Noisy intermediate-scale quantum algorithms

Kishor Bharti[*]

Centre for Quantum Technologies, National University of Singapore, 117543 Singapore

Alba Cervera-Lierta[†] and Thi Ha Kyaw[‡]

Department of Computer Science, University of Toronto, Toronto, Ontario M5S 2E4, Canada and Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, Ontario M5G 1Z8, Canada

Tobias Haug

QOLS, Blackett Laboratory, Imperial College London SW7 2AZ, United Kingdom

Sumner Alperin-Lea and Abhinav Anand

Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, Ontario M5G 1Z8, Canada

Matthias Degroote[§]

Department of Computer Science, University of Toronto, Toronto, Ontario M5S 2E4, Canada and Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, Ontario M5G 1Z8, Canada

Hermanni Heimonen

Centre for Quantum Technologies, National University of Singapore, 117543 Singapore

Jakob S. Kottmann

Department of Computer Science, University of Toronto, Toronto, Ontario M5S 2E4, Canada and Chemical Physics Theory Group, Department of Chemistry, University of Toronto, Toronto, Ontario M5G 1Z8, Canada

Tim Menke

Department of Physics, Harvard University, Cambridge, Massachusetts 02138, USA,
Research Laboratory of Electronics, Massachusetts Institute of Technology,
Cambridge, Massachusetts 02139, USA,
and Department of Physics, Massachusetts Institute of Technology,
Cambridge, Massachusetts 02139, USA

Wai-Keong Mok

Centre for Quantum Technologies, National University of Singapore, 117543 Singapore

Sukin Sim

Department of Chemistry and Chemical Biology, Harvard University, Cambridge, Massachusetts 02138, USA

Leong-Chuan Kwek[¶]

Centre for Quantum Technologies, National University of Singapore 117543, Singapore,
MajuLab, CNRS-UNS-NUS-NTU International Joint Research Unit UMI 3654, Singapore,
and National Institute of Education and Institute of Advanced Studies,
Nanyang Technological University, 637616 Singapore

Alán Aspuru-Guzik[**]

Department of Computer Science, University of Toronto, Toronto, Ontario M5S 2E4, Canada,
Chemical Physics Theory Group, Department of Chemistry, University of Toronto,

Toronto, Ontario M5G 1Z8, Canada, Vector Institute for Artificial Intelligence,
Toronto, Ontario M5S 1M1, Canada,
and Canadian Institute forthat Advanced Research, Toronto, Ontario M5G 1Z8, Canada

(published 15 February 2022)

A universal fault-tolerant quantum computer that can efficiently solve problems such as integer factorization and unstructured database search requires millions of qubits with low error rates and long coherence times. While the experimental advancement toward realizing such devices will potentially take decades of research, noisy intermediate-scale quantum (NISQ) computers already exist. These computers are composed of hundreds of noisy qubits, i.e., qubits that are not error corrected, and therefore perform imperfect operations within a limited coherence time. In the search for achieving quantum advantage with these devices, algorithms have been proposed for applications in various disciplines spanning physics, machine learning, quantum chemistry, and combinatorial optimization. The overarching goal of such algorithms is to leverage the limited available resources to perform classically challenging tasks. In this review, a thorough summary of NISQ computational paradigms and algorithms is provided. The key structure of these algorithms and their limitations and advantages are discussed. A comprehensive overview of various benchmarking and software tools useful for programming and testing NISQ devices is additionally provided.

DOI: 10.1103/RevModPhys.94.015004

CONTENTS

[*] Present address: Joint Center for Quantum Information and Computer Science and Joint Quantum Institute, NIST/University of Maryland, College Park, Maryland 20742, USA.
kishor.bharti1@gmail.com

[†] a.cervera.lierta@gmail.com

[‡] thihakyaw.phy@gmail.com

[§] Present address: Quantum Lab, Boehringer Ingelheim, 55218 Ingelheim am Rhein, Germany.

[¶] cqtklc@gmail.com

[**] alan@aspuru.com

I. INTRODUCTION

Quantum computing originated in the 1980s when physicists started to speculate about computational models that integrate the laws of quantum mechanics (Kaiser, 2011). Starting with the pioneering works of Benioff and Deutsch, which involved the study of quantum Turing machines and the notion of universal quantum computation (Benioff, 1980; Deutsch, 1985), the field continued to develop toward its natural application: the simulation of quantum systems (Manin, 1980; Feynman, 1982; Lloyd, 1996). Arguably, the drive for quantum computing took off in 1994 when Peter Shor provided an efficient quantum algorithm for finding prime factors of composite integers, rendering most classical cryptographic protocols unsafe (Shor, 1994). Since then, the study of quantum algorithms has matured as a subfield of quantum computing with applications in search and optimization, machine learning, simulation of quantum systems, and cryptography (Montanaro, 2016).

In the last 40 years, many scientific disciplines have converged toward the study and development of quantum algorithms and their experimental realizations. Quantum computers are, from the perspective of computational complexity, fundamentally different tools available for computationally intensive fields. The implementation of quantum algorithms requires that the minimal quantum information units, *qubits*, are as reliable as classical bits. Qubits need to be protected from environmental noise that induces decoherence but, at the same time, their states have to be controlled by external agents. This control includes the interaction that generates entanglement among qubits and the measurement operation that extracts the output of the quantum computation. It is technically possible to tame the effects of noise without compromising the quantum information process by developing quantum error-correction (QEC) protocols (Shor, 1995; Lidar and Brun, 2013; Terhal, 2015). The overhead of QEC in terms of the number of qubits is presently still far from current experimental capabilities. To achieve the goal of fault-tolerant quantum computation, the challenge is to scale up the number of qubits while maintaining sufficiently high qubit quality and fidelity in operations such as quantum gate implementation and measurement (Knill, Laflamme, and Zurek, 1998; Kitaev, 2003; Aharonov and Ben-Or, 2008). As the system size grows, it becomes highly challenging to contain the errors associated with cross talk and measurements below the required error-correction threshold.

Most quantum algorithms with performance guarantees require millions of physical qubits to successfully incorporate QEC techniques. Building such fault-tolerant quantum computers may take decades. Existing quantum devices contain on the order of 100 physical qubits. They are sometimes denoted as *noisy intermediate-scale quantum* (NISQ) devices (Preskill, 2018), in which qubits and quantum operations supported on NISQ devices are not error corrected and are, therefore, imperfect. One of the goals in the NISQ era is to extract the maximum quantum computational power from current devices while also continuing to develop techniques toward fault-tolerant quantum computation (Terhal, 2015).

A. Computational complexity theory in a nutshell

Defining a new computational paradigm enables one to solve or approach problems that could not be tackled with previously existing paradigms. With the development of quantum computing, new computational complexity classes have been recognized and proposed algorithms and goals have to be developed within well-known mathematical boundaries.

In this review, we often use some computational complexity-theoretic ideas to establish the domain and efficiency of the quantum algorithms covered. For this reason, we provide in this section a synopsis for a general audience and refer the interested reader to Arora and Barak (2009) for a more comprehensive treatment.

Complexity classes are groupings of problems by hardness, namely, the scaling of the cost of solving the problem with respect to a given resource, as a function of the "size" of an instance of the problem. We informally describe several of the most well-known classes here. (i) P: problems that can be solved in polynomial time with respect to input size by a deterministic classical computer. (ii) NP: a problem is said to be in NP if the problem of verifying the correctness of a proposed solution lies in P, irrespective of the difficulty of obtaining a correct solution. (iii) PH: refers to polynomial hierarchy. This class is a generalization of NP. It contains all the problems that one gets if one starts with a problem in the class NP and adds additional layers of complexity using quantifiers; i.e., it exists ($\exists$) and is present for all ($\forall$). As we add more quantifiers to a problem, it becomes more complex and is placed higher up in the polynomial hierarchy. We denote the classes in PH by Σ_i such that PH $=\cup_i \Sigma_i$. We have $\Sigma_1 =$ NP. The class Σ_i in PH can be interpreted in the context of two-player games where problems correspond to asking whether there is a winning strategy in $i/2$ rounds for player 1 in a game. Here one can interpret the quantifiers by asking whether there exists a move k_1 such that, no matter what move k_2 is played, there is a move k_3, etc., for $i/2$ rounds such that player 1 wins the two-player game. With increasing i, one would expect the problem to become more complex and hence

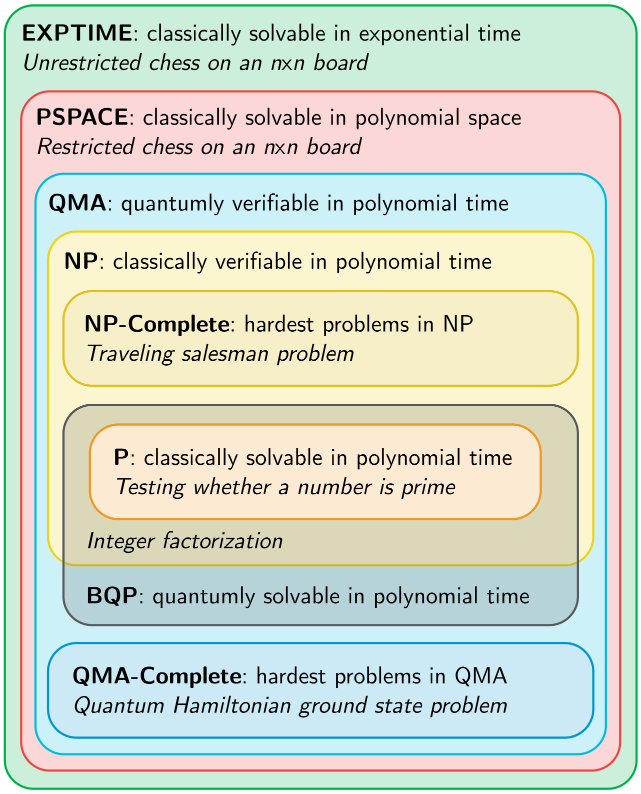
EXPTIME: classically solvable in exponential time
Unrestricted chess on an nxn board
PSPACE: classically solvable in polynomial space
Restricted chess on an nxn board
QMA: quantumly verifiable in polynomial time
NP: classically verifiable in polynomial time
NP-Complete: hardest problems in NP
Traveling salesman problem
P: classically solvable in polynomial time
Testing whether a number is prime
Integer factorization
BQP: quantumly solvable in polynomial time
QMA-Complete: hardest problems in QMA
Quantum Hamiltonian ground state problem

FIG. 1. Relevant complexity classes together with problem examples. For the chess example, the word "restricted" refers to a polynomial upper bound on the number of moves. The containment relations are suggestive. Some of them have not been mathematically proven, as it is a well-known open problem whether P is equal to NP.

$\Sigma_i \subseteq \Sigma_{i+1}$. (iv) BPP stands for bounded-error probabilistic polynomial time. A problem is said to be in BPP if it can be solved in polynomial time in the input size using a probabilistic classical computer. (v) BQP stands for bounded-error quantum polynomial time. Such problems can be solved in polynomial time in the input size using a quantum computer. (vi) PSPACE stands for polynomial space. The problems in PSPACE can be solved in polynomial space in the input size using a deterministic classical computer. Each class in PH is contained in PSPACE. However, it is not known whether PH is equal to PSPACE. (vii) EXPTIME stands for exponential time. The problems in EXPTIME can be solved in exponential time in the input size using a deterministic classical computer. (viii) QMA stands for quantum Merlin Arthur and is the quantum analog of the complexity class NP. A problem is said to be in QMA if, given "yes" as an answer, the solution can be verified in polynomial time (in the input size) using a quantum computer. Widely believed containment relations for some of the complexity classes are shown schematically in Fig. 1.

To understand the internal structure of complexity classes, the idea of "reductions" can be quite useful. One says that problem A is reducible to problem B if a method for solving B implies a method for solving A; one denotes the same by $A \leq B$. It is a common practice to assume the reductions as polynomial-time reductions. Intuitively, it could be thought that solving B is at least as difficult as solving A. Given a class C, a problem X is said to be C *hard* if every problem in class C reduces to X. We say that a problem X is C *complete* if X is C hard and also a member of C. The C-complete problems could be understood as capturing the difficulty of class C since any algorithm that solves one C-complete problem can be used to solve any problem in C.

A canonical example of a problem in the class BQP is integer factorization, which can be solved in polynomial time with a quantum computer using Shor's factoring algorithm (Shor, 1994). However, no classical polynomial-time algorithm is known for the aforementioned problem. Thus, the integer factorization problem is in BQP but not believed to be in P (Arora and Barak, 2009). While analyzing the performance of algorithms, it is prudent to perform complexity-theoretic sanity checks. For example, although quantum computers are believed to be powerful, they are not widely expected to be able to solve NP-complete problems, such as the traveling-salesman problem, in polynomial time. The quantum algorithms, however, could provide a speedup with respect to classical algorithms for NP-complete problems.

B. Experimental progress

In this section, we present a summary of recent quantum hardware and experiments. Interested readers should consult Acín *et al.* (2018) and references therein for further information on various quantum computing architectures.

Experimental progress in quantum computation can be measured using various figures of merit. The number of physical qubits must exceed a certain threshold to solve problems beyond the capabilities of a classical computer. However, there are several classical techniques capable of efficiently simulating certain quantum many-body systems. The success of several of these techniques, such as tensor networks (Verstraete, Murg, and Cirac, 2008; Orús, 2014), rely on the efficient representation of quantum states that are not highly entangled (Vidal, 2003, 2004). With universal quantum computers, one would expect to be able to generate and manipulate highly entangled quantum states. Hence, one imminent and practical direction toward demonstrating quantum advantage over classical machines consists of focusing on a region of the Hilbert space in which states cannot be efficiently represented with classical methods. Alternatively, one may implement particular computational tasks that are believed to be intractable using any classical computer, such as problems belonging only to quantum complexity classes.

Two recent experiments implemented the latter approach toward achieving quantum computational advantage. In 2019, the Google AI Quantum team implemented an experiment with the 53-qubit Sycamore chip (Arute *et al.*, 2019), which supported single-qubit gate fidelities of 99.85%, and two-qubit gate fidelities of 99.64% were attained on average. Quantum advantage was demonstrated against the best classical computers for the task of sampling the output of a pseudorandom quantum circuit.

Another quantum advantage experiment was implemented by Jian-Wei Pan's group using a *Jiuzhang* photonic quantum device performing Gaussian boson sampling (GBS) with 50 indistinguishable single-mode squeezed states (Zhong *et al.*, 2020). Here the quantum advantage was observed in sampling time complexity of a *Torontonian* of a matrix (Quesada, Arrazola, and Killoran, 2018) that scales exponentially with the photon click output. The Torontonian is a matrix function that determines the probability distribution of measurement outcomes, much like the permanent and *Hafnian* in other boson sampling models. Intuitively speaking, while the total number of perfect matchings in a bipartite graph is given by the permanent model, the Hafnian corresponds to the total number of perfect matchings in an arbitrarily given graph. Moreover, while the Hafnian is used in experiments counting the number of photons in each mode, the Torontonian corresponds to the case where one detects whether there are photons in each mode; see Sec. III.B for more details on GBS and the related terms.

There are several quantum computing platforms that researchers are actively developing at present to achieve scalable and practical universal quantum computers. The term "universal" is used to describe a quantum computer, using its native gate set, and can easily and accurately approximate any unitary gate; see Sec. V.B for more details. In addition to superconducting circuits and quantum optics, trapped-ion devices are also leading candidates for quantum computer architectures. In recent years, major advancements of trapped-ion devices include high-fidelity entangling gates reported by the Oxford group (Hughes *et al.*, 2020), all-to-all connectivity achieved by IonQ (Nam *et al.*, 2020), and transport and reordering capabilities in 2D trap array by the Boulder group (Wan *et al.*, 2020). In the last example, in addition to facilitating efficient transport of ions and quantum information exchange, the 2D architecture will allow implementations of more sophisticated QEC codes or surface codes (Lidar and Brun, 2013), and the smallest of it was realized in a superconducting qubit setup (Córcoles *et al.*, 2015).

Scientists and engineers are also developing hybrid quantum computing platforms trying to achieve similar feats to those previously described. These devices, which are built to solve specific problems, may not necessarily possess universal quantum gate sets. Coherent Ising machines (Utsunomiya, Takata, and Yamamoto, 2011; Wang *et al.*, 2013; Marandi *et al.*, 2014; Inagaki *et al.*, 2016; McMahon *et al.*, 2016), based on mutually coupled optical parametric oscillators, are a promising hybrid architecture for solving instances of hard combinatorial optimization problems. Recently, it was shown that the efficiency of these machines can be improved with error detection and correction feedback mechanisms (Kako *et al.*, 2020). See the recent review by Yamamoto *et al.* (2020) for an in-depth discussion about coherent Ising machines. Quantum annealing (Finnila *et al.*, 1994; Kadowaki and Nishimori, 1998) has been another prominent approach for achieving quantum advantage in the NISQ era (Perdomo-Ortiz *et al.*, 2018; Bouland *et al.*, 2020; Hauke *et al.*, 2020); see Sec. III.A for more details on quantum annealing.

Lastly, in recent years cloud-based quantum computers have become available, with which anyone with Internet access can control and manipulate delicate qubits and perform quantum computations on the fly. Examples of such platforms include IBM Quantum, Rigetti Computing, and Xanadu Quantum Cloud.

C. NISQ and near term

The experimental state of the art and the demand for QEC have encouraged the development of innovative algorithms capable of reaching the long-expected *quantum advantage*. Quantum advantage can be defined as a computation involving a quantum device that cannot be performed classically within reasonable amounts of time and energy resources. The term *near-term* quantum computation was coined to describe quantum algorithms tailored to be run on current quantum computing hardware or those that could be developed in the next few years. It is important to note that NISQ is a hardware-focused definition and does not necessarily imply a temporal connotation. NISQ devices can implement quantum circuits, in which all gates adhere to the topology of a specified graph G, in which the nodes of the graph correspond to qubits. The gates typically operate on one or two qubits. Because each gate operation involves a certain amount of noise, NISQ algorithms are naturally limited to employing circuits of shallow depths (Barak and Marwaha, 2021). Near-term algorithms, however, refer to those algorithms designed for quantum devices available in the next few years and carries no explicit reference to the absence of QEC. The phrase near term is subjective since different researchers may have other thoughts on how many years can be considered near term. Predicting experimental progress is always challenging, and such predictions are influenced by human bias. Algorithms developed for near-term hardware may be unfeasible if hardware advancement does not match the algorithm's experimental requirements.

D. Scope of the review

This review aims to accomplish three main objectives. The first is to provide a proper compilation of the available algorithms suited for the NISQ era. We present a summary of the crucial tools and techniques that have been proposed and harnessed to design such algorithms. The second objective is to discuss the implications of these algorithms in various applications such as quantum machine learning (QML), quantum chemistry, and combinatorial optimization. Finally, the third objective is to provide some perspective on potential future developments given the recent progress in quantum hardware.

Most of the current NISQ algorithms harness the power of quantum computers in a hybrid quantum-classical arrangement. Such algorithms delegate the classically difficult part of a computation to the quantum computer and perform the other on a sufficiently powerful classical device. These algorithms variationally update the variables of a parametrized quantum circuit and hence are referred to as *variational quantum algorithms* (VQAs) (Cao *et al.*, 2019; Cerezo, Sharma *et al.*, 2020; Endo, Cai *et al.*, 2020; McArdle *et al.*, 2020). The first proposals of VQAs were the variational quantum eigensolver (VQE) (Peruzzo *et al.*, 2014; Wecker, Hastings, and Troyer, 2015; McClean *et al.*, 2016), which was originally proposed to solve quantum chemistry problems, and the quantum approximate optimization algorithm (QAOA) (Farhi, Goldstone, and Gutmann, 2014), which was proposed to solve combinatorial optimization problems. While NISQ

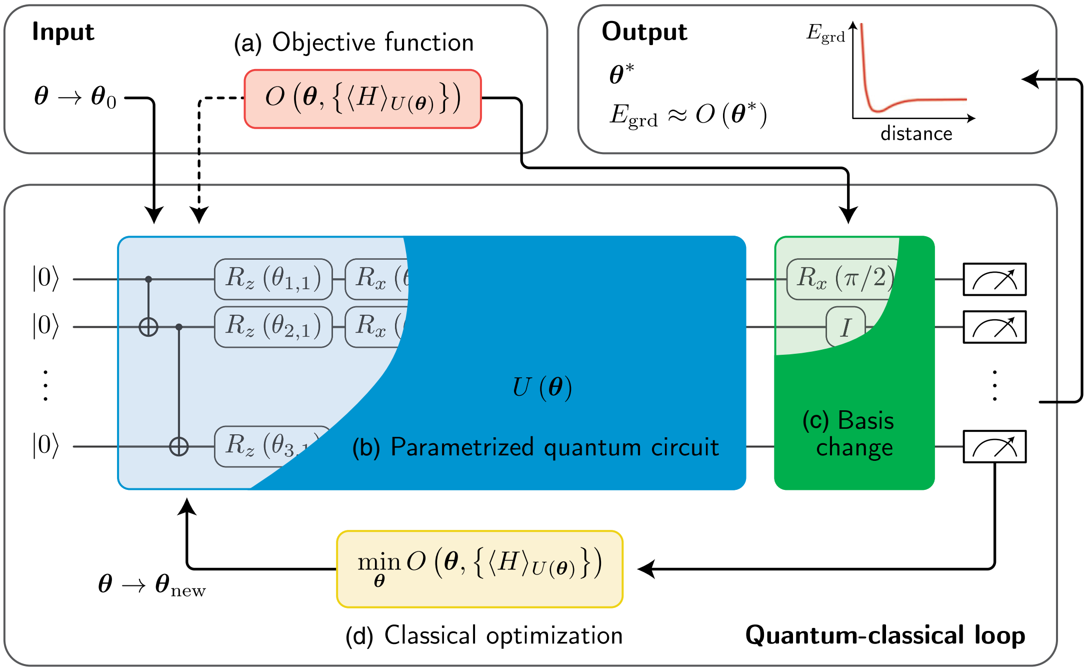
Input
(a) Objective function
$\boldsymbol{\theta} \to \boldsymbol{\theta}_0$
$O\left(\boldsymbol{\theta}, \{\langle H\rangle_{U(\boldsymbol{\theta})}\}\right)$
Output
$\boldsymbol{\theta}^*$
$E_{\rm grd} \approx O(\boldsymbol{\theta}^*)$
$E_{\rm grd}$
distance
$|0\rangle$
$|0\rangle$
$|0\rangle$
$R_z(\theta_{1,1})$
$R_z(\theta_{2,1})$
$R_z(\theta_{3,1})$
R_x
R_x
$U(\boldsymbol{\theta})$
(b) Parametrized quantum circuit
$R_x(\pi/2)$
I
(c) Basis change
$\min_{\boldsymbol{\theta}} O\left(\boldsymbol{\theta}, \{\langle H\rangle_{U(\boldsymbol{\theta})}\}\right)$
$\boldsymbol{\theta} \to \boldsymbol{\theta}_{\rm new}$
(d) Classical optimization
Quantum-classical loop

FIG. 2. Diagrammatic representation of a variational quantum algorithm (VQA). A VQA workflow can be divided into four main components: (a) the objective function O that encodes the problem to be solved; (b) the parametrized quantum circuit (PQC) U, in which variables $\boldsymbol{\theta}$ are tuned to minimize the objective; (c) the measurement scheme, which performs the basis changes and measurements needed to estimate expectation values that are used to evaluate the objective; and (d) the classical optimizer that minimizes the objective. The PQC can be defined heuristically, following hardware-inspired *Ansätze*, or designed from the knowledge about the problem Hamiltonian H. Inputs of a VQA are the circuit *Ansatz* $U(\boldsymbol{\theta})$ and the initial parameter values $\boldsymbol{\theta}_0$. Outputs include optimized parameter values $\boldsymbol{\theta}^*$and the minimum of the objective.

devices can arguably achieve quantum advantage for sampling problems, the question of their ability to provide advantage for optimization problems remains unanswered (Barak *et al.*, 2015; Barak and Marwaha, 2021). Despite its potential, it is important to note that there is currently no provable quantum advantage for VQAs with NISQ devices (Barak and Marwaha, 2021). In this review, we describe the building blocks of VQAs in Sec. II.

Other quantum computing paradigms propose different types of algorithms. They are inspired and hybridized with analog approaches. These paradigms include quantum annealing, digital-analog quantum computation, Gaussian boson sampling, and analog quantum computation. We present their fundamental properties in Sec. III.

In Sec. V, we examine the methods developed to best utilize NISQ algorithms as well as their theoretical and experimental challenges. We include the theoretical guarantees that some of these algorithms lie on as well as techniques to mitigate the errors coming from the use of noisy quantum devices. We also cover the possible trainability challenges that VQAs have and how to map theoretical NISQ circuits to real hardware. Section VI presents a range of applications of NISQ algorithms. Techniques to benchmark, compare, and quantify performance of current quantum devices are presented in Sec. VII. Like any other computational paradigm, quantum computing requires a language to establish human-machine communication. We explain the different levels of quantum programming and provide a list of open-source quantum software tools in Sec. V.C. Finally, we conclude this review in Sec. VIII by highlighting the increasing community involvement in this field and by presenting near-term and long-term goals of quantum computational research.

II. BUILDING BLOCKS OF VARIATIONAL QUANTUM ALGORITHMS

A VQA comprises several modular components that can be readily combined, extended, and improved with developments in quantum hardware and algorithms. These components include the objective function, the cost function to be variationally minimized; the parametrized quantum circuit (PQC), parametrized unitaries that are manipulated in the minimization of the objective; the measurement scheme, which estimates the expectation values needed to evaluate the objective; and the classical optimizer, the method used to obtain the optimal circuit parameters that minimize the objective. In Secs. II.A–II.D, we define each of these components, which are presented diagrammatically in Fig. 2.

A. Objective function

The Hamiltonian is a quantum operator that encodes information about a given physical system, such as a molecule or a spin chain. Its expectation value yields the energy of a quantum state, which is often used as the minimization target of a VQA; i.e., it obtains the Hamiltonian ground state. Problems that are not related to real physical systems can also be encoded in a Hamiltonian form such that they can be solved on a quantum computer. In general, any expectation value of a function written in an operational form (i.e., decomposed or encoded in a quantum operator) can also be evaluated on a quantum computer. After the Hamiltonian or operator of a problem has been determined, it must be decomposed into a set of particular operators that can be measured with a quantum processor. Such a decomposition, which is further discussed in Sec. II.A.1, is an important step in many quantum algorithms.

Within a VQA, one has access to measurements on qubits, in which the outcome probabilities are determined by the prepared quantum state. We consider only measurements on individual qubits in the standard computational basis and denote the probability to measure qubit q in the state $|0\rangle$ by p_0^q, where the qubit label q will be omitted whenever possible. The central element of a VQA is a parametrized cost or *objective function* O subject to a classical optimization algorithm, $\min_{\boldsymbol{\theta}} O(\boldsymbol{\theta}, \{\boldsymbol{p}_0(\boldsymbol{\theta})\})$. The objective function O and the measurement outcomes $\boldsymbol{p}_0$ of one or many quantum circuit evaluations depend on the set of parameters $\boldsymbol{\theta}$.

In practice, it is often inconvenient to directly work with the probabilities of the measurement outcomes when evaluating the objective function. Higher level formulations employ expectation value of the Hamiltonian H of the form

$$\langle H \rangle_{U(\boldsymbol{\theta})} \equiv \langle 0 | U^\dagger(\boldsymbol{\theta}) H U(\boldsymbol{\theta}) | 0 \rangle, \tag{1}$$

describing measurements on the quantum state generated by the unitary $U(\boldsymbol{\theta})$. This is in contrast to using the probabilities for individual qubit measurements. Arbitrary observables can be decomposed into basic measurements of the so-called Pauli strings, which can be evaluated in the computational basis, as explained here and in Sec. II.C. Restricting ourselves to expectation values instead of pure measurement probabilities, the objective function becomes

$$\min_{\boldsymbol{\theta}} O(\boldsymbol{\theta}, \{\langle H \rangle_{U(\boldsymbol{\theta})}\}). \tag{2}$$

This formulation often allows for more compact definitions of the objective function. For the original VQE (Peruzzo *et al.*, 2014) and QAOA (Farhi, Goldstone, and Gutmann, 2014) it can be described as a single expectation value $\min_{\boldsymbol{\theta}} \langle H \rangle_{U(\boldsymbol{\theta})}$, where the differences solely appear in the specific form and construction of the qubit Hamiltonian.

The choice of the objective function is crucial in a VQA to achieve the desired convergence. Vanishing gradient issues during the optimization, known as barren plateaus, are dependent on the cost function (Cerezo *et al.*, 2021); see Sec. IV.A for details.

1. Pauli strings

To extract the expectation value of the problem Hamiltonian, it is sufficient to express the Hamiltonian as a linear combination of primitive tensor products of Pauli matrices $\hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z$. We refer to these tensor products as *Pauli strings* $\hat{P} = \bigotimes_{j=1}^{n} \hat{\boldsymbol{\sigma}}$, where n is the number of qubits, $\hat{\boldsymbol{\sigma}} \in \{\hat{I}, \hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z\}$, and $\hat{I}$ is the identity operator. The Hamiltonian can then be decomposed as

$$H = \sum_{k=1}^{M} c_k \hat{P}_k, \tag{3}$$

where c_k is a complex coefficient of the kth Pauli string and the number of Pauli strings M in the expansion depends on the operator at hand. An expectation value in the sense of Eq. (1) then naturally decomposes into a set of expectation values, each defined by a single Pauli string

$$\langle H \rangle_U = \sum_{k=1}^{M} c_k \langle \hat{P}_k \rangle_U. \tag{4}$$

Examples of Hamiltonian objectives include molecules (by means of certain fermionic transformation to Pauli strings, as detailed in Sec. VI.A), condensed matter models written in terms of spin chains, and optimization problems encoded in a Hamiltonian form; see Sec. VI.C.

2. Fidelity

Instead of optimizing with respect to the expectation value of an operator, several VQAs require a subroutine to optimize the state obtained from the PQC $U(\boldsymbol{\theta})$, $|\Psi\rangle_{U(\boldsymbol{\theta})}$ with respect to a specific target state $|\Psi\rangle$. A commonly used cost function is the fidelity between the PQC and the target state

$$F(\Psi, \Psi_{U(\boldsymbol{\theta})}) \equiv |\langle \Psi | \Psi_{U(\boldsymbol{\theta})} \rangle|^2, \tag{5}$$

which is equivalent to the expectation value over the projector $\hat{\Pi}_\Psi = |\Psi\rangle\langle\Psi|$. The state preparation objective is then the minimization of the infidelity $1 - F(\Psi, \Psi_{U(\boldsymbol{\theta})})$ or the negated fidelity

$$\max_{\boldsymbol{\theta}} F(\Psi, \Psi_{U(\boldsymbol{\theta})}) = \min_{\boldsymbol{\theta}} (-\langle \hat{\Pi}_\Psi \rangle_{U(\boldsymbol{\theta})}). \tag{6}$$

If we know the efficient circuit U_Ψ that prepares the target state $|\Psi\rangle$, we can compute the fidelity with the inversion test by preparing the quantum state $U_\Psi^\dagger |\Psi_{U(\boldsymbol{\theta})}\rangle$ and measuring the projector onto the zero state $\hat{\Pi}_0 = |0\rangle^{\otimes n} \langle 0|^{\otimes n}$, with the fidelity given by $F(\Psi, \Psi_{U(\boldsymbol{\theta})}) = \langle \hat{\Pi}_0 \rangle_{U_\Psi^\dagger U(\boldsymbol{\theta})}$ (Havlíček *et al.*, 2019). If one wants to avoid optimizing with respect to a projector onto a single state, one can instead use a local observable that also becomes maximal for the target state, namely, $\hat{O} = (1/N) \sum_{k=1}^{N} |0_k\rangle\langle 0_k| \otimes I_{\bar{k}}$, where $I_{\bar{k}}$ is the identity matrix for all qubits except k and $|0_k\rangle$ is the zero state for qubit k (Barison, Vicentini, and Carleo, 2021; Cerezo *et al.*, 2021). Alternatively, one can use randomized measurements to measure the overlap $\mathrm{Tr}(\rho_1 \rho_2)$ of two density matrices ρ_1 and ρ_2 (van Enk and Beenakker, 2012; Elben *et al.*, 2019, 2020). First, one selects m unitaries $\{V_k\}_k$, which are chosen as the tensor product of Haar-random unitaries over the local d-dimensional subspace. These unitaries are applied on each quantum state $\rho_i = V_k \rho V_k^\dagger$ and ρ_i is sampled in the computational basis. One then estimates the probability $P_{V_k}^{(i)}(\mathbf{s})$ of measuring the computational basis state $\mathbf{s}$ for each quantum state ρ_i and unitary V_k. The overlap is given by

$$\mathrm{Tr}[\rho_1\rho_2] = \frac{d^N}{m}\sum_{k=1}^{m}\sum_{\mathbf{s},\mathbf{s}'}(-d)^{-\mathcal{D}[\mathbf{s},\mathbf{s}']}P^{(1)}_{V_k}(\mathbf{s})P^{(2)}_{V_k}(\mathbf{s}'), \quad (7)$$

where $\mathcal{D}[\mathbf{s},\mathbf{s}']$ is the Hamming distance between sampled computational basis states $\mathbf{s}$ and $\mathbf{s}'$. While the number of measurements scales exponentially with the number of qubits, the scaling is better than with state tomography. Moreover, importance sampling has been proposed to substantially reduce the number of samples necessary (Rath *et al.*, 2021).

Objective formulations over fidelities are prominent within state preparation algorithms in quantum optics (Kottmann *et al.*, 2020; Krenn, Erhard, and Zeilinger, 2020; Krenn *et al.*, 2020), excited state algorithms (Lee *et al.*, 2019; Kottmann, Anand, and Aspuru-Guzik, 2021), and QML (Cheng, Chen, and Wang, 2018; Benedetti, Garcia-Pintos *et al.*, 2019; Pérez-Salinas, Cervera-Lierta *et al.*, 2020; Huang *et al.*, 2021); see Sec. VI.B for more references and details. In these cases, the fidelities are often defined with respect to computational basis states e_i, such that $F_{e_i} = |\langle\Psi(\boldsymbol{\theta})|e_i\rangle|^2$.

