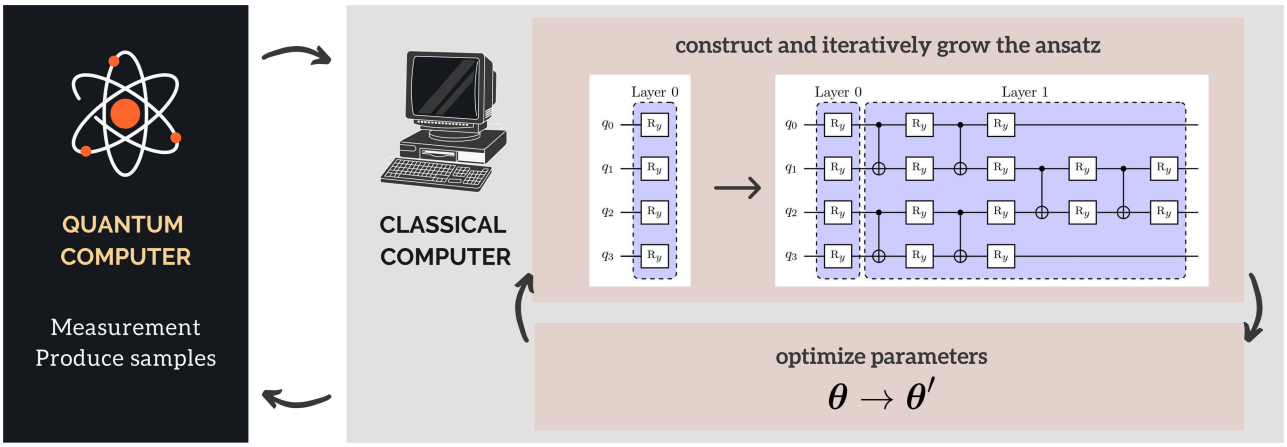

FIGURE 1. Layer-VQE: start from a simple and shallow ansatz with one R_Y act on each qubit; optimize and update the parameters; after some predefined number of iterations, increment the size of the ansatz, and optimize and update all parameters. The ansatz can be incremented multiple times.

II. BACKGROUND

We begin by briefly reviewing our notion of combinatorial optimization on quantum computers and relevant concepts. Suppose we have an objective function $C(x)$ defined on the Boolean cube $x = \{x_i\}_{i=1}^n \in \{0, 1\}^n$ and a corresponding optimization problem

$$\max_{x\in\{0,1\}^n} C(x) \tag{1}$$

where the objective function $C(x)$ can be formulated in the following form:

$$C(x) = \sum_q w_q \prod_{i\in q} x_i \prod_{j\notin q}(1 - x_j). \tag{2}$$

Here, $q \subset \{1, 2, \ldots, n\}$ are given index sets, and w_q are given coefficients. The objective function $C(x)$ is said to be *faithfully represented* by a Hamiltonian $\mathcal{H}$ if it acts as $\mathcal{H}|x\rangle = C(x)|x\rangle$ for each $x \in \{0, 1\}^n$. For a function given in the form (2), such a Hamiltonian representation can be constructed by substituting every x_i with the matrix $x_i \rightarrow \frac{1}{2}(\mathrm{I} - \mathrm{Z}_i)$, where I is the identity matrix and Z_i is the Pauli Z operator that acts on qubit i

$$\mathcal{H} = \sum_q w_q \prod_{i\in q} \frac{\mathrm{I} - \mathrm{Z}_i}{2} \prod_{j\in q^C} \frac{\mathrm{I} - \mathrm{Z}_j}{2}. \tag{3}$$

Note that the operator $\mathcal{H} \in \mathbb{C}^{2^n}$ is never constructed explicitly. Instead, we construct a compact representation of it as a combination of Pauli Z operators.

A. COMBINATORIAL OPTIMIZATION ON NEAR-TERM QUANTUM COMPUTERS

The two most prominent candidate algorithms for combinatorial optimization on noisy near-term quantum computers are the variational quantum eigensolver (VQE, originally proposed in the context of quantum chemistry [34]) and the QAOA [4]. Both algorithms are hybrid quantum-classical algorithms that combine a parameterized trial state $|\psi(\boldsymbol{\theta})\rangle$ prepared on a quantum computer with a classical routine used to find high-quality parameters $\boldsymbol{\theta}$. The goal is to find parameters $\boldsymbol{\theta}$ such that when the state $|\psi(\boldsymbol{\theta})\rangle$ is measured, the measurement result corresponds to a good solution to the classical optimization problem. The parameterized trial state $|\psi(\boldsymbol{\theta})\rangle$ is commonly called the ansatz.

In VQE, for optimization, the ansatz is frequently tailored to the hardware [8], [35], and the parameters $\boldsymbol{\theta}$ are found by using a classical outer-loop optimizer. The expectation value $\langle\psi(\boldsymbol{\theta})|\mathcal{H}|\psi(\boldsymbol{\theta})\rangle$ is commonly used as the metric for the optimizer, although other approaches have been suggested [36]. QAOA uses a problem-dependent ansatz given by

$$|\psi_p(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = e^{-i\beta_p B} e^{-i\gamma_p \mathcal{H}} \cdots e^{-i\beta_1 B} e^{-i\gamma_1 \mathcal{H}} |+\rangle^{\otimes n} \tag{4}$$

where $B = \sum_{i=1}^n \mathrm{x}_i$ is the mixing Hamiltonian, x_i is the Pauli X operator acting on qubit i, $\mathcal{H}$ is the Hamiltonian faithfully representing the objective function, and p is a parameter controlling the depth. The special structure of the QAOA ansatz enables finding high-quality parameters $\boldsymbol{\gamma}, \boldsymbol{\beta}$ purely classically in many settings [4], [37], [38] or using very few iterations of the outer-loop optimizer [25], [39], [40].

We evaluate the quality of the final quantum state $|\psi(\boldsymbol{\theta})\rangle$ by computing the approximation ratio ρ defined as follows:

$$\rho = \frac{\langle\psi(\boldsymbol{\theta})|\mathcal{H}|\psi(\boldsymbol{\theta})\rangle}{C_{\mathrm{bkv}}} \tag{5}$$

where C_{bkv} is either the global optimum of the objective function $C(x)$ if available, or the best known value otherwise. We defined approximation ratio with respect to the best known value since the global optimal $\max_{x\in\{0,1\}^n} C(x)$ may not be accessible for sufficiently large problem instances.

B. k-COMMUNITY DETECTION

The k-community detection, also known as modularity clustering, is a famous problem in network science. The goal is to partition network nodes into k communities (also known as clusters or parts) such that the modularity metric [41] defined in (6) is maximized. There are several graph partitioning problems whose goal is to split the graph nodes into

disjoint k parts in such a way that most edges will connect the nodes within the parts and the number of edges that span two parts is minimized [42]. The modularity optimization is one of them. The modularity metric measures how far is the number of edges that appear within the parts from that in the random graph model [43] with the same number of nodes and expected number of edges. This metric has been confirmed to reflect the properties of community existence in many applications. Intuitively, when the assignment of nodes to partitions produces large modularity, the partitions are likely to be real communities in many different applications, including social networks and biological and engineered systems.

For a formal definition, let $G = (V, E)$ be an undirected simple graph with $|V| = n$ nodes and $|E| = m$ edges. The adjacency matrix of G is denoted by $A = \{A_{u,v}\}_{1 \le u,v \le n}$, where $A_{u,v} = 1$ if there is an edge between node u and node v, and 0 otherwise. The degree of a node v is denoted by d_v. A k-community clustering $\mathcal{C} = \{C_1, \ldots, C_k\}$ is a partition of V into k disjoint sets, namely, $\bigcup_{i=1}^{k} C_i = V$, and $C_i \bigcap C_j = \varnothing$ for all $1 \le i \ne j \le k$. Furthermore, c_v denotes the membership of node v for a given clustering, i.e., if $v \in C_i$, then $c_v = i$. The modularity of a clustering $\mathcal{C}$ is given by

$$\mathcal{Q}(\mathcal{C}) = \frac{1}{2m} \sum_{u,v=1}^{n} B_{u,v} \delta(c_u, c_v) \tag{6}$$

where the modularity matrix B is given by $B_{u,v} = A_{u,v} - \frac{d_u d_v}{2m}$, $1 \le u, v \le n$, and δ is the Kronecker delta

$$\delta(c_u, c_v) = \begin{cases} 1, & \text{if } c_u = c_v \\ 0, & \text{otherwise} \end{cases}. \tag{7}$$

Our goal is to find the clustering $\mathcal{C}^*$ such that the modularity is maximized

$$\mathcal{C}^* = \text{argmax}_{\mathcal{C}}\, \mathcal{Q}(\mathcal{C}).$$

The problem has applications in chemistry [44], biology [45], social sciences [46], and other fields. The task of solving the modularity maximization problem to optimality is NP-complete [47].

Community detection has been extensively studied classically [41], [48], as well as by using the D-Wave quantum annealer [49]–[52] and QAOA [50], [52], [53]. In these hybrid quantum-classical approaches, the optimization problem is encoded as an Ising model Hamiltonian that has only two-body terms. In the formulations, for a graph with n nodes, solving the two-community modularity maximization problem requires n qubits, where each qubit encodes the membership of a node. For the k-community problem, to encode the membership of each node, one will need to associate k qubits to each node while introducing quadratic penalty constraints into the Ising Hamiltonian to enforce that each node belongs to only one community. The formulation requires kn qubits.

III. RELATED WORK

The question of ansatz choice is central to the success of hybrid quantum-classical methods introduced in Section II-A. In VQE, the choice of the ansatz determines the expressivity of the trial state and the hardness of finding parameters; therefore, the quality of VQE is only as good as the ansatz. Different strategies of parameterizing the ansatz and updating the parameters will also affect the performance of the algorithm. While being able to reach any state requires a circuit with exponential depth, shallow circuits are preferred in applications, especially if the goal is to run the circuits on modern NISQ devices. McClean *et al.* [11] show that with random parameterized circuit initialization, the exponential dimension of the Hilbert space and the gradient estimation complexity make the optimization impossible for deep circuits. Moreover, Wang *et al.* [54] show that another type of "barren plateau" is induced by hardware noise. More specifically, given local Pauli noise, the gradient vanishes exponentially with the depth of the circuits. Similar results have been demonstrated for QAOA [55].

There are fundamentally two ways to approach the problem of designing compact ansätze. We classify ansätze into two groups. The first way is to start with physics- or chemistry-inspired ansätze generated by a Hamiltonian. These circuits in the first group typically have high depth. For example, in quantum chemistry, we would use the unitary coupled-cluster method ansatz. In particular, unitary coupled clusters with singles and doubles (UCCSD) [56] ansatz can be used in VQE simulations [34]. Each parameter in UCCSD ansatz parametrizes a coupled-cluster amplitude for each fermionic excitation from a reference state, either single or double. Since it is an accurate ansatz, it has many redundant and unimportant parameters corresponding to the excited states that are not contributing to the ground energy giving a lot of room for optimization. One strong idea is to use symmetry to generate compact circuits [57] and another is to use MP2 amplitude to screen out UCCSD parameters [58]. The issue is that there is a limit to the reduction, and typically, such circuits are still too deep to execute on NISQ devices, especially for large and moderate size problems. In the second group, they are the so-called "hardware-efficient" (HE) ansätze. These ansätze contain sequences of parametrized single-qubit and two-qubit gates that can be easily implemented on NISQ devices because of their by-design compact nature [12], [57], [59]–[61]. The key aspect is that no information about the physics or chemistry of the system is used. The major plus of HE ansätze is that they are very expressible and the circuits to implement them are shallower compared to the first group of ansätze such as UCCSD and contain a much smaller number of two-qubit gates. The downside of HE ansätze is that they can have too many parameters to optimize and suffer from the barren plateaus [11], [15], [54] problem, which was discussed earlier in this article.

Another way to build ansätze is to dynamically generate them using some criteria (for example, using largest

IEEE Transactions on Quantum Engineering

gradients or resolutions of identity as a criterion to minimize total energy) [31], [62]–[67]. We will refer to these ansätze as "iterative ansätze." In quantum chemistry, one of the first proposed iterative ansätze is ADAPT-VQE [31], where fermionic operators are added to the ansatz based on the energy gradients with respect to variational parameters. Later, qubit-ADAPT-VQE [62] method was invented, where the fermionic operators are broken down into Pauli strings and used as building blocks for constructing an ansatz. Other important contributions to this field were made based on Yordanov *et al.* [68] work to use the so-called qubit excitations [69] instead of the fermionic excitations in ADAPT-VQE simulations and qubit-coupled cluster [63] method and its iterative version [64], where the ansatz is constructed directly in the qubit space. In general, the iterative circuits are somewhere in the middle between Hamiltonian ansätze and HE ansätze in terms of depth and the number of parameters. In the same spirit, Zhu *et al.* propose an adaptive version of QAOA, called ADAPT-QAOA [32]. Compared with the standard QAOA ansatz, which alternates between the predefined exponentiated cost and mixing Hamiltonian operator, ADAPT-QAOA grows the ansatz with two operators at a time. It also uses a gradient criterion to select the mixing operator from a predefined operator pool. On a class of MaxCut graph problems, ADAPT-QAOA demonstrates faster convergence while also reducing the number of optimization parameters and the CNOT gate counts, compared with standard QAOA. Our L-VQE approach can be considered as a new evolution of iterative ansätze. The traditional iterative ansätze are still based on using insights from the physics and chemistry of the problem with some criterion to build ansätze. In L-VQE, HE ideas are used to grow circuits. As a result, our circuits are more compact than Hamiltonian or iterative ansätze.

The optimization of the parameters is also a key component in hybrid quantum-classical algorithms. For example, in quantum machine learning, Skolik *et al.* [33] propose a layerwise learning strategy that grows the circuit depth incrementally during optimization and only updates subsets of parameters in training. However, a recent paper [22] shows that this type of layerwise training strategy, namely, training a circuit piecewise in sequence, could encounter abrupt transitions in the training process as the depth of the circuit grows.