3. Other objective functions

Hamiltonian expectation values are not the only objective functions that are used in VQAs. Any cost function that is written in an operational form can constitute a good choice. One such example is the conditional value at risk (CVaR). Given the set of energy basis measurements $\{E_1, \ldots, E_M\}$ arranged in a nondecreasing order, instead of using the expectation value from Eq. (1) as the objective function, it was proposed to use (Barkoutsos *et al.*, 2020)

$$\mathrm{CVaR}(\alpha) = \frac{1}{\lceil\alpha M\rceil}\sum_{k=1}^{\lceil\alpha M\rceil}E_k, \quad (8)$$

which measures the expectation value of the α tail of the energy distribution. Here $\alpha \in (0,1]$ is the confidence level. The CVaR (α) can be thought of as a generalization of the sample mean ($\alpha = 1$) and the sample minimum ($\alpha \to 0$).

Another proposal (Li *et al.*, 2020) is to use the Gibbs objective function

$$G = -\ln\langle e^{-\eta H}\rangle, \quad (9)$$

which is the cumulant generating function of the energy. The variable $\eta > 0$ is a hyperparameter to be tuned. For small η, the Gibbs objective function reduces to the mean energy in Eq. (1). Since both the CVaR and the Gibbs objective function can be reduced to the mean energy for suitable limits of the hyperparameters ($\alpha \to 1$ and $\eta \to 0$, respectively), their performances are guaranteed to be at least as good as using the mean energy $\langle H\rangle$. Empirically, by tuning the hyperparameters, both measures have been shown to outperform $\langle H\rangle$ for certain combinatorial optimization problems (Barkoutsos *et al.*, 2020; Li *et al.*, 2020).

B. Parametrized quantum circuits

Following the objective function, the next essential component of a VQA is the quantum circuit that prepares the state that optimizes the objective. This quantum circuit, also called a PQC, is a unitary operation that depends on a series of parameters. In this section, we describe how PQCs are defined and designed.

We define the state after application of the PQC as

$$|\Psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\Psi_0\rangle, \quad (10)$$

where $\boldsymbol{\theta}$ is a vector of parameters and $|\Psi_0\rangle$ is a given initial state. Typically, $|\Psi_0\rangle$ is a product state with all qubits in the $|0\rangle$ state, i.e., $|00\cdots0\rangle = |0\rangle^{\otimes n}$, where n is the number of qubits. In several VQAs, it is convenient to prepare that state in a particular form before applying the PQC. The state preparation operation would then depend on some other unitary operation P that may depend on variational parameters $\boldsymbol{\phi}$, $|\Psi_0\rangle = P(\boldsymbol{\phi})|0\rangle^{\otimes n}$. One example is the quantum feature maps defined in Sec. VI.B.1 that encode the data in the PQC. Any known property about the final state can also be used to obtain the initial guess. For instance, if we expect that the final state solution will contain all elements of the computational basis, or if we want to exploit a superposition state to seed the optimization, an initial state choice might be $P|0\rangle^{\otimes n} = H_d^{\otimes n}|0\rangle^{\otimes n}$, where H_d is the Hadamard gate. Applied to all qubits, H_d generates the even superposition of all basis states, i.e.,

$$|D\rangle = H_d^{\otimes n}|0\rangle^{\otimes n} = \frac{1}{\sqrt{n}}\sum_{i=1}^{n}|e_i\rangle, \quad (11)$$

where $|e_i\rangle$ are the computational basis states. In quantum chemistry algorithms, the initial state usually corresponds to the Hartree-Fock approximation; see Sec. VI.A for details. The choice of a good initial state allows the VQA to start the search in a region of the parameter space that is closer to the optimum.

The choice of the *Ansatz* U greatly affects the performance of a VQA. From the perspective of the problem, the *Ansatz* influences both the convergence speed and the closeness of the final state to a state that optimally solves the problem. On the other hand, the quantum hardware on which the VQA is executed has to be taken into account: Deeper circuits are more susceptible to errors, and some *Ansatz* gates are costly to construct from native gates. Accordingly, many of the *Ansätze* developed to date are classified as either more *problem inspired* or more *hardware efficient*, depending on their structure and application.

1. Problem-inspired *Ansätze*

An arbitrary unitary operation can be generated by a Hermitian operator $\hat{g}$ that, physically speaking, defines an evolution in terms of the t parameter as follows:

$$G(t) = e^{-i\hat{g}t}. \quad (12)$$

As an example, the generator $\hat{g}$ can be a Pauli matrix $\hat{\sigma}_i$, and thus $G(t)$ becomes a single-qubit rotation of the form

$$R_k(\theta) = e^{-i(\theta/2)\hat{\sigma}_k} = \cos(\theta/2)I - i\sin(\theta/2)\hat{\sigma}_k, \quad (13)$$

with $t = \theta$ and $\hat{g} = (1/2)\hat{\sigma}_k$, corresponding to the spin operator.

From a more abstract viewpoint, those evaluations can always be described as the time evolution of the corresponding quantum state, so the generator $\hat{g}$ is often referred to as a Hamiltonian. Note, however, that this Hamiltonian does not necessarily need to be the operator that describes the energy of the system of interest. In general, such generators can be decomposed into Pauli strings in the form of Eq. (3).

Within the so-called problem-inspired approaches, evolutions in the form of Eq. (12), with generators derived from properties of the system of interest, are used to construct the parametrized quantum circuits. The unitary coupled-cluster approach (discussed later), mostly applied to quantum chemistry problems, is one prominent example. The generators are elementary fermionic excitations, as shown in Eq. (17).

The *Suzuki-Trotter* (ST) expansion or decomposition (Suzuki, 1976) is a method to approximate a general unitary in the form of Eq. (12) as a function of the t parameter. This can be done by decomposing $\hat{g}$ into a sum of noncommuting operators $\{\hat{o}_k\}_k$, with $\hat{g} = \sum_k c_k \hat{o}_k$ and certain coefficients c_k. The operators $\hat{o}_k$ are chosen such that the evolution unitary $e^{-i\hat{o}_k t}$ can be easily implemented as Pauli strings $\hat{P}_k$. The full evolution over t can now be decomposed into integer m equal-sized steps as

$$e^{-i\hat{g}t} = \lim_{m\to\infty}\left(\prod\nolimits_k e^{-ic_k\hat{o}_k t/m}\right)^m. \quad (14)$$

For practical purposes, the time evolution can be approximated by a finite number m. When Pauli strings are used, this provides a systematic method to decompose an arbitrary unitary, generated by $\hat{g}$, into a product of multiqubit rotations $e^{-ic_k\hat{P}_k t/m}$, which themselves can be decomposed into primitive one- and two-qubit gates. Here we used the second-order ST decomposition to approximate the true unitary at each time step t. The error incurred from the approximation can be bounded by $||U_{\hat{g}}(\Delta_t) - U_{\hat{g}}^{\mathrm{ST}}(\Delta_t)|| \leq \sum_{k=1}^m ||[[H_k, H_{>k}], H_k]] + [[H_{>k}, H_k], H_{>k}]]||\Delta_t^3$, where $H_{>k} = \sum_{\beta>k} H_\beta$ and $H_k = c_k\hat{o}_k$ (Poulin et al., 2014).

Knowledge regarding the physics of the particular Hamiltonian to be *Trotterized* can substantially reduce the number of gates needed to implement this method. For instance, Kivlichan et al. (2018) showed that by using fermionic SWAP gates it is possible to implement a Trotter step for electronic structure Hamiltonians using first-neighbor connectivity circuits with $N^2/2$ two-qubit gate width and N depth, where N is the number of spin orbitals. They also showed that arbitrary Slater determinants can be efficiently implemented with $N/2$ gates of circuit depth.

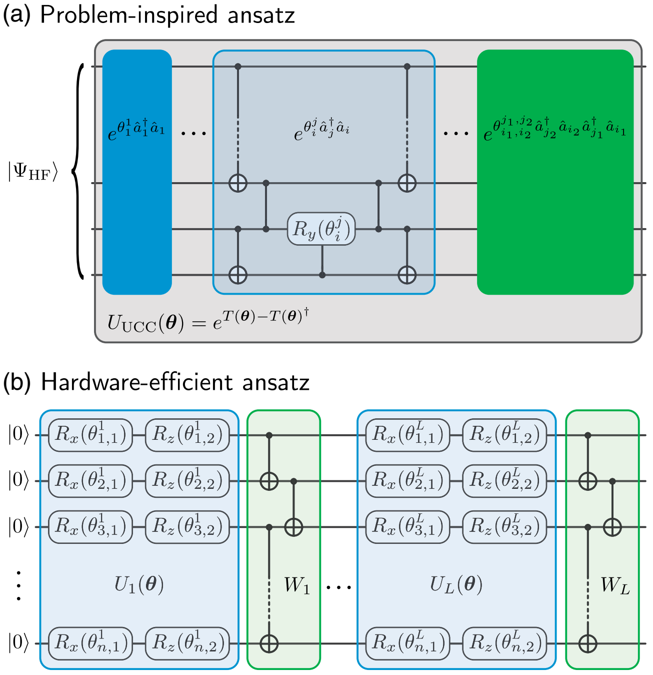
(a) Problem-inspired ansatz
$|\Psi_{\mathrm{HF}}\rangle$
$e^{\theta_1^1\hat{a}_1^\dagger\hat{a}_1}$
$e^{\theta_i^j\hat{a}_j^\dagger\hat{a}_i}$
$R_y(\theta_i^j)$
$e^{\theta_{i_1,i_2}^{j_1,j_2}\hat{a}_{j_2}^\dagger\hat{a}_{i_2}\hat{a}_{j_1}^\dagger\hat{a}_{i_1}}$
$U_{\mathrm{UCC}}(\boldsymbol{\theta}) = e^{T(\boldsymbol{\theta})-T(\boldsymbol{\theta})^\dagger}$
(b) Hardware-efficient ansatz

FIG. 3. Examples of circuits for problem-inspired and hardware-efficient *Ansätze*. (a) Circuit of the unitary coupled-cluster *Ansatz* with a detailed view of a fermionic excitation, as discussed by Yordanov, Arvidsson-Shukur, and Barnes (2020). (b) Hardware-efficient *Ansatz* tailored to a processor that is optimized for single-qubit x and z rotations and nearest-neighbor two-qubit CNOT gates.

Unitary coupled cluster.—Historically, problem-inspired *Ansätze* were proposed and implemented before hardware-efficient *Ansätze*; see Fig. 3. They arose from the quantum chemistry–specific observation that the unitary coupled-cluster (UCC) *Ansatz* (Taube and Bartlett, 2006), which adds quantum correlations to the Hartree-Fock approximation, is inefficient to represent on a classical computer (Yung et al., 2014). Leveraging quantum resources, the UCC *Ansatz* was instead realized as a PQC on a photonic processor (Peruzzo et al., 2014). It is constructed from the parametrized cluster operator $T(\boldsymbol{\theta})$ and acts on the Hartree-Fock ground state $|\Psi_{\mathrm{HF}}\rangle$ as

$$|\Psi(\boldsymbol{\theta})\rangle = e^{T(\boldsymbol{\theta})-T(\boldsymbol{\theta})^\dagger}|\Psi_{\mathrm{HF}}\rangle. \quad (15)$$

The cluster operator is given by $T(\boldsymbol{\theta}) = T_1(\boldsymbol{\theta}) + T_2(\boldsymbol{\theta}) + \cdots$, with

$$T_1(\boldsymbol{\theta}) = \sum_{\substack{i\in\mathrm{occ}\\ j\in\mathrm{virt}}}\theta_i^j\hat{a}_j^\dagger\hat{a}_i,$$
$$T_2(\boldsymbol{\theta}) = \sum_{\substack{i_1,i_2\in\mathrm{occ}\\ j_1,j_2\in\mathrm{virt}}}\theta_{i_1,i_2}^{j_1,j_2}\hat{a}_{j_2}^\dagger\hat{a}_{i_2}\hat{a}_{j_1}^\dagger\hat{a}_{i_1}, \quad (16)$$

and higher-order terms following accordingly (O'Malley et al., 2016). The operator $\hat{a}_k$ is the annihilation operator of the kth Hartree-Fock orbital, and the sets "occ" and "virt" refer to the occupied and unoccupied Hartree-Fock orbitals.

Owing to their decreasing importance, the series is usually truncated after the second or third term. The *Ansatz* is termed *UCCSD* or *UCCSDT*, respectively, referring to the inclusion of single, double, and triple excitations from the Hartree-Fock ground state. The k-UpCCGSD approach restricts the double excitations to pairwise excitations but allows k layers of the approach (Lee et al., 2019). After mapping to Pauli strings as described in Sec. II.A.1, the *Ansatz* is converted to a PQC, usually via the Trotter expansion in Eq. (14).

In its original form, the UCC *Ansatz* faces several drawbacks in its application to larger chemistry problems, as well as to other applications. For strongly correlated systems, the widely proposed UCCSD *Ansatz* is expected to have insufficient overlap with the true ground state and produces large circuit depths (Grimsley, Economou *et al.*, 2019; Lee *et al.*, 2019). Consequently, improvements and alternative *Ansätze* have been proposed to mitigate these challenges. We restrict our discussion here to provide a short overview of alternative *Ansatz* developments. For more details on the UCC *Ansatz*, see Sec. VI.A.

Factorized unitary coupled-cluster and adaptive approaches.—The noncommuting nature of the fermionic excitation generators, given by the cluster operators in Eq. (17), leads to difficulties in decomposing the canonical UCC *Ansatz* (15) into primitive one- and two-qubit unitaries. The first approaches employed the Trotter decomposition (14) using a single step (McClean *et al.*, 2016; Romero *et al.*, 2018). The accuracy of the obtained factorized *Ansatz* depends, however, on the order of the primitive fermionic excitations (Grimsley, Claudino *et al.*, 2019; Izmaylov, Díaz-Tinoco, and Lang, 2020).

Alternative approaches propose using factorized unitaries that are directly constructed from primitive fermionic excitations (Evangelista, Chan, and Scuseria, 2019; Izmaylov, Díaz-Tinoco, and Lang, 2020). Adaptive approaches are a special case of a factorized *Ansatz* where the unitary is iteratively grown by subsequently screening and adding primitive unitary operators from a predefined operator pool. The types of operator pools can be divided into two classes: adapt VQE (Grimsley, Economou *et al.*, 2019), which constructs the operator pool from primitive fermionic excitations, and qubit coupled cluster (Ryabinkin *et al.*, 2018), which uses Pauli strings. In both works, the screening process is based on energy gradients with respect to the prospective operator candidate. Since this operator is the trailing part of the circuit, the gradient can be evaluated through the commutator of the Hamiltonian with the generator of that operator. In contrast to a commutator-based gradient evaluation, direct differentiation, as proposed by Kottmann *et al.* (2021), allows gradient evaluations with similar cost as the original objective and generalizes the approach by allowing screening and insertion of operators at arbitrary positions in the circuit. This is necessary for excited state objectives, as discussed in Sec. VI.A.4.

Extended approaches include iterative methods (Ryabinkin *et al.*, 2020), operator pool construction from involutory linear combinations of Pauli strings (Lang, I. G. Ryabinkin, and Izmaylov, 2020), Pauli string pools from decomposed fermionic pools (Tang *et al.*, 2019), mutual information-based operator pool reduction (Zhang, Kyaw, Kottman *et al.*, 2021), measurement reduction schemes based on the density matrix reconstruction (Liu, Li, and Yang, 2020), and external perturbative corrections (Ryabinkin, Izmaylov, and Genin, 2021).

Variational Hamiltonian Ansatz.—Inspired by adiabatic state preparation, the variational Hamiltonian *Ansatz* (VHA) was developed to reduce the number of parameters and accelerate the convergence (Wecker, Hastings, and Troyer, 2015; McClean *et al.*, 2016). Instead of the Hartree-Fock operators, the terms of the fermionic Hamiltonian itself are used to construct the PQC. For this purpose, the fermionic Hamiltonian H is written as a sum of M terms $H = \sum_i \hat{h}_i$. The grouping of Hamiltonian terms depends on the problem. The PQC is then chosen as

$$U_{\mathrm{VHA}} = \prod_{i=1}^{M} e^{(i\theta_i \hat{h}_i)}, \tag{17}$$

with the operators in the product ordered by decreasing i. The unitary corresponds to n short time evolutions under different parts of the Hamiltonian, where the terms $\hat{h}_i$ of the Hamiltonian can be repeated multiple times. The initial state is chosen so that it is easy to prepare yet is related to the Hamiltonian. An example is the eigenstate of the diagonal part of H. The Fermi-Hubbard model, with a few simple interaction terms, is proposed as the most promising near-term application of the method. However, it is also shown that the VHA can outperform specific forms of the UCCSD *Ansatz* for strongly correlated model systems in quantum chemistry. In Sec. VI.A we discuss some VQE-inspired algorithms that also use adiabatic evolution to improve the performance of the algorithm.

Quantum approximate optimization algorithm.—QAOA is one of the canonical NISQ-era algorithms designed to provide approximate solutions to combinatorial optimization problems (Farhi, Goldstone, and Gutmann, 2014). QAOA has been studied in depth over the years both empirically and theoretically. As of now QAOA has not demonstrated any speedup over classical algorithms for any practically relevant task. Understanding the potential of QAOA with respect to classical algorithms is an active area of study.

The cost function C of a QAOA is designed to encode a combinatorial problem by means of bit strings that form the computational basis. With the computational basis vectors $|e_i\rangle$, one can define the problem Hamiltonian H_P as (see Sec. VI.C.1 for an example)

$$H_P \equiv \sum_{i=1}^{n} C(e_i)|e_i\rangle\langle e_i|, \tag{18}$$

and the mixing Hamiltonian H_M as

$$H_M \equiv \sum_{i=1}^{n} \hat{\sigma}_x^i. \tag{19}$$

The initial state in the QAOA is conventionally chosen to be the uniform superposition state $|D\rangle$ from Eq. (11). The final quantum state is given by alternately applying H_P and H_M on the initial state p times,

$$|\Psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle \equiv e^{-i\beta_p H_M} e^{-i\gamma_p H_P} \dots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_P}|D\rangle, \tag{20}$$

with $\boldsymbol{\gamma} \equiv (\gamma_1, \gamma_2, \dots, \gamma_p)$ and $\boldsymbol{\beta} \equiv (\beta_1, \beta_2, \dots, \beta_p)$. A quantum computer is used to evaluate the objective function

$$C(\boldsymbol{\gamma},\boldsymbol{\beta}) \equiv \langle \Psi(\boldsymbol{\gamma},\boldsymbol{\beta})|H_P(\boldsymbol{\gamma},\boldsymbol{\beta})|\Psi(\boldsymbol{\gamma},\boldsymbol{\beta})\rangle, \tag{21}$$

and a classical optimizer is used to update the $2p$ angles $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ until C is maximized, i.e., $C(\boldsymbol{\gamma}^*,\boldsymbol{\beta}^*) \equiv \max_{\boldsymbol{\gamma},\boldsymbol{\beta}} C(\boldsymbol{\gamma},\boldsymbol{\beta})$. Here p is often referred to as the QAOA level or depth. Since the maximization at level $p-1$ is a constrained version of the maximization at level p, the performance of the algorithm improves monotonically with p in the absence of experimental noise and infidelities.

In adiabatic quantum computing (see Sec. III.A), we start at the ground state of H_M and slowly move toward the ground state of H_P by slowly changing the Hamiltonian. In QAOA, we alternate between H_M and H_P. One can think of QAOA as a Trotterized version of quantum annealing. Indeed, the adiabatic evolution as used in quantum annealing can be recovered in the limit of $p \to \infty$.

For a combinatorial optimization problem with hard constraints to be satisfied, penalties can be added to the cost function. In practice, this may not be an efficient strategy, as it is still possible to obtain solutions that violate several of the hard constraints. A variation of the QAOA to deal with these constraints was also discussed in the Sec. VII from the original proposal (Farhi, Goldstone, and Gutmann, 2014). Building on previous work in quantum annealing (Hen and Sarandy, 2016; Hen and Spedalieri, 2016), Hadfield *et al.* (2017) proposed to encode the hard constraints directly into the mixing Hamiltonian. This approach yields the main advantage of restricting the state evolution to the feasible subspace where no hard constraints are violated. This speeds up the classical optimization routine to find the optimal angles. This framework was later generalized as the *quantum alternating operator Ansatz* to consider phase-separation and mixing unitary operators [$U_P(\gamma)$ and $U_M(\beta)$, respectively], which need not originate from the time evolution of a Hamiltonian (Hadfield *et al.*, 2019). The operators $e^{-i\beta H_M}$ and $e^{-i\gamma H_P}$ from Eq. (20) are replaced by $U_M(\beta)$ and $U_P(\gamma)$, respectively. It is worth noting that both the quantum approximate optimization algorithm and the quantum alternating operator *Ansatz* are abbreviated "QAOA" in the literature. In this case, we suggest "QuAltOA" as an acronym for the quantum alternating operator *Ansatz* to distinguish it from the quantum approximate optimization algorithm.

The use of QAOA for combinatorial optimization is presented in Sec. VI.C. Some theoretical guarantees of this *Ansatz* are introduced in Sec. IV.D.

2. Hardware-efficient *Ansätze*

Thus far we have described circuit *Ansätze* constructed from the underlying physics of the problem to be solved. Although it has been shown computationally that such *Ansätze* can ensure fast convergence to a satisfying solution state, they can be challenging to realize experimentally. Quantum computing devices possess a series of experimental limitations that include, among others, specific qubit connectivity, a restricted gate set, and limited gate fidelities and coherence times. Therefore, existing quantum hardware is not suited to implement deep and highly connected circuits required for the UCC and similar *Ansätze* for applications beyond basic demonstrations such as the H_2 molecule (Moll *et al.*, 2018).

A class of hardware-efficient *Ansätze* [see Fig. 3(b)] has been proposed to accommodate device constraints (Kandala *et al.*, 2017). The common trait of these circuits is the use of a limited set of quantum gates as well as a particular qubit connection topology. The gate set usually consists of a two-qubit entangling gate and up to three single-qubit gates. The circuit is then constructed from blocks of single-qubit gates and entangling gates, which are applied to multiple or all qubits in parallel. Each of these blocks is usually called a *layer*, and the *Ansatz* circuit generally has multiple such layers.

The quantum circuit of a hardware-efficient *Ansatz* with L layers is usually given by

$$U(\boldsymbol{\theta}) = \prod_{k=1}^{L} U_k(\boldsymbol{\theta}_k)W_k, \tag{22}$$

where $\boldsymbol{\theta} = (\boldsymbol{\theta}_1,, \ldots, \boldsymbol{\theta}_L)$ are the variational parameters, $U_k(\boldsymbol{\theta}_k) = \exp(-i\boldsymbol{\theta}_k V_k)$ is a unitary derived from a Hermitian operator V_k, and W_k represents nonparametrized quantum gates. Typically, the V_k operators are single-qubit rotation gates; i.e., V_k are Pauli strings acting locally on each qubit. In those cases, U_k becomes a product of combinations of single-qubit rotational gates, with each one defined as in Eq. (13). W_k is an entangling unitary constructed from gates that are native to the architecture at hand, such as CNOT or CZ gates for superconducting qubits or XX gates for trapped ions (Krantz *et al.*, 2019; Wright *et al.*, 2019). Following this approach, the so-called alternating layered *Ansatz* is a particular case of these hardware-efficient *Ansätze* that consists of layers of single-qubit rotations and blocks of entangling gates that entangle only a local set of qubits and are shifted every alternating layer.

The choice of these gates, their connectivity, and their ordering influences the portion of the Hilbert space that the *Ansatz* covers and how fast it converges for a specific problem. Some of the most relevant properties of hardware-efficient *Ansätze*, namely, expressibility, entangling capability, and the number of parameters and layers needed, were studied by Sim, Johnson, and Aspuru-Guzik (2019), Bravo-Prieto *et al.* (2020), Nakaji and Yamamoto (2020a), and Woitzik *et al.* (2020) and further discussed in Sec. IV.B.

Instead of choosing between the problem-inspired and hardware-efficient modalities, some PQC designers have chosen an intermediate path. One example is the use of an exchange-type gate, which can be implemented natively in transmons, to construct a PQC that respects the symmetry of the variational problem (Ganzhorn *et al.*, 2019; Sagastizabal *et al.*, 2019b). Such an *Ansatz* leads to particularly small parameter counts for quantum chemistry problems such as the H_2 and LiH molecules (Gard *et al.*, 2020). Another intermediate approach, termed QOCA for its inspiration from quantum optimal control, is to add symmetry-breaking unitaries, akin to a hardware-efficient *Ansatz*, into the conventional VHA circuit (Choquette *et al.*, 2020). This modification enables excursions of the variational state into previously restricted sections of the Hilbert space, which is numerically shown to yield shortcuts in solving particular fermionic problems.

C. Measurement

To gain information about the quantum state that has been prepared on the quantum hardware, one needs to estimate the expectation value of the objective function $\langle \hat{O} \rangle_{U_{\boldsymbol{\theta}}}$. The most direct approach to estimate expectation values is to apply a unitary transformation on the quantum state to the diagonal basis of the observable $\hat{O}$ and to obtain the probability of measuring specific computational states corresponding to an eigenvalue of $\hat{O}$. In other words, it is to determine whether a measured qubit is in the $|0\rangle$ or $|1\rangle$ state. For experimental details on this task, see existing reviews such as those on superconducting qubits (Krantz *et al.*, 2019) or ion traps (Häffner, Roos, and Blatt, 2008). However, on NISQ devices, the transformation to the diagonal basis mentioned before can be an overly costly one. As a NISQ friendly alternative, most observables of interest can be efficiently parametrized in terms of Pauli strings, as previously shown, and transformed into their diagonal basis by simple single-qubit rotations, as shown later.

Measurement of Pauli strings.—The expectation value of the $\hat{\sigma}_z$ operator on a particular qubit can be measured by reading out the probabilities of the computational basis state $\{|0\rangle, |1\rangle\}$ as

$$\langle \psi | \hat{\sigma}_z | \psi \rangle \equiv \langle \hat{\sigma}_z \rangle = |\alpha|^2 - |\beta|^2, \tag{23}$$

where $|\alpha|^2$ is the probability of measuring the qubit in state $|0\rangle$, $|\beta|^2$ is the probability to measure the qubit in state $|1\rangle$, and $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Measurements defined by $\hat{\sigma}_x$ and $\hat{\sigma}_y$ can be defined similarly by transforming them into the $\hat{\sigma}_z$ basis first. The transformation is given by primitive single-qubit gates

$$\hat{\sigma}_x = R_y^\dagger(\pi/2)\hat{\sigma}_z R_y(\pi/2) = H_\mathrm{d}\hat{\sigma}_z H_\mathrm{d}, \tag{24}$$

$$\hat{\sigma}_y = R_x^\dagger(\pi/2)\hat{\sigma}_z R_x(\pi/2) = S H_\mathrm{d}\hat{\sigma}_z H_\mathrm{d} S^\dagger, \tag{25}$$

where $S = \sqrt{\hat{\sigma}_z}$ and $H_\mathrm{d} = (\hat{\sigma}_x + \hat{\sigma}_z)/\sqrt{2}$ is the Hadamard gate. To measure $\hat{\sigma}_x$ on a quantum state $|\psi\rangle$, we then rotate $\hat{\sigma}_x$ into the z axis by applying H_d and measuring in a logical $\hat{\sigma}_z$ basis, i.e.,

$$\langle \hat{\sigma}_x \rangle \equiv \langle \psi | \hat{\sigma}_x | \psi \rangle = \langle \psi | H_\mathrm{d}\hat{\sigma}_z H_\mathrm{d} | \psi \rangle = \alpha\beta^* + \alpha^*\beta. \tag{26}$$

The same applies for $\langle \hat{\sigma}_y \rangle$. Arbitrary Pauli strings $\hat{P}$, with primitive Pauli operations $\hat{\sigma}_{f(k)} \in \{\sigma_x, \sigma_y, \sigma_z\}$ on qubits $k \in K$, can then be measured by the same procedure on each individual qubit as

$$\langle \hat{P} \rangle_U = \left\langle \prod_{k\in K} \sigma_z(k) \right\rangle_{\tilde{U}U}, \tag{27}$$

where $\tilde{U}$ is a product of single-qubit rotations according to Eqs. (24) and (25) depending on the Pauli operations $\hat{\sigma}_{f(k)}$ at qubit k.

Thus far we have discussed expectation values of a physical observable $\langle \hat{O} \rangle$, which is the mean value averaged over an infinite number of measurements. In practice, one can sample only a finite number of single-shot measurements N_s of the quantum state and estimate the expectation values within a certain finite error. For a Pauli string $\hat{P}$, the number of measurement samples N_s needed to estimate the expectation value $\langle \hat{P} \rangle_U$, with an additive error of at most ϵ and a failure probability of at most δ, is bounded by Hoeffding's inequality as follows (Huang, Bharti, and Rebentrost, 2019):

$$N_\mathrm{s} \geq \frac{2}{\epsilon^2} \log\left(\frac{2}{\delta}\right). \tag{28}$$

In particular, the error ϵ decreases with the inverse square root of the number of measurements $\epsilon \propto 1/\sqrt{N_\mathrm{s}}$.

For many problems, such as quantum chemistry–related tasks, the number of terms in the cost Hamiltonian to be estimated can become large. A naive way of measuring each Pauli string separately may incur a prohibitively large number of measurements. Recently several more efficient approaches have been proposed; see Bonet-Monroig, Babbush, and O'Brien (2020) for an overview. The common idea is to group different Pauli strings that can be measured simultaneously such that a minimal number of measurements need to be performed.

Pauli strings that commute qubitwise, i.e., the Pauli operators on each qubit commute, can be measured at the same time (McClean *et al.*, 2016; Kandala *et al.*, 2017). The problem of finding the minimal number of groups can be mapped to the minimum clique cover problem, which is NP hard in general, but good heuristics exist (Verteletskyi, Yen, and Izmaylov, 2020). One can collect mutually commuting operators and transform them into a shared eigenbasis, which adds unitary transformation to the measurement scheme (Crawford, van Straaten *et al.*, 2019; Gokhale *et al.*, 2019; Yen, Verteletskyi, and Izmaylov, 2020). Combinations of the single-qubit and Bell measurements have been proposed as well (Hamamura and Imamichi, 2020).

Alternatively, one can use a method called unitary partitioning to linearly combine different operators into a unitary and use the so-called Hadamard test (discussed later) to evaluate it (Izmaylov *et al.*, 2020; Zhao *et al.*, 2020). Izmaylov, Yen, and Ryabinkin (2019) decomposed the observables decomposed into the so-called mean-field Hamiltonians, which can be measured more efficiently if one measures one qubit after the other, and used information from previous measurement outcomes.

For specific problems such as chemistry and condensed matter systems, it is possible to use the structure of the problem to reduce the number of measurements (Gokhale and Chong, 2019; Huggins *et al.*, 2019; Cade *et al.*, 2020; Cai, 2020). In particular, Cai (2020), where a Fermi-Hubbard model is studied using VQE, reduced the number of measurements by considering multiple orderings of the qubit operators when applying the Jordan-Wigner transformation. In the context of quantum chemistry, the largest reduction to date could be achieved using the Cartan subalgebra approach of Yen and Izmaylov (2020). Other approaches use classical shadows (Hadfield *et al.*, 2020), a classical approximation of the quantum state of interest, or neural network estimators (Torlai *et al.*, 2020) to decrease the number of measurements. All those kinds of optimizations require an understanding of the underlying problem and are usually not applicable for every use of the VQE.

Measurement of overlaps.—Several VQAs require the measurement of overlap of a quantum state $|\psi\rangle$ with unitary U in the form of $\langle\psi|U|\psi\rangle$. This overlap is in general not an observable and has both real and imaginary parts. The Hadamard test can evaluate such a quantity on the quantum computer using a single extra qubit (Miquel *et al.*, 2002). The idea is to apply a controlled U operation, with control on that qubit, and target U on the quantum state. One can then measure from this single-qubit state both real and imaginary parts of the overlap. A downside of this method is the requirement to be able to implement a controlled unitary, which may require too many resources on current quantum processors. Alternative methods to measure the overlap without the use of control unitaries have been proposed (Mitarai and Fujii, 2019). One idea is to decompose U into a sum of Pauli strings, and then to measure the expectation value of each Pauli string individually. Another approach is possible if U can be rewritten as a product of unitaries U_q that act locally on only a few qubits. One can then find via classical means the diagonalization of $U_q = V_q^\dagger D V_q$, with the diagonal matrix D and V_q a unitary. The overlap can be found by applying the V_q unitaries on the state $|\psi\rangle$, measuring the outcomes on the computational basis, and conducting postprocessing of the results with the classically calculated eigenvalues of D.

Classical shadows.—This is a powerful technique to accurately predict M expectation values $\mathrm{Tr}(\hat{O}_i\rho)$, $1 \leq i \leq M$, of an unknown quantum state ρ (Huang, Kueng, and Preskill, 2020). The method was based on and inspired by shadow tomography (Aaronson, 2020). A random unitary U is first applied on the state $\rho \rightarrow U\rho U^\dagger$ and then all the qubits are measured on a computational basis. This step is repeated with several random unitaries U. Common choices for U are unitaries that can be efficiently computed on a classical computer such as random n-qubit Clifford circuits or tensor products of single-qubit rotations. By postprocessing the measurement results, one can gather a classical shadow, which is a classical representation of the quantum state ρ. There are performance guarantees that classical shadows with a size of the order of $\log M$ suffice to predict M expectation values simultaneously. For investigations involving classical shadow tomography protocols in the presence of noise, see Chen, Yu *et al.* (2020) and Koh and Grewal (2020). Experimental realizations have been performed recently as well (Struchalin *et al.*, 2021; Zhang, Sun *et al.*, 2021).