IV. LAYER VQE

We advocate an iterative hybrid approach to quantum optimization on NISQ devices: L-VQE. L-VQE combines ideas from recent developments in adaptive variational algorithms, such as [31]–[33]. In this section, we describe L-VQE in detail.

Suppose we use a problem encoding that requires n qubits. We start the algorithm with an ansatz with no entangling gates and one R_y gate acting on each qubit, where R_y is the single-qubit rotation through an angle θ around the y-axis, the unitary matrix is defined as $\mathrm{R}_y(\theta) \equiv e^{-i\frac{\theta}{2}\mathrm{Y}}$, and Y is the Pauli Y operator. The parameters of these R_y gates are initialized uniformly randomly on $[0, 2\pi]$. We denote the parameters for this layer of gates (Layer *0* in Fig. 2) as $\boldsymbol{\theta}_0$ and the layer as $U_0(\boldsymbol{\theta}_0)$. The quantum state after applying the circuit to the initial state $|0\rangle$ is denoted as $|\psi_0(\boldsymbol{\theta}_0)\rangle \equiv U_0(\boldsymbol{\theta}_0)|0\rangle$. We then proceed to the conventional VQE routine and iteratively update the parameters $\boldsymbol{\theta}_0$ to minimize the cost function $\langle\psi_0(\boldsymbol{\theta}_0)|\mathcal{H}|\psi_0(\boldsymbol{\theta}_0)\rangle$. In conventional VQE, this iterative procedure is run until convergence; but in L-VQE, we stop after a fixed number of iterations and then add another set of gates to the ansatz. The conventional strategy can indeed produce a better result at this step, but after adding the new set of gates, it may more easily get trapped in a local minimum in the subsequent optimization procedure. In our experiments, the number of iterations is picked empirically and increases linearly as the system size grows.

Algorithm 1: L-VQE With ℓ Layers.

1: Initialize the ansatz with one R_y acting on each qubit.
2: Update the parameters to minimize $\langle\psi_0(\boldsymbol{\theta})|\mathcal{H}|\psi_0(\boldsymbol{\theta})\rangle$; stop after k_0 iterations (before reaching convergence).
3: **for** $l = 1, \ldots, \ell$ **do**
4: Add a new layer to the ansatz and initialize it such that it evaluates to identity.
5: Update all parameters to minimize $\langle\psi_\ell(\boldsymbol{\theta})|\mathcal{H}|\psi_\ell(\boldsymbol{\theta})\rangle$; stop after k_ℓ iterations (before reaching convergence).
6: **end for**
7: Update all parameters to minimize $\langle\psi(\boldsymbol{\theta})|\mathcal{H}|\psi(\boldsymbol{\theta})\rangle$ until convergence.

The newly added set of gates includes the R_y gates and CNOT gates that act on nearest-neighbor qubits. Another way to describe this whole procedure is that we embed the obtained parameterized circuit into a deeper circuit. We denote this newly added layer of the circuit by $U_1(\boldsymbol{\theta}_1)$ (Layer *1* in Fig. 2). The newly added parameters $\boldsymbol{\theta}_1$ are initialized as zero. Note that here since $\mathrm{R}_y(0) = \mathrm{I}$ and $\mathrm{CNOT}^2 = \mathrm{I}$, where I is the identity matrix, the quantum state becomes

$$|\psi_1(\boldsymbol{\theta}_0, \boldsymbol{\theta}_1)\rangle \equiv U_1(\boldsymbol{\theta}_1)U_0(\boldsymbol{\theta}_0)|0\rangle = U_0(\boldsymbol{\theta}_0)|0\rangle = |\psi_0(\boldsymbol{\theta}_0)\rangle. \tag{8}$$

Therefore, initializing the newly added parameters as zeros guarantees that the cost function that we are optimizing will not change after adding this new layer. At this point, we can either let the optimization run until convergence or repeat the previous process, stop at a fixed number of iterations, and add another set of gates to the circuit and then optimize. The pseudocode of the algorithm is presented in Algorithm 1.

In simulations, the cost function $\langle\psi(\boldsymbol{\theta})|\mathcal{H}|\psi(\boldsymbol{\theta})\rangle$ can be evaluated exactly. When executing the algorithm on hardware, we have to repeat the state preparation and measurement multiple times to generate a number of samples and use the samples to estimate the cost function, introducing an error

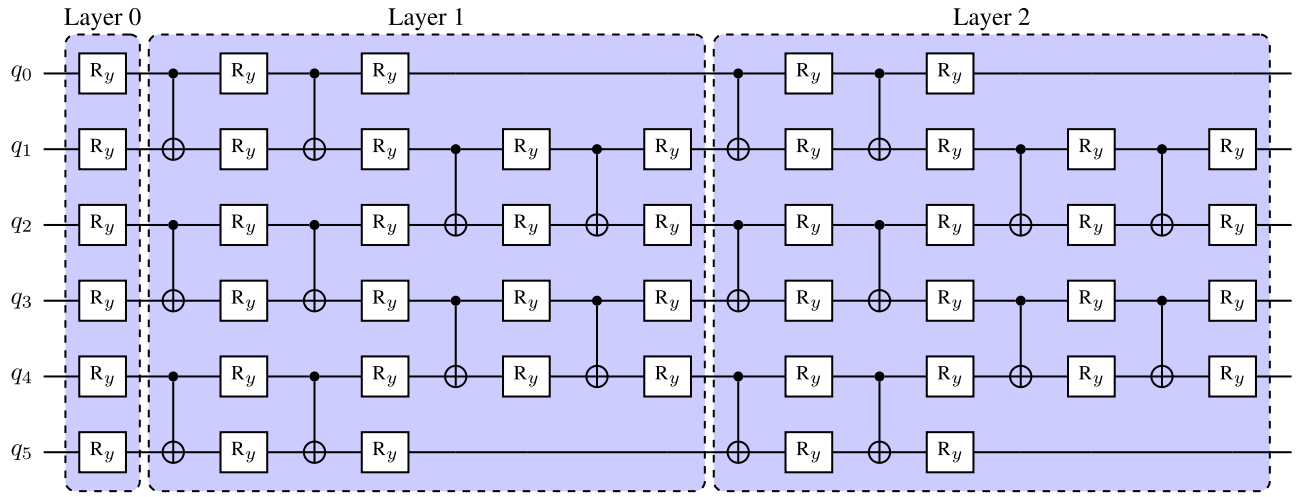

FIGURE 2. L-VQE ansatz for a six-qubit quantum state. R_y denotes rotation around the y-axis defined as $R_y(\theta) \equiv e^{-i\frac{\theta}{2}Y}$. Every R_y contains a parameter that is optimized over in the outer loop.

due to the finite number of samples. In our experiments, we investigate the performance of the algorithm in both cases.

The solution of a combinatorial optimization problem is classical, i.e., it is a computational basis state. Therefore, suppose we can find optimal parameters for the ansatz; the ansatz contains only single-qubit rotations on each qubit, thus should be able to prepare the state that contains the computational basis state we want. However, ansatz as such is prone to local minima. Therefore, in L-VQE, we start from an ansatz that only contains single-qubit rotation gates, then iteratively add entanglement to the ansatz to help the optimization process. Another motivation is to speed up the optimization process. Random initialization of deep circuits leads to difficult optimization, so we "pretrain" with shallower circuits. Importantly, we do not converge the optimization for a single layer (or other shallow circuits). Indeed, this would cause the algorithm to get stuck in a local minimum, which might be hard to escape after a new layer is added. Instead, we added a new layer before convergence is reached to avoid the above-mentioned issues. Empirical results show that this strategy increases the probability of finding the ground state or finding the state that is sufficiently close to the ground state.

Similar to ADAPT-VQE [31] and ADAPT-QAOA [32], we grow the size of the ansatz as we iteratively update the parameters. The added parameterized ansatz is initialized such that the new circuit parts evaluate to identity in order to avoid deterioration of the optimization. In ADAPT-VQE and ADAPT-QAOA, however, the algorithm will identify an operator that has the largest gradient from a collection of operators and then adds this operator to the ansatz. In L-VQE, we define the newly added ansatz upfront.

As discussed in [22], the conjecture that a circuit can be trained piecewise turns out not to always be true. In the finite setting, there are abrupt transitions in the ability of quantum circuits to be trained. In layerwise learning [33], when adding a new set of layers, a part of the previous layer's parameters is frozen, and additional optimization sweeps are performed on subsets of parameters. Each layer contains rotation gates on each qubit and also operators that connect the qubits. L-VQE, on the other hand, optimizes all parameters where none of the previous layers are fixed. Furthermore, the initial ansatz for L-VQE only contains rotation gates on each qubit. Therefore, we start from a product state, where no entanglement is involved. After some iterations, we add entanglement to help the optimization process. This is different from [33], where the initial layer already contains operators connecting qubits. This strategy may reduce the limitations of the lack of layerwise trainability [22]. In addition, layerwise learning is a general approach; for L-VQE, we focus on solving combinatorial optimization problems.

V. k-COMMUNITY DETECTION

We propose a novel qubit-frugal formulation for the k-community detection problem. When the problem is to divide the network into two communities, namely, with $k = 2$, we can associate a binary variable with each node $v \in V$ such that

$$x_v = \begin{cases} 1, & \text{if } c_v = 1 \\ 0, & \text{if } c_v = 2. \end{cases}$$

Then, we can rewrite the Kronecker delta (7) in terms of these binary variables

$$\delta(c_u, c_v) = \delta(x_u, x_v) = 2x_u x_v - x_u - x_v + 1. \tag{9}$$

Plugging (9) into (6) leads to the expression of modularity:

$$\mathcal{Q}(\mathcal{C}) = \frac{1}{2m} \sum_{u,v=1}^{n} B_{u,v}(2x_u x_v - x_u - x_v + 1).$$

For larger k, we can use a binary encoding by associating $N = \lceil \log_2 k \rceil$ binary variables $\{x_{j,v}\}_{j=1}^{N} \subset \{0, 1\}^N$ with each

IEEE Transactions on Quantum Engineering

node $v \in V$. We can rewrite the membership of node v as

$$c_v = \sum_{j=1}^{N} 2^{j-1} x_{j,v}.$$

Again, we can rewrite the Kronecker delta (7) in terms of these binary variables

$$\delta(c_u, c_v) = \prod_{j=1}^{N} \delta(x_{j,u}, x_{j,v}) = \prod_{j=1}^{N} (2x_{j,u}x_{j,v} - x_{j,u} - x_{j,v} + 1). \tag{10}$$

Plugging (10) into (6), we obtain for the modularity

$$\mathcal{Q}(\mathcal{C}) = \frac{1}{2m} \sum_{u,v=1}^{n} B_{u,v} \prod_{j=1}^{N} (2x_{j,u}x_{j,v} - x_{j,u} - x_{j,v} + 1). \tag{11}$$

Following the construction described in Section II, maximizing the modularity in (11) can be formulated in terms of finding the ground state of the following Hamiltonian:

$$\mathcal{H} = -\frac{1}{2m} \sum_{u,v=1}^{n} B_{u,v} \prod_{j=1}^{N} \frac{I + \mathrm{z}_{j,u}\mathrm{z}_{j,v}}{2} \tag{12}$$

where binary variables $x_{j,v}$ have been substituted with $\frac{1}{2}(I - \mathrm{z}_{j,v})$, $\forall j \in \{1, 2, \ldots, N\}$, $\forall v \in V$. Here, $\mathrm{z}_{j,v}$ is the Pauli z operator that acts on qubit (j, v).

Other formulations have been proposed to tackle the problem of specific quantum architecture. Ushijima-Mwesigwa *et al.* [51] use an Ising Hamiltonian formulation to detect two communities using quantum annealing on the D-Wave system, which requires n qubits. Negre *et al.* [49] extend it to detect k communities, which requires kn qubits. In contrast, the Hamiltonian we propose in this work requires only $n\lceil \log_2 k \rceil$ qubits, thanks to the encoding introduced above. Note that the many-body interactions present in the proposed Hamiltonian do not introduce significant overhead, as simulating a product of N Pauli z operators requires only $2(N-1)$ CNOTs.

VI. EXPERIMENTS

In this section, we present the numerical results. Since QAOA is considered the leading approach for combinatorial optimization on NISQ devices, we begin in Section VI-A with a numerical comparison of L-VQE and QAOA. We then compare L-VQE with the second leading approach, which is VQE in Section VI-B. To highlight the potential of the proposed L-VQE approach on NISQ devices, we present some further evidence in Section VI-C. This includes a scalability analysis and simulation results of L-VQE on a trapped ion noisy quantum simulator with a realistic noise level. To highlight the importance of entanglement for optimization, in Section VI-D, we present results comparing VQE with and without entanglement.

A. L-VQE AND QAOA

For the first set of experiments, we run simulations for the L-VQE and QAOA algorithms with the proposed Hamiltonian (12). The goal is to find a clustering of up to four communities that maximize the modularity. We are, thus, simulating $2n$ qubits for a graph with n nodes. For L-VQE, we run our simulations of the quantum circuits in MATLAB. We use matrix product states (MPS) techniques to simulate quantum circuits, which allow us to reach large system sizes (up to 40 qubits and 352 parameters). The Hamiltonian is also represented in the form of a matrix product operator [70], allowing access to full-precision energy computation. Our proprietary MPS simulator uses an exact representation of the wave function without any truncation. The complexity of simulations in the MPS simulator scales linearly with system size and exponentially with circuit depth. Since the ansatz in L-VQE is one-dimensional and shallow, the MPS simulator can simulate L-VQE circuits for relatively large system sizes. The QAOA circuits we consider are deep; therefore, there is no benefit to using the MPS simulator. We use the high-performance simulator Qiskit Aer [71] to simulate QAOA circuits due to its convenience. Because of the simulation complexity and the need to optimize parameters for the benchmark instances, we limit the simulations of QAOA to 20 qubits. The choice of a simulator (MPS or Qiskit Aer) is inconsequential as both methods simulate the quantum state exactly and produce the same outcomes.