D. Parameter optimization

In principle, the PQC parameter optimization to minimize the objective does not differ from any multivariate optimization procedures and standard classical methods can be applied (Lavrijsen, Tudor *et al.*, 2020). However, in the NISQ era, the coherence time is short, which means that high-depth analytical gradient circuits cannot be implemented. In addition, one of the greatest challenges in parameter optimization is the large number of measurements required for estimating the mean value of an observable to high precision. Because of this high sampling rate, the measurement process can become a significant bottleneck in the overall algorithm run-time. Thus, an effective optimizer for PQCs should try to minimize the number of measurements or function evaluations. As the last criterion, the optimizer should be resilient to noisy data coming from current devices and precision on expectation values that are limited by the number of shots in the measurement. These three requirements imply that certain existing algorithms are better suited for PQC optimization and are more commonly used and that new algorithms are being developed specifically for PQC optimization. Some intuitive concepts of the mechanisms behind the optimization of quantum problems were investigated by McClean, Harrigan *et al.* (2020). Recently Bittel and Kliesch (2021) showed that the classical optimization corresponding to VQAs is a NP-hard problem.

In this section, we first review two classes of optimization, gradient based and gradient free. We also consider resource-aware optimization methods and strategies that additionally minimize quantities associated with the quantum cost of optimization. While we reserve more detailed descriptions for the respective references and the Supplemental Material (824), we highlight the main features and advantages for each optimization strategy.

1. Gradient-based approaches

A common approach to optimize an objective function $f(\boldsymbol{\theta})$ is via its gradient, i.e., the change of the function with respect to a variation of its M parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_M)$. The gradient indicates the direction in which the objective function shows the greatest change. This is a local optimization strategy, as one uses information starting from a given initial parameter value $\boldsymbol{\theta}^{(0)}$ and iteratively updates $\boldsymbol{\theta}^{(t)}$ over multiple discrete steps t. A common update rule for each θ_i is

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \eta \partial_i f(\boldsymbol{\theta}), \tag{29}$$

or $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \boldsymbol{\nabla} f(\boldsymbol{\theta})$, where η is a small parameter called the learning rate and

$$\partial_i \equiv \frac{\partial}{\partial \theta_i}, \qquad \boldsymbol{\nabla} = (\partial_1, \ldots, \partial_M), \tag{30}$$

is the partial derivative with respect to the parameter θ_i and the gradient vector, respectively, using Einstein notation.

There are various ways of estimating the gradient on a quantum computer (Romero *et al.*, 2018). The most relevant of them are detailed in the Supplemental Material (824) and summarized as follows.

Finite difference.—One can compute the gradients using finite differences, i.e., $\partial_i f(\boldsymbol{\theta}) \approx [f(\boldsymbol{\theta} + \epsilon \mathbf{e}_i) - f(\boldsymbol{\theta} - \epsilon \mathbf{e}_i)]/2\epsilon$, where ϵ is a small number and $\mathbf{e}_i$ is the unit vector with 1 as its ith element and 0 otherwise. As the objective function $f(\theta)$ is obtained with limited accuracy, a good estimation of the gradient requires smaller ϵ, i.e., more samples taken from the quantum hardware.

Parameter-shift rule.—This strategy was proposed by Romero *et al.* (2018) and developed by Mitarai *et al.* (2018) and Schuld *et al.* (2019). This method computes the exact gradients and ϵ can be large (commonly $\epsilon = \pi/2$). This method assumes that the unitary to be optimized can be written as $U(\boldsymbol{\theta}) = VG(\theta_i)W$, where $G = e^{-i\theta_i g}$ is the unitary affected by the parameter θ_i, g is the generator of G, and V and W are unitaries independent of θ_i. If g has a spectrum of two eigenvalues $\pm\lambda$ only, the gradient can be calculated by measuring the observable at two shifted parameter values as follows:

$$\partial_i \langle f(\boldsymbol{\theta})\rangle = \lambda[\langle f(\boldsymbol{\theta}_+)\rangle - \langle f(\boldsymbol{\theta}_-)\rangle], \tag{31}$$

where $\boldsymbol{\theta}_\pm = \boldsymbol{\theta} \pm (\pi/4\lambda)\mathbf{e}_i$. This rule can be generalized to the case where the generator g does not satisfy the eigenspectrum condition; see the Supplemental Material (824) for details. It can also be adapted to calculate analytical gradients for fermionic generators of unitary coupled-cluster operators (Kottmann, Anand, and Aspuru-Guzik, 2021) and higher-order derivatives (Mari, Bromley, and Killoran, 2020).

Limited BFGS.—The limited Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is a quasi-Newton method that efficiently approximates the "inverse Hessian" using a limited history of positions and gradients (Liu and Nocedal, 1989; Fletcher, 2000). While effective in simulations, recent studies observed that BFGS methods do not perform well in experimental demonstrations of VQA due to the level of noise in the cost function and gradient estimates (Lavrijsen, Tudor *et al.*, 2020). Two heuristics were proposed to find quasioptimal parameters for QAOA using BFGS (Zhou *et al.*, 2020), INTERP and FOURIER, which are explained in the Supplemental Material (824). Efficient initialization of parameters has also been reported using the Trotterized quantum annealing protocol (Sack and Serbyn, 2021). These heuristic strategies can be easily extended to gradient-free optimization methods such as the Nelder-Mead algorithm.

Quantum natural gradient.—The update rule of standard gradient descent assumes that the parameter space is a flat Euclidean space. However, in general this is not the case, which can severely hamper the efficiency of gradient descent methods. In classical machine learning, a natural gradient was proposed that adapts the update rule to the non-Euclidean metric of the parameter space (Amari, 1998). Its extension, the quantum natural gradient (QNG), defines the following update rule (Stokes *et al.*, 2020):

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \eta\mathcal{F}^{-1}(\boldsymbol{\theta})\partial_i f(\boldsymbol{\theta}), \tag{32}$$

where $\mathcal{F}(\boldsymbol{\theta})$ is the Fubini-Study metric tensor or quantum Fisher information metric given by

$$\mathcal{F}_{ij} = \mathrm{Re}(\langle\partial_i\psi(\boldsymbol{\theta})|\partial_j\psi(\boldsymbol{\theta})\rangle - \langle\partial_i\psi(\boldsymbol{\theta})|\psi(\boldsymbol{\theta})\rangle\langle\psi(\boldsymbol{\theta})|\partial_j\psi(\boldsymbol{\theta})\rangle). \tag{33}$$

The superior performance of the QNG compared to other gradient methods has been reported (Yamamoto, 2019; Stokes *et al.*, 2020), and it has been shown that it can avoid becoming stuck in local minima (Wierichs, Gogolin, and Kastoryano, 2020). It can be generalized to noisy quantum circuits (Koczor and Benjamin, 2019). The QNG can be combined with adaptive learning rates $\eta(\theta_i^t)$ that change for every step of gradient descent to speed up training. For hardware-efficient PQCs, one can calculate adaptive learning rates using the quantum Fisher information metric (Haug and Kim, 2021b). While the full Fubini-Study metric tensor is difficult to estimate on quantum hardware, diagonal and block-diagonal approximations can be efficiently evaluated (Stokes *et al.*, 2020) and improved classical techniques to calculate the full tensor exist (Jones, 2020). A special type of PQC, the natural PQC, has a Euclidean quantum geometry such that the gradient is equivalent to the QNG close to a particular set of parameters (Haug and Kim, 2021a).

Quantum imaginary time evolution.—Instead of using the standard gradient descent for optimization, a variational imaginary time-evolution method was proposed by McArdle, Jones *et al.* (2019) to govern the evolution of parameters. They focused on many-body systems described by a k-local Hamiltonian and considered a PQC that encodes the state $|\psi(\tau)\rangle$ as a parametrized trial state $|\psi(\boldsymbol{\theta}(\tau))\rangle$. The evolution of $\boldsymbol{\theta}(\tau)$ with respect to all the parameters can then be obtained by solving a differential equation; see the Supplemental Material (824) for details. It was later shown by Stokes *et al.* (2020) that this method is analogous to the gradient descent via the QNG when one considers infinitesimal small step sizes.

Hessian-aided gradient descent.—A recent work (Huembeli and Dauphin, 2021) proposed computing the Hessian and its eigenvalues to help analyze the cost function landscapes of QML algorithms. Tracking the numbers of positive, negative, and zero eigenvalues provides insight on whether the optimizer is heading toward a stationary point. The Hessian can be computed by doubly applying the parameter-shift rule as shown by Mitarai and Fujii (2019) and reproduced in the Supplemental Material (824). While a deeper analysis is necessary to compare their performance, both QNG- and Hessian-based methods try to accelerate optimization by leveraging local curvature information.

Quantum analytic descent.—A method consisting of use of a classical model of the local energy landscape to estimate the gradients was proposed by Koczor and Benjamin (2020). In this hybrid approach, a quantum device is used to construct an approximate *Ansatz* landscape and the optimization toward the minima of the corresponding approximate surfaces can be carried out efficiently on a classical computer. Using this approximate *Ansatz* landscape, the full energy surface, gradient vector, and metric tensor can be expressed in terms of the *Ansatz* parameters. The analytic descent has been shown to achieve faster convergence than the QNG.

Stochastic gradient descent.—A major drawback of gradient-based methods is the high number of measurements. The stochastic gradient descent (SGD) algorithm addresses this issue by replacing the normal parameter update rule with the following modified version:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \alpha\boldsymbol{g}(\boldsymbol{\theta}^{(t)}), \tag{34}$$

where α is the learning rate and $\boldsymbol{g}$ is an unbiased estimator of the gradient of the cost function. As an estimator, one can take the measurement of the gradient with a finite number of shots (Harrow and Napp, 2019). This technique can be combined with a sampling of the parameter-shift rule terms (Sweke *et al.*, 2020) or by extending it to the doubly stochastic gradient. For the latter, the finite measurements are performed for only a subset of the expectation values of the Hamiltonian terms. This sampling can be performed in the extreme situation where only one Pauli term is evaluated at a single point in the quadrature. This is a powerful method that reduces the number of measurements drastically (Anand *et al.*, 2020). This method can be extended beyond circuits that allow the parameter-shift rule by expressing the gradient as an integral (Banchi and Crooks, 2020). To accelerate the convergence of SGD for VQA, different strategies have been proposed (Lyu, Montenegro, and Bayat, 2020) and are explained in the Supplemental Material (824).

2. Gradient-free approaches

In this section, we discuss optimization methods for VQA that do not rely on gradients measured on the quantum computer.

Evolutionary algorithms.—Evolutionary strategies (Schwefel, 1977; Rechenberg, 1978) are black-box optimization tools for high-dimensional problems that use a search distribution, from which they sample data, to estimate the gradient of the expected fitness to update the parameters in the direction of steepest ascent. More recently natural evolutionary strategies (NESs) (Wierstra *et al.*, 2014) demonstrated considerable progress in solving these high-dimensional optimization problems. They use natural gradient estimates for parameter updates instead of the standard gradients. They have been adapted for optimization of VQA (Anand, Degroote, and Aspuru-Guzik, 2020; Zhao *et al.*, 2021) and have been shown to have a performance similar to the state-of-the-art gradient-based method. Anand, Degroote, and Aspuru-Guzik (2020) showed that NESs, along with techniques like fitness shaping, local natural coordinates, adaptive sampling, and batch optimization, can be used for the optimization of deep quantum circuits.

Reinforcement learning.—Garcia-Saez and Riu (2019), Khairy *et al.* (2019), Wauters *et al.* (2020), Yao, Bukov, and Lin (2020), and Yao, Köttering *et al.* (2020) used reinforcement learning (RL) to optimize the QAOA parameters. This framework consists of a decision-making agent with policy $\pi_{\boldsymbol{\theta}}(a|s)$ parametrized by $\boldsymbol{\theta}$, which is a mapping from the state space $s \in \{S\}$ to an action space $a \in \{A\}$. In response to the action, the environment provides the agent with a reward r from the set of rewards $\{R\}$. The goal of RL is to find a policy that maximizes the expected total discounted reward. For more details, see Sec. VI.B.3. In the context of QAOA, $\{S\}$ can be the set of QAOA parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ used, a can be the value of γ and β for the next iteration, and the reward can be the finite difference in the QAOA objective function between two consecutive iterations. The policy can be parametrized by a deep neural network with the weights $\boldsymbol{\theta}$. The policy parameters $\boldsymbol{\theta}$ can be optimized using a variety of algorithms, such as Monte Carlo methods (Hammersley, 2013; Sutton and Barto, 2018), Q-learning (Watkins and Dayan, 1992), and policy gradient methods (Sutton and Barto, 2018).

Sequential minimal optimization.—In machine learning, the sequential minimal optimization method (Platt, 1998) has proven successful in optimizing the high-dimensional parameter landscape of support vector machines. The method breaks the optimization into smaller components for which the solution can be found analytically. This method has been applied to variational circuit optimization (Nakanishi, Fujii, and Todo, 2020), circuit optimization with classical acceleration (Parrish, Iosue *et al.*, 2019), and circuit optimization and learning with Rotosolve and Rotoselect (Ostaszewski, Grant, and Benedetti, 2019).

Surrogate model-based optimization.—When function evaluations are costly, it pays off not only to use the current function value to determine the next parameter value but also to use all previous evaluations to extract information about the search space. The function values in memory are used to build a surrogate model, an auxiliary function that represents the full expensive cost function based on the current information. All optimization happens on the surrogate cost landscape, so no explicit derivatives of the cost function are needed. Through the use of a fitted cost function, these methods are also expected to be more resilient to noise. Several classical surrogate models have been included in the SCIKIT-QUANT package (Lavrijsen *et al.*, 2020; Lavrijsen, Tudor *et al.*, 2020). In the bound optimization by quadratic approximation algorithm (Powell, 2009), a local quadratic model is formulated from the previous function values. It is then minimized in the trust region to obtain a new parameter value. When the evaluation at this new parameter value does not result in a lower function value, the trust region is altered and the quadratic model is optimized in this new parameter space. It was shown that this method works well when the PQC is initialized close to the optimal parameters but has more problems with shallow optimization landscapes and gets stuck in local minima (Lavrijsen, Tudor *et al.*, 2020). The stable noisy optimization by branch and fit (Huyer and Neumaier, 2008) algorithm uses a branching algorithm to explore new areas in parameter space.

3. Resource-aware optimizers

Optimization methods and strategies adopted for early demonstrations of VQAs are largely general purpose and black box, with a minimal emphasis on reducing the quantum resources used in the optimization. Therefore, they are more costly and prone to errors than their classical counterparts. Optimizers developed in recent years are tailored to additionally minimize quantities associated with the quantum cost of the optimization, such as the number of measurements or the real hardware properties. Additionally, one can use circuit compilation methods like the ones described in Sec. V.B.

ROSALIN.—While VQAs leverage low-depth circuits to execute on near-term quantum processors, a significant challenge in implementing these algorithms is the prohibitive number of measurements, or shots, required to estimate each expectation value that is used to compute the objective. To address the challenge, Arrasmith, Cincio *et al.* (2020) developed a shot-frugal optimizer called Random Operator

Sampling for Adaptive Learning with Individual Number of shots (ROSALIN) that effectively distributes fractions of a predefined number of shots to estimate each term of the Hamiltonian as well as each partial derivative. Given the expectation value of the Hamiltonian decomposed into the h_i terms as in Eq. (3), Arrasmith, Cincio *et al.* (2020) noted several strategies for allocating shots for estimating each term $\langle\cdot\rangle h_i$. While a naive strategy would allocate equal numbers of shots per term, they observed lower variance in the energies using weighted approaches in which the number of shots allocated to the ith term b_i is proportional to the corresponding Hamiltonian coefficient c_i.

SPSA.—In experimental realizations of VQAs, the optimizer is often hindered by statistical noise. Kandala *et al.* (2017) circumvented this issue by applying the simultaneous perturbation stochastic approximation (SPSA) algorithm (Spall, 1992), in which the algorithm hyperparameters are determined by experimental data on the level of statistical noise. Compared to the finite-difference gradient approximation, which requires $O(p)$ function evaluations for p parameters, SPSA requires only two evaluations, as explained in the Supplemental Material (824). The convergence of SPSA with various types of PQCs was studied by Woitzik *et al.* (2020).

III. OTHER NISQ APPROACHES

We now review some of the notable NISQ algorithms other than VQA. These algorithms do not require tuning the parameters of a PQC in an adaptive feedback manner and often exploit analog or hybrid paradigms that constitute alternatives to the digital quantum computation.

A. Quantum annealing

Quantum annealing (QA) (Finnila *et al.*, 1994; Kadowaki and Nishimori, 1998) derives its inspiration from simulated annealing (SA), a classical global optimization technique that is usually employed to solve combinatorial optimization problems. SA can be valuable in discovering global optima in optimization landscapes with many local optima. The word “annealing” comes from metallurgy and refers to heating and slow cooling. In SA, one identifies the objective function with the energy of a statistical-mechanical system. The system is assigned an artificially induced control parameter called temperature. Like annealing, SA starts at a given high temperature T, and the value of T is then decreased following a certain temperature variation function called the “annealing schedule” such that the final temperature is $T = 0$. The algorithm chooses a candidate state close to the current state randomly. If it improves the solution, it is always accepted with a probability of 1. If it does not, then the acceptance is determined based on a temperature-dependent probability function. The idea of tolerating worse solutions can be considered a virtue of the algorithm. In SA, the probability that a bad solution will be accepted slowly decreases as the solution space is explored. This relates to the notion of “slow cooling” in annealing.

In QA, one utilizes quantum-mechanical fluctuations like quantum tunneling to explore the solution space. This is related to the idea of using thermal fluctuations in SA to explore the solution space. In QA, artificial degrees of freedom of a quantum nature are introduced via noncommutative operators, which induces quantum fluctuations. The strength of these quantum fluctuations is controlled using an annealing schedule (similar to SA, where we decrease the temperature). The physical idea behind the annealing schedule in QA is to move the system from an initial Hamiltonian ground state to the ground state of the problem Hamiltonian. The concept of QA is related to the notion of quantum adiabatic evolution, which is being used for adiabatic quantum computation (Farhi *et al.*, 2000; Albash and Lidar, 2018).

We proceed to a formal discussion, starting with adiabatic quantum computation and then making the connection to QA. Adiabatic quantum computation is a model of computation based on quantum-mechanical processes operating under adiabatic conditions (Farhi *et al.*, 2000; Albash and Lidar, 2018). Before understanding adiabatic quantum computation, the concept of k-local Hamiltonians needs to be introduced.

Definition 1.—A k-local Hamiltonian is a Hermitian matrix of the form $H = \sum_{i=1}^{r} \hat{h}_i$, where each term is a Hermitian operator acting nontrivially on at most k qudits; i.e., $\hat{h}_i = h \otimes I$, where h is a Hamiltonian acting on at most k neighboring qudits and I is the identity operator.

We now consider a time-dependent Hamiltonian $H(s)$, with $s \equiv t/T \in [0,1]$ and a quantum system initialized in the ground state of $H(0)$. We assume that $H(s)$ varies smoothly as a function of s and that $H(s)$ has a unique ground state for $s \in [0,1]$. A quantum state initialized in $|\psi(t=0)\rangle$ evolves according to the following Schrödinger equation (setting $\hbar = 1$):

$$i\frac{d}{dt}|\psi(t)\rangle = H(t)|\psi(t)\rangle. \tag{35}$$

Equation (35) can be written equivalently as

$$i\frac{d}{ds}|\psi(s)\rangle = TH(s)|\psi(s)\rangle. \tag{36}$$

Assuming that $|\psi(0)\rangle$ is a ground state of $H(0)$, then in the limit $T \to \infty$ $|\psi(t)\rangle$ is a ground state of $H(1)$ obtained via evolution Eq. (35). Such an evolution will henceforth be referred to as the adiabatic evolution according to H for time T. We now define adiabatic quantum computation.

Definition 2.—An adiabatic quantum computation [adapted from Aharonov *et al.* (2008)] is specified by two k-local Hamiltonians H_0 and H_1 acting on n qudits and a map $s(t){:}\,[0,T] \to [0,1]$. The input of the computation is the ground state of H_0, which is unique and is a product state. The desired output is given by a quantum state that is ϵ close in l2-norm to the ground state of H_1. Furthermore, T is the smallest time such that the adiabatic evolution generated via $H(s) = (1-s)H_0 + sH_1$ for time T yields the desired output. The running time of the algorithm is given by $T\max_s \|H(s)\|$, where $\|\cdot\|$ denotes the spectral norm.

QA relaxes the strict requirement of adiabatic evolution, thus allowing diabatic transitions due to the finite temperature of the system, fast changes of Hamiltonian parameters, or the interaction with the noisy environment (Hauke *et al.*, 2020).

Because of diabatic transitions, QA is prone to get trapped in excited states.

QA has been investigated for problems in diverse areas including machine learning (O'Gorman *et al.*, 2015; Benedetti *et al.*, 2016, 2017; Benedetti, Realpe-Gómez, and Perdomo-Ortiz, 2018; Li *et al.*, 2018), protein folding (Perdomo *et al.*, 2008; Babbush *et al.*, 2012; Perdomo-Ortiz *et al.*, 2012; Babej *et al.*, 2018), fault diagnosis (Perdomo-Ortiz *et al.*, 2015, 2019), compressive sensing (Ayanzadeh *et al.*, 2019), finance (Marzec, 2016; Rosenberg *et al.*, 2016; Orús, Mugel, and Lizaso, 2019; Venturelli and Kondratyev, 2019; Bouland *et al.*, 2020; Cohen, Khan, and Alexander, 2020), fermionic simulation (Babbush, Love, and Aspuru-Guzik, 2014), and high-energy physics (Mott *et al.*, 2017; Das *et al.*, 2019). The protein folding problem entails calculating a protein's lowest free energy structure given its amino-acid sequence. The goal is to solve the protein folding problem by mapping it to a Hamiltonian and then using QA to identify low-energy conformations of the protein model. Perdomo-Ortiz *et al.* (2012) used five and eight qubits for the four-amino-acid sequence to encode and solve the protein folding problem for a short tetrapeptide and hexapeptide chain. QA has been one of the prominent approaches in the NISQ era in the search for quantum advantage (Perdomo-Ortiz *et al.*, 2018; Bouland *et al.*, 2020; Hauke *et al.*, 2020).

A major experimental implementation of QA is the D-Wave machine. It attempts to solve problems of a particular form called quadratic unconstrained binary optimization (QUBO) (Lucas, 2014). Optimization problems can be cast as a polynomial unconstrained binary optimization (PUBO) expressed in the form of a k-local interaction with $k \geq 3$ over binary variables $x_i \in \{0, 1\}$ (Perdomo-Ortiz *et al.*, 2019; Hauke *et al.*, 2020). QUBO is a special case of PUBO with $k = 2$. For a vector of n binary variables $\mathbf{x} \in \{0, 1\}^n$ and problem specified values of $Q \in \mathbb{R}^{n\times n}$ and $\mathbf{c} \in \mathbb{R}^n$, QUBO is defined as

$$\arg\min \mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x}. \tag{37}$$

Using the map $x_i \to (1 - \sigma_z^i)/2$, one can convert the problem in Eq. (37) to a ground state finding problem of the following diagonal n-qubit Ising Hamiltonian (up to a constant):

$$H_{\mathrm{QUBO}} = -\sum_{i,j} J_{i,j}\hat{\sigma}_z^i\hat{\sigma}_z^j - \sum_i h_i\hat{\sigma}_z^i, \tag{38}$$

where $\hat{J}_{i,j} = -Q_{i,j}/4$ and $h_i = (-c_i + \sum_j Q_{i,j})/2$.

Starting with the ground state of the base Hamiltonian $H_0 = -\sum_i \hat{\sigma}_x^i$, solving the QUBO problem on a quantum annealer corresponds to implementing the annealing schedule $A(t)$ and $B(t)$ for the Hamiltonian

$$H(t) = A(t)H_0 + B(t)H_{\mathrm{QUBO}}. \tag{39}$$

In Eq. (39) $A(0) = B(T) = 1$ and $A(T) = B(0) = 0$, where T is the computation time. Because annealing does not necessarily satisfy the constraints of adiabatic evolution, it is possible to get trapped in excited states, as mentioned earlier. However, one can run the annealing schedule multiple times and take the best answer, i.e., the one corresponding to the lowest energy. The qubits in an annealer are not necessarily all-to-all connected, necessitating additional engineering restrictions, such as the minor embedding problem (Choi, 2008, 2011; Klymko, Sullivan, and Humble, 2014).

The potential of QA has been studied extensively (Farhi, Goldstone, and Gutmann, 2002; Brady and van Dam, 2016; Denchev *et al.*, 2016; Hastings, 2020; Hauke *et al.*, 2020). The performance of D-Wave annealers has also been explored comprehensively (Shin *et al.*, 2014; Albash *et al.*, 2015; Cohen, Khan, and Alexander, 2020). In particular, an extensive study comparing the performance of quantum annealing with other quantum-inspired and classical optimization state-of-the-art strategies, and in the context of a real-world application, was conducted by Perdomo-Ortiz *et al.* (2019). For the details of QA, see Hauke *et al.* (2020) and the references therein. A review of adiabatic quantum computation was presented by Albash and Lidar (2018); see Sec. VI.B and Sec. II of the Supplemental Material (824) for a discussion regarding applications of QA in machine learning and finance.

B. Gaussian boson sampling

Boson sampling was first proposed as a candidate for quantum computational supremacy by Aaronson and Arkhipov (2011). The scenario consists of having n photons that enter an optical circuit comprising m modes. This state is then acted upon by a series of phase shifters and beam splitters. A phase shifter adds a phase $R(\theta) = e^{i\theta_j}$ with a given angle θ_j to the amplitude in mode j and acts as the identity in the other $m - 1$ modes. A beam splitter acts on two modes with a rotation $\begin{pmatrix}\cos\phi & -\sin\phi \\ \sin\phi & \cos\phi\end{pmatrix}$ for a given angle ϕ and as the identity in the other $m - 2$ modes. Finally, a measurement is made where the number of photons in each mode is found. An optical circuit with these elements is shown in Fig. 4. Each of these measurement outcomes represents a sample from the symmetric wave function that bosonic systems have. Aaronson and Arkhipov found that the existence of an efficient classical algorithm for sampling from the distribution implies the existence of a classically efficient algorithm for the calculation of the permanent of a related matrix. It is unlikely that such an algorithm exists, as that would imply the collapse of the polynomial hierarchy (see Sec. I.A) to the third order, which is believed to be unlikely (Arora and Barak, 2009).

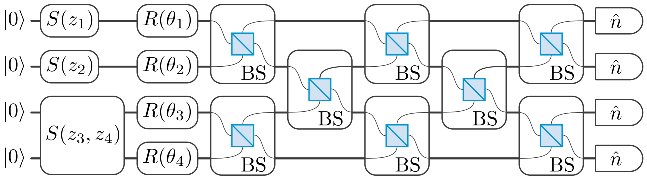

FIG. 4. Gaussian boson sampling circuit for a photonic setup. The qumodes are prepared in Gaussian states from the vacuum by squeezing operations $S(z_i)$, followed by an interferometer consisting of phase shifters $R(\theta) = e^{i\theta_j}$ and beam splitters (BSs). At the end, photon number resolving measurements are made in each mode.

GBS is a variant of boson sampling in which, instead of photon states as inputs to the optical circuit, Gaussian states are used as inputs (Hamilton *et al.*, 2017). Gaussian states are those whose Wigner quasiprobability distributions $W(q,p)$ have a Gaussian shape. A good introduction to the theory was given by Serafini (2017). Gaussian states have the advantage that they can be created deterministically (Hamilton *et al.*, 2017; Kruse *et al.*, 2019). They also provide additional degrees of freedom relative to boson sampling. Where boson sampling is equivalent to sampling from the permanent of a matrix, GBS is computationally equivalent to sampling from the Hafnian function of a matrix. Given a graph G with adjacency matrix E, the Hafnian of E is the number of perfect matchings of the graph G. A matching of a graph G is a subset of edges M such that no two edges in M have a vertex in common. A matching M is perfect if every vertex is incident to exactly one edge in M. While the permanent gives the number of perfect matchings for a bipartite graph, the Hafnian gives perfect matching for any graph. Thus, the Hafnian can be thought of as a generalization of the permanent. Using the adjacency matrix E, the relation between the Hafnian and the permanent is given by

$$\mathrm{Haf}\begin{pmatrix} 0 & E \\ E^T & 0 \end{pmatrix} = \mathrm{Per}(E). \tag{40}$$

The difficulty of simulating a noisy version of GBS has been studied (Qi *et al.*, 2020) and GBS recently became the second platform to show quantum computational supremacy (Zhong *et al.*, 2020). The latest experimental venture toward a dynamically programmable GBS nanophotonic chip was carried out by Arrazola *et al.* (2021).

1. The protocol

In GBS we consider m quantum modes (*qumodes*), which are represented by harmonic oscillators with canonically conjugate variables q and p. Gaussian states of the qumodes are those represented by a Wigner function $W(q,p)$ that has a Gaussian form. These states can be efficiently represented by the complex amplitude $\alpha = (1/\sqrt{2\hbar})(q+ip)$ and a covariance matrix $\Sigma \in \mathbb{C}^{2m\times 2m}$. A general pure Gaussian state can be generated from a vacuum with three steps: (i) single-mode squeezing, (ii) multimode linear interferometry, and (iii) single-mode displacements. In the GBS protocol, the state is then measured in the Fock basis, which is performed in practice using photon number resolving detectors. The optical circuit in Fig. 4 shows how the system is initialized in the vacuum state, followed by single-mode and multimode squeezing operators $S(z_i)$ and $S(z_i, z_j)$, respectively, and an interferometer with phase shifters $R(\theta_j)$ and beam splitters (BSs). At the end of the protocol, the photon number in each mode is measured.

For a Gaussian state with zero mean (of the Wigner function), the probability of detecting s_i photons in the ith qumode was given by Hamilton *et al.* (2017) and Kruse *et al.* (2019) as

$$P(s_1, s_2, \ldots, s_m) = \frac{1}{\det(Q)} \frac{\mathrm{Haf}(A_s)}{\sqrt{s_1! s_2! \cdots s_m!}}, \tag{41}$$

where all the matrices are defined in terms of the following covariance matrix Σ:

$$\begin{aligned} Q &= \Sigma + \mathbb{1}/2, \\ A &= X(\mathbb{1} - Q^{-1}), \\ X &= \begin{bmatrix} 0 & \mathbb{1} \\ \mathbb{1} & 0 \end{bmatrix}. \end{aligned}$$

The A_s matrix is a matrix created from A such that if $s_i = 0$ we delete the rows and colums i and $i+m$ of the matrix, and if $s_i \neq 0$ we repeat the rows and columns s_i times. This means that, by manipulating the covariance matrix Σ, we control the matrix from which we sample the Hafnian. For a pure Gaussian state, it can be shown that the A matrix is symmetric (Bromley *et al.*, 2020).

A simpler form of the experiment, where instead of counting the number of photons in each mode we detect only whether there are photons in each mode, can be used to sample from the so-called Torontonian function of a matrix (Quesada, Arrazola, and Killoran, 2018). If the probability of observing more than one photon per output mode remains low enough, this model has been shown to stay classically intractable to simulate. A more general experiment instead, where the mean of the Gaussian states is nonzero, can be used to sample from the loop Hafnian (Björklund, Gupt, and Quesada, 2019).

2. Applications

Several algorithms for applications of GBS have been investigated and were reviewed by Bromley *et al.* (2020). Here we only summarize that work. Typically, GBS algorithms are based on heuristics, and GBS devices are often used to provide a seed for starting points of classical algorithms. GBS can also be viewed as directly giving access to a statistical distribution, as in the case of point processes (Jahangiri *et al.*, 2020).

Problems in chemistry have been approached using GBS. Vibrational spectra of molecules have been computed using GBS by mapping the phononic modes of the molecule to the qumodes of the GBS device (Huh *et al.*, 2015) and, by extension, electron-transfer reactions have been studied (Jahangiri, Arrazola, and Delgado, 2020). The technique of sampling high-weight cliques has also been applied to predicting molecular docking configurations (Banchi *et al.*, 2020).

The largest number of GBS algorithms are for graph problems since the adjacency matrix of a graph is a natural fit as the symmetric A matrix. The Hafnian function computes the number of perfect matchings of a graph, so the samples from the GBS device are with high likelihood from subgraphs with high density. This is how GBS is used to identify dense subgraphs (Arrazola and Bromley, 2018), and to get good initial guesses for classical search algorithms to compute the maximum clique of a graph (Banchi *et al.*, 2020).

GBS can also be used to build succinct feature vectors, or "fingerprints," of larger graphs via coarse-graining techniques. These feature vectors can then be used as inputs to statistical methods or machine learning to classify graphs. One such problem is to measure the similarity between graphs (Schuld, Brádler *et al.*, 2020), which has applications in tasks such as checking fingerprint comparison or detecting mutations of molecules.

GBS can also be used as a type of importance sampling device to speed up algorithms requiring randomness. This is how stochastic search algorithms have been sped up by sampling from a GBS device encoding the graph to be searched, instead of sampling uniformly (Arrazola, Bromley, and Rebentrost, 2018).

Recently variational methods have been used within the GBS framework (Banchi, Quesada, and Arrazola, 2020) and applied to stochastic optimization and unsupervised learning. The method is based on varying the squeezing and interferometer parameters in the device and updating based on the measurement outcomes.

C. Analog quantum simulation

Simulating a quantum system is a hard problem for classical computers as the Hilbert space increases exponentially with the size of the system. As a solution to this long-standing problem, Feynman suggested the groundbreaking idea of harnessing those physical systems given to us by nature that are quantum mechanical. He proposed using quantum systems that are well controlled in the lab to simulate other quantum systems of interest (Feynman, 1982). This concept has spurred the field of analog quantum simulation (Trabesinger, 2012; Georgescu, Ashhab, and Nori, 2014).

The core idea differs from digital quantum simulation (Lloyd, 1996). Digital quantum simulators decompose the quantum dynamics to be simulated into a circuit of discrete gate operations that are implemented on a quantum processor. The quantum processor is a well-controlled quantum system that is engineered to be able to efficiently apply a set of specific quantum gates that are universal, i.e., a sequential application of those gates can realize arbitrary unitaries; see Sec. V.B.1.

With this universal approach, a wide range of quantum problems can be simulated to the desired accuracy with a polynomial increase in quantum resources only (Lloyd, 1996). However, current quantum processors have limited coherence time and cannot correct errors that inevitably appear during the computation, thus severely limiting the range of dynamics that can be reliably simulated. In contrast, the idea of analog quantum simulators is to map the problem Hamiltonian to be simulated $\hat{H}_{\mathrm{sys}}$ to the Hamiltonian of the quantum simulator $\hat{H}_{\mathrm{sim}}$, which can be controlled to some degree, $\hat{H}_{\mathrm{sys}} \leftrightarrow \hat{H}_{\mathrm{sim}}$. One then runs the quantum simulator and maps the results back to the problem.

The range of problems that can efficiently be mapped to the simulator is limited. However, as one uses the native quantum dynamics of the simulator, the accessible system size, coherence length, and errors are often more favorable relative to current digital quantum simulators.

1. Implementations

A wide range of implementations in various controlled quantum systems has been achieved, ranging from solid-state superconducting circuits (Houck, Türeci, and Koch, 2012) to quantum dot arrays (Hensgens *et al.*, 2017) to nitrogen-vacancy centers (Yao *et al.*, 2012) to atomic and molecular physics–based platforms such as trapped ions (Blatt and Roos, 2012), interacting photons (Chang, Vuletić, and Lukin, 2014; Hartmann, 2016), Rydberg atoms (Adams, Pritchard, and Shaffer, 2020), and cold atoms (Bloch, Dalibard, and Nascimbene, 2012; Gross and Bloch, 2017; Amico *et al.*, 2021).

Concepts of analog quantum simulation have been used within VQAs as well, such as problem-inspired *Ansätze* (see Sec. II.B.1) or protocols inspired by quantum control (Yang *et al.*, 2017; Meitei *et al.*, 2020). Experimental results for a quantum many-body problem beyond current classical computational capabilities have been reported for 2D systems (Choi *et al.*, 2016).

2. Programmable quantum simulators

An analog quantum system, such as a superconducting circuit, can be adapted to simulate arbitrary dynamics (Bastidas *et al.*, 2020). The idea is to drive the parameters of the Hamiltonian $H(t)$ that describes the analog quantum simulator in time t. This can be achieved by adjusting the physical parameters of the quantum simulator in time. The driving protocol is engineered via machine learning methods (Haug *et al.*, 2021) such that the effective dynamics of the driven system over a time T corresponds to the evolution of a problem Hamiltonian that one wants to simulate. The effective dynamics that are generated can realize long-range interactions as well as complicated many-body terms, which are not natively supported by the quantum simulator and are often hard to simulate on digital quantum simulators. By periodically driving the analog quantum simulator with the aforementioned driving protocol, various problem Hamiltonians can be simulated (Oka and Kitamura, 2019). One can realize complicated many-body dynamics or chemistry problems and solve combinatorial tasks such as SAT-3. Trapped-ion-based analog quantum simulators were recently used for the implementation of the quantum approximate optimization algorithm (Pagano *et al.*, 2020).

Highly controllable analog quantum simulators have also been proposed for engineering quantum chemistry Hamiltonians by combining different cold atom species embedded within cavity modes, which mediate long-range interactions required to simulate Coulomb repulsion. Optical fields can be used to modify the potential and interaction parameters to simulate large-scale chemistry problems (Argüello-Luengo *et al.*, 2019), as well as a quantum spin model with tunable interactions for system sizes ranging from 64 to 256 qubits (Ebadi *et al.*, 2020). For ion traps, a programmable quantum simulator can be designed by light fields that are applied to manipulate the internal degrees of freedom as well as the interaction between different ions. This allows one to simulate various types of spin Hamiltonians with a high degree of control over the parameters (Monroe *et al.*, 2019).

D. Digital-analog quantum simulation and computation

As opposed to analog simulators that are limited by the Hamiltonians they can simulate (Goldman and Dalibard, 2014; Kyriienko and Sørensen, 2018), digital quantum simulators can simulate any system's Hamiltonian but sometimes with costly quantum resources. To benefit from a combination of the two approaches, the digital-analog method to quantum computation (Dodd *et al.*, 2002; Parra-Rodriguez *et al.*, 2020) and simulation (Mezzacapo *et al.*, 2014; Yung *et al.*, 2014) have been proposed. These schemes combine the application of digital single-qubit gates with the underlying analog Hamiltonian of the quantum processor. This approach allows for universal simulation of quantum dynamics while two-qubit gates for an analog Hamiltonian are replaced and has been argued to be more resilient against certain types of noise than digital quantum computing (Martin *et al.*, 2020; Parra-Rodriguez *et al.*, 2020; García-Molina, Martin, and Sanz, 2021).

Digital-analog quantum simulation has been proposed to simulate the Rabi model (Mezzacapo *et al.*, 2014), the Dicke model (Mezzacapo *et al.*, 2014; Lamata, 2017), and fermionic systems (García-Álvarez *et al.*, 2015; Céleri *et al.*, 2021). Digital-analog quantum simulation was reviewed by Lamata, Parra-Rodriguez *et al.* (2018b); digital-analog quantum computing is more recent. The implementation of digital-analog quantum computing has been proposed for superconducting platforms (Gonzalez-Raya *et al.*, 2021; Yu *et al.*, 2021). Gonzalez-Raya *et al.* (2021) employed cross-resonance gate interaction between two superconducting qubits to implement digital-analog quantum computation. Thus far the computing framework has been used to simulate Ising models (Parra-Rodriguez *et al.*, 2020), where the analog blocks can be used to enhance the effective connectivity of the qubits to simulate graphs that have different connectivities from the native connectivity of the quantum device (Galicia *et al.*, 2020). The analog blocks have also been applied to reduce the operation count required to perform the quantum Fourier transform (Martin *et al.*, 2020).

The digital-analog approach has also been combined with VQAs (see Sec. II), resulting in a digital-analog QAOA algorithm, where the two-qubit gates were replaced by analog blocks (Headley *et al.*, 2020). This also has two versions: (i) where a layer of entangling gates is replaced by an analog block, and (ii) where an analog block is applied continuously with single-qubit operations overlaid.

E. Iterative quantum assisted eigensolver

Almost all of the VQAs update a PQC's parameters in a feedback loop. However, there are alternative algorithms that can circumvent this approach, with the *Ansatz* given by (McClean *et al.*, 2017; Huang, Bharti, and Rebentrost, 2019; Bharti and Haug, 2021a)

$$|\phi(\boldsymbol{\alpha}(t),\boldsymbol{\theta})\rangle = \sum_{i=0}^{m-1} \alpha_i(t)|\psi_i(\boldsymbol{\theta}_i)\rangle, \tag{42}$$

where $\alpha_i \in \mathbb{C}$ and $\theta_i \in \mathbb{R}^{k_i}$ for non-negative integers k_i. This *Ansatz* is a linear combination of quantum states, where the α_i parameters are stored on a classical device. In the special case in which $m = 1$, it corresponds to the usual PQC, whereas for $m > 1$ this *Ansatz* subsumes it. This *Ansatz* has been used for finding the ground state of Hamiltonians (Bharti, 2020; Bharti and Haug, 2021a), the excited state (Parrish, Hohenstein *et al.*, 2019; Parrish and McMahon, 2019; Huggins *et al.*, 2020; Stair, Huang, and Evangelista, 2020), the simulation of quantum dynamics (Haug and Bharti, 2020; Bharti and Haug, 2021b), error mitigation (McClean *et al.*, 2017), nonlinear dynamics (Haug and Bharti, 2020; Bharti and Haug, 2021b), linear systems (Huang, Bharti, and Rebentrost, 2019), and semidefinite programming (Bharti *et al.*, 2021). If one keeps the parameters of the PQC θ_i fixed and varies only the α_i, the algorithm can be considered a borderline non-VQA algorithm. Updates of θ_i parameters have been shown to cause trainability issues in VQAs (see Sec. IV.A), and thus by fixing θ_i one can by construction circumvent these issues. We present here the iterative quantum assisted eigensolver (IQAE) algorithm as an illustration, and in Sec. VI the quantum assisted simulator for closed systems (see Sec. VI.A.5), open systems (see Sec. VI.A.7), and Gibbs state preparation (see Sec. VI.A.9).

The IQAE algorithm provides an approximation to the ground state of a Hamiltonian H. Without loss of generality, the N-qubit Hamiltonian H is assumed to be a linear combination of unitaries

$$H = \sum_{i=1}^{m} \beta_i U_i. \tag{43}$$

In Eq. (43) $\beta_i \in \mathbb{C}$ and $U_i \in \mathrm{SU}(2^N)$ for $i \in \{1, 2, \ldots, m\}$. The unitaries U_i act on at most $\mathcal{O}(\mathrm{poly}(\log N))$ qubits. This condition can be relaxed if the unitaries are Pauli strings; see Sec. II.A.1. The *Ansatz* state is taken as a linear combination of "cumulative K-moment states" $\mathbb{CS}_K$, which is generated using some efficiently preparable quantum states and the unitaries defining the Hamiltonian in Eq. (43). For pedagogical reasons, we present the definition of K-moment states and cumulative K-moment states.

Definition 3 [adapted from Bharti and Haug (2021a)].— For a given positive integer K, a set of unitaries $\mathbb{U} \equiv \{U_j\}_{j=1}^m$, and a quantum state $|\psi\rangle$, K-moment states are the set of quantum states of the form $\{U_{j_K} \cdots U_{j_2} U_{j_1} |\psi\rangle\}_j$ for $U_{j_l} \in \mathbb{U}$. We denote the aforementioned set by $\mathbb{S}_K$. We define the singleton set $\{|\psi\rangle\}$ as the zero-moment state (denoted by $\mathbb{S}_0$). Finally, we define the cumulative K-moment states $\mathbb{CS}_K$ as $\mathbb{CS}_K \equiv \cup_{i=0}^{K} \mathbb{S}_i$.

As an instructive example, we use the set of one-moment states $\{U_j|\psi\rangle\}_{j=1}^m$, where the unitaries $\{U_j\}_{j=1}^m$ make up the Hamiltonian H. The set of cumulative one-moment states is $\mathbb{CS}_1 = \{|\psi\rangle\} \cup \{U_j|\psi\rangle\}_{j=1}^m$, and the set of cumulative K-moment states is $\mathbb{CS}_K = \{|\psi\rangle\} \cup \{U_{j_1}|\psi\rangle\}_{j_1=1}^m \cup \cdots \cup \{U_{j_K} \cdots U_{j_1}|\psi\rangle\}_{j_1=1,\ldots,j_K=1}^r$.

The *Ansatz* is then given by $|\xi(\alpha)\rangle^{(K)} = \sum_{|\chi_j\rangle \in \mathbb{CS}_K} \alpha_j |\chi_j\rangle$. The ground state problem reduces to the following optimization program:

$$\min_{\alpha} \alpha^\dagger \mathcal{D}^{(K)} \alpha,$$
$$\text{subject to } \alpha^\dagger \mathcal{E}^{(K)} \alpha = 1. \tag{44}$$

In Eq. (44) the overlap matrices $\mathcal{D}^{(K)}$ and $\mathcal{E}^{(K)}$ are given by $\mathcal{D}^{(K)}_{nm} = \sum_i \beta_i \langle \chi_n | U_i | \chi_m \rangle$ and $\mathcal{E}^{(K)}_{nm} = \langle \chi_n | \chi_m \rangle$. These overlap matrices can be computed on a quantum computer without the requirement of any complicated measurements involving multiqubit controlled unitaries. For example, for a Hamiltonian composed of Pauli strings the product of Pauli strings is a Pauli string up to a phase factor ±1 or $\pm\imath$. Thus, the overlap matrices are simply expectation values $\langle\psi|\hat{P}|\psi\rangle$ of a certain Pauli string $\hat{P}$, which can be easily measured; see Sec. II.C. The optimization program (44) is a quadratically constrained quadratic program (QCQP) with a single equality constraint. The algorithm proceeds in three serial and disjoint steps.

(1) Select *Ansatz*, which can be done on paper.
(2) Estimate overlap matrices on a quantum computer, which can be done efficiently in a parallel fashion.
(3) Postprocessing on a classical computer to solve the QCQP based on the overlap matrices from step 2.

As a major speedup compared to standard VQA, there is no feedback loop between classical and quantum computers such that the calculations can be easily parallelized. The *Ansatz* can be improved by changing K to $K+1$. The *Ansatz* construction is systematic and there is no trainability issue such as the barren plateau problem; see Sec. IV.A. For the QCQP, there are conditions that tell whether a local minimum is a global minimum as a stopping criterion for the classical solver. Moreover, the Lagrangian relaxation of the program (44) is a semidefinite program and is efficiently solvable.

IV. THEORETICAL CHALLENGES

A. Barren plateaus

It was recently shown that the expectation value of the gradient of the objective function corresponding to randomly initialized PQCs (RPQCs) decays exponentially to zero as a function of the number of qubits (McClean *et al.*, 2018). The mathematical basis of this result hinges on the fact that the PQC from Eq. (22) becomes a unitary 2-design as the circuit depth increases polynomially with the circuit width, i.e., the number of qubits. The notion of a unitary 2-design has been used extensively in the recent proofs of barren plateau in RPQCs, which necessitates a small discussion about their mathematical structure.

Using the notion of 2-design, the appearance of barren plateaus in the training landscape has been established for various kinds of *Ansätze*. Barren plateaus can be thought of as a consequence of the Hilbert-space dimension increasing exponentially with the number of qubits and of the variational circuit being a 2-design for randomly initialized parameters. Consequently, the strategies proposed to tackle this problem focus on reducing the dimension of this unitary or breaking the randomness properties related to the 2-designs. Another way to think of the origin of the barren plateau issue could be the problem-agnostic nature of the *Ansatz* when faced with an exponentially large parameter space. Thus, one could attempt to devise *Ansätze* as well as the optimization methodology in a problem-aware manner by using physically inspired or problem-specific *Ansätze* such as the ones presented in Sec. II.B or those proposed in Sec. III.E.

Besides the exponential parameter space that induces barren plateaus, other physical phenomena can also generate them. In particular, the noise and decoherence present in the quantum computing experiments can also generate this problem in VQAs (Wang, Fontana *et al.*, 2020). Entanglement-induced barren plateaus have also been reported recently (Marrero, Kieferová, and Wiebe, 2020).

While certain *Ansätze* can be assumed or proven to form approximate 2-designs, such proofs are challenging for general *Ansätze*. To numerically verify the presence of barren plateaus, past studies often considered computing the gradients and variances of a local observable using a particular *Ansatz* over increasing system sizes (McClean *et al.*, 2018; Skolik *et al.*, 2020).

Another attempt to avoid a barren plateau is to initialize the variational circuit with a particular state choice. Intuitively, the algorithm will start in a particular region of the Hilbert space, thereby allowing the optimization subroutine to potentially find the minima in a nearby region. This strategy includes all physically inspired methods mentioned in Sec. II.B. The use of clever encodings for the algorithm parameters can also be understood as an initialization strategy (Cervera-Lierta, Kottmann, and Aspuru-Guzik, 2021); see Sec. VI.B. Classical algorithms such as neural networks can also be used to learn the proper circuit encodings (Verdon, Broughton *et al.*, 2019; Sauvage *et al.*, 2021; Wilson *et al.*, 2021).

A good choice for the initial state is often not enough to reduce the size of the Hilbert space. Although expressive circuit *Ansätze* are usually a requirement for the success of a VQA (see Sec. IV.B for more details), such *Ansätze* can expand the parameter space that the optimizer has to explore. Several works propose circuit structures that reduce the space by introducing correlations between the variational parameters of the circuit (Volkoff and Coles, 2021), blockwise initialization of parameters (Grant *et al.*, 2019), or exploring particular *Ansatz* structures (Sharma *et al.*, 2020).

The mentioned works require a circuit design that is not necessarily hardware efficient. Other ideas focus on the classical parts of the VQA instead of the quantum circuit designs. One example involves using local instead of global cost functions for the optimization. It has been shown (Cerezo *et al.*, 2021) that barren plateaus also emerge in shallow depth circuits and that the use of local cost functions reduces the exponential decay tendency to a polynomial one. The optimization strategy may also reduce the effect of the vanishing gradients, such as by training the circuit layer by layer (Lyu, Montenegro, and Bayat, 2020; Skolik *et al.*, 2020) or by measuring low-depth gradients (Harrow and Napp, 2019). Certain variational quantum algorithms for quantum simulation can be free of barren plateaus when at every training step the state to be learned is close to the state of the circuit (Haug and Kim, 2021b).

Barren plateaus are a roadblock in trainability, and hence any PQC *Ansatz* that suffers from this phenomenon will likely fail to properly train the parameters in its search for the near-optimal (or optimal) solution. As shown by Arrasmith, Cerezo *et al.* (2020), even the family of gradient-free approaches that perform a local search, therefore mimicking gradient-based optimization, appear to face similar challenges. However, one can circumvent this issue by using hybrid quantum states of the form of Eq. (42) or the hybrid density matrices introduced by Haug and Bharti (2020). The idea is to write the overall *Ansatz* as a classical combination of quantum states, i.e., $|\phi(\boldsymbol{\alpha}(t),\boldsymbol{\theta})\rangle = \sum_{i=0}^{m-1}\alpha_i(t)|\psi_i(\boldsymbol{\theta}_i)\rangle$. Tuning the $\boldsymbol{\theta}_i$ can often lead to barren plateaus. One can avoid such issues by fixing $\boldsymbol{\theta}_i$ by harnessing the structure of the problem to find the basis states of the *Ansatz*, i.e., $\{|\psi_i(\boldsymbol{\theta}_i)\rangle\}$; see Sec. III.E for more details. Quantum convolutional neural networks also do not exhibit barren plateaus (Pesah *et al.*, 2021).

B. Expressibility of variational *Ansätze*

A cornerstone of the success of VQA is choosing the proper *Ansatz* for the problem. In addition to trainability, i.e., how well the *Ansatz* can be optimized, another major quality is expressibility. This concerns whether a given PQC is able to generate a rich class of quantum states. The number of PQC layers, parameters, or entangling gates required to achieve a given accuracy is also linked to the expressibility of the circuit.

Expressibility.—Sampling states from a PQC $|\psi_{\boldsymbol{\theta}}\rangle$ for randomly chosen $\boldsymbol{\theta}$ generates a distribution of states. Expressibility is defined as follows as the deviation of this distribution from the Haar measure, which samples uniformly from the full Hilbert space:

$$A^{(t)} = \left\| \int_{\text{Haar}} (|\psi\rangle\langle\psi|)^{\otimes t} d\psi - \int_{\boldsymbol{\theta}} (|\psi_{\boldsymbol{\theta}}\rangle(\langle\psi_{\boldsymbol{\theta}}|)^{\otimes t} d\psi_{\boldsymbol{\theta}} \right\|_{\text{HS}}^2, \quad (45)$$

where $\int_{\text{Haar}} d\psi$ denotes the integration over a state $|\psi\rangle$ distributed according to the Haar measure and $\|A\|_{\text{HS}}^2 = \text{Tr}(A^\dagger A)$ indicates the Hilbert-Schmidt norm. An *Ansatz* circuit U with small $A_U^{(t)}$ is more expressive, with $A_U^{(t)} = 0$ corresponding to being maximally expressive, as it generates quantum states with a distribution closer to the Haar measure. The PQC samples uniformly from the full Hilbert space and thus is able to approximate any possible state. This is especially important in the case where one wants to train the PQC to represent a particular quantum state while having little prior information about the state. A highly expressive PQC is more likely to be able to represent the target state.

Entangling capability.—This measure denotes the power of a PQC to create entangled states and can be used as another quantifier of the expressiveness of an *Ansatz*. Sim, Johnson, and Aspuru-Guzik (2019) proposed the Meyer-Wallach Q measure (Meyer and Wallach, 2002) to estimate the number and types of entangled states a particular PQC can generate. One defines a linear mapping $\iota_j(e)$ that acts on the computational basis $\iota_j(b)|b_1\cdots b_n\rangle = \delta_{bb_j}|b_1\cdots\tilde{b}_j\cdots b_n\rangle$, where $b_j \in \{0,1\}$ and $\tilde{b}_j$ denotes absence of the jth qubit. The entanglement measure Q is then defined as

$$Q(|\psi\rangle) \equiv \frac{4}{n}\sum_{j=1}^{n} D(\iota_j(0)|\psi\rangle, \iota_j(1)|\psi\rangle), \quad (46)$$

where D is the generalized distance defined by the coefficients of two states $|u\rangle = \sum u_i|e_i\rangle$ and $|v\rangle = \sum v_i|e_i\rangle$,

$$D(|u\rangle, |v\rangle) = \frac{1}{2}\sum_{i,j}|u_i v_j - u_j v_i|^2. \quad (47)$$

Equation (47) can be rewritten as follows as the average of the purity of each qubit (Brennen, 2003):

$$Q(|\psi\rangle) = 2\left(1 - \frac{1}{n}\sum_{k=1}^{n}\text{Tr}[\rho_k^2]\right), \quad (48)$$

where ρ_k is the density matrix of the kth qubit. Thus, $Q(|\psi\rangle)$ is an entanglement monotone (Scott, 2004) and can be interpreted as the average of the entanglement of each qubit with the rest of the system. Only if the state is a product state do we find that $Q = 0$, whereas $Q = 1$ is reached for certain entangled states such as the Greenberger-Horne-Zeilinger (GHZ) state. The entangling capability of a PQC is then defined as follows as the average Q of states randomly sampled from the circuit:

$$\text{Ent} = \frac{1}{|S|}\sum_{\boldsymbol{\theta}_i \in S} Q(|\psi_{\boldsymbol{\theta}_i}\rangle), \quad (49)$$

where $S = \{\boldsymbol{\theta}_i\}_i$ is the set of sampled circuit parameters.

Parameter dimension.—The parameter dimension D_{C} is the number of independent parameters of the quantum state that is generated by the PQC (Haug, Bharti, and Kim, 2021). From this measure, one can calculate the redundancy of a PQC, i.e., the fraction of parameters that can be removed without loss of expressive power. A further local measure of expressibility is the effective quantum dimension $G_{\text{C}}(\boldsymbol{\theta})$, which can be used to calculate the expressive power of initialization strategies for the PQC. Under a small variation of the PQC parameter $\boldsymbol{\theta}$, it measures how many independent directions in the parameter space exist for the quantum state. Both measures can be calculated as the number of nonzero eigenvalues of the Fubini-Study metric tensor defined in Eq. (33).

Sim, Johnson, and Aspuru-Guzik (2019), Nakaji and Yamamoto (2020a), and Haug, Bharti, and Kim (2021) investigated a wide class of circuits with the aforementioned expressibility measures. It has been found that certain types of *Ansätze* are more expressive: e.g., layered PQCs consisting of CNOT or $\sqrt{\text{iSWAP}}$ gates are more expressive than CZ ones . There is a trade-off between an *Ansatz* being expressive and trainable. Making an *Ansatz* more expressive most likely will result in reducing the gradient of the objective function. Holmes *et al.* (2021) suggested several strategies for reducing expressibility and improving trainability, including correlating parameters or restricting rotation angles of parametrized gates. Interpolating the PQC parameters between fixed and random angles has been proposed as another method (Haug, Bharti, and Kim, 2021). Expressibility of PQCs has been further explored using classical Fisher information (Abbas *et al.*, 2021) and memory capacity (Wright and McMahon, 2019).

It has been shown that an alternating layered *Ansatz* (see Sec. II.B.2) is relatively expressive and does not exhibit barren plateaus in certain regimes (Nakaji and Yamamoto, 2020a). In VQE algorithms, there is a trade-off between the number of layers in this *Ansatz* and the correlation length of critical

Hamiltonians. However, in the critical phase, the number of layers must exceed a certain threshold dictated by the system size to show an exponential improvement. The circuit depth unravels an effective correlation length that can be used as an estimation of the number of free parameters in the *Ansatz* (Bravo-Prieto *et al.*, 2020).

C. Reachability

Reachability discusses the question of whether a given PQC $|\Psi(\boldsymbol{\theta})\rangle$ with parameters $\boldsymbol{\theta}$ is capable of representing a quantum state that minimizes a certain objective function. This can be quantified by the reachability deficit over finding the minimum of an objective function $\hat{O}$ (Akshay, Philathong, Morales, and Biamonte, 2020) as

$$f_{\mathrm{R}} = \min_{\psi\in\mathcal{H}}\langle\psi|O|\psi\rangle - \min_{\boldsymbol{\theta}}\langle\Psi(\boldsymbol{\theta})|O|\Psi(\boldsymbol{\theta})\rangle, \tag{50}$$

where the first term on the right-hand side is the minimum over all states $|\psi\rangle$ of the Hilbert space, whereas the second term is the minimum over all states that can be represented by the PQC. The reachability deficit is equal to or greater than zero ($f_{\mathrm{R}} \geq 0$), with $f_{\mathrm{R}} = 0$ when the PQC can generate a state $|\Psi(\boldsymbol{\theta}^*)\rangle$, where $\boldsymbol{\theta}^*$ are the parameters that minimize the objective function.

Reachability has been studied in depth for QAOA. Although QAOA has been shown to exhibit quantum computational universality (Lloyd, 2018; Morales, Biamonte, and Zimborás, 2019), which implies that any unitary operator is reachable under the QAOA *Ansatz*, this statement does not hold true for finite fixed depths p. In fact, it was shown that QAOA exhibits reachability deficits for the MAX-2-SAT and MAX-3-SAT problems, where the optimal value of the objective function cannot be found using a fixed circuit depth p beyond a critical clause density (defined as the ratio between the number of clauses and the number of variables in the problem) (Akshay, Philathong, Morales, and Biamonte, 2020). In other words, for problems with a certain clause density, there is a critical depth p^* for which the optimal solution can be found (up to a threshold) only if $p \geq p^*$. As p^* grows with the clause density, this limits the performance of QAOA for problem instances with high clause density.

Similar reachability deficits have also been found in the variational Grover search problem (Akshay, Philathong, Morales, and Biamonte, 2020). Moreover, by reanalyzing the experimental data from Google's Sycamore quantum processor on the application of QAOA to various graph optimization problems (Harrigan *et al.*, 2021), Akshay, Philathong, Zacharov, and Biamont (2020) also discovered reachability deficits in this case, where the graph density (defined as the ratio between the number of graph edges and the number of graph nodes) replaces the clause density as the order parameter.

Note that the reachability deficits are distinct from the barren plateau problem, where the gradients of the objective function concentrate to zero for many choices of initial variational parameters, thus slowing down the optimization process. On the other hand, the reachability deficit for $p < p^*$ is independent of the initial parameters.

D. Theoretical guarantees of QAOA

QAOA has several key analytical results that have contributed to its considerable interest in recent years. The quantum advantage of QAOA was studied by Farhi and Harrow (2016), who showed that the efficient sampling of the output distribution of QAOA, even for the lowest depth case of $p = 1$, implies the collapse of the polynomial hierarchy; see Sec. I.A. Following the conjecture from complexity theory that the polynomial hierarchy does not collapse, this result propels QAOA as a possible candidate for establishing a quantum advantage in a sampling task. In particular, it has been shown that, for $p = 1$, 420 qubits would suffice to demonstrate quantum advantage (Dalzell *et al.*, 2020).

QAOA has yet to demonstrate any speedup over classical algorithms for practical applications. Understanding the potential of QAOA relative to classical algorithms is an active research topic (Barak *et al.*, 2015; Farhi and Harrow, 2016; Wecker, Hastings, and Troyer, 2016; Yang *et al.*, 2017; Bravyi *et al.*, 2019; Hastings, 2019). For specific instances of the max-cut problem, QAOA for $p = 1$ was shown to perform as well as or worse than classical algorithms (Bravyi *et al.*, 2019; Hastings, 2019). For more discussion of QAOA for the max-cut problem, see Sec. VI.C.1. For QAOA of depth p, the measurement outcomes of a qubit depend on the p neighborhood of that qubit. Thus, if p is too small, it does not "see" the entire graph (Farhi, Gamarnik, and Gutmann, 2020a, 2020b). For large p, the QAOAs can see the entire graph with no known indications regarding the performance limitations.

For the case where the problem Hamiltonian H_P takes the form

$$H_P = \sum_i \omega_A\hat{\sigma}_z^{2i} + \omega_B\hat{\sigma}_z^{2i+1} + \gamma_{AB}\hat{\sigma}_z^{2i}\hat{\sigma}_z^{2i+1} + \gamma_{BA}\hat{\sigma}_z^{2i+1}\hat{\sigma}_z^{2i+2}, \tag{51}$$

where $\omega_{A(B)}$ are the coefficients for the even (odd) sites and $\gamma_{AB(BA)}$ are the interaction strengths between first (second) neighbor spins. Taking H_M as defined in Eq. (19) for a 1D lattice, Lloyd (2018) showed that QAOA can be used to implement universal quantum computation. This result was proven and generalized in a later work (Morales, Biamonte, and Zimborás, 2019) to include a larger class of problems and mixing Hamiltonians that can provide computational universality for QAOA.

By connecting VQA with optimal control theory, Pontryagin's minimum principle of optimal control shows that the bang-bang protocol in which the evolution switches abruptly between two Hamiltonians is optimal for a fixed total time T (Yang *et al.*, 2017). Since QAOA can be regarded as a bang-bang *Ansatz* by switching between unitary evolution under H_P and H_M, respectively, this suggests the optimality of QAOA as a VQA. However, recent works have challenged this claim. By generalizing the argument of Yang *et al.* (2017), it has been shown that the optimal protocol actually possesses the "bang-anneal-bang" structure (Brady *et al.*, 2020). Such protocols begin and end with a bang, with regions of smoothly varying control function akin to quantum annealing in between. It was also shown that when the total time T is large, a bang-bang QAOA suffers from the proliferation of local minima in the control parameters, rendering it difficult to find optimal or near-optimal QAOA parameters.

V. PROGRAMMING AND MAXIMIZING NISQ UTILITY

Current NISQ devices have a limited number of qubits (~50–100) available. In addition, owing to their noisy nature and short coherence time, one can perform only a restricted number of gate operations. To make maximal use of the currently available quantum resources, there are two approaches from the operational point of view: the bottom-up and the top-down approach. In the bottom-up approach one has full control over the design of the quantum computing platform to keep improving the performance quality by such means as gate fidelity and coherence time within the given hardware constraints. The top-down approach implies that one does not get involved in hardware design and simply makes use of what has already been made or fabricated in the experimental labs. In this section, we focus on the latter approach, which extends the utility of current and near-term quantum devices from an algorithmic perspective. Finally, we present a summary of software tools to control, program, and maximize the utility of NISQ algorithms.

A. Quantum error mitigation (QEM)

Sensitivity to errors and noise are the two most prominent roadblocks facing scalable universal quantum computers. Fault-tolerant quantum computing can be attained by encoding non-Abelian anyons into topological materials (Kitaev, 2003) or applying QEC codes (Raussendorf and Harrington, 2007). While the former is still in its infancy, the latter mandates physical resources exceeding our current experimental capabilities. In the NISQ era of running hybrid quantum-classical algorithms, it is desirable to use all the restricted and available qubits as logical qubits without applying QEC techniques. As we discuss throughout this review, the hybrid quantum-classical algorithms rely on computations of the expectation value of physical observables using quantum processors. QEM techniques discussed here do not need extra qubits in general and can suppress errors via classical postprocessing techniques and multiple runs of quantum circuits. In QEM, we aim not to recover the ideal quantum output state $\hat{\rho}^{(0)}$ but to estimate the ideal expectation value $E[\mu^{(0)}] = \langle \hat{A}^{(0)} \rangle = \mathrm{Tr}(\hat{\rho}^{(0)}\hat{A})$ of an observable $\hat{A}$ (Li and Benjamin, 2017; Temme, Bravyi, and Gambetta, 2017; Kandala *et al.*, 2019). This approach can surpass the break-even point, where the effective gates are superior to their physical building blocks with an affordable cost in quantum resources for near-term quantum hardware (Zhang, Lu *et al.*, 2020). Here μ is the outcome of measurements and we use the superscript (0) to denote an ideal noise-free realization of a state, operation, or observable quantity. Recently it was also shown how to achieve stochastic error mitigation for a continuous-time evolution (Sun *et al.*, 2020). For a comprehensive treatment of quantum error mitigation, see Endo, Cai *et al.*, 2020).

1. Zero-noise extrapolation

Li and Benjamin (2017) and Temme, Bravyi, and Gambetta (2017) independently proposed the Richardson extrapolation QEM or zero-noise extrapolation (ZNE) technique. Here a quantum program operates at various effective noise levels of a quantum processor, where the output is then extrapolated to gain an estimated expectation value without noise.

Formally, a quantum circuit in the presence of noise can be modeled as an open quantum system (Breuer and Petruccione, 2002) using the Gorini-Kossakowski-Sudarshan-Lindblad equation or, in short, the Lindblad master equation

$$\frac{d}{dt}\hat{\rho}(t) = -i[\hat{K}(t), \hat{\rho}(t)] + \hat{\hat{\mathcal{L}}}[\hat{\rho}(t)], \tag{52}$$

where we set $\hbar = 1$, $\hat{K}(t)$ acts as time-dependent driving Hamiltonian, and $\hat{\hat{\mathcal{L}}}[\cdot] = \sum_k \Gamma_k(\hat{O}_k[\cdot]\hat{O}_k^\dagger - \frac{1}{2}\{\hat{O}_k\hat{O}_k^\dagger, [\cdot]\})$ is a superoperator. Equation (52) describes the Markovian dynamics for $\Gamma_k \geq 0$. Whenever the loss rate Γ_k becomes negative (Rivas, Huelga, and Plenio, 2010; Fleming and Hu, 2012), Eq. (52) can also describe non-Markovian dynamics (Tan, Kyaw, and Yeo, 2010; Bastidas *et al.*, 2018; Kyaw *et al.*, 2020). To ensure complete positivity, we require $\int_0^t \Gamma(t')dt' > 0 \;\forall\; t$. In general, Γ_k are fixed by the nature of the noise affecting the quantum system. For the ZNE, we parametrize Γ_k with a dimensionless scalar λ, i.e., $\Gamma_k \to \lambda\Gamma_k$. When $\lambda = 0$, there is no noise and the loss term $\hat{\hat{\mathcal{L}}}[\hat{\rho}(t)]$ in Eq. (52) is zero, thus resulting in pure unitary dynamics. When $\lambda = 1$, the noise matches the actual quantum device. In summary, ZNE involves two steps.

(1) Noise scaling: we measure several instances of $E[\mu^{(\lambda_j)}]$ for $\lambda_j \geq 1$.