In variational algorithms, the choice of the classical outer-loop optimizer is central to the performance of the method. However, in this work, we do not specifically investigate the performance of various optimizers and do not perform any hyperparameter tuning on the optimizers. We only test L-VQE with a sequential minimal optimizer (SMO) [72] and COBYLA [73], [74]. SMO is implemented using the recommended settings [72], and COBYLA is implemented in the SciPy [75] package with the default setting. We observe that SMO performs slightly better than COBYLA; therefore, we advocate for SMO over COBYLA. For optimization in QAOA, we also use COBYLA with the default setting (we do not consider SMO as it is not designed for QAOA). Furthermore, we exhaustively optimized the parameters by using COBYLA as a local optimizer in the libEnsemble [76] implementation of APOSMM [77], [78]. Given a fixed number of iterations, APOSMM, as a multistart method, will run the local optimizer until convergence and then restart the optimization. This approach has been shown to work well in our previous study [40]. Additional details of the QAOA experiments are provided in Appendix A.

We run QAOA experiments on four `gnp random` graphs, with 7–10 nodes, respectively, and with p up to 10. All graphs are generated with `Networkx`. We give APOSMM a limit of 30 000 iterations. The limit is chosen based on an empirical observation that with this parameter choice, APOSMM will restart COBYLA at least ten times, usually much more. To compare, we run our L-VQE on each

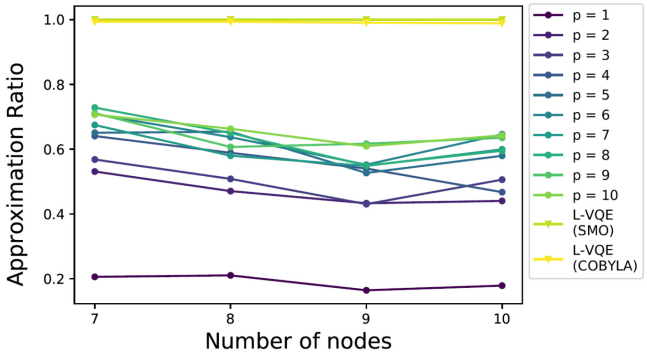

FIGURE 3. Compare QAOA and L-VQE on graphs of size from 7 to 10, simulating 14–20 qubits. L-VQE ansatz is iteratively increased up to $\ell = 2$ layers. L-VQE finds the ground state or a state that is close (with an approximation ratio of at least 0.99) to the ground state for each graph.

TABLE 1. Assuming Full Connectivity and Compiling the Higher Order Terms in the Hamiltonian (12) Into Gate Sets {R_Z, CNOT}, the Gate Count of QAOA Scales Quadratically With *n*, While L-VQE Scales Linearly

	QAOA with p steps	L-VQE with ℓ layers
CNOT	$5n(n-1)p$	$\ell(4n-2)$
single qubit gates	$\frac{1}{2}(3n^2+n)p$	$\ell(8n-4)+2n$

In our experiments presented in Fig. 3, QAOA circuits with p steps consists of $77p$ single-qubit gates and $210p$ CNOT gates, while L-VQE with ℓ layers contains $52\ell + 14$ single-qubit gates and 26ℓ CNOT gates. Thus, we expect that the L-VQE approach will be more robust to noise in real-life experiments. CNOT count of QAOA can be decreased by further circuit optimizations and more efficient native gates. On the other hand, it would be increased if the connectivity is not full.

graph ten times given different random seeds using SMO and COBYLA as the optimizer. Each run is given a limit of 3000 iterations, and we report the best result found by L-VQE. The results of the experiments are shown in Fig. 3. For each graph, L-VQE finds an estimate of the ground state with an approximation ratio ρ of at least 0.99, despite having a much lower budget of function evaluation.

For a graph with n nodes, our approach requires $2n$ qubits in order to detect four communities. Assuming full connectivity and compiling the higher order terms in the Hamiltonian (12) into gate sets {R_z, CNOT}, the product of four Pauli Z operators is decomposed into six CNOTs and one R_z in the middle. Single-qubit gates are cheap, so one is primarily interested in the CNOT gate counts. The gate counts of QAOA and L-VQE circuits are summarized in Table 1. When p is small, the Hamiltonian evolution ansatz used in QAOA is less expressive as compared with the hardware-efficient ansatz used in L-VQE. Therefore, a large number of QAOA layers and large circuit depth are needed to achieve the required overlap with the target state. At the same time, the cost function landscape of QAOA is highly nonconvex and contains many low-quality local optima, which makes finding high-quality parameters difficult for larger p. In addition, the Hamiltonian (12) contains many-body terms, which can be hard to compile into gates in practice due to the limited connectivity of the hardware. In contrast, L-VQE follows the connectivity of the hardware as the Hamiltonian structure does not enter the ansatz explicitly.

TABLE 2. Graph Information of the `Networkx` Generated Instances for Comparing VQE and L-VQE

Graph class	# instances	$\|V\|$
`relaxed caveman`	2	20
`gaussian random`	2	20
`random partition`	4	20
`windmill`	1	17
`gnp random`	4	20
`power law cluster`	3	20

TABLE 3. Best Approximation Ratio Achieved by VQE and L-VQE Using SMO

graph	VQE ρ_{best}		L-VQE ρ_{best}		
	1 Layer	2 Layer	0 Layer	1 Layer	2 Layer
caveman	0.99	0.93	1.0	1.0	1.0
caveman2	0.98	0.93	0.99	0.99	0.99
gaussian	0.98	0.9	1.0	1.0	1.0
gaussian2	0.99	0.93	0.99	1.0	1.0
gnp1	0.99	0.93	1.0	1.0	1.0
gnp2	0.98	0.92	0.95	1.0	1.0
gnp3	0.99	0.91	0.99	1.0	1.0
gnp4	0.99	0.88	0.97	0.99	1.0
power	0.96	0.9	1.0	1.0	1.0
power2	0.99	0.93	0.99	1.0	1.0
power3	0.98	0.91	0.9	0.96	1.0
random1	0.98	0.93	1.0	1.0	1.0
random2	0.99	0.89	0.95	1.0	1.0
random3	0.99	0.96	1.0	1.0	1.0
random4	0.92	0.93	0.96	1.0	0.98
windmill	0.99	0.96	1.0	1.0	1.0

As the number of layers in the ansatz increases, results of VQE deteriorates. L-VQE does not suffer from that problem, and we achieve better results as the number of layers grows.

B. VQE AND L-VQE

To further examine the performance of L-VQE, we compare the results of VQE with fixed ansatz and L-VQE on larger problems. In Section VI-B1, we compare the performance of VQE and L-VQE, and in Section VI-B2, we compare the performance with full-precision energy computation, that is, with energy computed as the expectation of the problem Hamiltonian with the full simulated quantum state. The results are summarized in Section VI-B3.

We generated 16 graph instances with `NetworkX`. Graph information is summarized in Table 2. The goal is to find a clustering of up to four communities that maximize the modularity; thus, we are simulating 34 qubits for `windmill` and 40 qubits for all other graphs.

For VQE, we define a fixed form of the ansatz upfront and then iteratively optimize and update over all parameters. We compare three sets of ansätze, which are shown in Fig. 2 as Layer 0 only ($\ell = 0$), Layer 0 to 1 only ($\ell = 1$), and Layer 0 to 2 ($\ell = 2$), respectively. For L-VQE with $\ell = 0$, the ansatz will not grow; thus, the algorithm is the same as VQE with one R_y gate acting on each qubit. For L-VQE with $\ell = 1$ and $\ell = 2$, we set the parameter $k_0 = 200$ in Algorithm 1. In other words, we first run L-VQE with Layer 0 ansatz for 200 iterations and then reuse the parameters to the ansatz with 1 layer and 2 layers, respectively. Again, we run our simulations of the quantum circuits in the MPS simulator. For optimization, we use the SMO [72] and COBYLA [73].

IEEE Transactions on Quantum Engineering

TABLE 4. Average Approximation Ratio Achieved by VQE and L-VQE Using SMO

graph	VQE $\rho_{\text{average}} \pm \sigma$		L-VQE $\rho_{\text{average}} \pm \sigma$		
	1 Layer	2 Layer	0 Layer	1 Layer	2 Layer
caveman	0.91 ± 0.08	0.83 ± 0.08	0.83 ± 0.15	0.92 ± 0.1	0.91 ± 0.1
caveman2	0.95 ± 0.03	0.86 ± 0.06	0.92 ± 0.07	0.99 ± 0.0	0.99 ± 0.0
gaussian	0.83 ± 0.08	0.77 ± 0.09	0.86 ± 0.08	0.94 ± 0.09	0.91 ± 0.09
gaussian2	0.92 ± 0.06	0.87 ± 0.03	0.9 ± 0.07	0.95 ± 0.06	0.95 ± 0.05
gnp1	0.87 ± 0.07	0.82 ± 0.08	0.87 ± 0.07	0.93 ± 0.05	0.94 ± 0.06
gnp2	0.89 ± 0.07	0.8 ± 0.09	0.87 ± 0.07	0.92 ± 0.06	0.94 ± 0.04
gnp3	0.89 ± 0.06	0.81 ± 0.06	0.86 ± 0.07	0.94 ± 0.06	0.95 ± 0.04
gnp4	0.92 ± 0.05	0.82 ± 0.04	0.92 ± 0.05	0.95 ± 0.04	0.94 ± 0.06
power	0.9 ± 0.04	0.82 ± 0.05	0.87 ± 0.07	0.9 ± 0.07	0.89 ± 0.08
power2	0.93 ± 0.08	0.85 ± 0.06	0.9 ± 0.06	0.92 ± 0.05	0.93 ± 0.05
power3	0.85 ± 0.07	0.79 ± 0.07	0.84 ± 0.06	0.89 ± 0.04	0.9 ± 0.05
random1	0.91 ± 0.1	0.84 ± 0.05	0.86 ± 0.13	0.98 ± 0.02	0.98 ± 0.02
random2	0.93 ± 0.07	0.81 ± 0.06	0.85 ± 0.1	0.93 ± 0.07	0.93 ± 0.06
random3	0.95 ± 0.05	0.84 ± 0.1	0.9 ± 0.11	0.95 ± 0.04	0.97 ± 0.03
random4	0.85 ± 0.05	0.83 ± 0.07	0.82 ± 0.08	0.9 ± 0.08	0.9 ± 0.07
windmill	0.93 ± 0.06	0.9 ± 0.05	0.84 ± 0.12	0.92 ± 0.06	0.94 ± 0.06

As the number of layers in the ansatz increases, results of VQE deteriorate; but for L-VQE, we achieve better results.

TABLE 5. Best Approximation Ratio Achieved by VQE and L-VQE Using COBYLA

graph	VQE ρ_{best}		L-VQE ρ_{best}		
	1 Layer	2 Layer	0 Layer	1 Layer	2 Layer
caveman	0.96	0.89	0.97	1.0	1.0
caveman2	0.95	0.92	0.97	0.99	0.99
gaussian	0.96	0.81	0.92	1.0	1.0
gaussian2	0.91	0.84	0.91	1.0	1.0
gnp1	0.94	0.85	0.91	1.0	1.0
gnp2	0.86	0.84	0.87	0.97	0.97
gnp3	0.83	0.84	0.97	1.0	1.0
gnp4	0.89	0.86	0.96	1.0	0.98
power	0.89	0.89	0.92	0.99	0.99
power2	0.91	0.77	0.9	1.0	1.0
power3	0.92	0.83	0.91	1.0	1.0
random1	0.93	0.88	0.96	1.0	1.0
random2	0.92	0.91	0.93	1.0	1.0
random3	0.97	0.91	0.98	1.0	1.0
random4	0.91	0.88	0.91	1.0	1.0
windmill	0.97	0.95	0.99	1.0	1.0

As the number of layers in the ansatz increases, results of VQE deteriorate; but for L-VQE, we achieve better results.

For each graph and each approach, we initialize the ansatz with ten different random seeds.

1) VQE AND L-VQE

We report the results of VQE and L-VQE in Tables 3–6. To evaluate the cost function $\langle\psi(\boldsymbol{\theta})|\mathcal{H}|\psi(\boldsymbol{\theta})\rangle$, we execute the circuit and generate 2000 samples and use the mean of the samples as an estimator. Having a finite number of samples is a realistic setup since when the scale of the system gets larger, the exact computation of the cost function becomes intractable.

In Table 3, we report the best approximation ratio (ρ_{best}) achieved from the ten runs using SMO for each graph. In Table 4, we report the average and standard deviation ($\rho_{\text{average}} \pm \sigma$) of the approximation ratio from the ten runs for each graph. We additionally report the results that use COBYLA as the optimizer in Tables 5 and 6.

2) VQE AND L-VQE WITH FULL-PRECISION ENERGY COMPUTATION

We report the results of VQE and L-VQE with full-precision energy computation in Tables 7 and 8. In each iteration, we evaluate the cost function exactly. In Table 7, we report the best approximation ratio (ρ_{best}) achieved from the ten runs using SMO for each graph. In Table 8, we report the average and standard deviation ($\rho_{\text{average}} \pm \sigma$) of the approximation ratio from the ten runs for each graph.

3) SUMMARY OF VQE AND L-VQE

Across all instances, we set the threshold of approximation ratio to 0.99, 0.95, and 0.90, respectively, and in Table 9, we report the percentage of the local optimizer runs that find the quantum state with a higher approximation ratio at the end of the algorithm. The rows in blue are the experiments with a finite number of samples (i.e., the cost function is estimated by the mean of the samples), and the rows in white are the experiments with full-precision energy computation (i.e., the cost function is evaluated exactly).