(2) Extrapolation: using the previous measurements, we estimate $E[\mu^{(0)}]$ by extrapolating to $\lambda = 0$.

Noise scaling can be accomplished in three ways. First, Temme, Bravyi, and Gambetta (2017) proposed using a *time-scaling approach* to take $\lambda > 1$, which means that the time-dependent driving Hamiltonian $\hat{K}(t)$ is now rescaled by $(1/\lambda)\hat{K}(t/\lambda)$. This approach is possible only if the user has full control over the quantum processor, as the control pulses for each quantum gate have to be recalibrated and applied for a longer duration. Second, one can apply a technique called *circuit folding* (Giurgica-Tiron *et al.*, 2020). Suppose that a quantum circuit is composed of d unitary layers such that $U = L_d \cdots L_2 L_1$, where d refers to the circuit depth and each L_j represents either a single layer of gate operations or simply a single quantum gate. The circuit folding is then achieved by

$$U \to U(U^\dagger U)^n, \tag{53}$$

where n is a positive integer. Since $U^\dagger U$ is an identity, this action has no effect on an ideal circuit. However, in a noisy circuit U is imperfect and the $1 + 2n$ circuit operations increase the noise level. Third, instead of folding the entire circuit, one can use the *gate folding* technique where individual gates are folded (Giurgica-Tiron *et al.*, 2020) as follows:

$$L_j \to L_j(L_j^\dagger L_j)^n. \tag{54}$$

The second and third techniques do not require users to have full control of the pulses applied to the quantum computer, and thus these methods are more suitable when only a limited control of the quantum computer is possible.

The *extrapolation step* of the ZNE method can be considered a regression problem if we choose to consider a generic model for calculating the expectation value $E_{\rm model}[\mu^{(\lambda;\Upsilon)}]$, where the meaning of *model* becomes clear shortly and Υ corresponds to the model parameters. We note that the expectation value E is a real number that can be obtained for an infinite amount of measurements. With a limited number of measurement samples $N_{\rm s}$, the statistical estimation is given by $\hat{E}[\mu^{(\lambda)}] = E[\mu^{(\lambda)}] + \hat{\delta}$,[1] where $\hat{\delta}$ is a random variable with zero mean and variance $\sigma^2 = \mathbb{E}(\hat{\delta}^2) = \sigma_0^2/N_{\rm s}$. Here σ_0^2 is the single-shot variance. Given a set of m scaling parameters $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ with $\lambda_j \geq 1$ and the corresponding measurement outcomes $\mu = \{\mu_1, \mu_2, \ldots, \mu_m\}$, the ZNE corresponds to building a good estimator $\hat{E}[\mu^{(0)}]$ for $E[\mu^{(0)}]$ such that its bias $\mathbb{E}(\hat{E}[\mu^{(0)}] - E[\mu^{(0)}])$ and its variance $\mathbb{E}(\hat{E}[\mu^{(0)}]^2) - \mathbb{E}(\hat{E}[\mu^{(0)}])^2$ are both reasonably small. From now on, we adopt a simplified notation of $E[\mu^{(\lambda)}] = E(\lambda)$.

We now discuss various statistical models for extrapolation. The expectation value $E(\lambda)$ cannot be an arbitrary function, which would make ZNE impossible to extrapolate back to $E(0)$. Depending on the underlying noise model assumption, one can apply various statistical models.

The *polynomial extrapolation* is based on the polynomial model of degree d such that

$$E_{\rm poly}^{(d)}(\lambda) = c_0 + c_1\lambda + \cdots + c_d\lambda^d, \tag{55}$$

where c_j are $d+1$ unknown real parameters. This extrapolation is justified in a weak noise limit and we need the number of data points m to be equal to or larger than $d+1$. Consequently, we can obtain two other variants: the *linear extrapolation* ($d = 1$) and the *Richardson extrapolation* ($d = m-1$) (Temme, Bravyi, and Gambetta, 2017). By construction, the error with respect to the true expectation value is $O(m)$ when we have a large sample size $N_{\rm s} \to \infty$. By using the interpolating *Lagrange polynomial*, the estimator is explicitly given by

$$\hat{E}_{\rm Rich}(0) = \hat{c}_0 = \sum_{k=1}^{m} \mu_k \prod_{i\neq k} \frac{\lambda_i}{\lambda_i - \lambda_k}, \tag{56}$$

with the assumption that all λ_j are different. One important observation is that the Richardson model–based ZNE is dictated by a statistical uncertainty that is scaling exponentially with the number of data points. There are also other statistical models such as *polyexponential extrapolation* (Giurgica-Tiron *et al.*, 2020) and *exponential extrapolation* (Endo, Benjamin, and Li, 2018). Various exponential extrapolation methods were proposed and investigated by Cai (2021a) and applied to depolarizing noise by Vovrosh *et al.* (2021). One can combine the ZNE-, quasiprobability-, and stabilizer-based approaches for further improvements (Cai, 2021a).

The ZNE scheme suffers from a few limitations. The scheme works by extrapolation, and hence it is challenging to obtain result guarantees in general. The number of measurement shots required to obtain the mitigated expectation value can be relatively high relative to the unmitigated case. The fundamental drawback of both ZNE and probabilistic error cancellation (PEC) (Temme, Bravyi, and Gambetta, 2017) and the quasiprobability method (which is discussed next) is that one needs to know the precise physical noise model in advance, which in itself is a difficult proposition. Experimentalists in the lab will have imperfect knowledge about the real noise, which typically differs from the canonical one. We also later discuss a more practical approach based on gate set tomography that was proposed by Endo, Benjamin, and Li (2018), which does not require explicit knowledge of the noise model and mitigates any localized Markovian errors, such that the error in the final output is due only to unbiased statistical fluctuation.

2. Probabilistic error cancellation

We now familiarize ourselves first with the notations used in quantum tomography (Merkel *et al.*, 2013; Greenbaum, 2015), which we adopt here. A quantum state is represented by a density matrix $\hat{\rho}$, and a physical observable is denoted by a Hermitian $\hat{A}$ operator. An operation is a map on the states space such that one can use the Kraus representation to denote it as $\hat{\hat{\mathcal{L}}}[\hat{\rho}] = \sum_j \hat{K}_j \hat{\rho} \hat{K}_j^\dagger$. We note that this equivalence with Eq. (52) is valid only when we have Markovian dynamics. Here $\hat{K}_j$ are Kraus operators. In terms of the Pauli transfer matrix representation, $\hat{\rho}$ in Eq. (52) can be written as a column vector denoted as $|\rho\rangle\rangle$ (Navarrete-Benlloch, 2015). Similarly, the Lindblad superoperator $\hat{\hat{\mathcal{L}}}$ can be recast as a square matrix using the Pauli transfer matrix representation. For simplicity and without loss of generality, we absorb the unitary dynamics [the first term on the right-hand side of Eq. (52)] into $\hat{\hat{\mathcal{L}}}$. A physical observable $\hat{A}$ is now written as a row vector $\langle\langle A|$. Consequently, the expectation value is $\langle\hat{A}\rangle = {\rm Tr}[\hat{A}\,\hat{\rho}] = \langle\langle A|\rho\rangle\rangle$. Likewise, the expectation of $\hat{A}$ after the state $\hat{\rho}$ passing through a series of linear maps is written as ${\rm Tr}[\hat{A}\hat{\hat{\mathcal{L}}}_N \circ \cdots \circ \hat{\hat{\mathcal{L}}}_1(\hat{\rho})] = \langle\langle A|\mathcal{L}_N \cdots \mathcal{L}_1|\rho\rangle\rangle$.

The central theme of PEC or the quasiprobability decomposition introduced by Temme, Bravyi, and Gambetta (2017) is that one can estimate the expectation value of an observable by sampling from a set of erroneous circuits, labeled by $\mathcal{L}_{\rm tot}^{(l)}$ for $l = 1, 2, \ldots$, such that

$$\langle\hat{A}^{(0)}\rangle = \sum_l q_l \langle\langle A^{(l)}|\mathcal{L}_{\rm tot}^{(l)}|\rho^{(l)}\rangle\rangle. \tag{57}$$

The expectation value of the observable differs from its ideal value due to the presence of noise. Given specific error models (assuming the experimentalist has full knowledge of them),

[1] The hat notation used is in accordance with statistics notation and should not be confused with a quantum operator.

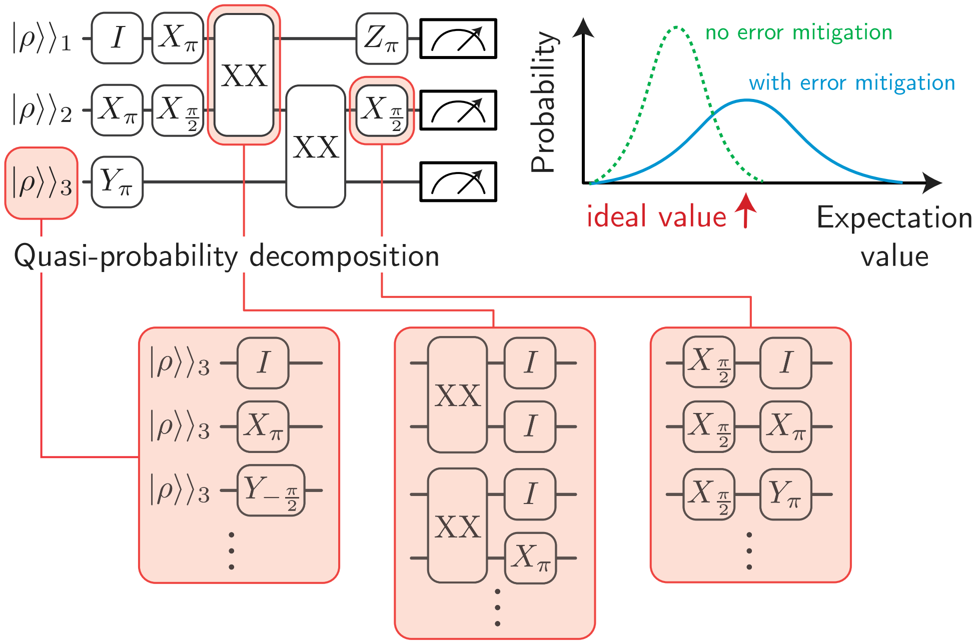

FIG. 5. Quantum computing of the expectation value of an observable using gate set tomography-based PEC. Quasiprobability decomposition of the initial state preparation is performed, and single- and two-qubit processes are computed. Implementing the resulting decomposition is done using a Monte Carlo approach. With QEM, the probability distribution of the expected value of the physical observable is centered around an ideal value with a larger variance than the one without QEM. Adapted from Zhang, Lu *et al.*, 2020.

the real numbers q_l, which represent quasiprobabilities, can be efficiently derived. Here each $\mathcal{L}_{\mathrm{tot}}^{(l)}$ represents the total sequence of noisy gates in the lth circuit. Monte Carlo sampling could be used to compute $\langle \hat{A}^{(0)} \rangle$ by randomly choosing the lth circuit with a probability $p_l = |q_l|/C$, where $C = \sum_l |q_l|$. Lastly, the computed result is given by the expected value of effective measurement outcomes $\langle A^{(0)} \rangle = CE[\mu_{\mathrm{eff}}]$, where the effective outcome is $\mu_{\mathrm{eff}} = \mathrm{sgn}(q_l)\mu^{(l)}$ if the lth circuit is chosen and $\mu^{(l)}$ is the outcome from the lth circuit. As a consequence, the mean value of the PEC outcome centers around the ideal one with a larger variance due to C; see the right corner of Fig. 5.

The previously mentioned PEC method relies on correct knowledge of the error model $\mathcal{L}_{\mathrm{tot}}^{(l)}$, as is apparent from Eq. (57). To enable practical implementations, Endo, Benjamin, and Li (2018) proposed combining linearly independent basis set operations and gate set tomography to fully remove the impact of localized Markovian errors by systematically measuring the effect of errors to design efficient QEM circuits. The set of operations including measurement and single-qubit Clifford gates is universal in computing the expectation values of observables. For the single-qubit case, any operation $\mathcal{L}$ that is a 4×4 real matrix in the Pauli transfer matrix representation can be expressed as a linear combination of 16 basic operations $\mathcal{L} = \sum_{i=1}^{16} q_i \mathcal{B}_i^{(0)}$, which are composed of $\{\pi, H, S, R_x, R_y\}$ gates (Endo, Benjamin, and Li, 2018). The same decomposition can be applied to the two-qubit case; see Fig. 5 for examples of decompositions.

A way to systematically measure errors is through gate set tomography (GST), with which one can even mitigate state preparation and measurement errors. In short, the purpose of GST is to measure noisy individual quantum circuit performance *a priori*. For a single-qubit gate, one prepares initial states $|0\rangle, |1\rangle, |+_x\rangle$, and $|+_y\rangle$, where $|+_x\rangle$ and $|+_y\rangle$ are the eigenstates of Pauli operators $\hat{\sigma}_x$ and $\hat{\sigma}_y$ with $+1$ eigenvalue, respectively. For noisy devices, these four states are denoted as $\bar{\rho}_1, \bar{\rho}_2, \bar{\rho}_3$, and $\bar{\rho}_4$, accordingly. We also use $\bar{\mathcal{L}}$ (superoperator) to denote a noisy or imperfect gate to be measured. Since we care about the expectation value of physical observables, for the single-qubit case we have observables $\hat{I}, \hat{\sigma}_x, \hat{\sigma}_y, \hat{\sigma}_z$, which are denoted as $\bar{A}_1, \bar{A}_2, \bar{A}_3, \bar{A}_4$. The mean value of the observables, the 4×4 matrix $\tilde{A}$, is simply $\tilde{A}_{j,k} = \mathrm{Tr}[\bar{A}_j \bar{\mathcal{L}} \bar{\rho}_k]$. Similarly, we can construct the 4×4 matrix g without applying any gate to the initial states as $g_{j,k} = \mathrm{Tr}[\bar{A}_j \bar{\rho}_k]$. This is repeated for each qubit and each single-qubit gate. Statistical estimation of the initial states $\bar{\rho}_k$ and the observables $\bar{A}_j$ are then given by

$$|\hat{\rho}_k\rangle\rangle = T_{\bullet,k}, \tag{58}$$

$$\langle\langle \hat{A}_j| = (gT^{-1})_{j,\bullet}, \tag{59}$$

where we note that the hat symbol represents the statistical estimate and $T_{\bullet,k}$ ($T_{j,\bullet}$) denotes the kth column (jth row) of the matrix T, where T is an invertible 4×4 matrix with the following relationship: $\hat{\mathcal{L}} = Tg^{-1}\tilde{\mathcal{L}}T^{-1}$. The same procedure applies for the two-qubit case, with the only difference being that there is a total of 16 initial states $\bar{\rho}_{k_1} \otimes \bar{\rho}_{k_2}$ and 16 observables $\bar{A}_{j_1} \otimes \bar{A}_{j_2}$ to be measured. Similarly, we have $g = g_1 \otimes g_2$ and $T = T_1 \otimes T_2$. We have to implement a two-qubit gate GST for each qubit pair involved in the quantum program.

The quasiprobability decomposition is then computed based on the previously mentioned GST results. From GST, we have an estimation of initial states $|\hat{\rho}_k\rangle\rangle$, observables to be measured $\langle\langle\hat{A}_j|$, and gates $\hat{\mathcal{L}}$. We denote $\mathcal{L}^{(0)}$ as the Pauli transfer matrix of the ideal gate with no error. The main idea of decomposition is that a noisy gate operation can be modeled as an ideal operation followed by a noise operation, i.e., $\mathcal{L} = \mathcal{N}\mathcal{L}^{(0)}$. Hence, the inverse of the noise is given by $\mathcal{N}^{-1} = \mathcal{L}^{(0)}\hat{\mathcal{L}}^{-1} = \sum_i q_{\mathcal{L},i}\hat{\mathcal{B}}_i$. By applying the inverse of the noise after the operation, we can obtain the operation without error $\mathcal{L}^{(0)} = \mathcal{N}^{-1}\mathcal{L}$. Notice that the matrices in the previous equation are obtained from the first GST step. The remaining task is to determine the quasiprobabilities $q_{\mathcal{L},i}$ for each qubit and gate involved by solving the previous equation. We note that instead of a quasiprobabilistic decomposition of quantum gates, one could use the randomized compiling technique proposed by Wallman and Emerson (2016).

GST-based PEC experiments have recently been realized in trapped-ion systems (Zhang, Lu *et al.*, 2020) and superconducting circuits (Song *et al.*, 2019). Lastly, a similar strategy was recently applied to mitigating errors in measurement readout (Kwon and Bae, 2021).

3. Other QEM strategies

We have seen that the quantum error-mitigation techniques discussed thus far do not require extra qubits, with the caveat that one needs to perform more measurements. At the same time, one is interested only in information about the expectation value. Along this line of thought, there are several proposals, which we outline next. However, some of the methods might require extra qubits.

The *subspace expansion method* (McClean *et al.*, 2017; Colless *et al.*, 2018; McArdle, Yuan, and Benjamin, 2019; Sagastizabal *et al.*, 2019a; Barron *et al.*, 2020; McClean, Jiang *et al.*, 2020) is designed to mitigate errors in the VQE routine, where we want to find an approximate ground state $|\psi_a\rangle$ of a system Hamiltonian H. However, the state found by VQE may differ from the true ground state $|\psi_g\rangle$ due to noise. In general, we do not know which error occurred in the state. The subspace expansion method works by resolving the action of H on the linear combination of quantum states *Ansatz* (42). The subspace is spanned by a set of operators $\hat{\mathcal{O}}_i$, i.e., $\{|\hat{\mathcal{O}}_i\psi_a\rangle\}$. Now one proceeds to evaluate $H_{ij} = \langle\psi_a|\hat{\mathcal{O}}_i H\hat{\mathcal{O}}_j|\psi_a\rangle$ and $S_{ij} = \langle\psi_a|\hat{\mathcal{O}}_i\hat{\mathcal{O}}_j|\psi_a\rangle$. The latter is needed since the subspace states are in general not orthogonal to each other. By solving the generalized eigenvalue problem $HC = SCE$, with eigenvectors C and diagonal matrix of eigenvalues E, we can obtain the Hamiltonian spectra including the excited states; see Sec. VI.A.4. This method requires an appropriate choice of subspace operators to mitigate errors due to noise. In general, without knowing the noise models of the quantum device, it would require an exponential number of expansion operators to obtain the optimal ground state.

The *stabilizer-based approach* (Bonet-Monroig *et al.*, 2018; McArdle, Yuan, and Benjamin, 2019; Sagastizabal *et al.*, 2019a; Cai, 2021b) relies on the information associated with conserved quantities in the *Ansatz* such as spin or particle number. If any change in these quantities is detected, one can pinpoint an error in the circuit, which is similar to stabilizer measurements in QEC schemes. We can implement the stabilizer measurements by adding ancilla qubits to the qubit registers or taking additional measurements and postprocessing.

The *individual error reduction* method was proposed by Otten and Gray (2019). As we saw earlier, Markovian noise can be modeled using the Lindblad master equation [Eq. (52)], where $d\hat{\rho}/dt = \hat{\hat{\mathcal{L}}}(\hat{\rho}) = \sum_i \mathcal{L}_i(\hat{\rho})$, with each $\mathcal{L}_i$ denoting the presence of a noise channel. Here we have absorbed the unitary component into $\hat{\hat{\mathcal{L}}}$. It was shown that

$$\tilde{\rho}(T) = \hat{\rho}(T) - \sum_{j=1}^{m}\frac{1}{g_j}[\hat{\rho}(T) - \hat{\rho}_j(T)] \tag{60}$$

$$= \hat{\rho}^{(0)}(T) + \mathcal{O}(\tau^2). \tag{61}$$

In Eqs. (60) and (61) $\hat{\rho}(T)$ is the density matrix after quantum gates are applied with the presence of all associated noise channels at the final evolution time T. In contrast, $\hat{\rho}_j(T)$ is the state under the influence of all the noise channels but one less $\mathcal{L}_j$ according to the ratio g_j. Notice that if $g_j = 1$, we have fully removed the entire channel $\mathcal{L}_j$. $\hat{\rho}^{(0)}(T)$ is the ideal output state without any error, while τ is the evolution time for each noise process after the application of the gate. As a result, the first-order error $\mathcal{O}(\tau)$ is removed. Using Eq. (60), we can now obtain the mitigated expectation value $\langle\hat{A}\rangle = \mathrm{Tr}[\hat{\rho}^{(0)}(T)\hat{A}]$ for a physical observable $\hat{A}$. However, this method assumes that it is possible to remove individual noise channels. Hence, it may be challenging to achieve this on current quantum hardware compared to other strategies.

Dynamic error suppression and robust control techniques aim to suppress experimental gate errors at the pulse control level, which can be performed in a passive or an active way. The pulse shaping technique is a strategy for passive cancellation of system-bath interactions. This method, which is commonly known as the derivative removal of adiabatic gate scheme (Motzoi *et al.*, 2009; Gambetta *et al.*, 2011; De, 2015), builds upon techniques to obtain high-fidelity quantum gates in nonlinear qubits such as transmons. On the other hand, *dynamical decoupling* (DD) (Viola, Knill, and Lloyd, 1999; Santos and Viola, 2005; Viola and Knill, 2005) is a well-known and widely used quantum control technique in the literature that is designed to suppress decoherence via pulses to the system so that it cancels the system-bath interaction to a given order in time-dependent perturbation theory (Lidar, 2014) in an active manner. Recently DD experiments were performed on the 16-qubit IBMQX5, 5-qubit IBMQX4, and 19-qubit Rigetti Acorn chips (Pokharel *et al.*, 2018), where the gain in substantial gate fidelity relative to unprotected, free evolution of individual transmon qubits was demonstrated. One may combine DD and pulse shaping techniques to obtain dynamically corrected gates (Khodjasteh and Viola, 2009; Edmunds *et al.*, 2020) composed of shaped pulses that actively drive state evolution within a Hilbert space in order to cancel certain system-bath couplings. With IBM's Qiskit

Pulse (Alexander *et al.*, 2020), which allows users to control back-end pulse shapes and sequences of a quantum processor on the fly, a recent study (Carvalho *et al.*, 2020) using robust control techniques substantially improved the performance of a NISQ computer. Carvalho *et al.* (2020) achieved an ~10 times single-qubit gate coherent-error reduction, an ~5 times average coherent-error reduction, an ~10 times increase in calibration window to one week of valid pulse calibration, an ~12 times reduction gate-error variability across qubits and over time, and up to an ~9 times reduction in single-qubit gate error including cross talk. These improvements have implications for the performance of multiqubit gates in trapped ions (Milne *et al.*, 2020). With these tools, we envisage the possibility of realizing holonomic quantum gates (Zanardi and Rasetti, 1999; Zhang *et al.*, 2016; Zhang, Kyaw, Filipp *et al.*, 2021), which are robust against parameter fluctuations and attain even better gate fidelity and performance.

A *Lanczos-inspired approach* (Suchsland *et al.*, 2020) estimates the expectation value of a physical observable $\mathrm{Tr}[\hat{\rho}^{(0)}\hat{A}]$ by constructing a basis of the order-m Krylov subspace $\mathcal{K}^{(m)}$ spanned by $\{|\Psi\rangle, H|\Psi\rangle, H^2|\Psi\rangle, \ldots, H^m|\Psi\rangle\}$. This can be used to systematically construct the objective function to be minimized. For the mth order, the objective function is given by

$$E_{L,k,n,m} = \min_{a\in\mathbb{R}^m} \sqrt[k]{\frac{\langle\Psi|H^k(\sum_{i=0}^{m-1} a_i H^i)^n|\Psi\rangle}{\langle\Psi|(\sum_{i=0}^{m-1} a_i H^i)^n|\Psi\rangle}}. \tag{62}$$

With the Krylov expansion, this technique can reduce the impact of different sources of noise by performing additional measurements without the need to increase the circuit depth. Calculating dynamic quantities such as Hamiltonian moments (Vallury *et al.*, 2020) and a quantum power method based on a higher-order Suzuki-Trotter expansion (Seki and Yunoki, 2021) on near-term quantum computers are two recent examples of this approach.

Learning-based and artificial intelligence–inspired methods employ machine learning techniques such as regression for error mitigation. The process consists of training different candidate circuit variants with non-Clifford gates substituted by gates with efficient classical simulability (Czarnik *et al.*, 2020; Strikis *et al.*, 2020). A recent approach suggests merging zero-noise extrapolation with learning-based methods for near-Clifford circuits (Lowe *et al.*, 2020). There are also genetic algorithms to mitigate errors in quantum simulations (Las Heras *et al.*, 2016a; Spagnolo *et al.*, 2017).

B. Circuit compilation

As discussed in Sec. V.C, a quantum computer is composed of its hardware (quantum) and software (classical). The software translates a quantum algorithm into a set of instructions that implement the desired quantum operations and read out the qubit states. This process can be understood as *quantum compilation* (Chong, Franklin, and Martonosi, 2017), but the term is not limited to this application. When mapping a quantum circuit to a specific device architecture, one needs to consider the available quantum gates, the qubit connectivity that allows two-qubit gate implementation, and experimental limitations such as decoherence time, which imposes a certain maximum circuit depth in terms of the number of gates. For these reasons, it has become indispensable to develop tools that allow for circuit simplifications and efficient mappings of the general algorithm to a specific hardware. These tools are also known as quantum compilers since they translate the theoretical circuit to a realistic simulator or device. In the following, we describe some of these tools. Many of them are suited to both NISQ and fault-tolerant quantum computation.

1. Native and universal gate sets

The available gates that can be implemented experimentally on a particular hardware platform are sometimes referred to as the *native gate set*. With a universal gate set $\mathcal{G} \in SU(d)$ (also called a *instruction gate set*), any unitary operation can be constructed efficiently. More formally, the *Solovay-Kitaev theorem* (Dawson and Nielsen, 2006) states that, given this universal set $\mathcal{G}$, any unitary operation $U \in SU(d)$ can be approximated with ϵ accuracy with a finite sequence S of gates from $\mathcal{G}$. This sequence scales logarithmically as $\mathcal{O}(\log^c(1/\epsilon))$, where c is a constant that depends on the theorem proof. For $d = 2^n$ this theorem guarantees that qubit quantum circuits can be decomposed using a finite gate sequence. Although this is one of the most important theorems in quantum computation, it is an existence theorem; i.e., it does not provide the decomposition that it predicts. It also requires that the gate set contains the inverse of all gates. Further developments presented by Bouland and Ozols (2018) try to remove this assumption.

The Clifford group is an important object in quantum information science because of its applications in QEC, randomized benchmarking, and investigations for quantum advantage. The generalized Pauli operators in prime dimension p are given by

$$T_{(a,b)} = \begin{cases} \omega^{-ab/2} Z^a \mathcal{X}^b, & (a,b) \in \mathbb{Z}_p \times \mathbb{Z}_p, p \neq 2, \\ \imath^{ab} \mathcal{Z}^a \mathcal{X}^b, & (a,b) \in \mathbb{Z}_2 \times \mathbb{Z}_2, p = 2, \end{cases} \tag{63}$$

where $\omega = \exp(2\pi i/p)$ and $\mathbb{Z}_p$ denotes an integer modulo p. The $\mathcal{Z}$ and $\mathcal{X}$ operators are defined via their action on computational basis states $\{|k\rangle\}_k$, with $\mathcal{X}|k\rangle = |k + 1 \bmod p\rangle$ and $\mathcal{Z}|k\rangle = \omega^k|k\rangle$. The unitaries that map the set of generalized Pauli operators to themselves up to a phase are called *Clifford unitaries*. We denote the set of p-dimensional Clifford unitaries by $\mathcal{C}_p$. Mathematically speaking,

$$U \in \mathcal{C}_p \Leftrightarrow \exists\,\phi : UT_{(a_1,b_1)}, \qquad U^\dagger = \exp(i\phi) T_{(a_2,b_2)}, \tag{64}$$

where $T_{(a_1,b_1)}$ and $T_{(a_2,b_2)}$ are generalized Pauli operators. The set of Clifford unitaries $\mathcal{C}_p$ forms a group called the *Clifford group*. In this review, we focus on $p = 2$, i.e., the qubit Clifford group.

For qubits, the generators of the Clifford group are the Clifford gates H, $S = \exp(-i\pi/4\sigma^z)$, and CNOT. Any circuit composed of Clifford gates can be simulated efficiently with a classical computer, as stated in the *Gottesman-Knill theorem* (Aaronson and Gottesman, 2004). The states generated by the

Clifford gates are also called stabilizer states and can contain a high amount of entanglement. However, not all unitaries can be decomposed into Clifford gates. A universal gate set is a set of gates that can perform arbitrary quantum computations. The Clifford gates combined with the $T = \exp(-i\pi/8\sigma^z)$ gate form an example of such a universal gate set. It is a necessary condition for a quantum advantage that a circuit contains T gates. As such, the computational difficulty of simulating a quantum circuit with classical computers increases with the number of T gates in the circuit. For this reason, many algorithms try to reduce quantum circuits to the minimal number of T gates to give an estimation of the difficulty of classically simulating a circuit (Amy *et al.*, 2013; Gosset *et al.*, 2013; Heyfron and Campbell, 2018; Amy and Mosca, 2019; Kissinger and van de Wetering, 2019).

Other basic decompositions besides these minimal reduction algorithms are useful. Even if only a native gate set is available experimentally, other basic gates can be constructed and used in algorithms. As an example, S and T gates are particular cases of the single-qubit rotational gate R_z, and the H gate can be obtained from R_y and R_x gates as $H = R_y(-\pi/2)R_x(\pi)$. Any single-qubit unitary can be decomposed into the gate sequence $U(\theta, \phi, \lambda) = R_z(\phi)R_y(\theta)R_z(\lambda)$. This motivates using single-qubit rotational gates and at least one entangling gate (such as a CNOT or CZ gate) as native gate sets. Any two-qubit gate can be obtained from this minimal set by using circuit decompositions (Barenco *et al.*, 1995; Blaauboer and De Visser, 2008; Watts, O'Connor, and Vala, 2013; de Guise, Di Matteo, and Sánchez-Soto, 2018; Peterson, Crooks, and Smith, 2020). The choice of the entangling gate can be motivated by the experimental platform. Depending on the technology used to construct the quantum device, a natural two-qubit gate implementation could be better suited. Some examples are the use of CZ gates in tunable superconducting circuits (Krantz *et al.*, 2019), cross-resonance gates in fixed frequency superconducting qubits (Krantz *et al.*, 2019; Kjaergaard *et al.*, 2020), and the XX gates in trapped ions (Häffner, Roos, and Blatt, 2008). More expressive gate sets with continuous gate parameters or long-range interactions can be achieved by means of further control over the hardware parameters in time (Bastidas *et al.*, 2020; Foxen *et al.*, 2020; Krinner *et al.*, 2020; Lacroix *et al.*, 2020). The complexity of the circuit decomposition into CNOT and R_z gates was analyzed by Amy, Azimzadeh, and Mosca (2018).

2. Circuit decompositions

Once the native gate set is fixed, the next step consists of decomposing the theoretical unitary circuit into this basic set. A raw translation of all single- and two-qubit gates into the native set might imply a large circuit depth, which would reduce the effectiveness of that decomposition. Moreover, finding the decomposition of gates acting on more than one qubit might prove challenging in general. Besides the common circuit decompositions mentioned before, one may need mathematical tools to understand and derive general circuit reductions to particular smaller pieces.

One of these mathematical tools is the so-called ZX-calculus. It is a graphical language that maps quantum circuits to particular graph representations and derives a set of rules to manipulate these graphs. Its application range goes from measurement-based quantum computation to QEC. For a complete review about ZX-calculus and its variety of applications, see van de Wetering (2020). For the purpose of this review, we are interested in the quantum circuit simplification applications (Cowtan, Dilkes, Duncan, Simmons, and Sivarajah, 2019; de Beaudrap, Bian, and Wang, 2019; Duncan *et al.*, 2020; Hanks *et al.*, 2020; Kissinger and van de Wetering, 2020).

Other approaches use well-known artificial intelligence algorithms to find optimal circuit decompositions such as reinforcement learning (Pirhooshyaran and Terlaky, 2020; Zhang, Zheng *et al.*, 2020). Evolutionary algorithms such as genetic algorithms have been widely studied as well (Williams and Gray, 1999; Massey, Clark, and Stepney, 2004, 2006; Bang and Yoo, 2014; Las Heras *et al.*, 2016b; Li, Alvarez-Rodriguez *et al.*, 2017; Lamata, Alvarez-Rodriguez *et al.*, 2018; Potoček *et al.*, 2018). In these approaches, multiple random circuits composed of the native gate set are generated and evolved. The evolution strategy includes the definition of possible mutations, such as the introduction of a new gate on a particular qubit, the swap between circuit gates, or the deletion of a particular gate. A multiobjective loss function is then used to estimate the success of each circuit family under a given convergence criterion, after which the circuit with the best performance is selected. These works can be applied to finding the optimal PQC for a given VQA, as discussed in Sec. II.B. A VQA for circuit compilation using a genetic algorithm as optimization subroutine, called the quantum assisted quantum compiler, was presented by Khatri *et al.* (2019).

3. The qubit mapping problem

After decomposing and simplifying the quantum circuit into the native gates, a hardware-specific task remains: mapping the resultant circuit to the particular qubit connectivity or topology, a task also known as the *qubit routing problem*. In general, because of experimental limitations, not all qubits are connected such that two-qubit gates can be applied between them. A naive approach to circumventing this limitation consists of swapping each qubit state with its neighbor (by using SWAP gates) until we find a connected pair, perform the desired two-qubit operation, and swap back the states of the qubits involved, returning to the original state with the intended two-qubit gate applied to it. This translates into a significant growth of the circuit depth for circuit topologies with a sparse qubit connectivity.

Some NISQ algorithms presented in this review may include the qubits' connectivity by means of the loss function or the available rules used to decompose the unitaries. However, quantum compilation is a hardware-specific transformation, and it might be more useful to apply this step independently of the quantum circuit and instead depending on the chip architecture. The qubit mapping problem is NP complete (Botea, Kishimoto, and Marinescu, 2018). Several heuristic approaches based on dynamic programming, depth partitioning (Zulehner, Gasser, and Wille, 2017; Siraichi *et al.*, 2018; Cowtan, Dilkes, Duncan, Krajenbrink *et al.*, 2019; Li, Ding, and Xie, 2019; Zulehner, Paler, and Wille, 2019; Zulehner and Wille, 2019), and reinforcement learning (Pozzi *et al.*, 2020) have been explored. Exact methodologies based on reasoning engines such as Boolean satisfiability solvers have also been proposed (Wille, Burgholzer, and Zulehner, 2019; Tan and Cong, 2020). The so-called Lechner-Hauke-Zoller architecture is an approach that solves the connectivity issue at the cost of increasing the number of qubits (Lechner, 2020). The same framework can be applied to quantum annealing systems as well (Lechner, Hauke, and Zoller, 2015). Encoding this problem into a QUBO (see Sec. III.A) to solve it using classical simulated annealing was proposed by Dury and Di Matteo (2020). Approaches for circuit compilation based on the commutation algebra of quantum gates were suggested by Itoko *et al.* (2019, 2020).

There can be many possible qubit mappings of a given algorithm to a particular device if not all qubits are required. In those cases, one can put some extra effort into finding the best performing qubits in terms of error rates and coherence times (Nishio *et al.*, 2020; Niu *et al.*, 2020). In that direction, finding the mapping with the lowest circuit depth may prove valuable for reducing the errors due to decoherence (Zhang, Chen *et al.*, 2020).

Finally, the use of circuit synthesis with connectivity constraints has also been proposed. Some of these works are based on Gaussian elimination processes where one takes the matrix representation of the circuit transformation and manipulates it to extract the basic transformations (in particular, the CNOT gates that respect the connectivity) (Kissinger and de Griend, 2019; Nash, Gheorghiu, and Mosca, 2020). The strategy of Gheorghiu *et al.* (2020) consisted of slicing the circuit into smaller parts that could be adapted and transformed to fit into the particular topology. One can also adapt this problem to the syndrome decoding problem (de Brugière *et al.*, 2020). de Griend and Duncan (2020) solved the qubit routing problem for phase polynomial circuits, which are circuits that contain only CNOT and R_z gates.

4. Resource-aware circuit design

As described in Sec. II.B, there are different strategies for designing a circuit *Ansatz*. However, many of them require circuit depths, qubit connectivity, and a number of parameters that are beyond the capabilities of current quantum hardware. Next we discuss strategies for designing and adapting PQCs and VQAs to the characteristics of devices.

ADAPT VQE.—Early VQA employed a fixed *Ansatz* design with its parameters tuned using a classical optimizer. The adaptive derivative-assembled pseudo-Trotter *Ansatz* variational quantum eigensolver (ADAPT VQE) was introduced as a more scalable and efficient way to simultaneously design and optimize a parametrized *Ansatz* (Grimsley, Economou *et al.*, 2019). At each iteration, ADAPT VQE constructs an *Ansatz* by adding an operator corresponding to the largest gradient from a carefully designed operator pool. That is, given an operator $\hat{\tau}_i$ from the operator pool, the gradient of the energy with respect to the corresponding parameter θ_i is defined as

$$\partial_i E = \langle\psi|[H, \hat{\tau}_i]|\psi\rangle, \tag{65}$$

where $|\psi\rangle$ is the *Ansatz* at the current iteration to be updated. After computing the gradient components and choosing the operator corresponding to the largest gradient, the gate operation implementing $\hat{\tau}_i$ is added to the *Ansatz* with its parameter value initialized at 0. The *Ansatz* is then optimized before adding the next operator. The ADAPT-VQE algorithm terminates when the norm of the gradient vector falls below a predefined threshold.

In the case of fermionic ADAPT VQE, the operator pool consists of fermionic operators that are transformed into quantum gate operations through the Jordan-Wigner mapping. A more hardware-efficient variant of the ADAPT-VQE algorithm is the qubit ADAPT VQE, in which the pool consists of gate operators acting directly on qubits (Tang *et al.*, 2019). Both versions of ADAPT VQE were able to generate optimized circuits with reduced depths and CNOT counts relative to previous *Ansatz* construction and optimization methods.

MI ADAPT- VQE.—The mutual information-assisted ADAPT VQE (MI ADAPT VQE), introduced by Zhang, Kyaw, Kottman *et al.* (2021), leverages the density matrix renormalization group (DMRG) (White, 1992; Hallberg, 2006) method to accelerate the circuit constructions for the ADAPT-VQE routine. Instead of gradients, it uses mutual information to guide the construction of circuits. At the beginning of the algorithm, the pairwise quantum mutual information is approximated using DMRG, which is then applied to construct a reduced pool of entangling gates. In each iteration of the method, new circuits are generated in which quantum gates are distributed mainly among pairs of qubits corresponding to considerable mutual information. This avoids allocating quantum resources on pairs of qubits that are less important to entangle. Numerical experiments suggest that the number of new circuits needed in each step of the adaptive construction can be significantly reduced using MI ADAPT VQE, thereby saving both time and quantum resources. The number of trial circuits in certain cases can be reduced to about 5% for H_2 and 10% for H_2O relative to ADAPT VQE, using an operation pool based on the qubit coupled-cluster method (Ryabinkin *et al.*, 2020).

MoG VQE.—To reduce two-qubit gate counts for near-term experiments, the multiobjective genetic variational quantum eigensolver (MoG VQE) optimizes for both the energy and the number of CNOT gates in the quantum circuit (Chivilikhin *et al.*, 2020). The MoG-VQE algorithm combines the two following evolutionary strategies: (i) NSGA-II (Deb *et al.*, 2000), a multiobjective genetic algorithm for proposing a circuit structure to minimize both the energy and the CNOT count, and (ii) CMA-ES (Hansen, Müller, and Koumoutsakos, 2003), which tunes the parameters and evaluates the optimized energies for the qubit topologies suggested by the NSGA-II algorithm. MoG VQE initializes a diverse population by sampling a checkerboard pattern of two-qubit circuit blocks. To vary the populations over different generations, the three possible mutation operators are (i) inserting a two-qubit circuit block in a random position, (ii) removing a two-qubit circuit block in a random position, and (iii) adding or removing ten

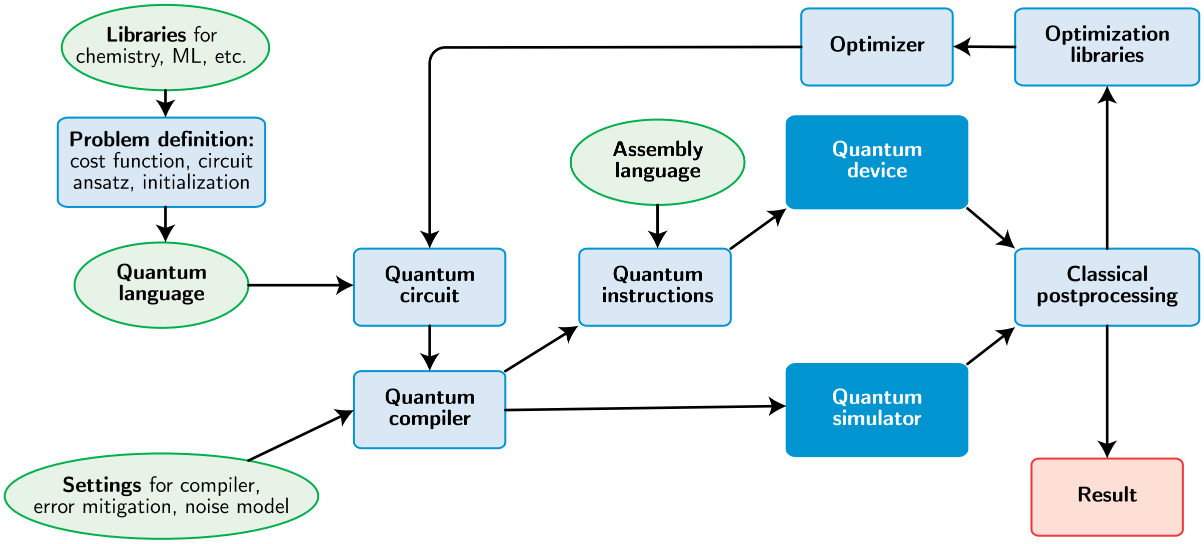

FIG. 6. Schematic representation of a standard NISQ programming workflow. Green circular boxes represent the libraries and languages used for designing, optimizing, and running a quantum algorithm in a real quantum device or a simulator. External libraries can be used to define the problem or to improve the performance of the algorithm by simplifying the circuit or providing error-mitigation techniques. An assembly language is needed to translate the theoretical algorithm to a set of physical operations on the quantum hardware. Classical postprocessing is necessary to manipulate the result of the computation and to either obtain the final result or send the provisional one to a classical optimizer (VQA).

circuit blocks to help escape from local minima. Chivilikhin *et al.* (2020) noted that (iii) is selected with a lower probability than mutation operators (i) and (ii). Parents are selected using the tournament selection method. For each circuit topology, the corresponding energy is evaluated using the CMA-ES optimizer. These steps are repeated until certain termination criteria are satisfied. Using MoG VQE, Chivilikhin *et al.* reported significant reductions in the CNOT counts relative to those of other hardware-efficient circuits when estimating the ground state energies of several molecules. For example, for a 12-qubit LiH Hamiltonian, MoG VQE generated a circuit corresponding to estimating the ground state energy within chemical precision using only 12 CNOT gates.

PECT.—An alternative approach for adaptively constructing and optimizing an *Ansatz* was introduced by the parameter-efficient circuit training (PECT) scheme (Sim *et al.*, 2020). PECT enables optimizations of predefined *Ansätze*, such as unitary coupled-cluster or the low-depth circuit *Ansatz* (LDCA) (Dallaire-Demers *et al.*, 2019), by dynamically pruning and adding back parametrized gates during an optimization. After selecting an *Ansatz* U, a subset of gate operations from A is chosen while other parametrized gate operations are tuned to identity operations. This results in an *Ansatz* substructure A' with reduced circuit depth and gate count. Parameters of A' are then optimized, following what Sim *et al.* (2020) called a "local optimization" step. After local optimization, to refine or reparametrize the *Ansatz* substructure, parameters with small magnitudes are pruned or removed. A heuristic growth rule is used to grow back the same number of parameters that were pruned. Steps of local optimization and *Ansatz* reparametrization are repeated until termination criteria are met. Because PECT optimizes parameter subsets at any iteration, circuits that are executed on the quantum computer have reduced depths and CNOT counts relative to the original *Ansatz*. Using PECT, Sim *et al.* were able to optimize 12-qubit LDCA circuits, naively equipped with hundreds to low thousands of parameters, to estimate ground state energies of LiH and H_2O. Previous optimizations of LDCA were limited to eight qubits.

C. Quantum software tools

A quantum computer is a hybrid device composed of quantum hardware and classical software that controls it by sending a list of instructions and processing the results of the computation. This hybrid nature is accentuated in the NISQ era, as explained in the review. Thus, the classical subroutines are part of the core in state-of-the-art NISQ algorithms and a language to communicate with the quantum device is a necessity. On top of that, almost all progress in quantum algorithms is tested in quantum simulators, making it essential to perform proof-of-concept simulations before or until the algorithm is applicable to real devices.

Figure 6 represents diagrammatically the typical workflow of a NISQ algorithm. The individual parts of the problem, such as the objective function to optimize, the quantum circuit design, or the initialization parameters, are translated into quantum circuits by a classical precomputation step. The syntax of this language includes quantum gates, qubit initialization, objective function definition, etc. The theoretical circuit is then compiled to fulfill experimental limitations such as qubit connectivity, a native quantum gate set, or circuit depth. To accomplish this task, compilers that allow for circuit simplification (see Sec. V.B) or noise models (for simulation purposes) might be useful. After this preprocessing step, the algorithm is ready to enter into the quantum-classical loop. The quantum circuit can be run on a quantum simulator or real hardware. In the latter case, an assembly language (Smith, Curtis, and Zeng, 2016; Cross et al., 2017; Khammassi et al., 2018; Killoran et al., 2019) will translate the quantum circuit into a set of instructions for the device. After the qubits are measured, the result can be postprocessed and techniques such as error mitigation might be used. Either the algorithm finishes or the result is sent to the classical optimizer that computes the next loop variables (such as for VQA).

We define a *quantum software library* as a library or a set of libraries written in a classical programming language (such as PYTHON or C++) that allows for writing quantum programs. In some cases, these libraries are open source and can be used directly on real hardware or on a quantum simulator. The proliferation of all these libraries, simulators, and devices has also created a necessity for some multiplatform languages. These are those that can use multiple quantum software libraries as a back end, thus reducing the programming efforts substantially by unifying the language syntax. Some of these packages include built-in sublibraries suited for particular applications, from chemistry to QML, or particular well-known algorithms such as VQE and QAOA.

We provide a list of some open-source libraries suited for NISQ computation in Table VII of the Supplemental Material (824). This list represents just a snapshot of the state-of-the-art of quantum software ecosystem as new tools are being developed and some projects are being abandoned. An updated list of quantum software resources was given by QOSF (2020a),[2] and detailed comparative analyses between some of these languages and libraries were given by Gay (2006), Heim et al. (2020), Nguyen et al. (2020), and Garhwal, Ghorani, and Ahmad (2021). Because of the broad applications of NISQ algorithms, specific libraries used in other fields beyond quantum computation can also be required, e.g., quantum chemistry and machine learning libraries or external compilers and simulators. These libraries are used for other applications besides quantum computing, so we consider and list them as external libraries in Table VIII of the Supplemental Material (824), although most of them are integrated within the quantum software libraries.

VI. APPLICATIONS

A. Many-body physics and chemistry

Understanding the static and dynamic properties of quantum-mechanical systems is a core challenge at the heart of many fields, such as chemistry and physics. Classical numerical methods often struggle to solve these problems due to the exponential increase of resources needed with a growing number of particles to simulate. Owing to their quantum-mechanical nature, quantum computers offer a way to simulate even large-scale many-body systems (Feynman, 1982; von Burg et al., 2020). The initial application for chemistry was to obtain molecular energies via quantum phase estimation on a quantum computer (Aspuru-Guzik et al., 2005). Besides the molecular energy, properties that can be extracted from a successfully prepared ground state, such as energy derivatives with respect to the nuclear framework, are of similar interest (Kassal and Aspuru-Guzik, 2009; O'Brien et al., 2019). Fault-tolerant quantum algorithms have the potential to become *killer applications* in the computational discovery of chemical reaction mechanisms (Reiher et al., 2017), and NISQ algorithms could play a major role in their realization. Here we review various NISQ algorithms that have been proposed to tackle quantum chemistry and many-body physics-related problems. We start by introducing concepts on mapping physical problems onto the quantum computer. We then introduce algorithms for common challenges, such as finding the static and the dynamic properties of quantum systems in various settings. All NISQ algorithms discussed in this section are listed in Table I of the Supplemental Material (824).

1. Qubit encodings

In general, any physical system can be written in terms of a Hamiltonian that is the sum of its kinetic and potential energy. In quantum theory, each physical system is associated with a language of operators and an algebra establishing such language. Depending on the system constituents, there are three types of particles (operators) in nature: fermions, bosons, and anyons. The first two are elementary particles obeying Fermi-Dirac and Bose-Einstein statistics, respectively. Anyons are quasiparticles obeying continuous or anyonic statistics that exist only in two-dimensional confinement. Quantum computers operate in the language of qubits (a distinguishable set of spin-$1/2$ particles). Hence, the quantum simulation of a physical system refers to performing a one-to-one mapping from the system operator to the quantum computing language, and thus preserving the underlying statistics. For a recent review on hardware-dependent mappings of spin Hamiltonians into their corresponding quantum circuit, see Tacchino, Chiesa et al. (2020).

In the standard model of quantum computation, a two-level system or spin-$1/2$ particle is denoted by its spin orientation $|\uparrow\rangle = |0\rangle = (1,0)^T$ and $|\downarrow\rangle = |1\rangle = (0,1)^T$. An N-qubit system is then constructed from the standard Pauli matrices $\hat{\sigma}_x^i, \hat{\sigma}_y^i, \hat{\sigma}_z^i$, where the superscript i refers to the ith local qubit site. These operators satisfy the commutation relations of a $\bigoplus_{i=1}^N \mathrm{su}(2)_i$ algebra $[\hat{\sigma}_\mu^l, \hat{\sigma}_\nu^m] = 2i\delta_{lm}\epsilon_{\mu\nu\lambda}\hat{\sigma}_\lambda^l$, where $\epsilon_{\mu\nu\lambda}$ is the totally antisymmetric Levi-Civita symbol with $\mu, \nu, \lambda \in \{x, y, z\}$.

Fermions.—In the second quantized notation, N fermions are denoted by fermionic operators $\hat{f}_i^\dagger$ $(\hat{f}_i)$, the creation (annihilation) operators of a fermion in the ith mode or site $(i = 1, \ldots, N)$. The fermionic operators obey Pauli's exclusion principle and the antisymmetric nature of the fermion wave function. Hence, the fermionic algebra is defined by the anticommutators $\{\hat{f}_i, \hat{f}_j\} = 0, \{\hat{f}_i^\dagger, \hat{f}_j\} = \delta_{ij}$. There are a number of well-known mappings that allow the description of a fermionic system by the standard model of quantum computers. They are the Jordan-Wigner transformation (Jordan and Wigner, 1928), Bravyi-Kitaev transformation (Bravyi and Kitaev, 2002), and Ball-Verstraete-Cirac transformation (Ball, 2005; Verstraete and Cirac, 2005). Steudtner and Wehner (2019) took the two-dimensional topology of most proposed qubit architectures explicitly into account and

[2] See also https://quantumcomputingreport.com/tools/.

compared them to some of the aforementioned one-dimensional mappings. More advanced mappings using the interaction graph of the Hamiltonian (Setia and Whitfield, 2018; Setia *et al.*, 2019) or customized quasilocal and local encodings (Havlíček, Troyer, and Whitfield, 2017; Chien and Whitfield, 2020; Derby and Klassen, 2020; Jiang *et al.*, 2020) have been introduced as well. Other approaches try to lower the qubit requirements of the mapped fermionic operators by taking inspiration from classical error-correction codes and the internal symmetries of the system (Bravyi *et al.*, 2017; Steudtner and Wehner, 2018). Other examples are mappings based on point-group symmetries of molecular Hamiltonians (Setia *et al.*, 2020). Recently mappings of SU(N) fermions to qubits were proposed (Consiglio *et al.*, 2021).

In the following, we outline the oldest and most intuitive mapping: the Jordan-Wigner transformation. In this mapping, the qubit states are equivalent to the second-quantized occupation number vectors, and fermionic creation and annihilation operators are transformed to qubit raising and lowering operators $\hat{\sigma}^j_\pm = (\hat{\sigma}^j_x \pm i\hat{\sigma}^j_y)/2$ combined with strings of $\hat{\sigma}_z$ operators that ensure the correct anticommutation properties

$$\hat{f}_j \to \left(\prod_{l=1}^{j-1} -\hat{\sigma}^l_z\right)\hat{\sigma}^j_-, \qquad f^\dagger_j \to \left(\prod_{l=1}^{j-1} -\hat{\sigma}^l_z\right)\hat{\sigma}^j_+. \tag{66}$$

In Eq. (66), one can verify that $\hat{f}^\dagger_j$ and $\hat{f}_j$ satisfy the previously mentioned anticommutation relations, while $\hat{\sigma}^j_\mu$ satisfies the previously shown commutation relations; see Somma *et al.* (2003), Aspuru-Guzik *et al.* (2005), Seeley, Richard, and Love (2012), Tranter *et al.* (2015), and Tranter *et al.* (2018) and the respective original references for details and comparisons regarding the other transformations.

Bosons.—Bosonic operators satisfy the commutation relations $[\hat{\tilde{b}}_i, \hat{\tilde{b}}_j] = 0, [\hat{\tilde{b}}_i, \hat{\tilde{b}}^\dagger_j] = \delta_{ij}$ in an infinite-dimensional Hilbert space. At first it seems impossible to simulate bosonic systems due to the nature of infinite dimensions. However, sometimes we are interested in studying some finite modes of excitations above the ground state. Hence, the use of the entire infinite-dimensional Hilbert space is unnecessary. In a finite-dimensional basis, the bosons $\hat{b}^\dagger_i, \hat{b}_i$ obey the following commutation relations (Batista and Ortiz, 2004):

$$[\hat{b}_i, \hat{b}_j] = 0, \quad [\hat{b}_i, \hat{b}^\dagger_j] = \delta_{ij}\left[1 - \frac{N_b + 1}{N_b!}(\hat{b}^\dagger_i)^{N_b}(\hat{b}_i)^{N_b}\right], \tag{67}$$

where $\hat{b}^\dagger_i\hat{b}_i|n_i\rangle = n_i|n_i\rangle$, with $n_i = 0, \ldots, N_b$, and N_b is the maximum truncated excitation number corresponding to the ith bosonic site or mode. A direct consequence is that one can then write the creation and annihilation operators as

$$\hat{b}^\dagger_i = \sum_{n=0}^{N_b - 1} \sqrt{n+1}\,|n+1\rangle\langle n| \tag{68}$$

and $\hat{b}_i$, which is the complex conjugate of $\hat{b}^\dagger_i$. There are infinite means to translate such truncated operators into the quantum computing language, the so-called Pauli words. A commonly used method is known as standard binary or compact encoding (Somma *et al.*, 2003; Veis *et al.*, 2016; McArdle, Mayorov *et al.*, 2019; Sawaya and Huh, 2019; Sawaya *et al.*, 2020), where $\{\alpha, \beta \in \mathbb{W}\}$ in $|\alpha\rangle\langle\beta|$ are now written in terms of binary strings. Using the following identities, $|0\rangle\langle 1| \equiv \hat{\sigma}_-; |1\rangle\langle 0| \equiv \hat{\sigma}_+; |0\rangle\langle 0| \equiv (I + \hat{\sigma}_z)/2; |1\rangle\langle 1| \equiv (I - \hat{\sigma}_z)/2$, Pauli words translation can be accomplished. Recently detailed studies on various encodings (binary, Gray, unary, block unary), have been studied and Gray code, in particular, is found to be resource efficient (in terms of number of qubits and two-qubit entangling gates) in simulating some specific bosonic and spin Hamiltonians (Sawaya *et al.*, 2020).

Anyons.—As previously discussed, we can now proceed to simulate more general particle statistics, in particular, hard-core anyons. Regarding the "hard-core" ones, we refer to Pauli's exclusion principle, where only zero or one particle can occupy a single mode. The anyonic operators $\hat{a}_i$ and $\hat{a}^\dagger_i$ obey the commutation relations $[\hat{a}_i, \hat{a}_j]_\theta = [\hat{a}^\dagger_i, \hat{a}^\dagger_j]_\theta = 0$, $[\hat{a}_i, \hat{a}^\dagger_j]_{-\theta} = \delta_{ij}(1 - (e^{-i\theta+1})\hat{n}_j)$, and $[\hat{n}_i, \hat{a}^\dagger_j] = \delta_{ij}\hat{a}^\dagger_j$, where $\hat{n}_j = \hat{a}^\dagger_j\hat{a}_j, [\hat{A}, \hat{B}]_\theta = \hat{A}\,\hat{B} - e^{i\theta}\hat{B}\,\hat{A}$, with $(i \le j)$ and $0 \le \theta < 2\pi$. Specifically, $\theta = \pi \bmod (2\pi)$ gives rise to canonical fermions, and $\theta = 0 \bmod (2\pi)$ would recover hard-core bosons. By simply applying the following isomorphic mapping between algebras (Somma *et al.*, 2003),

$$\hat{a}^\dagger_j = \prod_{i<j}\left(\frac{e^{-i\theta} + 1}{2} + \frac{e^{-i\theta} - 1}{2}\hat{\sigma}^i_z\right)\hat{\sigma}^j_+,$$

$$\hat{a}_j = \prod_{i<j}\left(\frac{e^{i\theta} + 1}{2} + \frac{e^{i\theta} - 1}{2}\hat{\sigma}^i_z\right)\hat{\sigma}^j_-, \qquad \hat{n}_j = \frac{1 + \hat{\sigma}^j_z}{2}, \tag{69}$$

we obtain Pauli words for the quantum computer. This mapping also ensures the previously shown anyonic algebra.

2. Constructing electronic Hamiltonians

The electronic structure problem is one of the most prominent tasks within VQAs [see the reviews by Cao *et al.* (2019) and McArdle *et al.* (2020)] and was the pioneering task for VQE (Peruzzo *et al.*, 2014; McClean *et al.*, 2016). In this section, we illustrate how the original continuous many-electron problem can be discretized to a second-quantized formulation that can itself be encoded in qubits by the techniques introduced at the beginning of Sec. VI.A. This encoded qubit systems define then the central problem of the VQAs further described in Sec. VI.A.3.

The electronic structure problem aims to approximate eigenfunctions of electronic Hamiltonians

$$H_\mathrm{e} = \mathcal{T}_\mathrm{e} + \mathcal{V}_\mathrm{ee} + \mathcal{V}_\mathrm{ext}, \tag{70}$$

describing a system of N_e electrons through their accumulated kinetic energies $\mathcal{T}_\mathrm{e} = -(1/2)\sum_{k=1}^{N_\mathrm{e}}\Delta_{\boldsymbol{r}_k}$, the electronic Coulomb repulsion $\mathcal{V}_\mathrm{ee} = \sum_{k\neq l} V_\mathrm{ee}(\boldsymbol{r}_k - \boldsymbol{r}_l) = \sum_{k\neq l} 1/|\boldsymbol{r}_k - \boldsymbol{r}_l|$, and an external potential $\mathcal{V}_\mathrm{ext} = \sum_{k=1}^{N_\mathrm{e}} V_\mathrm{ext}(\boldsymbol{r}_k)$ that is usually given by the accumulated Coulomb potential of nuclear point charges. If the external potential is not explicitly spin dependent, the electronic Hamiltonian acts only on the spatial coordinates $\boldsymbol{r}_k \in \mathbb{R}^3$ of the electrons and, to ensure proper electronic wave functions, the fermionic antisymmetry is achieved via restrictions in the Hilbert-space; see Rohwedder (2010), Yserentant (2010), Herbst (2018), and Kottmann (2018) for the direct construction and discretization of these continuous Hilbert spaces.

A more compact but formally equivalent definition is offered through second quantization by introducing the abstract anticommuting field operators $\hat{\psi}^\dagger(x)$ and $\hat{\psi}(x)$ that create and annihilate electrons at spin coordinate $x_k \in \mathbb{R}^3 \times \{\pm 1/2\}$ (Jordan and Klein, 1927; Jørgensen, 2012; Surján, 2012). The electronic Hamiltonian can then be written as

$$H_\mathrm{e} = \int dx\, \hat{\psi}^\dagger(x)[T(x) + V_\mathrm{ext}(x)]\hat{\psi}(x) + \int dx\, dy \hat{\psi}^\dagger(x)\hat{\psi}^\dagger(y)V_\mathrm{ee}(x-y)\hat{\psi}(y)\hat{\psi}(x), \quad (71)$$

where the potential operators still act only on the spatial part of the spin components. Although direct approaches for real-space grids are possible (Kivlichan *et al.*, 2017; Kottmann and Bischoff, 2017; Kottmann, 2018; Kunitsa and Hirata, 2020), the majority of VQAs employ a fixed set of three-dimensional functions (so-called orbitals) to capture the spatial part of the electronic Hilbert space. The orbitals are usually determined by solving a mean-field (Hartree-Fock) problem within a set of globally defined atomic orbitals. Alternatives to the standard representation include direct determination of system adapted orbitals (Kottmann *et al.*, 2021), compactification of basis sets through intrinsic atomic orbitals (Barison, Galli, and Motta, 2020), and optimized virtual orbitals represented by plane waves (Bylaska *et al.*, 2020).

For the formal description of the discretized second-quantized electronic Hamiltonian, the origin of the orbitals is not important as long as they form an orthonormal set of $H^1(\mathbb{R}^3)$ functions. Using such a set of spatial orbitals we can formally expand the field operators in the corresponding spin orbitals

$$\hat{\psi}^\dagger(x) = \sum_k \phi_k^*(x) f_k^\dagger, \qquad \hat{\psi}(x) = \sum_k \phi_k(x) f_k, \quad (72)$$

where $f_k^\dagger$ and f_k are fermionic creation and annihilation operators obeying the anticommutation relations shown in Sec. VI.A.1. Using the expansion from Eq. (71) leads to the following common discretized second-quantized Hamiltonian:

$$H_\mathrm{e} = \sum_{kl} h_{kl} f_k^\dagger f_l + \sum_{klmn} g_{klmn} f_k^\dagger f_l^\dagger f_n f_m, \quad (73)$$

with the molecular integrals (Fermann and Valeev, 2020)

$$h_{kl} = \int \phi_k^*(x)[T(x) + V_\mathrm{ext}(x)]\phi_l(x)dx,$$
$$g_{klmn} = \int \phi_k^*(x)\phi_l^*(y)V_\mathrm{ee}(x-y)\phi_m^*(x)\phi_n^*(y)dxdy. \quad (74)$$

Note that the indices of the two-body integrals are denoted in the standard Dirac notation $g_{klmn} \equiv \langle kl|V_\mathrm{ee}|mn\rangle$, but other notations such as Mulliken $(km|lm) = \langle kl|V_\mathrm{ee}|mn\rangle$ are sometimes used. Generally speaking, an arbitrary set of spatial orbitals that can in principle be any set of orthonormal $H^1(\mathbb{R}^3)$ functions defines a discretized second-quantized Hamiltonian as in Eq. (73) over the corresponding molecular integrals [Eq. (74)]. This discretized Hamiltonian can then be encoded in a qubit Hamiltonian by the corresponding fermion to qubit mappings that are discussed in Sec. VI.A.

3. Variational quantum eigensolver

Estimating the ground state and its energy of Hamiltonians is an important problem in physics, which has numerous applications ranging from solid-state physics to combinatorial optimization; see Sec. VI.C. While this problem is in general QMA hard and even quantum computers are not expected to be able to efficiently solve it in general (Kempe, Kitaev, and Regev, 2006), there is hope that approximate solutions of the ground state could be found faster and for larger system sizes than what is possible with classical computers.

To this end, VQE (Peruzzo *et al.*, 2014; McClean *et al.*, 2016) has been proposed to find the ground state of a Hamiltonian H in a manner that is suited for NISQ devices (Wecker, Hastings, and Troyer, 2015). Following the concept introduced in Secs. II.A and II.B, a parametrized circuit $U(\boldsymbol{\theta})$ is minimized with respect to the objective function, which in general is the expectation value of the energy of the Hamiltonian in Eq. (1). The approximated ground state is given by the quantum state $|\psi_\mathrm{min}\rangle = U(\boldsymbol{\theta}_\mathrm{min})|0\rangle$, which minimizes the energy $\min_{\boldsymbol{\theta}}\langle H_{\boldsymbol{\theta}}\rangle \geq E_\mathrm{g}$ upper bounded by the true ground state energy E_g as guaranteed by the Rayleigh-Ritz variational principle (Gould, 2012). VQE has been intensively studied in both theory and experiments, and various adaptions and extensions have been proposed, which we now discuss.

Self-verification.—Whether the variational quantum simulator has converged to an actual eigenstate of the Hamiltonian can be checked directly on the quantum processor by verifying that the variance of the energy $\mathrm{var} = \langle (H - \langle H\rangle)^2\rangle$ is zero. This has been demonstrated for solving a many-body Hamiltonian on eight qubits on an ion trap (Kokail *et al.*, 2019); see also Sec. VI.E.6.

Accelerated VQE.—A key computational effort in VQE lies in estimating the cost function, which is achieved by repeatedly running the circuit and taking measurements of the Pauli strings; see Sec. II.C. For a given desired additive error bounded by ϵ, it takes $O(1/\epsilon^2)$ number of samples. This can be improved by using the quantum phase estimation algorithm to estimate the expectation value, which takes only $O(\log(1/\epsilon))$ samples, however, at the cost of additional computation, which may be hard in the NISQ era. To leverage a trade-off between the advantages and disadvantages of the two methods, an accelerated version of VQE that interpolates between regular measurements and quantum phase estimation has been proposed (Wang, Higgott, and Brierley, 2019).

Measurement-based VQE.—Ferguson *et al.* (2020) presented two strategies to implement the VQE algorithm on a measurement-based quantum computer, an alternative quantum computing paradigm that uses entanglement as a resource and achieves the desired computation by performing particular sets of local measurements; see Briegel *et al.* (2009) for a review. They proposed a way to generate the needed variational state families using measurements on a highly entangled state and provide equivalence between the measurement- and gate-based schemes.

Reusing qubits in VQE.—A recent proposal suggested a VQE method that relies on fewer qubits by reusing some of them (Liu *et al.*, 2019). The core idea is to represent a virtual N-qubit state by $R + V < N$ physical qubits, where R qubits have to be reusable qubits; e.g., they can be measured and reinitialized during the circuit run-time. These intermittent measurements are possible on current ion-trap hardware (Pino *et al.*, 2020). The $R + V$ qubits are entangled by a PQC, and then R qubits are measured and the outcome is recorded. The R qubits are reinitialized to the $|0\rangle^{\otimes R}$ state and again entangled with the V other qubits by another PQC. This procedure is repeated until in total N qubits have been measured. The concept and expressiveness of this type of *Ansatz* is the same as tensor network methods such as the matrix product state one, which have been highly useful for the classical calculation of many-body problems and open up a way to perform quantum computing of many qubits on devices with a limited number of qubits.