Intuitively, when we increase the size of the ansatz, the ansatz becomes more expressive, and we should have a better chance of finding the ground state. However, we can see that for VQE, when the cost function is estimated with finite number of samples, as the number of layers in the ansatz increase, the results deteriorate. But for L-VQE, as we increase the size of the ansatz, the results improve. Moreover, it is not practical to evaluate the energy exactly in applications when the size of the system gets larger. In L-VQE, by iteratively growing and reoptimizing the ansatz, we can achieve a higher probability of finding the ground state or a state that is sufficiently close to the ground state. By comparing the results of our L-VQE with or without full-precision energy computation, we see no significant difference, which suggests that our approach is relatively robust to finite sampling errors. We expect that this behavior is caused by the complicated landscape of VQE spanned by many parameters. A large circuit, even though

TABLE 6. Average Approximation Ratio Achieved by VQE and L-VQE Using COBYLA

graph	conventional VQE $\rho_{\text{average}} \pm \sigma$		L-VQE $\rho_{\text{average}} \pm \sigma$		
	1 Layer	2 Layer	0 Layer	1 Layer	2 Layer
caveman	0.86 ± 0.07	0.81 ± 0.06	0.8 ± 0.1	0.9 ± 0.07	0.88 ± 0.06
caveman2	0.84 ± 0.09	0.77 ± 0.12	0.82 ± 0.13	0.98 ± 0.02	0.99 ± 0.0
gaussian	0.76 ± 0.15	0.63 ± 0.1	0.75 ± 0.08	0.89 ± 0.08	0.87 ± 0.07
gaussian2	0.78 ± 0.1	0.69 ± 0.09	0.8 ± 0.08	0.96 ± 0.05	0.97 ± 0.04
gnp1	0.8 ± 0.08	0.72 ± 0.09	0.79 ± 0.09	0.9 ± 0.06	0.91 ± 0.07
gnp2	0.75 ± 0.07	0.72 ± 0.1	0.78 ± 0.07	0.91 ± 0.06	0.92 ± 0.05
gnp3	0.69 ± 0.1	0.71 ± 0.1	0.79 ± 0.11	0.93 ± 0.06	0.94 ± 0.07
gnp4	0.81 ± 0.08	0.71 ± 0.11	0.85 ± 0.06	0.95 ± 0.03	0.95 ± 0.03
power	0.79 ± 0.07	0.67 ± 0.14	0.76 ± 0.1	0.92 ± 0.05	0.92 ± 0.03
power2	0.78 ± 0.1	0.67 ± 0.08	0.78 ± 0.09	0.95 ± 0.03	0.93 ± 0.07
power3	0.77 ± 0.1	0.67 ± 0.07	0.79 ± 0.09	0.9 ± 0.08	0.86 ± 0.08
random1	0.82 ± 0.09	0.74 ± 0.12	0.86 ± 0.09	0.98 ± 0.03	0.97 ± 0.03
random2	0.78 ± 0.12	0.69 ± 0.15	0.82 ± 0.08	0.93 ± 0.05	0.95 ± 0.04
random3	0.84 ± 0.1	0.79 ± 0.1	0.9 ± 0.06	0.94 ± 0.04	0.94 ± 0.05
random4	0.8 ± 0.09	0.71 ± 0.08	0.77 ± 0.09	0.93 ± 0.06	0.94 ± 0.05
windmill	0.9 ± 0.06	0.82 ± 0.09	0.87 ± 0.09	0.92 ± 0.06	0.92 ± 0.06

As the number of layers in the ansatz increases, results of VQE deteriorate, but for L-VQE, we achieve better results.

TABLE 7. Best Approximation Ratio With Full-Precision Energy Computation Achieved by VQE and L-VQE Using SMO

graph	VQE ρ_{best}		L-VQE ρ_{best}		
	1 Layer	2 Layer	0 Layer	1 Layer	2 Layer
caveman	1.0	1.0	0.95	1.0	1.0
caveman2	0.99	0.99	0.99	0.99	0.99
gaussian	1.0	1.0	1.0	1.0	1.0
gaussian2	1.0	1.0	1.0	1.0	1.0
gnp1	1.0	0.94	1.0	1.0	1.0
gnp2	0.97	1.0	0.92	0.97	1.0
gnp3	1.0	1.0	1.0	1.0	0.99
gnp4	1.0	1.0	0.99	1.0	0.98
power	0.94	0.93	0.98	0.99	1.0
power2	1.0	1.0	0.99	1.0	1.0
power3	0.96	0.96	0.9	1.0	1.0
random1	1.0	1.0	0.97	1.0	1.0
random2	1.0	1.0	0.98	1.0	1.0
random3	1.0	1.0	1.0	1.0	1.0
random4	0.98	0.97	1.0	1.0	1.0
windmill	1.0	1.0	1.0	1.0	1.0

Comparing this table with Table 3, L-VQE is clearly more robust to finite sampling errors.

it is shallow, is hard to optimize if one does not use any mitigation strategies. L-VQE can be understood as one such strategy in which the circuit is carefully grown. This provides a good starting point for the optimization of a deeper circuit. Once the deeper circuit is initialized closer to the solution, the optimizer is less likely to hit the local minimum or spend more shots to escape from the neighborhood of the local minimum.

Additional evidence of L-VQE performance

We now provide additional evidence for the effects of reusing parameters and adding layers of the ansatz in L-VQE to complement the high-level statistics given in Tables 3–8. We observe that for the runs of experiments that start from the same initial Layer 0 ansatz by reusing the parameters obtained from that ansatz, in most cases, the results improve. Across all runs of the experiments, with finite samples, for one layer, 147 out of the 160 (91.88%) runs to find a state with a better or equal approximation ratio compared with the ansatz with 0 layer only. For two layers, 151 out of the 160 (94.38%) runs to find a state with a better or equal approximation ratio compared with the ansatz with 0 layers. Similarly, with full-precision energy computation, across all runs of the experiments, for one layer, 150 out of the 160 (93.75%) runs to find a state with a better or equal approximation ratio compared with the ansatz with 0 layers. For two layers, 151 out of the 160 (94.38%) runs find a quantum state with a better or equal approximation ratio compared with the ansatz with 0 layers. We present the violin plot of a representative instance `caveman` here in Fig. 4. Additional violin plots of the experiments for each graph can be found in Appendix C. Finally, we present the average number of iterations needed for L-VQE and VQE on all 20 nodes graph (40 qubits) in Fig. 5. When using SMO as the optimizer, L-VQE needs fewer iterations in general. At the same time, VQE requires fewer iterations if it is used with COBYLA optimizer. It should be pointed out, however, that L-VQE obtains higher quality results. In near-term applications, it is reasonable to use a slightly more expensive technique if it gives more accurate results. As demonstrated here, this is the case with L-VQE versus VQE.

C. FURTHER EVIDENCE OF THE POTENTIAL OF L-VQE

To provide further evidence of the potential of L-VQE, we present a scaling analysis of L-VQE in Section VI-C1 and discuss the simulation results of L-VQE on a trapped ion noisy quantum simulator in Section VI-C2.

1) SCALING ANALYSIS

In this set of experiments, we generate random graphs with nodes ranging from 8 to 20. This means that in our application of finding a clustering up to four communities that maximize the modularity, we need to simulate qubits

IEEE Transactions on Quantum Engineering

TABLE 8. Average Approximation Ratio With Full-Precision Energy Computation Using SMO

graph	VQE $\rho_{\text{average}} \pm \sigma$		L-VQE $\rho_{\text{average}} \pm \sigma$		
	1 Layer	2 Layer	0 Layer	1 Layer	2 Layer
caveman	0.89 ± 0.12	0.92 ± 0.06	0.76 ± 0.11	0.9 ± 0.09	0.92 ± 0.09
caveman2	0.96 ± 0.04	0.96 ± 0.07	0.92 ± 0.06	0.99 ± 0.0	0.99 ± 0.0
gaussian	0.94 ± 0.07	0.92 ± 0.08	0.81 ± 0.08	0.94 ± 0.09	0.85 ± 0.08
gaussian2	0.97 ± 0.05	0.99 ± 0.02	0.86 ± 0.1	0.99 ± 0.04	0.97 ± 0.04
gnp1	0.94 ± 0.04	0.89 ± 0.04	0.88 ± 0.08	0.92 ± 0.05	0.9 ± 0.05
gnp2	0.9 ± 0.05	0.94 ± 0.04	0.89 ± 0.03	0.92 ± 0.05	0.92 ± 0.05
gnp3	0.92 ± 0.08	0.88 ± 0.06	0.89 ± 0.06	0.9 ± 0.07	0.92 ± 0.06
gnp4	0.94 ± 0.03	0.95 ± 0.05	0.88 ± 0.07	0.96 ± 0.03	0.92 ± 0.06
power	0.91 ± 0.04	0.9 ± 0.03	0.87 ± 0.07	0.92 ± 0.08	0.89 ± 0.08
power2	0.95 ± 0.04	0.93 ± 0.06	0.91 ± 0.08	0.94 ± 0.05	0.92 ± 0.08
power3	0.84 ± 0.08	0.89 ± 0.06	0.82 ± 0.08	0.91 ± 0.05	0.92 ± 0.05
random1	0.87 ± 0.14	0.97 ± 0.02	0.92 ± 0.06	0.97 ± 0.02	0.98 ± 0.02
random2	0.96 ± 0.04	0.95 ± 0.06	0.9 ± 0.06	0.96 ± 0.03	0.97 ± 0.02
random3	0.96 ± 0.08	0.95 ± 0.07	0.9 ± 0.11	0.97 ± 0.04	0.96 ± 0.04
random4	0.89 ± 0.06	0.91 ± 0.05	0.8 ± 0.12	0.94 ± 0.08	0.94 ± 0.06
windmill	0.96 ± 0.06	0.94 ± 0.06	0.82 ± 0.12	0.96 ± 0.06	0.96 ± 0.06

Comparing this table with Table 4, L-VQE is clearly more robust to finite sampling errors.

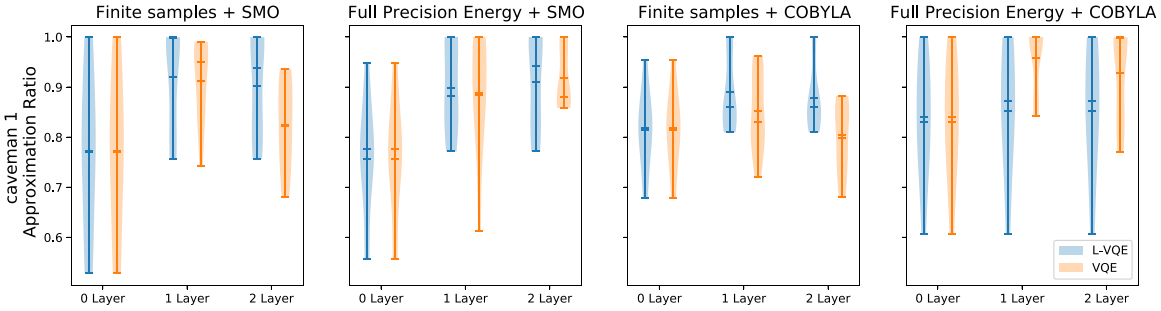

FIGURE 4. Violin plots of L-VQE versus VQE on graph `caveman`, and finite samples versus full-precision energy computation, using SMO and COBYLA as optimizers. The plots show the probability density of the results, with the kernel density estimator truncated to (min(ρ), max(ρ)) (since the approximation ratio cannot exceed 1). Comparing the results of L-VQE and with finite samples or full-precision energy computation, we see no significant difference. But for VQE with finite samples, as the number of layers in the ansatz increases, the results deteriorate. L-VQE is relatively robust to sampling noise.

TABLE 9. Percentage of Runs of Local Optimizers That Reach a Given Approximation Ratio. Blue Rows Show Results From Experiments With the Energy Computed From a Finite Number of Samples (Mean of 2000 Samples). White Rows are From Experiments With Full-Precision Energy Computation

Approximation ratio > 0.99	0 Layer	1 Layer	2 Layer
VQE	11.875%	0.625%	0.0%
L-VQE	11.875%	29.375%	30.625%
VQE	7.5%	26.875%	24.375%
L-VQE	7.5%	31.25%	27.5%
Approximation ratio > 0.95	0 Layer	1 Layer	2 Layer
VQE	21.25%	33.75%	1.25%
L-VQE	21.25%	49.375%	48.125%
VQE	18.75%	45.0%	48.125%
L-VQE	18.75%	57.5%	58.125%
Approximation ratio > 0.90	0 Layer	1 Layer	2 Layer
VQE	40.0%	55%	19.375%
L-VQE	40.0%	66.875%	67.5%
VQE	42.5%	72.5%	66.25%
L-VQE	42.5%	71.25%	71.25%

The optimizer is SMO. With finite sampling errors, as the number of layers in the ansatz increases, the results of VQE deteriorate. But for L-VQE, we achieve better results. Thus, L-VQE is more robust to finite sampling errors compared with VQE.

ranging from 16 to 40. For each graph and each approach, we run the experiments ten times and record the average number of iterations needed for convergence of each graph. The results are shown in Fig. 6. We can see that the number of iterations scales up polynomially as the number of nodes increases. Here, since within each iteration the number of R_y gates in the ansatz scales linearly with respect to the number of qubits needed (ansatz shown in Fig. 2), the number of parameters that need to be optimized, therefore, scales up linearly. In addition, the number of samples produced for evaluating the cost function is fixed as constant. Thus, the resources required for the entire algorithm scale polynomially. We point out, however, that our algorithm is heuristic by design and there is no guarantee of obtaining a solution with specified quality.