Adiabatically assisted VQE.—The ground state of more challenging Hamiltonians can be difficult to find for standard VQE due to convergence to local minima instead of the global minima of the energy. To alleviate this, quantum annealing (see Sec. III.A) can be used to adiabatically assist the optimization procedure, as proposed for the adiabatically assisted VQE (Garcia-Saez and Latorre, 2018). This approach uses an objective function $O(s) = \langle 0|U^{\dagger}(\boldsymbol{\theta})H(s)U(\theta)|0\rangle$, where $H(s) = (1 - s)H_0 + sH_1$. Here H_0 is a Hamiltonian with an easily preparable ground state and the goal is to find the ground state of a Hamiltonian H_1. In this algorithm, VQE is run for multiple discrete steps s_n. One starts with $s_0 = 0$ and finds the minimal parameters $\boldsymbol{\theta}_0^*$ of the objective function $O(s_0)$. $\boldsymbol{\theta}_0^*$ is then used as an initial guess for the VQE for the next increasing step $s_1 = s_0 + \Delta s$ with the objective function $O(s_1)$. This procedure is repeated until $s = 1$ is reached. This approach eases the optimization task, as the initial Hamiltonian H_0 is a simple Hamiltonian with a ground state that can be easily found via optimization. For small steps Δs, the ground state of the Hamiltonian $H(s)$ and $H(s + \Delta s)$ will not differ too much, making the optimization task at every step less challenging than directly solving for $H(1)$. Previous works (McClean *et al.*, 2016) also suggested using adiabatically prepared states as initial states of a VQE algorithm; see Sec. II.B.

4. Variational quantum eigensolver for excited states

The methods of VQE have been extended to obtain excited states of a given Hamiltonian. Finding excited states or the spectrum of a Hamiltonian is an important objective in quantum chemistry and many-body physics. Various proposals have been put forward.

Folded spectrum method.—A straightforward way to calculate excited states is the folded spectrum method proposed by Peruzzo *et al.* (2014). To find an excited state of a Hamiltonian H with approximate energy λ, the previously defined VQE method is applied here to the objective function $C(\boldsymbol{\theta}) = \langle\cdot\rangle(H - \lambda)^2 U_{(\boldsymbol{\theta})}$. VQE will target the eigenstate with an energy that is closest to λ. This method requires approximate knowledge of the energy of the excited state that one wants to find, as well as an estimate of $\langle H^2\rangle$, which may require an excessively large number of measurements to be performed.

An extension of this method can also be used to find states that are constrained to a specific value of the conserved quantity of the problem, such as the total particle number or magnetization (Ryabinkin, Genin, and Izmaylov, 2019). Here one defines the objective function $C(\boldsymbol{\theta}) = \langle H\rangle_{\mathcal{U}(\boldsymbol{\theta})} + \sum_i \mu_i(\langle S_i\rangle_{\mathcal{U}(\boldsymbol{\theta})} - s_i)^2$, where S_i is the operator corresponding to the conserved quantity and s_i is the target value of that quantity. Note that this does not restrict the target space to be an eigenstate of S_i.

Orthogonally constrained VQE.—Excited states can be found by constraining the VQE objective function such that it penalizes the ground state (Higgott, Wang, and Brierley, 2019). One first finds an approximation to the ground state of Hamiltonian H via VQE with $\boldsymbol{\theta}_0 = \arg\min_{\boldsymbol{\theta}}\langle H\rangle_{\mathcal{U}(\boldsymbol{\theta})}$ and the approximated ground state $|\psi(\boldsymbol{\theta}_0)\rangle = \mathcal{U}(\boldsymbol{\theta}_0)|0\rangle$. One then uses this information to formally project out the approximate ground state to find the next highest excited state. One defines the Hamiltonian $H_1 = H + a|\psi(\boldsymbol{\theta}_0)\rangle\langle\psi(\boldsymbol{\theta}_0)|$ with a given sufficiently large positive parameter a. The ground state of H_1 then corresponds to the first excited state of H and can be found with a VQE. This procedure can be repeated to find higher excited states up to any order by sequentially accumulating the projector terms of all states found. The Hamiltonian for the kth excited state is then given by $H_k = H + \sum_i^{k-1} a_i|\psi(\boldsymbol{\theta}_i)\rangle\langle\psi(\boldsymbol{\theta}_i)|$. Combined with the unitary coupled-cluster *Ansatz*, the orthogonally constrained VQE can find excited states of small molecules (Higgott, Wang, and Brierley, 2019; Lee *et al.*, 2019). It was further extended for adaptive circuit construction (Kottmann, Anand, and Aspuru-Guzik, 2021) and imaginary time evolution (Jones *et al.*, 2019).

The projector term requires one to calculate the overlap $|\langle\psi(\theta)|\psi(\theta_0)\rangle|$, which can be achieved using the SWAP test, by applying the inverse of the circuit that generated the ground state $|\langle 0|U^{\dagger}(\theta)U(\theta_0)|0\rangle|^2$ or randomized measurements (Elben *et al.*, 2020). An alternative approach that relies on a discriminator circuit that is trained in parallel to distinguish between the excited state to be learned and previously found lower-lying states has been proposed (Tilly *et al.*, 2020) and demonstrated on a small model system. Scalable proposals still remain an open research question. Since the projector term requires not the overlap itself but rather the absolute square of it, it can be computed with the help of Eq. (5) by computing the fidelity of the current trial state with the previously found states (Lee *et al.*, 2019; Kottmann *et al.*, 2021).

Subspace expansion.—The subspace expansion method was introduced in Sec. V.A.3 for error mitigation. This method can be also used to find excited states (McClean *et al.*, 2017), and it was demonstrated for a small molecule by Colless *et al.*

(2018). After finding the ground state of a Hamiltonian H with VQE, one follows the steps that were detailed in Sec. V.A.3. One expands the prepared quantum state with different appropriate operators that match the low-energy excitations of H and generates a set of states that span the low-energy subspace. Overlaps between the states are measured, which are then used to solve a generalized eigenvalue problem on a classical computer. The eigenvalues and eigenstates give the excited states of the Hamiltonian. For quantum chemistry problems, the subspace expansion method was also proposed for including dynamical correlations to ground states over external corrections (Takeshita *et al.*, 2020), in the spirit of classical-quantum chemistry methods like CAS-CI (Roos, Taylor, and Sigbahn, 1980).

An alternative approach, expansion in the subspace, can also be accomplished by real-time evolving a reference state and picking states at different evolution times as a basis for expansion (Stair, Huang, and Evangelista, 2020). This is motivated by the fact that the time evolution can be seen as an approximate Krylov expansion of the quantum state. One then proceeds to solve the generalized eigenvalue problem to find eigenstates and eigenvalues of the Hamiltonian.

Subspace-search VQE and state-averaged VQE.—The core idea of a subspace-search VQE (SSVQE) (Nakanishi, Mitarai, and Fujii, 2019) or state-averaged VQE (Arimitsu *et al.*, 2021; Yalouz *et al.*, 2021) is to minimize the energy of a PQC $U(\boldsymbol{\theta})$ over a set of orthogonal quantum states. The goal is to find the kth eigenstates with the lowest eigenenergy of a Hamiltonian H. In the weighted SSVQE the cost function is

$$L(\boldsymbol{\theta}) = \sum_{j=1}^{k} w_j \langle \varphi_j | U^\dagger(\theta) H U(\boldsymbol{\theta}) | \varphi_j \rangle, \tag{75}$$

where $\{|\varphi_j\rangle\}_{j=0}^{k}$ is a set of k easily preparable mutually orthogonal quantum states (with $\langle \varphi_i | \varphi_j \rangle = \delta_{i,j}$) and $\{w_j\}_j$ are positive real numbers with $w_i > w_j$ for $i < j$. Minimizing $\boldsymbol{\theta}^* = \arg\ \min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$ to its global minimum gives us the ground state and excited states $|\psi_j\rangle = U(\boldsymbol{\theta})|\varphi_j\rangle$, where $j = 1$ is the ground state and $j > 1$ are the excited states sorted in ascending order. This algorithm gives all k eigenstates in a single optimization routine. Note, however, that the more states there are to be optimized, the more complex the optimization landscape and the effort to minimize become. An alternative formulation of the algorithm to specifically find the kth lowest eigenstate is the unweighted SSVQE. Here one minimizes $L_1(\boldsymbol{\theta}) = \sum_{j=1}^{k} \langle \varphi_j | U^\dagger(\boldsymbol{\theta}) H U(\boldsymbol{\theta}) | \varphi_j \rangle$. However, because of the absence of weight, the found states $|\psi'_j\rangle = U(\boldsymbol{\theta}^*)|\varphi_j\rangle$ for minimal $\boldsymbol{\theta}^*$ are not proper eigenstates of H but are instead superposition states that span the subspace of the k lowest energies. As a final step in finding the kth eigenstate, one fixes $\boldsymbol{\theta} = \theta^*$ to its minimized value and then maximizes $\boldsymbol{\phi}^* = \max_{\boldsymbol{\phi}} L_2(\boldsymbol{\phi})$, with $L_2(\boldsymbol{\phi}) = \sum_{j=1}^{k} \langle \varphi_j | V^\dagger(\boldsymbol{\phi}) U^\dagger(\theta^*) H U(\boldsymbol{\theta}^*) V(\boldsymbol{\phi}) | \varphi_j \rangle$ and $V(\boldsymbol{\phi})$ a unitary that acts only on the Hilbert space of the k lowest eigenstates. For the maximized $\boldsymbol{\phi}^*$, the kth lowest eigenstate is given by $|\psi_k\rangle = U(\boldsymbol{\theta}^*) V(\boldsymbol{\phi}^*) |\varphi_k\rangle$. Besides general applications that involve excited states, state-averaged orbital-optimized VQEs were proposed to treat chemical systems that require a "democratic description of multiple states," as necessary in the vicinity of conical intersections (Yalouz *et al.*, 2021). Here "democratic description" corresponds to treating degenerate or quasidegenerate states on the same footing.

Multistate contracted VQE.—This algorithm combines the nonweighted SSVQE with the subspace expansion to find the ground state and excited states (Parrish, Hohenstein *et al.*, 2019). One first runs the nonweighted SSVQE routine to find the unitary $U(\boldsymbol{\theta}^*)$ to find k states that span the subspace of the k smallest eigenvalues $|\psi'_j\rangle = U(\boldsymbol{\theta}^*)|\varphi_j\rangle$. To find the correct eigenstates, one then runs the subspace expansion and measures the overlap matrix $H_{ij} = \langle \psi'_i | H | \psi'_j \rangle$ and diagonalizes it to find estimates of the k lowest eigenenergies and eigenstates.

Fourier transform of evolution.—Recent experiments have determined the spectra of molecular and many-body Hamiltonians using superconducting processors (Roushan *et al.*, 2017; Aleiner *et al.*, 2020; Quantum *et al.*, 2020). A particular method to determine the eigenenergies of Hamiltonians via Fourier transforming the dynamics of observables was applied by Roushan *et al.* (2017) and Aleiner *et al.* (2020). The idea is to prepare a Fock state that has overlap with the eigenstates whose eigenvalues one wants to calculate. The Fock state then evolves in time with the Hamiltonian and specific observables are measured over a range of time. The Fourier transform of the time evolution of the observables can be used to deduce the eigenenergies of the Hamiltonian.

Witness-assisted variational eigenspectra solver (WAVES).—WAVES's core idea is to use a single reference qubit as an eigenstate witness to variationally find the ground state and excited states (Santagati *et al.*, 2018). A variational *Ansatz* applied to a reference state is chosen. The time-evolution operator $U(t) = \exp(-iHt)$ is then evolved on the *Ansatz* state as a control unitary $CU(t)$, with the control being the single qubit in a superposition state. Full tomography is then performed on the single qubit to read out its von Neumann entropy. If the variational state is an eigenstate of the Hamiltonian H, then the entropy is zero. Further, the energy of the state can be estimated from the state of the qubit as well. The *Ansatz* is variationally updated using the information from the qubit in an iterative fashion until the ground state is found. Excited states can be found by applying an appropriate excitation operator on the found ground state and then variationally minimizing the von Neumann entropy of the qubit. As the last step, the iterative phase estimation algorithm is used to further improve the accuracy of the excited state as well as to determine its eigenvalue. This method requires implementation of a controlled time-evolution operator similar to nonvariational proposals (Jensen *et al.*, 2020), which are considered challenging for larger systems on NISQ devices.

5. Hamiltonian simulation

A major application for quantum computers is the simulation of the dynamics of Hamiltonians for problems such as many-body physics and chemistry. One standard approach for quantum simulation of Hamiltonians is based on the Trotter-Suzuki expansion from Eq. (14), where the evolving unitary is split up into small discrete time steps of efficiently implementable unitaries, which can be run on the quantum computer. Naturally, the depth of the quantum circuit increases polynomially with the desired time to be evolved and the target accuracy, which may not be feasible on NISQ devices without access to error correction. The relevant algorithms are reviewed in the following. We remark that some necessary tools to simulate many-body interaction Hamiltonians (Bravyi *et al.*, 2008; Menke *et al.*, 2019) have also been proposed.

Variational quantum simulator.—VQA has been proposed to solve dynamical problems in the NISQ era (Li and Benjamin, 2017). The core idea is to iteratively update an efficiently implementable variational quantum state $|\psi(\boldsymbol{\theta})\rangle$ with a new set of parameters $\boldsymbol{\theta} \to \boldsymbol{\theta}'$ such that it minimizes the error between the actual time evolution $\exp(-iH\delta t)|\psi(\boldsymbol{\theta})\rangle$ for a time step δt and the updated variational state $|\psi(\boldsymbol{\theta}')\rangle$. The rules to update the parameters $\boldsymbol{\theta}$ to solve the Schrödinger equation $id/dt|\psi(t)\rangle = H|\psi(t)\rangle$ can be found by using the variational McLachlan principle $\delta\|(d/dt + iH)|\psi(\boldsymbol{\theta})\rangle\| = 0$, with $\||\psi\rangle\| = \sqrt{\langle\psi|\psi\rangle}$, and demanding that $\boldsymbol{\theta}$ remains real valued. One finds a set of linear equations of motion $\boldsymbol{A}\dot{\boldsymbol{\theta}} = \boldsymbol{C}$ with

$$A_{i,j} = \mathrm{Re}[\partial_{\theta_i}\langle\psi(\boldsymbol{\theta})|\partial_{\theta_j}|\psi(\boldsymbol{\theta})\rangle],$$
$$C_i = \mathrm{Im}[\partial_{\theta_i}\langle\psi(\boldsymbol{\theta})|H|\psi(\boldsymbol{\theta})\rangle]. \quad (76)$$

At a given step of the iteration, one needs to measure the elements of $\boldsymbol{A}$ and $\boldsymbol{C}$ using the Hadamard test or the methods of Mitarai and Fujii (2019) (see Sec. II.C), and then update $\boldsymbol{\theta}$ with the solution of the linear equation of motion using a small time step δt. The solver can be combined with adaptive strategies to reduce the complexity of the *Ansatz* circuit (Yao, Gomes *et al.*, 2020; Zhang, Sun *et al.*, 2020).

VQS has been applied on the IBM Quantum processor to simulate energy transfer in molecules (Lee *et al.*, 2021), as well as to simulate a time-dependent Hamiltonian (Lau, Bharti *et al.*, 2021). A straightforward extension of the variational quantum simulator can be applied to solve the Schrödinger equation in imaginary time (McArdle, Jones *et al.*, 2019), for time-dependent problems (Yuan *et al.*, 2019), or for general linear differential equations (Endo, Sun *et al.*, 2020; Kubo *et al.*, 2020). Its implementation in open quantum systems (Yuan *et al.*, 2019; Endo, Sun *et al.*, 2020) is discussed in Sec. VI.A.7. Using the hardware-efficient structure of the PQC, it is possible to reduce the cost of measuring the **A** and **C** matrices (Benedetti, Fiorentini, and Lubasch, 2020). Alternatively, the projected–variational quantum dynamics method was proposed to bypass the measurement of aforementioned matrices (Otten, Cortes, and Gray, 2019; Barison, Vicentini, and Carleo, 2021). Here one variationally maximizes the fidelity between the PQC $|\psi(\boldsymbol{\theta})\rangle$ and the Trotter evolved state $\exp(-iH\delta t)|\psi(\boldsymbol{\theta}')\rangle$. The optimized PQC yields the state evolved by a time δt. This algorithm is then repeated to gain evolution for longer times. By appropriately choosing the evolution time δt, barren plateaus can be avoided (Haug and Kim, 2021b).

Subspace variational quantum simulator.—The subspace variational quantum simulator (SVQS) (Heya *et al.*, 2019) builds upon the idea of the SSVQE (Nakanishi, Mitarai, and Fujii, 2019), which was introduced in Sec. VI.A.4. The core idea is to rotate the initial state to be evolved onto the low-energy subspace found by the weighted SSVQE, evolve it in time within the subspace, and then apply the reverse mapping. The weighted SSVQE is first run by preparing k initial states $\{|\varphi_j\rangle = \sigma_j^x|0\rangle\}_{j=0}^k$, which are orthogonal with each other ($\langle\varphi_i|\varphi_j\rangle = \delta_{i,j}$) and lie in the computational subspace, as well as a PQC $U(\boldsymbol{\theta})$. Now, as in the weighted SSVQE, minimize Eq. (75). Then prepare an initial state $|\psi_{\mathrm{in}}\rangle$ to be evolved, which is encoded in the computational subspace by applying the Hermitian conjugate of the obtained circuit $U^\dagger(\boldsymbol{\theta})$. Here the evolution of the state in time is performed by applying single-qubit rotations on each qubit, $\mathcal{T}(t) = \otimes_j R_Z(-E_j t)$, where $\{E_j\}_j$ are the eigenenergies of the eigenstates $\{|E_j\rangle\}_j$ that were previously obtained using SSVQE. Finally, the state $\mathcal{T}(t)U^\dagger(\boldsymbol{\theta})|\psi_{\mathrm{in}}\rangle$ in the computational subspace is reverse mapped by applying $U(\boldsymbol{\theta})$, giving the evolved state

$$|\psi(t)\rangle = U(\boldsymbol{\theta})\mathcal{T}(t)U^\dagger(\boldsymbol{\theta})|\psi_{\mathrm{in}}\rangle. \quad (77)$$

This method has the key advantage that since the evolution is directly implemented as simple rotations in the computational subspace, the circuit depth is independent of the evolution time to be simulated. However, the initial SSVQE optimization can be difficult, especially when one considers the many eigenstates k.

Variational fast-forwarding.—Similar to the idea of the SVQS, variational fast-forwarding relies on the idea of evolving a quantum state in time $\exp(-iHt)$ within a diagonal subspace such that an enhanced evolution time can be achieved (Cirstoiu *et al.*, 2020). A circuit that implements a small time step of the desired evolution is first implemented as $V(\delta t) = \exp(-iH\delta t)$. An approximate diagonal factorization of $V(\delta t)$ is then trained on a particularly structured variational circuit

$$U(\boldsymbol{\theta}, \boldsymbol{\gamma}, \delta t) = W(\boldsymbol{\theta})D(\boldsymbol{\gamma}, \delta t)W^\dagger(\boldsymbol{\theta}). \quad (78)$$

In Eq. (78) $D(\boldsymbol{\gamma}, \delta t)$ is composed of commuting unitaries and chosen to parametrize the eigenvalues of unitary $V(\delta t)$, whereas $W(\boldsymbol{\theta})$ represents its eigenvectors. The evolution to an arbitrary time $T = N\delta t$, where N is a given integer, is then found by fast-forwarding with $U(\boldsymbol{\theta}, \boldsymbol{\gamma}, N\delta t) = W(\boldsymbol{\theta})D^N(\boldsymbol{\gamma}, \delta t)W^\dagger(\boldsymbol{\theta})$. For the training of the variational *Ansatz*, the fidelity between $V(\delta t)$ and $U(\boldsymbol{\theta}, \boldsymbol{\gamma}, \delta t)$ is maximized by a quantum-classical feedback loop with a cost function that uses the local Hilbert-Schmidt test (Khatri *et al.*, 2019). As an alternative approach, it was proposed to diagonalize the Hamiltonian H instead of the unitary $V(\delta t)$, and fast-forward via $U(\boldsymbol{\theta}, \boldsymbol{\gamma}, T) = W(\boldsymbol{\theta})\exp[-iD(\boldsymbol{\gamma})T]W^\dagger(\boldsymbol{\theta})$ (Commeau *et al.*, 2020). Fast-forwarding can also be performed without a requirement to train via a feedback loop using the linear combination of states approach [Eq. (42)] (Lim *et al.*, 2021).

Quantum assisted simulator.—The VQS algorithm employs a classical-quantum feedback loop to update the parameters of the PQC. Until the classical processor has calculated its output, the classical-quantum feedback loop delays any use of the quantum device, slowing the algorithm on the current cloud computing framework. The VQS algorithm, as well as its VQE-based variant, i.e., SVQS, shares similarities with and most of the concerns faced by VQE, such as the barren plateau issue; see Sec. IV.A. Further, the VQS algorithm requires controlled unitaries, which makes it difficult to realize for current-term devices. To tackle the issues faced by VQS, the quantum assisted simulator (QAS) was suggested recently (Bharti and Haug, 2021b). The QAS algorithm does not need any classical-quantum feedback loop, can be parallelized, evades the barren plateau problem by construction, supplies a systematic approach to constructing the *Ansatz*, and does not require any complicated unitaries.

The QAS algorithm shares its approach with IQAE; see Sec. III.E. The *Ansatz* is given as a linear combination of states $|\phi(\boldsymbol{\alpha}(t))\rangle = \sum_{|\psi_i\rangle \in \mathbb{CS}_K} \alpha_i(t)|\psi_i\rangle$ [see Eq. (42)], with classical coefficients $\boldsymbol{\alpha}(t)$ for the *Ansatz* state $|\psi_i\rangle$, which can be systematically constructed; see Definition 3. The Hamiltonian H is given as a linear combination of unitaries; see Eq. (43). The QAS algorithm employs the Dirac-Frenkel principle to obtain the following classical evolution equation for $\boldsymbol{\alpha}(t)$:

$$\mathcal{E}\frac{\partial \boldsymbol{\alpha}(t)}{\partial t} = -\imath \mathcal{D}\boldsymbol{\alpha}(t). \tag{79}$$

In Eq. (79) $\mathcal{E}_{i,j} = \langle \psi_i|\psi_j\rangle$ and $\mathcal{D}_{i,j} = \sum_k \beta_k \langle \psi_i|U_k|\psi_j\rangle$ are overlap matrices that can be efficiently measured on a quantum computer; i.e., for H given as a combination of Pauli strings, the overlaps are a measurement of the Pauli strings.

Recently QAS was run on the IBM Quantum computer and showed superior performance relative to Trotter and VQS for a time-dependent Hamiltonian (Lau, Bharti *et al.*, 2021). A novel Hamiltonian simulation algorithm based on a truncated Taylor series was proposed recently (Lau, Haug *et al.*, 2021). The classical postprocessing in the aforementioned algorithm corresponds to a QCQP.

6. Quantum information scrambling and thermalization

Quantum information scrambling is a quantum phenomenon occurring when initially local states become increasingly nonlocal with the time evolution of the system. It can be analyzed by computing the so-called out-of-time-ordered correlation function and has strong implications in thermalization in closed quantum system dynamics. Recent experiments have been carried out to study this phenomenon in few-qubit trapped-ion devices and simulators (Landsman *et al.*, 2019; Joshi *et al.*, 2020), and in a 53 superconducting qubit processor (Mi *et al.*, 2021). The algorithms proposed are based on the well-known teleportation algorithm and use single- and two-qubit gates to reproduce the scrambling process.

In the context of VQAs, a variation of the VQE algorithm has been proposed to obtain the thermal evolution of quantum systems (Verdon, Marks *et al.*, 2019). Verdon, Marks *et al.* presented quantum Hamiltonian-based models (QHBMs), an extension of the VQA's PQC to mixed states instead of pure states. Within this approach, QHBMs are classically trained to learn a mixed state distribution as a function of the optimization parameters. A direct application of such a model is the variational quantum thermalizer (VQT), an algorithm whose goal is to prepare a fixed-temperature thermal state of a given Hamiltonian.

The limitations of using variational QML algorithms to learn a scrambling unitary were also studied by Holmes *et al.* (2020), who found trainability issues related to barren plateaus; see Sec. IV.A.

7. Simulating open quantum systems

In the following, we deal with the physics of open quantum systems (Huh *et al.*, 2014), which are well described by the Lindblad master equation from Eq. (52). By sampling from a mixture of pure state trajectories evolved using a non-Hermitian Hamiltonian and random quantum jumps, one recovers the Lindblad dynamics.

Trotter simulation of open systems.—NISQ quantum hardware can be used to directly simulate the dynamics of small-scale open systems by using ancillae combined with measurements in the spirit of the quantum jump method (Hu, Xia, and Kais, 2020; Koppenhöfer, Bruder, and Roulet, 2020). Here the unitary part of the dynamics is implemented via a Suzuki-Trotter decomposition; see Sec. II.B. The nonunitary part of the dynamics that encodes the interaction with the external degrees of freedom is simulated by entangling the circuit with ancillae and subsequently measuring them. For every time step of the dynamics, a new set of ancilla qubits has to be provided. Current quantum computers based on superconducting circuits do not allow one to measure and reuse qubits, thus requiring a linear increase in the number of qubits with every time step. Further, in general the circuit depth scales polynomially with simulation time.

Generalized variational quantum simulator.—Endo, Sun *et al.* (2020) extended the VQS algorithm to simulate the method of quantum jumps in a variational setting. They implemented the algorithm for 2D Ising Hamiltonians for six qubits and observed a dissipation-induced phase transition. Yuan *et al.* (2019) extended VQS to mixed states and simulated the Lindblad dynamics fully without the need of stochastic sampling. The idea is to write the density matrix as $\rho = \rho(\boldsymbol{\theta}(t))$ and simulate the evolution of ρ via evolution of the parameters $\boldsymbol{\theta}(t)$. One can reexpress Eq. (52) as $(d/dt)\rho = \sum_i g_i S_i \rho T_i^\dagger$, where S_i and T_i are unitaries and g_i are coefficients. Using the Dirac-Frenkel equation, the evolution of parameters is given by

$$\sum_j M_{i,j}\dot{\theta}_j = V_i, \tag{80}$$

$$M_{i,j} = \mathrm{Tr}(\{\partial_i \rho[\boldsymbol{\theta}(t)]\}^\dagger \partial_j \rho[\boldsymbol{\theta}(t)]), \tag{81}$$

$$V_i = \mathrm{Tr}\left[\{\partial_i \rho[\boldsymbol{\theta}(t)]\}^\dagger \sum_j g_j S_j \rho T_j^\dagger\right]. \tag{82}$$

This method can also be extended to deep quantum neural network–type *Ansätze* (Liu, Duan, and Deng, 2020).

These algorithms, however, suffer from the canonical drawbacks of the VQS algorithm, such as the requirement of a feedback loop, trainability issues, and the necessity of controlled unitaries.

Generalized quantum assisted simulators.—Recently the generalized quantum assisted simulator (Haug and Bharti, 2020) was proposed as an extension of the quantum assisted simulator to tackle the previously mentioned issues; see Sec. VI.A.5. Instead of using a density matrix, the generalized quantum assisted simulator algorithm introduced the concept of a "hybrid density matrix"

$$\hat{\rho} = \sum_{k,l} \beta_{k,l} |\psi_k\rangle\langle\psi_l| \tag{83}$$

for $\beta_{k,l} \in \mathbb{C}$, where $|\psi_l\rangle$ are chosen from the set of cumulative K moment states; see Definition 3. A classical device stores the coefficients $\boldsymbol{\beta}$ and the quantum states correspond to a certain quantum register. A hybrid density matrix is a valid density matrix if $\mathrm{Tr}(\hat{\rho}) = 1$ and $\hat{\rho} \succeq 0$. Note that the normalization condition is fulfilled when $\mathrm{Tr}(\hat{\rho}) = \mathrm{Tr}(\boldsymbol{\beta}\mathcal{E}) = 1$, where $\mathcal{E}_{k,l} = \langle\psi_k|\psi_l\rangle$. Using the Dirac-Frenkel principle, the simulation of open system dynamics for the hybrid density matrix is given by

$$\begin{aligned}\mathcal{E}\frac{d}{dt}\boldsymbol{\beta}(t)\mathcal{E} = &-\imath[\mathcal{D}\boldsymbol{\beta}(t)\mathcal{E} - \mathcal{E}\boldsymbol{\beta}(t)\mathcal{D}] \\ &+ \sum_{n=1}^{K} \gamma_n [\mathcal{R}_n \boldsymbol{\beta}(t) \mathcal{R}_n^\dagger - \tfrac{1}{2}\mathcal{F}_n \boldsymbol{\beta}(t)\mathcal{E} - \tfrac{1}{2}\mathcal{E}\boldsymbol{\beta}(t)\mathcal{F}_n],\end{aligned} \tag{84}$$

where $\mathcal{D}_{k,l} = \langle\psi_k|H|\psi_l\rangle$, $\mathcal{R}^n_{k,l} = \langle\psi_k|L_n|\psi_l\rangle$, and $\mathcal{F}^n_{k,l} = \langle\psi_k|L_n^\dagger L_n|\psi_l\rangle$. For a given choice of *Ansatz*, the quantum computers have to compute the overlap matrices only as measurements of Pauli strings. The classical computer then uses this information to simulate the dynamics. There is no quantum-classical feedback loop, which on the currently available quantum computers can speed up the computations substantially.

8. Nonequilibrium steady state

Unlike in Sec. VI.A.7, we are concerned here with the physics of an open quantum system that is out of equilibrium in nature, which is common in designing devices for molecular-scale electronics (Xiang *et al.*, 2016), excitonic transport (Kyaw *et al.*, 2017) as well as quantum thermodynamics (Vinjanampathy and Anders, 2016). By "out of equilibrium" we mean that a quantum system and bath are constantly driven by external forces such as voltage differences, during which the composite particles of the system and bath are also interacting with each other.

Notice that the method used in Sec. VI.A.7 would also lead to extremely high-dimensional matrices in the Lindblad-like master equation approach $d\hat{\rho}/dt = \hat{\hat{\mathcal{L}}}\hat{\rho}$ [see Sec. V.A.1, Eq. (52)], and it is deemed impossible to capture all the degrees of freedom involved. However, one may relax some of the constraints involved in the problem setup, say, time-independent dissipation and noninteraction among particles with a small system size. The steady-state density matrix of a quantum system $\hat{\rho}_{\mathrm{SS}}$ at the limit $t \to \infty$ is then given by solving

$$\hat{\hat{\mathcal{L}}}|\hat{\rho}_{\mathrm{SS}}\rangle = 0, \tag{85}$$

or equivalently $\hat{\hat{\mathcal{L}}}^\dagger \hat{\hat{\mathcal{L}}}|\hat{\rho}_{\mathrm{SS}}\rangle = 0$. A recent study (Yoshioka *et al.*, 2020) showed that with ancilla qubits the previously mentioned non-Hermitian superoperator $\hat{\hat{\mathcal{L}}}$ can be simulated. The main idea is to map the density matrix of N qubits onto a vector of twice the number of qubits $2N$ as follows:

$$\hat{\rho} = \sum_{ij} \rho_{ij} |i\rangle\langle j| \to |\hat{\rho}\rangle = \sum_{ij} \frac{\rho_{ij}}{C} |i\rangle_P |j\rangle_A, \tag{86}$$

where $C = \sqrt{\sum_{ij} |\rho_{ij}|^2}$. By using a digital quantum computer and the variational approach, one iteratively minimizes the expectation value of a parametrized density matrix $|\rho_{\boldsymbol{\theta}}\rangle = U(\boldsymbol{\theta})$ with $\min_{\boldsymbol{\theta}} \langle 0|^{\otimes 2N} U^\dagger(\boldsymbol{\theta}) \hat{\hat{\mathcal{L}}}^\dagger \hat{\hat{\mathcal{L}}} U(\boldsymbol{\theta}) |0\rangle^{\otimes 2N}$. A drawback of this approach is that measuring expectation values from the parametrized density matrix directly is difficult and thus requires an additional transformation.

Beyond the Lindblad master equation, to capture and describe truly the out-of-equilibrium processes, the nonequilibrium Green's function (NEGF) formalism (Dalla Torre *et al.*, 2013; Stefanucci and Van Leeuwen, 2013; Sieberer, Buchhold, and Diehl, 2016) is commonly used. These existing Green's function techniques are complicated to solve. Many assumptions need to be made in order to have a closed form and do some calculations. In particular, it requires that the interaction among particles is weak such that one does not need to find higher-order Feynman diagrams in finding the self-energy functional.

Since some of the existing quantum algorithms provide promising speedup relative to classical ones, one may wonder whether to use quantum algorithms to solve the NEGF, with a strategy of leaving classically hard computational tasks to the quantum processor and feeding its output back to a classical computer, which could be done in a variational fashion. There are a number of proposals (Kreula, Clark, and Jaksch, 2016; Endo, Kurata, and Nakagawa, 2020; Jaderberg *et al.*, 2020) in the literature that undertake such a hybrid quantum-classical approach. However, these methods assume no interaction among composite particles. In a generic open quantum system in which many-body effects cannot be neglected, one wants to go beyond those assumptions. No quantum advantage of those near-term quantum algorithms over existing methods has yet been seen (Härtle, Benesch, and Thoss, 2008; Li, Petruccione, and Koch, 2016; Fitzpatrick *et al.*, 2017) for solving a nonequilibrium steady-state solution of an extremely complex physical setup such as vibrationally coupled electron transport with multiple electronic levels (Härtle, Benesch, and Thoss, 2008).

9. Gibbs state preparation

Finding the ground state of quantum Hamiltonians is known to be QMA hard. Under reasonable assumptions, preparing a Gibbs state corresponding to arbitrarily small temperatures is as challenging as the Hamiltonian ground state problem. Gibbs state preparation has applications in many areas, including quantum annealing, quantum semidefinite programming solvers, Boltzmann training, and simulations of equilibrium physics. For a Hamiltonian H, the Gibbs state at temperature T (with $k_B = 1$) is given by

$$\rho(T) = \frac{\exp(-H/T)}{\mathrm{Tr}[\exp(-H/T)]}. \tag{87}$$

Some of the approaches to prepare the Gibbs state are mentioned in the following.

(1) Starting with the d-dimensional maximally mixed state I_d/d, under imaginary time evolution for time τ, one gets the Gibbs state corresponding to temperature $T = 1/2\tau$ (Verstraete, Garcia-Ripoll, and Cirac, 2004).

(2) One can start with the maximally entangled state $|\xi\rangle_d = (1/\sqrt{d})\sum_j |j, j\rangle_{AB}$ of a system combined of two equally sized subsystems A and B and can evolve it under imaginary time evolution using the Hamiltonian $H \otimes I$. After tracing out system B, the state of system A at time τ is given by the Gibbs state corresponding to temperature $T = 1/2\tau$.