2) NOISY SIMULATIONS

The experiments described in the preceding sections are simulated in a setting that has no gate noise. For demonstration purposes, in the next set of experiments, we also investigate the performance of L-VQE using a trapped ion noisy quantum simulator. We use realistic error rates in our simulations. Details of the noise model are given in Appendix B. We

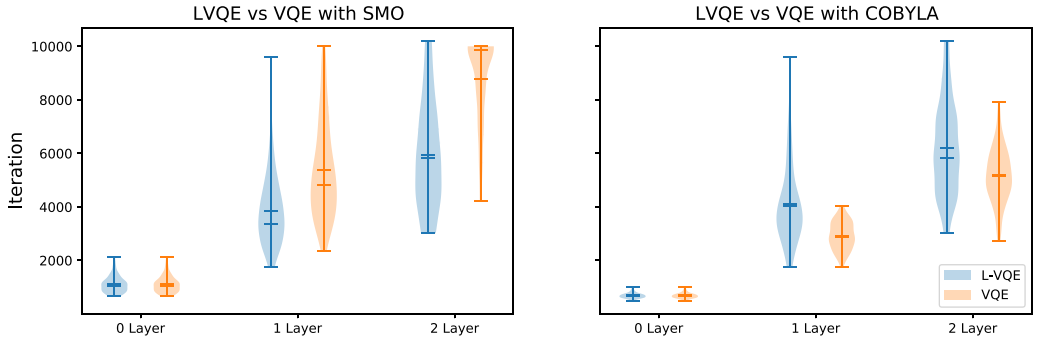

FIGURE 5. Average number of iterations needed for L-VQE and VQE on all 20 nodes graph (40 qubits), using SMO and COBYLA as optimizers. When using SMO as the optimizer, L-VQE needs fewer iterations in general. When using COBYLA, VQE needs fewer iterations. However, there is a tradeoff between the quality of the solution and the number of iterations.

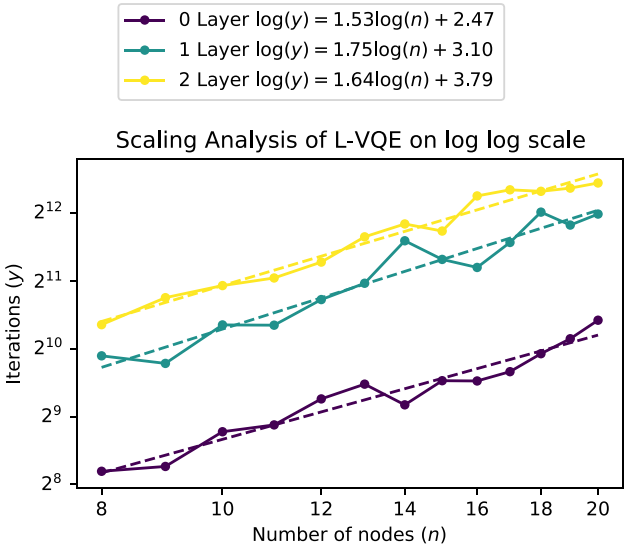

FIGURE 6. Average number of iterations until convergence scales up polynomially with respect to the size of the graph.

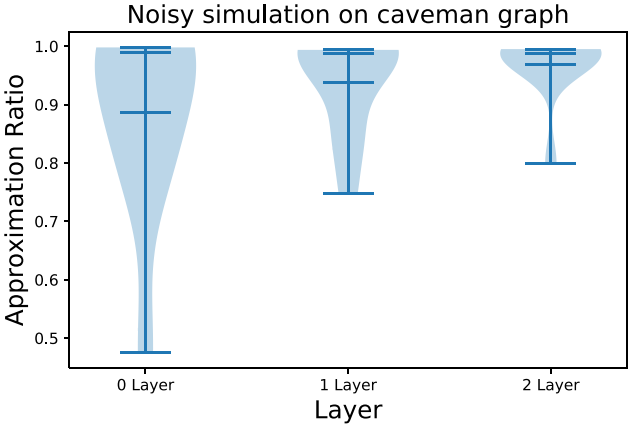

FIGURE 7. Violin plot of L-VQE performance on a trapped ion noisy quantum simulator. The plot shows the probability density of the results, with the kernel density estimator truncated to (min(ρ), max(ρ)) (since the approximation ratio cannot exceed 1). As the size of the ansatz increases, the probability of finding the ground state or a state that is sufficiently close increases.

run the experiments on all 16 graphs and show a representative instance `caveman` here; more results can be found in Appendix D. For L-VQE with Layers 1 and 2, we run the experiments ten times each. Fig. 7 gives a violin plot of the results. We observe that, in general, as the size of the ansatz increases, the probability of finding the ground state or a state that is sufficiently close increases. This suggests that L-VQE is also relatively robust to hardware noise and can be adapted to different quantum architecture.

TABLE 10. Percentage of Experiments Given the Approximation Ratio Threshold

Approximation ratio > 0.99	Entanglement	No entanglement
Finite samples	15%	0%
Full precision energy computation	37.5%	32.5%
Approximation ratio > 0.95	Entanglement	No entanglement
Finite samples	45%	37.5%
Full precision energy computation	57.5%	47.5%
Approximation ratio > 0.90	Entanglement	No entanglement
Finite samples	65%	60%
Full precision energy computation	70%	57.5%

D. ENTANGLEMENT VERSUS NO ENTANGLEMENT

Our next experiment is aimed at understanding the role of entanglement in VQE. We use the same methodology as proposed in [8]. That is, the experiment is based on replacing the entanglement gates CNOT with a T gate acting on both qubits. Compared with previous work, with our simulator we can investigate the algorithm's performance on larger problems. We run the experiments on four graphs: (`caveman`, `gnp`, `random`, and `gaussian`). For each graph, we repeat the experiments ten times with a different random seed. For the set of experiments with entanglement, we use the ansatz described in Fig. 2 with Layer 0 and Layer 1. For the set of experiments without entanglement, we replace all CNOT gates with a T gate acting on both qubits. The results are summarized in Table 10, where we report the percentage of runs that reach the approximation threshold 0.99, 0.95, and 0.90, respectively. As we can see from the results, under both cases, with and without full-precision energy computation,

IEEE Transactions on Quantum Engineering

using the ansatz with entanglement performs better than using the ansatz without entanglement.

VII. CONCLUSIONS AND DISCUSSION

Combinatorial optimization on near-term quantum devices is a leading candidate to demonstrate quantum advantage, and hybrid quantum-classical algorithms have been developed to solve this problem. In this work, we propose an iterative L-VQE approach inspired by VQE. We specifically studied the application of k-communities detection. In existing works, for a graph with n nodes, solving the k-communities modularity maximization problem requires kn qubits that encode the problem as an Ising model Hamiltonian. We propose a novel qubit-frugal formulation that requires only $n\lceil \log k \rceil$ qubits.

We compared the performance of L-VQE with QAOA, which is widely considered to be a strong candidate for quantum advantage in applications with NISQ computers. However, the many-body terms in the Hamiltonian make it harder to implement in the QAOA setting. Moreover, the numerical results show that the optimization indeed gets harder, thus suggesting that L-VQE provides a practical alternative to QAOA for combinatorial optimization on noisy near-term quantum computers.

Unlike VQE, which has an ansatz fixed upfront, L-VQE starts from a simple and shallow hardware efficient ansatz with a small number of parameterized gates and then adds layers to the ansatz systematically. This strategy allows us to make the ansatz more expressive and reduces the optimization overhead. Our numerical results suggest that adding layers of the ansatz indeed increases the probability of finding the ground state or finding the state that is sufficiently close to the ground state. With finite samples, however, VQE is more likely to fail. We empirically observe L-VQE to be more robust to finite sampling errors, making it a promising approach for NISQ devices. We use MPS representation to perform large-scale simulations of the quantum circuits in MATLAB. Doing so allowed us to explore problems of larger size (simulations up to 40 qubits and 352 parameters). We also studied the performance of L-VQE using a simulator of a noisy trapped-ion quantum computer. The results suggest that our approach is relatively robust to hardware noise and can be adapted and generalized to different quantum architecture. Finally, we present numerical results of the role of entanglement in VQE. The results clearly show that the ansatz with entanglement performs better than the ansatz without entanglement.

Our results are the first indication that the introduction of additional entangling parameters in VQE for classical problems, as proposed in [79, Sec. V-B], break down the barriers in the optimization landscape, making it more convex and, therefore, more amenable to simple local outer-loop optimizers to find a minimum. This is in sharp contrast with the previous results of Nannicini [8], who did not observe any beneficial effects of entanglement. The difference in findings between our results and those presented in [8] suggests the importance of the parameterization choice and the overall VQE procedure design to the success of such methods. We hope that this work will lead to even better algorithms to design ansätze for NISQ devices.

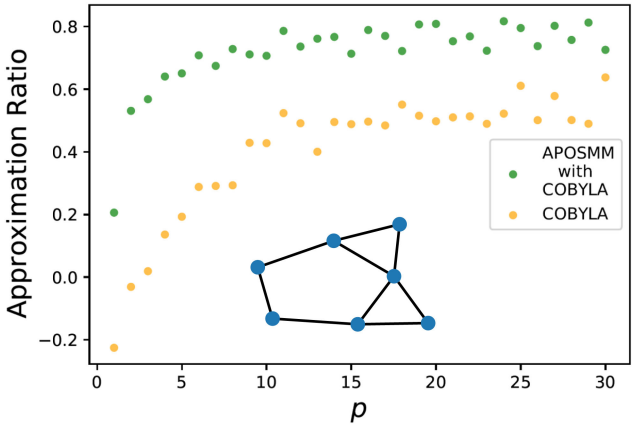

FIGURE 4. Best approximation ratio QAOA found for the seven-node graph (shown in the inset) with p ranging from 1 to 30. Even with the multistart method APOSSM to improve the optimizer COBYLA, we at most find an estimate of the ground state up to approximation ratio 0.817.

APPENDIX A
QAOA EXPERIMENTS

In this Appendix, we provide some additional details of the QAOA experiments on the `gnp random` graph with seven nodes, as shown in the inset of Fig. 4, simulating 14 qubits. The maximal modularity with up to four communities of this graph can be found by brute force (0.1790). We report the approximation ratio ρ [defined in (5)] found by QAOA in Fig. 4. We first run QAOA with p ranging from 1 to 30 for ten times for each p, using COBYLA to optimize the parameters. Each run is given a different random seed and is run until convergence. In Fig. 3, we report the best approximation ratio we find from the ten runs. Note that local optimizers such as COBYLA cannot guarantee to find the optimal parameters, especially as p increases. This is the reason that the data points of approximation ratio do not grow monotonically with p. Therefore, to further improve the optimizer, we use the multistart method APOSMM with COBYLA, which uses an ensemble of local optimization solvers. We use COBYLA as the local optimization solver within APOSMM. We give APOSMM a limit of 30 000 iterations. The limit is chosen based on an empirical observation that with this parameter choice, APOSMM will restart COBYLA at least ten times, usually much more. Using the multistart method, the results improve compared with using only COBYLA. We observe that with this small graph, even if we increase p up to 30, QAOA at most finds an estimate of the ground state up to approximation ratio 0.817.

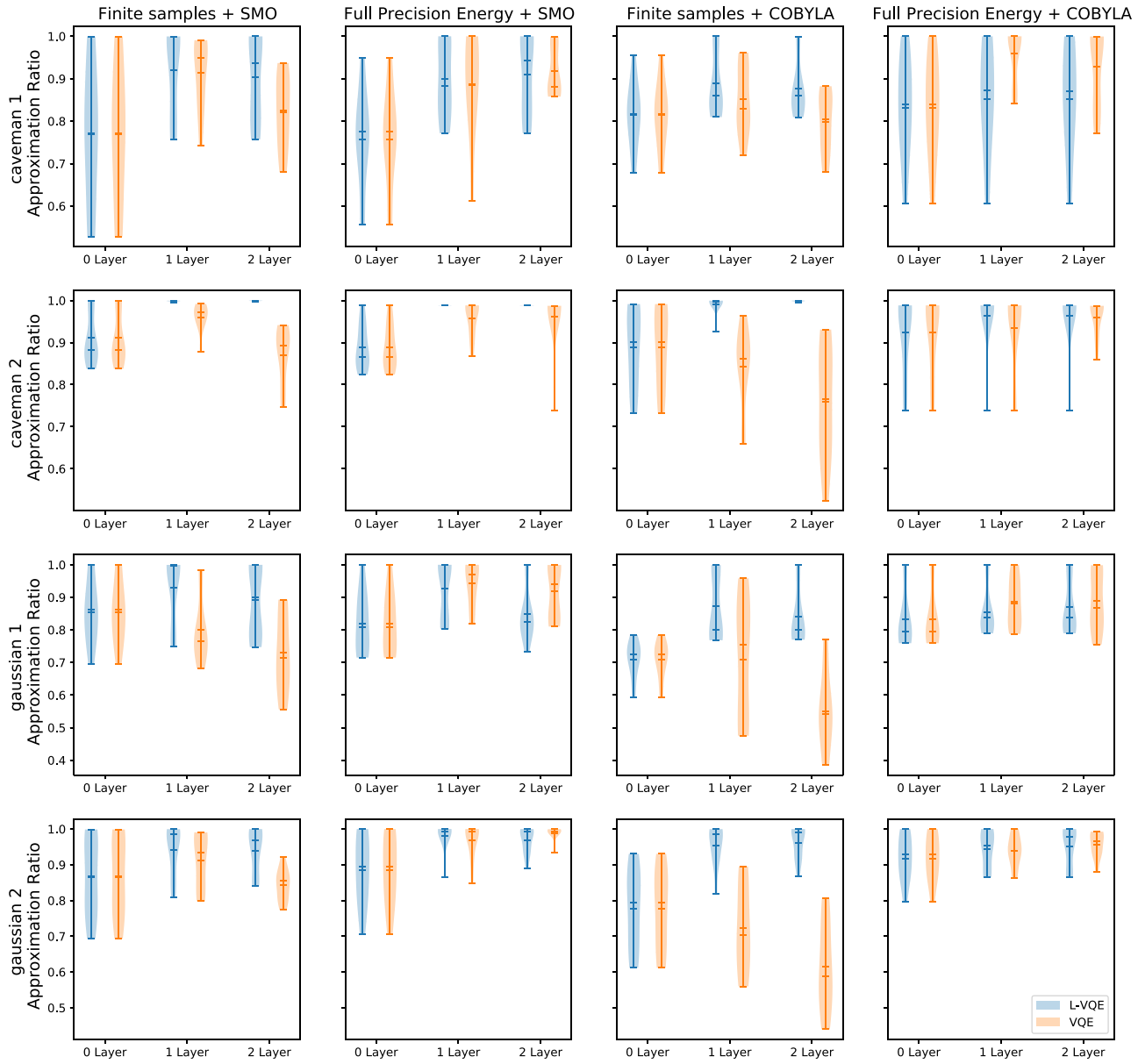

FIGURE 5. **Violin plots of L-VQE versus VQE: finite samples versus full-precision energy computation, using SMO and COBYLA as optimizers. The plots show the probability density of the results with the kernel density estimator truncated to (min(ρ), max(ρ)) (since the approximation ratio cannot exceed 1). In general, as the number of layers in the ansatz increases, results of VQE deteriorate, but for L-VQE, we achieve better results.**

APPENDIX B
TRAPPED-ION QUANTUM COMPUTER NOISE MODEL

In this Appendix, we give details on the error model that was used in Section VI-C2 of the main text. The error model is derived for a near-term trapped-ion quantum computer with realistic error rates specified in the following. Trout *et al.* [80] develop error maps to model errors that accumulate during the execution of single-qubit rotations $X(\theta) = e^{i\theta X}$, $Y(\theta) = e^{i\theta Y}$, and $Z(\theta) = e^{i\theta Z}$, as well as during the execution of Molmer–Sørensøn gate $XX(\theta) = e^{i\theta X\otimes X}$. X, Y, and Z denote spin-$\frac{1}{2}$ Pauli matrices. The quantum channels representing the noisy action of the above-mentioned gates take the following form:

$$\mathcal{E}_X(\theta) = \mathcal{D}(p_{\mathrm{d}}) \circ \mathcal{W}(p_{\mathrm{dep}}) \circ \mathcal{R}_X(p_\alpha) \circ \mathcal{U}_X(\theta) \tag{13}$$

$$\mathcal{E}_Y(\theta) = \mathcal{D}(p_{\mathrm{d}}) \circ \mathcal{W}(p_{\mathrm{dep}}) \circ \mathcal{R}_Y(p_\alpha) \circ \mathcal{U}_Y(\theta) \tag{14}$$

$$\mathcal{E}_Z(\theta) = \mathcal{D}(p_{\mathrm{d}}) \circ \mathcal{W}(p_{\mathrm{dep}}) \circ \mathcal{R}_Z(p_\alpha) \circ \mathcal{U}_Z(\theta) \tag{15}$$

$$\mathcal{E}_{XX}(\theta) = \big(\mathcal{D}(p_{\mathrm{d},1}) \otimes \mathcal{D}(p_{\mathrm{d},2})\big) \circ \big(\mathcal{W}(p_{\mathrm{dep}}) \otimes \mathcal{W}(p_{\mathrm{dep}})\big) \circ \mathcal{H}(p_{\mathrm{xx}}) \circ \mathcal{H}(p_{\mathrm{h}}) \circ \mathcal{U}_{XX}(\theta). \tag{16}$$

IEEE Transactions on Quantum Engineering

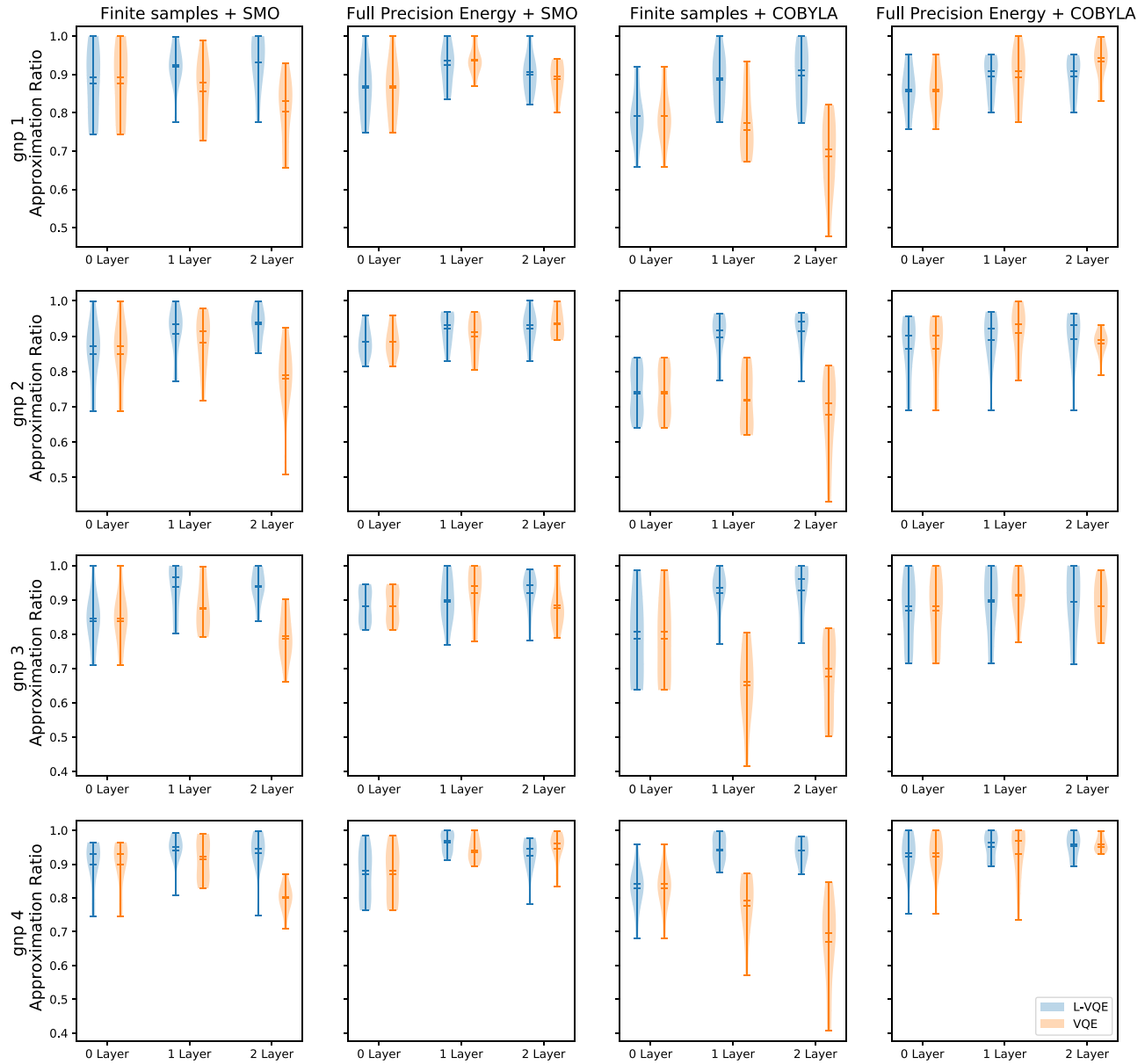

FIGURE 6. **Violin plots of L-VQE versus VQE: finite samples versus full-precision energy computation, using SMO and COBYLA as optimizers. The plots show the probability density of the results with the kernel density estimator truncated to (min(ρ), max(ρ)) (since the approximation ratio cannot exceed 1). In general, as the number of layers in the ansatz increases, results of VQE deteriorate, but for L-VQE, we achieve better results.**

Here, $\mathcal{U}_V(\theta)$ represents an ideal unitary evolution according to unitary V. That is

$$\mathcal{U}_V(\theta)\rho = e^{-i\theta V}\rho e^{i\theta V}. \tag{17}$$

$\mathcal{D}(p_d)$ is a dephasing channel defined as

$$\mathcal{D}(p_{\rm d})\rho = (1-p_{\rm d})\rho + p_{\rm d} Z\rho Z. \tag{18}$$

Note that in the definition of $\mathcal{E}_{XX}(\theta)$, we use separate dephasing channels for each qubit with (potentially different) error rates $p_{{\rm d},1}$, $p_{{\rm d},2}$.

Depolarizing channel $\mathcal{W}(p_{\rm dep})$ is defined as follows:

$$\mathcal{W}(p_{\rm dep})\rho = (1-p_{\rm dep})\rho + \frac{p_{\rm dep}}{3}X\rho X + \frac{p_{\rm dep}}{3}Y\rho Y + \frac{p_{\rm dep}}{3}Z\rho Z. \tag{19}$$

Imprecise rotation is implemented with $\mathcal{R}_V(p_\alpha)$. It is defined with

$$\mathcal{R}_V(p_\alpha) = (1-p_\alpha)\rho + p_\alpha V^\dagger \rho V \tag{20}$$

where $V = X, Y, Z$.

Finally, $\mathcal{H}(p_{\rm xx})$ represents the effects of two-qubit imprecise rotation and $\mathcal{H}(p_{\rm h})$ implements ion heating. The channel

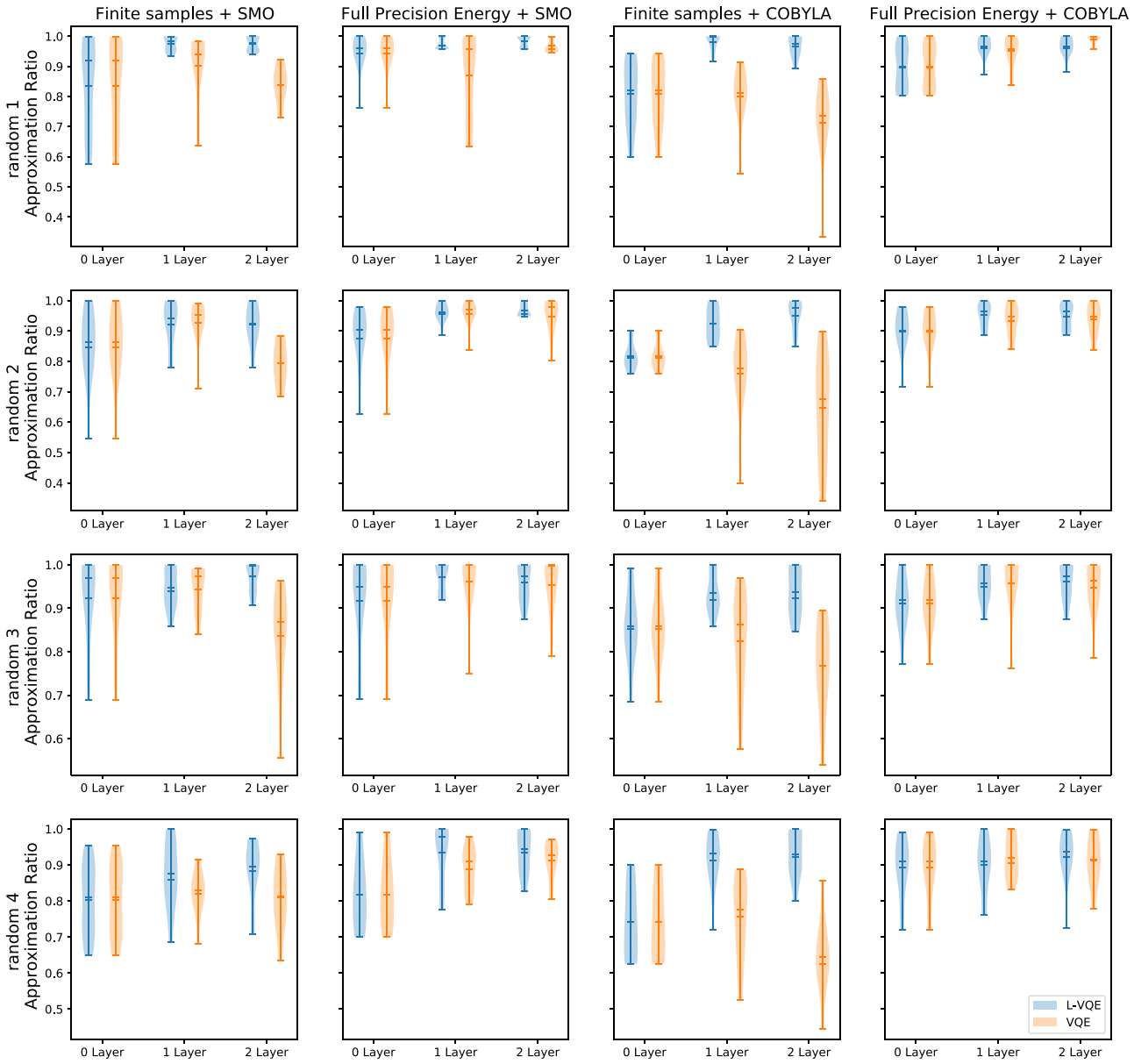

FIGURE 7. Violin plots of L-VQE versus VQE: finite samples versus full-precision energy computation using SMO and COBYLA as optimizers. The plots show the probability density of the results, with the kernel density estimator truncated to (min(ρ), max(ρ)) (since the approximation ratio cannot exceed 1). In general, as the number of layers in the ansatz increases, results of VQE deteriorate, but for L-VQE, we achieve better results.

is defined in the following way:

$$\mathcal{H}(p)\rho = (1-p)\rho + p(X\otimes X)\rho(X\otimes X). \tag{21}$$

We also model the effects of noisy initial state preparation. In our simulations, the perfect state $\rho_0 = \text{diag}(1, 0)$ is replaced with the state affected by depolarizing channel: $\mathcal{W}(p_{\text{dep}})\rho_0$. Similarly, the measurement error is modeled with the depolarizing channel. It is implemented by preceding the ideal POVM element with an action of depolarizing channel.

We used the following realistic values of noise rates:

$$p_{\text{d}} = 1.5\times 10^{-4}$$

$$p_{\text{dep}} = 8\times 10^{-4}$$

$$p_{\text{d},1} = p_{\text{d},2} = 7.5\times 10^{-4}$$

$$p_{\alpha} = 1\times 10^{-4}$$

$$p_{\text{xx}} = 1\times 10^{-3}$$

$$p_{\text{h}} = 1.25\times 10^{-3}. \tag{22}$$

APPENDIX C
ADDITIONAL L-VQE AND VQE SIMULATION RESULTS

In this Appendix, we present more detailed simulation results of L-VQE and VQE using MPS simulator for all 16 graphs. Figs. 5–8 show the violin plots of L-VQE versus VQE, finite samples versus full-precision energy computation, using

IEEE Transactions on Quantum Engineering

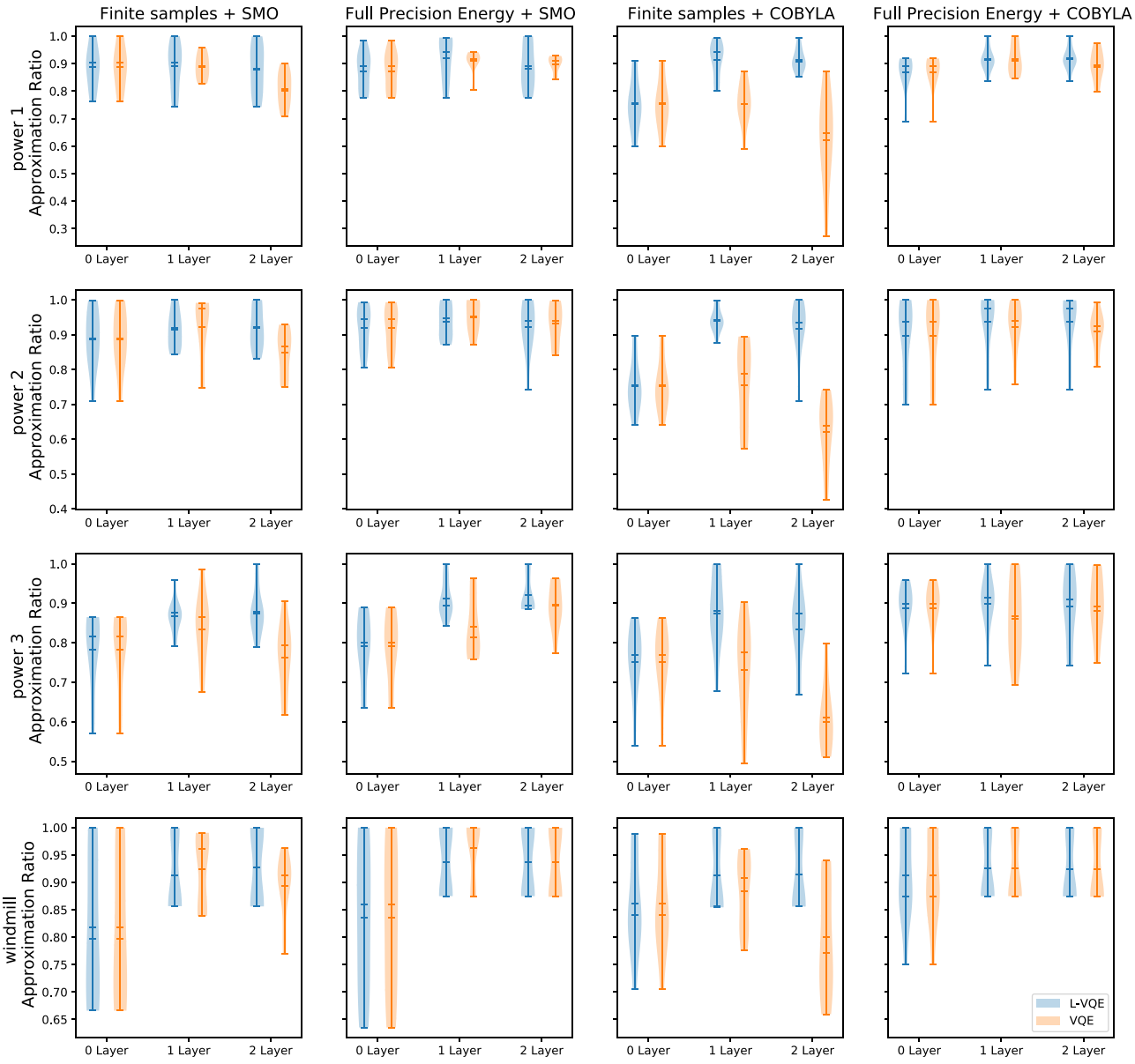

FIGURE 8. Violin plots of L-VQE versus VQE: finite samples versus full-precision energy computation using SMO and COBYLA as optimizers. The plots show the probability density of the results, with the kernel density estimator truncated to (min(ρ), max(ρ)) (since the approximation ratio cannot exceed 1). In general, as the number of layers in the ansatz increases, results of VQE deteriorate, but for L-VQE, we achieve better results.

SMO and COBYLA as optimizers. In general, as the number of layers in the ansatz increases, results of VQE deteriorate, but for L-VQE, we achieve better results. This suggests that L-VQE is more robust to finite sampling errors as compared with VQE.

APPENDIX D
ADDITIONAL L-VQE SIMULATION RESULTS ON TRAPPED ION NOISY SIMULATOR

In this Appendix, we present the simulation results of L-VQE using trapped ion noisy quantum simulator for all 16 graphs in Fig. 9. In general, as the size of the ansatz increases, the probability of finding the ground state or a state that is sufficiently close increases. Therefore, we suggest that L-VQE is relatively robust to hardware noise.

The submitted manuscript has been created by UChicago Argonne, LLC, Operator of Argonne National Laboratory ("Argonne"). Argonne, a U.S. Department of Energy Office of Science laboratory, is operated under Contract DE-AC02-06CH11357. The U.S. Government retains for itself, and others acting on its behalf, a paid-up nonexclusive, irrevocable worldwide license in said article to reproduce, prepare derivative works, distribute copies to the public, and perform publicly and display publicly, by or on behalf of the Government. The Department of Energy will provide public access to these results of federally sponsored

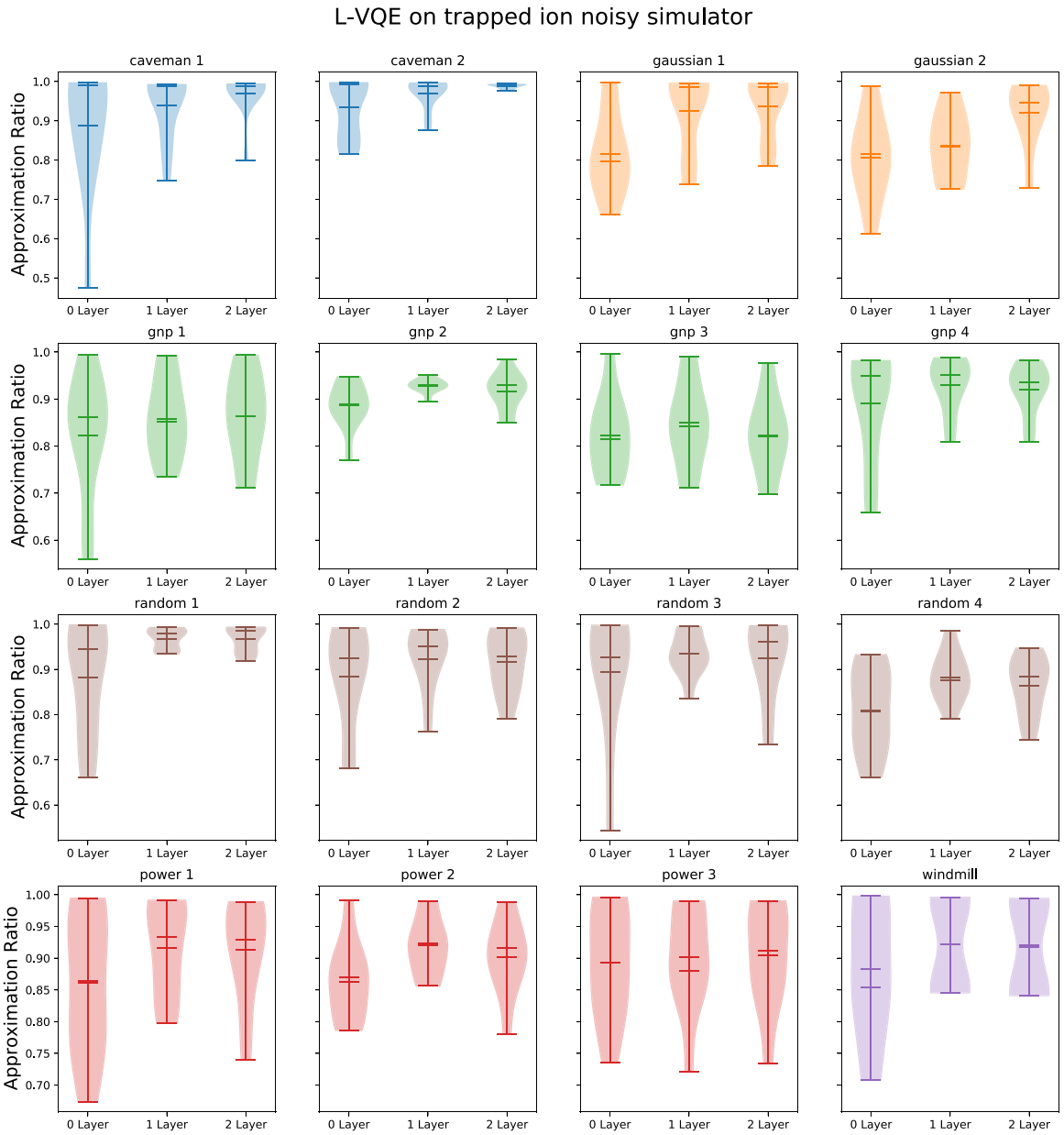

FIGURE 9. Violin plot of L-VQE performance on a trapped ion noisy quantum simulator. The plot shows the probability density of the results with the kernel density estimator truncated to (min(ρ), max(ρ)) (since the approximation ratio cannot exceed 1). In general, as the size of the ansatz increases, the probability of finding the ground state or a state that is sufficiently close increases.

research in accordance with the DOE Public Access Plan http://energy.gov/downloads/doe-public-access-plan.

ACKNOWLEDGMENT

The authors would like to thank J. Larson for help with tuning APOSMM for QAOA parameter optimization. Clemson University is acknowledged for generous allotment of computing time on the Palmetto cluster. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of the U.S. Department of Energy under Contract 89233218CNA000001.

REFERENCES

[1] Y. Alexeev *et al.*, "Quantum computer systems for scientific discovery," *PRX Quantum*, vol. 2, no. 1, Feb. 2021, Art. no. 017001, doi: 10.1103/PRXQuantum.2.017001.

[2] R. Shaydulin, H. Ushijima-Mwesigwa, C. F. A. Negre, I. Safro, S. M. Mniszewski, and Y. Alexeev, "A hybrid approach for solving optimization problems on small quantum computers," *Computer*, vol. 52, no. 6, pp. 18–26, Jun. 2019, doi: 10.1109/MC.2019.2908942.

[3] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[4] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv:1411.4028*, 2014. [Online]. Available: https://arxiv.org/abs/arXiv:1411.4028

[5] E. Farhi, J. Goldstone, S. Gutmann, and H. Neven, "Quantum algorithms for fixed qubit architectures," *arXiv:1703.06199*, 2017. [Online]. Available: https://arxiv.org/abs/arXiv:1703.06199

[6] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," *Algorithms*, vol. 12, no. 2, p. 34, 2019, doi: 10.3390/a12020034.

[7] N. Moll *et al.*, "Quantum optimization using variational algorithms on near-term quantum devices," *Quantum Sci. Technol.*, vol. 3, no. 3, 2018, Art. no. 030503, doi: 10.1088/2058-9565/aab822

[8] G. Nannicini, "Performance of hybrid quantum-classical variational heuristics for combinatorial optimization," *Phys. Rev. E*, vol. 99, no. 1, 2019, Art. no. 013304, doi: 10.1103/PhysRevE.99.013304.

[9] M. Cerezo *et al.*, "Variational quantum algorithms," *Nature Reviews Phys.*, vol. 3, p. 625644, 2021, doi: 10.1038/s42254-021-00348-9.

[10] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, "Connecting ansatz expressibility to gradient magnitudes and barren plateaus," *PRX Quantum*. American Physical Society, vol. 3, no. 1, Jan. 2022, Art. no. 010313, doi: 10.1103/PRXQuantum.3.010313.

[11] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Commun.*, vol. 9, no. 1, pp. 1–6, 2018, doi: 10.1038/s41467-018-07090-4.

[12] A. Kandala *et al.*, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," *Nature*, vol. 549, no. 7671, pp. 242–246, 2017, doi: 10.1038/nature23879.

[13] R. Shaydulin, S. Hadfield, T. Hogg, and I. Safro, "Classical symmetries and the Quantum Approximate Optimization Algorithm," *Quantum Inf. Process.*, vol. 20, 2021, Art. no. 359, doi: 10.1007/s11128-021-03298-4.

[14] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, "Obstacles to state preparation and variational optimization from symmetry protection," *Phys. Rev. Lett.*, vol. 125, no. 26, Dec. 2020, Art. no. 260505, doi: 10.1103/PhysRevLett.125.260505.

[15] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost-function-dependent barren plateaus in shallow quantum neural networks," *Nature Commun.*, Nature Publishing Group, vol. 12, no. 1, pp. 1–12, 2021

[16] K. Sharma, M. Cerezo, L. Cincio, and P. J. Coles, "Trainability of dissipative perceptron-based quantum neural networks," *arXiv:2005.12458*, 2020. [Online]. Available: https://arxiv.org/abs/arXiv:2005.12458

[17] A. Pesah, M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, "Absence of barren plateaus in quantum convolutional neural networks," *Phys. Rev. X*, vol. 11, no. 4, Oct.–Dec. 2021, Art. no. 041011, doi: 10.1103/PhysRevX.11.041011.

[18] M. Cerezo and P. J. Coles, "Higher order derivatives of quantum neural networks with barren plateaus," *Quantum Sci. Technol.*, vol. 6, no. 3, 2021, Art. no. 035006, doi: 10.1088/2058-9565/abf51a/pdf.

[19] S. Wang *et al.*, "Noise-induced barren plateaus in variational quantum algorithms," *Nature Commun.*, vol. 12, 2021, Art. no. 6961, doi: 10.1038/s41467-021-27045-6.

[20] Z. Holmes, A. Arrasmith, B. Yan, P. J. Coles, A. Albrecht, and A. T. Sornborger, "Barren plateaus preclude learning scramblers," *Phys. Rev. Lett.*, vol. 126, no. 19, May 2021, Art. no. 190501, doi: 10.1103/PhysRevLett.126.190501.

[21] T. J. Volkoff, "Efficient trainability of linear optical modules in quantum optical neural networks," *J. Russ. Laser Res.*, vol. 42, p. 250260, 2021, doi: 10.1007/s10946-021-09958-1.

[22] E. Campos, A. Nasrallah, and J. Biamonte, "Abrupt transitions in variational quantum circuit training," *Phys. Rev. A*, vol. 103, Mar. 2021, Art. no. 032607, doi: 10.1103/PhysRevA.103.032607.

[23] C. Ortiz Marrero, M. Kieferová, and N. Wiebe, "Entanglement-induced barren plateaus," *PRX Quantum*, vol. 2, no. 4, Oct.–Dec. 2021, Art. no. 040316, doi: 10.1103/PRXQuantum.2.040316.

[24] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," *Nature Comput. Sci.*, vol. 1, p. 403409, 2021, doi: 10.1038/s43588-021-00084-1.

[25] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, "Learning to optimize variational quantum circuits to solve combinatorial problems," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 2367–2375, doi: 10.1609/aaai.v34i03.5616.

[26] M. Wilson, S. Stromswold, F. Wudarski, S. Hadfield, N. M. Tubman, and E. Rieffel, "Optimizing quantum heuristics with meta-learning," *Quantum Mach. Intell.*, vol. 3, 2021, Art. no. 13, doi: 10.1007/s42484-020-00022-w.

[27] G. Verdon *et al.*, "Learning to learn with quantum neural networks via classical neural networks," *arXiv:1907.05415*, 2019. [Online]. Available: https://arxiv.org/abs/arXiv:1907.05415

[28] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices," *Phys. Rev. X*, vol. 10, no. 2, 2020, Art. no. 021067, doi: 10.1103/PhysRevX.10.021067.

[29] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," *arXiv:1811.08419*, 2018. [Online]. Available: https://arxiv.org/abs/arXiv:1811.08419

[30] G. B. Mbeng, R. Fazio, and G. Santoro, "Quantum annealing: A journey through digitalization, control, and hybrid quantum variational schemes," *arXiv:1906.08948*, 2019. [Online]. Available: https://arxiv.org/abs/arXiv:1906.08948

[31] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, "An adaptive variational algorithm for exact molecular simulations on a quantum computer," *Nature Commun.*, vol. 10, no. 1, pp. 1–9, 2019, doi: 10.1038/s41467-019-10988-2.

[32] L. Zhu, H. L. Tang, G. S. Barron, N. J. Mayhall, E. Barnes, and S. E. Economou, "An adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer," *arXiv:2005.10258*, 2020. [Online]. Available: https://arxiv.org/abs/arXiv:2005.10258

[33] A. Skolik, J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib, "Layerwise learning for quantum neural networks," *Quantum Mach. Intell.*, vol. 3, 2021, Art. no. 5, doi: 10.1007/s42484-020-00036-4.

[34] A. Peruzzo *et al.*, "A variational eigenvalue solver on a photonic quantum processor," *Nature Commun.*, vol. 5, 2014, Art. no. 4213, doi: 10.1038/ncomms5213.

[35] M. P. Quinones and C. Junqueira, "Modeling linear inequality constraints in quadratic binary optimization for variational quantum eigensolver," Jul. 2020. [Online]. Available: https://arxiv.org/abs/2007.13245

[36] P. K. Barkoutsos, G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, "Improving variational quantum optimization using CVaR," *Quantum*, vol. 4, p. 256, Apr. 2020, doi: 10.22331/q-2020-04-20-256.

[37] M. Streif and M. Leib, "Training the quantum approximate optimization algorithm without access to a quantum processing unit," *Quantum Sci. Technol.*, vol. 5, no. 3, May 2020, Art. no. 034008, doi: 10.1088/2058-9565/ab8c2b.

[38] R. Shaydulin and S. M. Wild, "Exploiting symmetry reduces the cost of training QAOA," *IEEE Trans. Quantum Eng.*, vol. 2, 2021, Art. no. 3101409, doi: 10.1109/TQE.2021.3066275.

[39] F. G. Brandao, M. Broughton, E. Farhi, S. Gutmann, and H. Neven, "For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances," *arXiv:1812.04170*, 2018. [Online]. Available: https://arxiv.org/abs/arXiv:1812.04170

[40] R. Shaydulin, I. Safro, and J. Larson, "Multistart methods for quantum approximate optimization," in *Proc. IEEE High Perform. Extreme Comput. Conf.*, 2019, pp. 1–8, doi: 10.1109/HPEC.2019.8916288.

[41] M. E. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci.*, vol. 103, no. 23, pp. 8577–8582, 2006, doi: 10.1073/pnas.0601602103.

[42] A. Buluç, H. Meyerhenke, I. Safro, P. Sanders, and C. Schulz, "Recent advances in graph partitioning," in *Algorithm Engineering*. New York, NY, USA: Springer, 2016, pp. 117–158, doi: 10.1007/978-3-319-49487-6_4.

[43] P. Erdös and A. Rényi, "On the evolution of random graphs," in *The Structure and Dynamics of Networks*. Princeton, NJ, USA: Princeton Univ. Press, 2011, pp. 38–82, doi: 10.1515/9781400841356.38.

[44] A. M. Niklasson *et al.*, "Graph-based linear scaling electronic structure theory," *J. Chem. Phys.*, vol. 144, no. 23, 2016, Art. no. 234101, doi: 10.1063/1.4952650.

[45] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási, "The large-scale organization of metabolic networks," *Nature*, vol. 407, no. 6804, pp. 651–654, 2000, doi: 10.1038/35036627.

[46] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow, "The anatomy of the Facebook social graph," 2011. [Online]. Available: https://arxiv.org/abs/arXiv:1111.4503

[47] U. Brandes *et al.*, "Maximizing modularity is hard," 2006. [Online]. Available: https://arxiv.org/abs/physics/0608255

[48] M. C. Nascimento and A. C. De Carvalho, "Spectral methods for graph clustering-a survey," *Eur. J. Oper. Res.*, vol. 211, no. 2, pp. 221–231, 2011, doi: 10.1016/j.ejor.2010.08.012.

[49] C. F. Negre, H. Ushijima-Mwesigwa, and S. M. Mniszewski, "Detecting multiple communities using quantum annealing on the D-Wave system," *PLoS One*, vol. 15, no. 2, 2020, Art. no. e0227538, doi: 10.1371/journal.pone.0227538.

[50] R. Shaydulin, H. Ushijima-Mwesigwa, I. Safro, S. Mniszewski, and Y. Alexeev, "Network community detection on small quantum computers," *Adv. Quantum Technol.*, vol. 2, no. 9, 2019, Art. no. 1900029, doi: 10.1002/qute.201900029.

[51] H. Ushijima-Mwesigwa, C. F. Negre, and S. M. Mniszewski, "Graph partitioning using quantum annealing on the D-Wave system," in *Proc. 2nd Int. Workshop Post Moores Era Supercomput.*, 2017, pp. 22–29, doi: 10.1145/3149526.3149531.

[52] H. Ushijima-Mwesigwa, R. Shaydulin, C. F. Negre, S. M. Mniszewski, Y. Alexeev, and I. Safro, "Multilevel combinatorial optimization across quantum architectures," *ACM Trans. Quantum Comput.*, vol. 2, no. 1, pp. 1–29, 2021, doi: 10.1145/3425607.

[53] R. Shaydulin, H. Ushijima-Mwesigwa, I. Safro, S. Mniszewski, and Y. Alexeev, "Community detection across emerging quantum architectures," in *Proc. 3rd Int. Workshop Post Moore's Era Supercomput.*, 2018. [Online]. Available: https://arxiv.org/abs/arXiv:1810.07765

[54] S. Wang *et al.*, "Noise-induced barren plateaus in variational quantum algorithms," *Nature Commun.*, vol. 12, 2021, Art. no. 6961, doi: 10.1038/s41467-021-27045-6.

[55] C. Xue, Z.-Y. Chen, Y.-C. Wu, and G.-P. Guo, "Effects of quantum noise on quantum approximate optimization algorithm," *Chinese Phys. Lett.*, vol. 38, no. 3, 2021, Art. no. 030302, doi: 10.1088/0256-307X/38/3/030302.

[56] R. J. Bartlett, S. A. Kucharski, and J. Noga, "Alternative coupled-cluster ansätze. II. The unitary coupled-cluster method," *Chem. Phys. Lett.*, vol. 155, no. 1, pp. 133–140, 1989, doi: 10.1016/S0009-2614(89)87372-5.

[57] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, "Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm," *npj Quantum Inf.*, vol. 6, no. 1, pp. 1–9, 2020, doi: 10.1038/s41534-019-0240-1.

[58] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, "Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz," *Quantum Sci. Technol.*, vol. 4, no. 1, 2018, Art. no. 014008, doi: 10.1088/2058-9565/aad3e4.

[59] 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*, vol. 567, no. 7749, pp. 491–495, 2019, doi: 10.1038/s41586-019-1040-7.

[60] P. K. Barkoutsos *et al.*, "Quantum algorithms for electronic structure calculations: Particle-hole hamiltonian and optimized wave-function expansions," *Phys. Rev. A*, vol. 98, no. 2, 2018, Art. no. 022322, doi: 10.1103/PhysRevA.98.022322.

[61] M. Ganzhorn *et al.*, "Gate-efficient simulation of molecular eigenstates on a quantum computer," *Phys. Rev. Appl.*, vol. 11, Apr. 2019, Art. no. 044092, doi: 10.1103/PhysRevApplied.11.044092.

[62] H. L. Tang *et al.*, "Qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor," *PRX Quantum*, vol. 2, no. 2, 2021, Art. no. 020310, doi: 10.1103/PRXQuantum.2.020310.

[63] I. G. Ryabinkin, T.-C. Yen, S. N. Genin, and A. F. Izmaylov, "Qubit coupled cluster method: A systematic approach to quantum chemistry on a quantum computer," *J. Chem. Theory Comput.*, vol. 14, no. 12, pp. 6317–6326, 2018, doi: 10.1021/acs.jctc.8b00932.

[64] I. G. Ryabinkin, R. A. Lang, S. N. Genin, and A. F. Izmaylov, "Iterative qubit coupled cluster approach with efficient screening of generators," *J. Chem. Theory Comput.*, vol. 16, no. 2, pp. 1055–1063, 2020, doi: 10.1021/acs.jctc.9b01084.

[65] R. A. Lang, I. G. Ryabinkin, and A. F. Izmaylov, "Unitary transformation of the electronic hamiltonian with an exact quadratic truncation of the Baker–Campbell–Hausdorff expansion," *J. Chem. Theory Comput.*, vol. 17, no. 1, pp. 66–78, 2020, doi: 10.1021/acs.jctc.0c00170.

[66] D. Claudino, J. Wright, A. J. McCaskey, and T. S. Humble, "Benchmarking adaptive variational quantum eigensolvers," *Front. Chem.*, vol. 8, 2020, Art. no. 1152, doi: 10.3389/fchem.2020.606863.

[67] S. Sim, J. Romero, J. F. Gonthier, and A. A. Kunitsa, "Adaptive pruning-based optimization of parameterized quantum circuits," *Quantum Sci. Technol.*, vol. 6, no. 2, 2021, Art. no. 025019, doi: 10.1088/2058-9565/ABE107.

[68] Y. S. Yordanov, D. R. M. Arvidsson-Shukur, and C. H. W. Barnes, "Efficient quantum circuits for quantum computational chemistry," *Phys. Rev. A*, vol. 102, no. 6, 2020, Art. no. 062612, doi: 10.1103/PhysRevA.102.062612.

[69] Y. S. Yordanov, V. Armaos, C. H. W. Barnes, and D. R. M. Arvidsson-Shukur, "Qubit-excitation-based adaptive variational quantum eigensolver," *Commun. Phys.*, Nature Publishing Group, vol. 4, no, 1, pp. 1–11, 2020.

[70] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," *Ann. Phys.*, vol. 349, pp. 117–158, 2014, doi: 10.1016/j.aop.2014.06.013.

[71] "Qiskit: An open-source framework for quantum computing," 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2562111

[72] K. M. Nakanishi, K. Fujii, and S. Todo, "Sequential minimal optimization for quantum-classical hybrid algorithms," *Phys. Rev. Res.*, vol. 2, no. 4, Oct.–Dec. 2020, Art. no. 043158, doi: 10.1103/PhysRevResearch.2.043158.

[73] 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*. Dordrecht, The Netherlands: Springer, 1994, pp. 51–67, doi: 10.1007/978-94-015-8330-5_4.

[74] M. J. D. Powell, "Direct search algorithms for optimization calculations," *Acta Numerica*, vol. 7, pp. 287–336, Jan. 1998, doi: 10.1017/S0962492900002841.

[75] E. Jones *et al.*, "SciPy: Open source scientific tools for Python," 2001. [Online]. Available: http://www.scipy.org/

[76] S. Hudson, J. Larson, S. M. Wild, and D. Bindel, "libEnsemble users manual," 2019. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/libensemble/latest/libensemble.pdf

[77] J. Larson and S. M. Wild, "A batch, derivative-free algorithm for finding multiple local minima," *Optim. Eng.*, vol. 17, no. 1, pp. 205–228, 2016, doi: 10.1007/s11081-015-9289-7.

[78] J. Larson and S. M. Wild, "Asynchronously parallel optimization solver for finding multiple minima," *Math. Program. Comput.*, vol. 10, no. 3, pp. 303–332, 2018, doi: 10.1007/s12532-017-0131-4.

[79] J. R. McClean *et al.*, "Low-depth mechanisms for quantum optimization," *PRX Quantum*, vol. 2, no. 3, Jul.–Sep. 2021, Art. no. 030312, doi: 10.1103/PRXQuantum.2.030312.

[80] C. J. Trout *et al.*, "Simulating the performance of a distance-3 surface code in a linear ion trap," *New J. Phys.*, vol. 20, no. 4, 2018, Art. no. 043038, doi: 10.1088/1367-2630/aab341.