(3) The Gibbs state of a system is the density matrix that corresponds to a minimum of its free energy. Thus, one can variationally tune the parameters of a parametrized density matrix such that it leads to minimization of the free energy.

Recently a few NISQ algorithms for Gibbs state preparation were proposed that apply the aforementioned ideas. Yuan *et al.* (2019) used VQS-based imaginary time evolution to prepare the Gibbs state following the second approach. The first approach does not work in VQS-based imaginary time evolution. Chowdhury, Low, and Wiebe (2020) used the third approach to prepare Gibbs states. The aforementioned works require complicated controlled unitaries and a classical-quantum feedback loop. Haug and Bharti (2020) suggested QAS-based imaginary time evolution (see Sec. VI.A.5) to prepare the Gibbs state with either the first or second approach. The QAS approach does not require a classical-quantum feedback loop or any complicated controlled unitaries. Using random circuits as the initial state, Richter and Pal (2020) suggested an approach based on imaginary time evolution for preparing the Gibbs state.

10. Simulation of topological phases and phase transitions

NISQ devices can be used to study the ground states of quantum Hamiltonians for understanding topological phases and phase transitions. An important example is the one-dimensional cluster Ising Hamiltonian, describing a symmetry-protected topological phase of matter. The ground state of this Hamiltonian is the one-dimensional cluster state, which can be created by applying Hadamard gates to all qubits, followed by controlled-Zgates on each pair of neighboring qubits. State tomography and symmetry arguments were used to study the entanglement measures of this state and to highlight its topological nature (Choo *et al.*, 2018; Azses *et al.*, 2020). A modified algorithm was implemented to simulate an enlarged family of Hamiltonians and study the quantum phase transition between topological and topologically trivial phases of matter (Smith, Jobst *et al.*, 2019). NISQ devices were also used to simulate the dynamics of fundamental models of quantum magnetism (Smith, Kim *et al.*, 2019; Bassman *et al.*, 2020) and topological phases in one and two dimensions (Mei *et al.*, 2020).

11. Many-body ground state preparation

The preparation of nontrivial many-body quantum states is crucial for many applications in quantum metrology and quantum information processing (Kyaw, Li, and Kwek, 2014). QAOA has been used as a resource-efficient scheme for many-body quantum state preparation. In this context, the state $|\psi\rangle$ for a system with linear dimension L (L can refer to the number of spins in a 1D spin chain) is nontrivial if there is no local unitary circuit U with depth $\mathcal{O}(1)$ that can generate $|\psi\rangle$ from a product state $|\phi\rangle$: $|\psi\rangle = U|\phi\rangle$ (Ho and Hsieh, 2019). The GHZ state, which is an essential resource in several quantum metrology proposals (Dür *et al.*, 2014; Tóth and Apellaniz, 2014), is an example of a nontrivial quantum state due to its highly entangled nature and is the ground state of the 1D Ising Hamiltonian with periodic boundary conditions, i.e., $H_P = -\sum_{i=1}^{L} \hat{\sigma}_z^i \hat{\sigma}_z^{i+1}$.

Using QAOA, it has been shown that the GHZ state can be prepared efficiently with perfect fidelity using $p = L/2$, where p is the QAOA depth (Ho and Hsieh, 2019). Ho and Hsieh conjectured that the ground state of the 1D transverse-field Ising model, with L even and periodic boundary conditions, can be prepared perfectly at any point in the phase diagram using QAOA with $p = L/2$. The ground state of the antiferromagnetic Heisenberg model with open boundary conditions $H_P = \sum_{i=1}^{L-1} \hat{\boldsymbol{\sigma}}^i \cdot \hat{\boldsymbol{\sigma}}^{i+1}$, where $\hat{\boldsymbol{\sigma}}^i \equiv (\hat{\sigma}_x^i, \hat{\sigma}_y^i, \hat{\sigma}_z^i)$, has also been prepared with near perfect fidelity using QAOA. Using a long-range 1D Ising Hamiltonian $H_P = -\sum_{i<j} J_{ij} \hat{\sigma}_z{}^i \hat{\sigma}_z^j$, where $J_{ij} = J_0/|i-j|^\alpha$, QAOA can achieve an ultrafast preparation of a GHZ state with a circuit depth of $\mathcal{O}(1)$ (for $\alpha = 0$) (Ho, Jonay, and Hsieh, 2019). This result was generalized as follows by Wauters, Mbeng, and Santoro (2020), who showed that QAOA can prepare the ground states of the fully connected ferromagnetic q-spin model (note that q is used here instead of the conventional p in order to avoid confusion with the QAOA depth p):

$$H = -\frac{1}{N^{q-1}}\left(\sum_{i=1}^{N} \hat{\sigma}_z^i\right)^q - h\left(\sum_{i=1}^{N} \hat{\sigma}_x^i\right), \tag{88}$$

with resources scaling polynomially with the number of spins N. Since the system can encounter a first-order phase transition where the spectral gap becomes small, QAOA greatly outperforms quantum annealing in this instance since an exponentially long annealing time is needed.

12. Quantum autoencoder

The quantum autoencoder (QAE) (Romero, Olson, and Aspuru-Guzik, 2017) is a VQA for the compression of data on a quantum computer. It finds a new data state representation that requires fewer qubits than the data were originally defined upon. This new encoding is said to be a representation in the *latent space*. The process of transforming the data into the latent space is referred to as *encoding*, whereas the converse, the transformation of states in the latent space back into the original, is known as *decoding*.

Training a QAE requires the minimization of an objective defined over several related quantum states. For a set of n-qubit states $\{|\psi\rangle_i\}$, the goal of the QAE is to find a unitary circuit $\mathcal{E}(\boldsymbol{\theta})$ that accomplishes the following transformation:

$$E{:}\mathcal{H}^n \to \mathcal{H}^k \otimes \mathcal{H}^{n-k}|E|\psi\rangle_i = |\phi\rangle_i \otimes |0\rangle^{\otimes(n-k)}, \qquad (89)$$

where k is the dimension of the latent space. Thus, the application of a perfectly trained autoencoder to any state of the relevant set yields a product state that consists of the transformed state on k qubits with a $(n-k)$-qubit "trash" state. In principle, the trashed state could be any state, but the all-zero state is chosen for simplicity.

The loss function of the QAE may be defined in several ways. It is a fidelity loss function (see Sec. II.A) in which minimization is performed by increasing the overlap between a partial measurement of the state resulting from the application of the encoder and a known state. The most practical definition for training the autoencoder, called "trash training," uses as its objective the overlap between the trash qubits and the $|0\rangle^{\otimes(n-k)}$ state. Formulated in the density matrix picture, the objective of minimization is

$$O = -\mathrm{Tr}(I^{\otimes k} \otimes |0\rangle\langle 0|^{\otimes(n-k)}\rho_i), \qquad (90)$$

where $\rho_i = \sum_i p_i|\psi_i\rangle\langle\psi_i|$, with all states in the set generally equally weighted.

The QAE can be trained by training only the encoding circuit due to the unitarity of the encoder. The decoding operation is achieved using the complex conjugate of the encoder circuit. Improvements in the encoding results translate to improvements in the decoder, a boon not realized by classical autoencoders.

After the successful training of a QAE, the encoder and decoder circuits may be used for data transformation in further algorithms. Here processing the data in the latent space, in which the data are represented more densely, may be beneficial for further applications.

A data reuploading strategy to construct a QAE encoder was presented by Bravo-Prieto (2020), who trained the so-called enhanced feature quantum autoencoder to compress the ground state of the 1D Ising model as a function of the external field and samples of handwritten digits. The QAE has also been deployed experimentally for the compression of qutrits on a photonic device (Pepper, Tischler, and Pryde, 2019). Small states were experimentally compressed without loss on photonic devices by C.-J. Huang *et al.* (2020). Bondarenko and Feldmann (2020) designed a QAE capable of denoising entangled quantum states such as GHZ or W states that are subject to spin-flip errors and random unitary noise.

13. Quantum computer-aided design

Two recent proposals used the computing power of NISQ devices to improve the NISQ processors themselves. Here techniques were developed to simulate quantum hardware on a quantum computer (Kottmann *et al.*, 2020; Kyaw *et al.*, 2021). They established the paradigm of "quantum computer-aided design," indicating that classically intractable simulations of quantum hardware properties can be performed on a quantum computer and thereby improving the prediction of device performance and reducing experimental testing cycles.

In the first approach, optical path modes are mapped to sets of qubits and quantum optical elements are mapped to digital quantum circuits for simulating photonic setups (Kottmann *et al.*, 2020). The framework is used to simulate a boson sampling experiment and the optimization of a setup for preparing high-dimensional multipartite entangled states.

The second proposal introduces quantum simulation techniques for superconducting circuit hardware (Kyaw *et al.*, 2021). In this work, a circuit module consisting of coupled transmon qubits is designed. The corresponding superconducting circuit Hamiltonian, which is written on a basis of multilevel operators, is efficiently mapped to a set of data qubits (Sawaya *et al.*, 2020). Simulations of a multilevel extension to the VQE algorithm (Higgott, Wang, and Brierley, 2019) are used to determine the spectrum of the superconducting circuit. The resulting states and eigenenergies are directly related to experimentally relevant device characteristics and can be used to seed the simulation of time evolution.

Device and setup design is a key challenge for improving and scaling quantum systems. Therefore, digital quantum simulation of quantum processors will be a relevant application for NISQ quantum computers when classical resources become too small to capture the relevant Hilbert space of the hardware.

B. Machine learning

Machine learning aims to enable a computer to act without being explicitly programmed to do so. As per Mitchell (1997), given a certain class of tasks $\mathcal{T}$ and performance metric $\mathcal{P}$ a computer program is said to learn from experience $\mathcal{E}$ if

$$\mathcal{P}(\mathcal{T}) \propto \mathcal{E}. \qquad (91)$$

In other words, its performance measured by $\mathcal{P}$ for task $\boldsymbol{\mathcal{T}}$ increases with $\mathcal{E}$.

Depending on the kind of experience that $\mathcal{E}$ is permitted to have during the learning process, the machine learning algorithms are classified into the following three categories:

(1) *Supervised learning*.—Given a function $y = f(x)$, the goal is to learn f so that it returns the label y for the unlabeled data x. A canonical example would be pictures of cats and monkeys, with the task to recognize the correct animal. Given training examples from the joint distribution $P(Y, X)$, the task of supervised learning is to infer the probability of a label y given sample data x, i.e., $P(Y = y|X = x)$.

(2) *Unsupervised learning*.—The data are provided without any label. The task is to recognize an underlying pattern in these data. Given access to several examples $x \in X$, the algorithm goal is to learn the probability distribution $P(X)$ or some important properties of the aforementioned distribution.

(3) *Reinforcement learning*.—In this case, neither data nor label is provided. The machine has to generate data and improve the aforementioned data generation process via the optimization of a given reward function. This is similar to how a human child learns to walk. If it fails, the output acts as a negative reward.

Machine learning has uncovered applications in physics such as Monte Carlo simulation (Huang and Wang, 2017; Liu *et al.*, 2017), many-body physics (Carleo and Troyer, 2017), phase transition (Wang, 2016), quantum foundations (Bharti, Haug, Vedral, and Kwek, 2019), and state tomography (Torlai *et al.*, 2018) For a meticulous review on machine learning for physics, see Dunjko and Briegel (2018), Carleo *et al.* (2019), and Bharti *et al.* (2020).

Most of the success in machine learning comes from the use of artificial neural networks, structures capable of learning sophisticated distributions that encompass multiple features that can be fine-tuned depending on the problem tackled. In that direction, there are several proposals to define a model for quantum neural networks with different kinds of activation functions (Schuld, Sinayskiy, and Petruccione, 2014; Wan *et al.*, 2017; Torrontegui and García-Ripoll, 2019). For implementations of artificial neurons and artificial neural networks on the NISQ hardware, see Tacchino *et al.* (2019) and Tacchino, Barkoutsos *et al.* (2020).

The merger of quantum theory and machine learning has recently led birth to a new discipline known as QML. Both algorithms that deal classically with data from a quantum origin and quantum algorithms that process quantum and classical data are usually known as QML applications. However, in this review, we focus on those algorithms that process data quantum mechanically, in particular, those that use quantum algorithms that can be run on NISQ computers. For a QML review that focuses mainly on fault-tolerant quantum algorithms, see Biamonte *et al.* (2017). For a survey of quantum computational learning theory, see Arunachalam and de Wolf (2017). Analyses of QML from a classical ML perspective were given by Ciliberto *et al.* (2018) and Dunjko and Briegel (2018), and for near-term devices analyses were given by Perdomo-Ortiz *et al.* (2018), Benedetti, Lloyd *et al.* (2019), and Li and Deng (2021).

It might be surprising that a linear theory such as quantum physics can generate the nonlinearities that a machine learning model needs. However, the linearity of quantum mechanics comes from the dynamical part (quantum state evolution) and one can encounter multiple sources of nonlinearities arising from measurement, postselection, or coupling of the system to the environment. Quantum operations in the Hilbert space can also encode nonlinear behavior, as we later show with kernel methods.

In Secs. VI.B.1–VI.B.3, we present the quantum-mechanical analogs of the three previously defined machine learning categories. The algorithms discussed are listed in Table II of the Supplemental Material (824).

1. Supervised learning

The two prominent methods to perform a supervised learning classification task using a NISQ computer are *quantum kernel estimation* (Havlíček *et al.*, 2019; Kusumoto *et al.*, 2019; Schuld and Killoran, 2019; Huang *et al.*, 2021) and the *variational quantum classifier* (VQC) (Farhi and Neven, 2018; Mitarai *et al.*, 2018).

Classical kernel methods include well-known machine learning algorithms such as support vector machines (Cortes and Vapnik, 1995), principal component analysis, and Gaussian processes. The rich theoretical structure of kernel methods can be expanded to the quantum world by defining and working in the Hilbert space with the quantum equivalent of feature vectors (Schuld and Killoran, 2019). To that aim, one needs to modify and adapt the well-known theorems to work in a quantum feature space. For more details about classical kernel methods, see Hofmann, Schölkopf, and Smola (2008). A review on kernel methods in the context of QML was given by Mengoni and Di Pierro (2019). We directly describe the quantum versions of them next. The basics of supervised learning with quantum computers were presented by Schuld and Petruccione (2018).

Given an input set $\mathcal{X}$ and quantum Hilbert space $\mathcal{H}$, data $\boldsymbol{x} \in \mathcal{X}$ are encoded in a quantum state (*quantum feature vector*) $|\Phi(\boldsymbol{x})\rangle$ by means of the *quantum feature map*, i.e., $\Phi: \mathcal{X} \rightarrow \mathcal{H}$. The inner product of two quantum feature vectors defines a kernel

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv \langle \Phi(\boldsymbol{x}_i) | \Phi(\boldsymbol{x}_j) \rangle_{\mathcal{H}} \tag{92}$$

for $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{X}$. The inner product is defined in a Hilbert space by replacing the standard definition $\langle \cdot, \cdot \rangle$ with the Dirac brackets $\langle \cdot | \cdot \rangle$. For a map Φ, the *reproducing kernel Hilbert space* takes the form

$$\mathcal{R}_\phi = \{ f: \mathcal{X} \rightarrow \mathbb{C} | f(\boldsymbol{x}) = \langle w | \Phi(\boldsymbol{x}) \rangle_{\mathcal{H}} \;\; \forall \;\; \boldsymbol{x} \in \mathcal{X}, |w\rangle \in \mathcal{H} \}. \tag{93}$$

The orthogonality of $|w\rangle$ with respect to $|\Phi(\boldsymbol{x})\rangle$ defines a decision boundary; i.e., depending on the sign of the inner product, $\boldsymbol{x}$ lies on one side of the hyperplane. The function f is thus a linear function in $\mathcal{H}$. The *representer theorem* (Schölkopf, Herbrich, and Smola, 2001) states that this function can be approximated by the linear function $f^\star$ by using the previously defined kernel, i.e.,

$$f^\star(\boldsymbol{x}) = \sum_{i=1}^{\mathcal{D}} \alpha_i \kappa(\boldsymbol{x}, \boldsymbol{x}_i) \tag{94}$$

for an input dataset $\mathcal{D}$. Using Eq. (94), one can solve a convex optimization problem to get the coefficients α_i. The analysis thus far entails the connection between linear models in reproducing kernel Hilbert space with kernelized models in the input space.

One can use a quantum computer to calculate the inner product of feature mapped quantum states to obtain the kernel κ. This kernel can be fed to a classical device, which can use

Eq. (94) to obtain the coefficients α_i, for instance, by maximizing a cost function of the form (Havlíček *et al.*, 2019)

$$C(\boldsymbol{\alpha}) = \sum_{i=1}^{\mathcal{D}} \alpha_i - \frac{1}{2}\sum_{i,j}^{\mathcal{D}} y_i y_j \alpha_i \alpha_j \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{95}$$

where y_i are the labels of the training points and are constrained to $\sum_{i=1}^{\mathcal{D}} \alpha_i y_i = 0$. Ideas based on connections between kernel methods and quantum circuit–based machine learning has been used to justify the contention that the models for QML can be framed as kernel methods (Schuld, 2021). For some of the other relevant works on quantum kernel methods, see Blank *et al.* (2020) and Park, Blank, and Petruccione (2020). A high-dimensional data classification experiment with quantum kernel methods was carried out recently (Peters *et al.*, 2021). The encoding of data into quantum circuits is characterized by the quantum Fisher information metric (Haug, Self, and Kim, 2021). For hardware-efficient PQCs, the kernel can be related to radial basis function kernels (Haug and Kim, 2021b; Haug, Self, and Kim, 2021). Measuring the quantum kernel using the SWAP or inversion test scales as $\mathcal{D}^2$. Using randomized measurements (Elben *et al.*, 2020), the kernel can be computed in a time that scales linearly with the dataset size $\mathcal{D}$, which allows one to process large datasets with quantum computers (Haug, Self, and Kim, 2021). Note also that a Gaussian boson sampling device (see Sec. III.B) can be used for computing kernel functions (Schuld, Brádler *et al.*, 2020).

Another approach is to use a variational circuit $U(\boldsymbol{\theta})$ and directly perform the classification task in the reproducing kernel Hilbert space without using Eq. (94). This approach is sometimes referred to as a variational quantum classification. Data are also embedded into the state $|\Phi(\boldsymbol{x})\rangle$ and then processed with a PQC $U(\boldsymbol{\theta})$. The resultant state becomes

$$|\Psi(\boldsymbol{x},\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|\Phi(\boldsymbol{x})\rangle, \tag{96}$$

whose parameters are estimated by training it to match the target states $|y_i\rangle$ that represent the y_i labels of the training points, i.e., by minimizing the infidelity

$$C(\boldsymbol{\theta}) = \sum_{i=1}^{\mathcal{D}} [1 - |\langle y_i|\Psi(\boldsymbol{x}_i,\boldsymbol{\theta})\rangle|^2]. \tag{97}$$

Both methods require a way to encode the data into a quantum state. There are several strategies to define the quantum feature map. It is a key step in the success of the classification task, as the needed nonlinearities must come from it. Furthermore, to eventually obtain any quantum advantage, one should search from the set of classically intractable feature maps. One of the first proposed approaches was the amplitude encoding (Schuld, Sinayskiy, and Petruccione, 2016) that is also required in other quantum algorithms (Harrow, Hassidim, and Lloyd, 2009). This approach encodes the classical data points into the amplitudes of a quantum state, i.e., $|\Phi(\boldsymbol{x})\rangle = \sum_i x_i|e_i\rangle$, where $|e_i\rangle$ are the basis states. However, this raw encoding requires (i) knowing which gates can be used to perform this operation for general data points and (ii) having an efficient way to extract and process these amplitudes. Although the first point can eventually be overcome by using approaches similar to the ones used to define a PQC, the second one requires tools such as quantum random access memory (QRAM) (Giovannetti, Lloyd, and Maccone, 2008), which is experimentally challenging for the NISQ era. The studies toward QRAM of Park, Petruccione, and Rhee (2019) proposed an approach to update classical data consisting of M entries of n bits each using $O(n)$ qubits and $O(Mn)$ steps. A *forking-based* sampling scheme was suggested by Park *et al.* (2019) to reduce the resource requirements for state preparation for tasks involving repeated state preparation and sampling. Building a QRAM remains challenging and further investigations are required.

In general, the encoding strategies used in state-of-the-art algorithms are based on introducing classical data points into the parameters of the quantum circuit gates. As mentioned in Sec. II.B, one designs a state preparation circuit E that encodes the data points as follows:

$$|\Phi(\boldsymbol{x})\rangle = E(\boldsymbol{x},\boldsymbol{\phi})|0\rangle. \tag{98}$$

The use of $\boldsymbol{\phi}$ parameters is optional and they can be subject to the optimization subroutine too.

Typically, the encoding gates are designed using the same structure of a layerwise PQC from Eq. (22). Data points are introduced in layers of single-qubit rotational gates R, as defined in Eq. (13), followed by an entangling gate unitary W such as

$$E(\boldsymbol{x}) = \prod_{k=1}^{L_E}\left(\bigotimes_{i=1}^{n} R_k(\boldsymbol{x}_i)\right)W_k, \tag{99}$$

with L_E the total number of encoding layers. The entire VQC is composed of the encoding circuit and a processing circuit $U_{\mathrm{VQC}}(\boldsymbol{\theta},\boldsymbol{x}) = E(\boldsymbol{x})U(\boldsymbol{\theta})$, which is optimized for the respective task.

Alternatively, some works propose removing the distinction between the encoding E and processing U circuits and introducing the data values along the circuit (Vidal and Theis, 2019; Lloyd *et al.*, 2020; Nghiem, Chen, and Wei, 2020; Pérez-Salinas, Cervera-Lierta *et al.*, 2020; Schuld, Sweke, and Meyer, 2020). This strategy, sometimes called *input redundancy* or *data reuploading*, introduces the data into all circuit layers as follows:

$$U_{\mathrm{VQC}}(\boldsymbol{\theta},\boldsymbol{x}) = \prod_{k=1}^{L}\left(\bigotimes_{i=1}^{n} R_k(\boldsymbol{x}_i,\boldsymbol{\theta})\right)W_k, \tag{100}$$

where L is now the total number of circuit layers. This strategy has proven to be universal when applied to one qubit (Pérez-Salinas, Cervera-Lierta *et al.*, 2020) and can reconstruct the coefficients of the Fourier series (Vidal and Theis, 2019; Schuld, Sweke, and Meyer, 2020).

The inclusion of encoding strategies and data reuploading can help well-known VQAs such as the VQE. In general, one of the final goals of a VQE can be the identification of interesting points on a potential energy surface generated by a parametrized Hamiltonian. Commonly, one is interested in the ground state energy as a function of a Hamiltonian parameter λ such as the interatomic distance; however, other properties such as the energy gap between ground state and first excited state are of interest as well (Kyaw *et al.*, 2021). To learn the ground state energy as a function of the parameter λ, one often scans discretely over λ for a particular interval and runs a VQE to obtain the ground state energy for each of these points. This can become an additional computational cost, especially if we are interested only in a particular region of the ground state profile, such as the parameter $\lambda_{\min}$, where the ground state energy is minimal. In that vein, some proposals suggest encoding the parameters of the Hamiltonian into the PQC and learning the energy profiles (Mitarai, Yan, and Fujii, 2019). In particular, the *meta-VQE* algorithm (Cervera-Lierta, Kottmann, and Aspuru-Guzik, 2021) proposes to encode the λ into the PQC gates together with the optimization parameters. One then optimizes an objective function that corresponds to the sum of expectation values for some M training λ parameters, i.e., $\langle \hat{O} \rangle = \sum_{i=1}^{M} \langle H \rangle_{U(\boldsymbol{\theta},\lambda_i)}$. Once the circuit has been optimized, one can run it again with the new λ_i to directly extract an estimation of the ground state without having to optimize the full circuit again. An extension of this approach is the *optimized meta-VQE* algorithm, which consists of using the optimized parameters from the meta-VQE algorithm as the starting points for a standard VQE. This approach tries to avoid the vanishing gradients problem (see Sec. IV.A) by starting in a particular region of the parameter space instead of at a random initialization.

Some VQC additionally use the definition of the target state $|y_i\rangle$ to construct the objective function and optimize the fidelity with respect to these states. The goal of the quantum circuit is to divide and push the quantum states that encode the data points into two or more regions of the Hilbert space. To that aim, the parameters of the circuit are trained to match every encoded state into a particular representative of one of these regions. Therefore, the more separated that these regions are, the lower the number of expected misclassified points. As discussed in Sec. II.C, measuring qubits implies a certain computational cost. For that reason, many proposals suggest using the state of only a single qubit to train the entire circuit (Farhi and Neven, 2018; Schuld, Bocharov *et al.*, 2020). The cost function estimation reduces to measuring the probability distribution of one qubit. Other works use a more sophisticated definition of these target states by selecting the most orthogonal states of the qubit space (Lloyd *et al.*, 2020; Pérez-Salinas, Cervera-Lierta *et al.*, 2020). This strategy is inspired by optimal state discrimination (Helstrom, 1969).

Using the nonlinear character of quantum-mechanical processes as the "reservoir," the notion of *quantum reservoir computing* has been suggested. The reservoir is a highly nonlinear system whose parameters are arbitrary but fixed. One can perform reservoir computing by employing a basic training algorithm such as linear regression at the readout stage. Since the reservoir parameters are fixed, only the readout stage parameters are trained. The aforementioned idea utilizes the high nonlinearities of the reservoir without requiring the high computational cost of training. The concept of employing quantum systems as quantum reservoirs was introduced by Fujii and Nakajima (2017) and Nakajima *et al.* (2019). Quantum reservoir computing has been proposed for many experimental platforms, such as Gaussian states in the optical setup (Nokkala *et al.*, 2020), two-dimensional fermionic lattices (Ghosh *et al.*, 2019), and nuclear spins (Negoro *et al.*, 2018). Quantum gate–based implementation of quantum reservoir computing for NISQ devices has also been discussed (Chen, Nurdin, and Yamamoto, 2020). A Gaussian boson sampler (see Sec. III.B) can also be used for quantum reservoir computing, as suggested by Wright and McMahon (2019), to perform machine learning tasks such as classification. NISQ devices have also been used for regression (Mitarai *et al.*, 2018). Further, distance-based classifiers using quantum interference circuits were proposed by Schuld, Fingerhuth, and Petruccione (2017).

Quantum annealing has been also applied to supervised learning to predict biological data (Li *et al.*, 2018). Here the quantum annealer is used to train the parameters of the classification model, which is done by mapping the problem of finding the optimal parameters to the minimization of a QUBO.

2. Unsupervised learning

The use of quantum devices to speed up different unsupervised learning tasks has been investigated thoroughly and had led to different algorithms for generative modeling (Benedetti *et al.*, 2016, 2017; Benedetti, Garcia-Pintos *et al.*, 2019), clustering (Otterbach *et al.*, 2017), etc. (Lloyd, Mohseni, and Rebentrost, 2013). An analysis of quantum speedup in unsupervised learning for fault-tolerant algorithms was presented by Aïmeur, Brassard, and Gambs (2013). The task of learning probabilistic generative models, in particular, has been of interest to the QML community because of the potential advantage quantum computers may exhibit over their classical counterparts in the near future (Perdomo-Ortiz *et al.*, 2018). For the advantages rendered by quantum correlations such as *contextuality* and *Bell nonlocality* for generative modeling, see Gao *et al.* (2021).

Generative modeling involves learning the underlying probability distribution from a finite set of samples from a dataset and generating new samples from the distribution. There have been several proposals for using PQCs as models for generative learning (Amin *et al.*, 2018; Benedetti, Realpe-Gómez, and Perdomo-Ortiz, 2018; Benedetti, Garcia-Pintos *et al.*, 2019), including quantum Boltzmann machines, quantum circuit Born machines, quantum assisted Helmholtz machines, and quantum generative adversarial networks (Amin *et al.*, 2018; Benedetti, Realpe-Gómez, and Perdomo-Ortiz, 2018; Benedetti, Garcia-Pintos *et al.*, 2019; Benedetti, Lloyd *et al.*, 2019). We now discuss some of these proposals in detail.

Quantum Boltzmann machines.—The quantum Boltzmann machine (QBM) (Amin *et al.*, 2018) extends the classical Boltzmann machine (Ackley, Hinton, and Sejnowski, 1985), a neural architecture capable of several tasks, including generative modeling of data. These models are inspired by the Boltzmann distribution over the Ising model in the classical case and the Boltzmann distribution over the transverse-field Ising model for the quantum case. Such a network consists of a mixture of *visible* and *hidden* vertices connected by weighted edges. The visible vertices function as both inputs and outputs to the network, while the hidden vertices add extra degrees of freedom to the network.

The QBM can be modeled with the Hamiltonian

$$H = -\sum_{a}^{N}(b_a\hat{\sigma}_z^a + \Gamma_a\hat{\sigma}_x^a) - \sum_{a,b}\omega_{ab}\hat{\sigma}_z^a\hat{\sigma}_z^b, \qquad (101)$$

where b_a, Γ_a, and ω_{ab} are the parameters to be fine-tuned to generate the training data. The marginal probability that the visible variables are in a given state $\boldsymbol{v}$ is given by $P_v = \mathrm{Tr}(\Lambda_v\rho)$. Here we define $\Lambda_v = [\bigotimes_\nu(1 + v_\nu\hat{\sigma}_z^\nu)/2]\otimes I_h$ as a projector onto the subspace spanned by the visible variables tensored with the identity acting on the hidden variables and $\rho = e^{-H}/Z$ as a density matrix, with Z the usual partition function $Z = \mathrm{Tr}(e^{-H})$. The objective of training the QBM is to match the family of probability distributions P_v with the family inherent to the data P_v^{data} for arbitrary $\boldsymbol{v}$. This is achieved by minimizing the following negative log-likelihood measure:

$$\mathcal{L} = -\sum_v P_v^{\mathrm{data}} \log\frac{\mathrm{Tr}\Lambda_v e^{-H}}{\mathrm{Tr}e^{-H}}. \qquad (102)$$

The gradients of $\mathcal{L}$ with respect to the Hamiltonian parameters are difficult to calculate by sampling the Boltzmann machine, either classically or in the quantum variant. Methodologies of approximating these gradients are necessary to advance the deployment of QBMs.

The QBM can be trained to be a generator or a discriminator with respect to the distribution it is trained to mimic. Consider the joint distribution of input and output variables x and y, respectively. In the discriminative case, the objective is to minimize negative log-likelihood with respect to $P_{y|x}$. For generative learning, the goal is to learn the joint distribution $P_{x,y}$ directly.

The implementation of the QBM designed by Amin *et al.* (2018) found that a ten-qubit QBM with only visible vertices was able to learn a mixture of randomly generated Bernoulli distributions more effectively than a classical Boltzmann machine and performed better in generative applications. Kieferová and Wiebe (2017) found that a QBM outperformed classical Boltzmann machines in generative training to reproduce small Haar-random states. Extensions of the QBM, such as the variational quantum Boltzmann machine (VQBM) (Zoufal, Lucchi, and Woerner, 2020), have improved upon trainability. Using ideas similar to those of Zoufal, Lucchi, and Woerner (2020), Shingu *et al.* (2020) also proposed VQBM. Additionally, QBMs have shown potential in reinforcement learning (Crawford, Levit *et al.*, 2019), where they achieve better fidelity than restricted Boltzmann machines or deep Boltzmann machines with multiple layers of hidden vertices. To suit NISQ devices, Verdon, Broughton, and Biamonte (2017) suggested that QBMs can be approximated using QAOA as a subroutine and Anschuetz and Cao (2019) proposed an efficient method for training QBMs with NISQ devices based on the eigenstate thermalization hypothesis.

Quantum circuit Born machines.—PQCs can function as generative models to sample from probability distributions. The quantum circuit Born machine (QCBM) (Benedetti, Garcia-Pintos *et al.*, 2019) outputs bit strings $\boldsymbol{x}$ sampled from measurements in the computational basis of a quantum circuit $U(\boldsymbol{\theta})$, with the probability of each bit string given by the Born rule $[p_{\boldsymbol{\theta}}(x) \sim |\langle\boldsymbol{x}|U(\boldsymbol{\theta})|0\rangle|^2]$. The goal is that the distribution of the QCBM matches the one from a given target distribution.

QCBMs can prepare classical probability distributions as well as entangled quantum states by training the QCBM to match the probability distribution corresponding to the desired quantum state (Benedetti, Garcia-Pintos *et al.*, 2019). Liu and Wang (2018) proposed the training of QCBMs using the gradients of a PQC using the maximum mean discrepancy loss, which calculates the difference of the sampled output from the quantum circuit and the desired distribution in a kernel feature space.

QCBMs are well suited to be run on current NISQ hardware and can serve as benchmarks (Hamilton, Dumitrescu, and Pooser, 2019; Leyton-Ortega, Perdomo-Ortiz, and Perdomo, 2019; Zhu *et al.*, 2019) and have been applied to tasks such as the generation of images (Rudolph *et al.*, 2020) or financial data (Alcazar, Leyton-Ortega, and Perdomo-Ortiz, 2020; Coyle *et al.*, 2021). It has been shown that QCBMs can potentially outperform classical computers, as they are able to sample from probability distributions that are difficult for classical computers (Coyle *et al.*, 2020; Du *et al.*, 2020).

Quantum generative adversarial networks.—Generative adversarial learning (Goodfellow *et al.*, 2014) was a major recent breakthrough in machine learning and has become a powerful tool in the machine learning community for image and video generation as well as materials discovery. Generative adversarial networks (GANs) consist of two networks, a generator $F_G(z;\theta_g)$ and a discriminator $F_D(x;\theta_d)$. They are parametrized with θ_g and θ_d, respectively, and play the following adversarial game:

$$\min_{\theta_g}\max_{\theta_d}\big[E_{x\sim p_{\mathrm{data}}(x)}\{\log[F_D(x)]\} + E_{z\sim p_z(z)}\big(\log\{1 - F_D[F_G(z)]\}\big)\big]. \qquad (103)$$

In Eq. (103) $p_z(z)$ is a fixed prior distribution, $p_{\mathrm{data}}(x)$ is the target distribution, x is the data sampled from $p_{\mathrm{data}}(x)$, and z is the noise sampled from $p_z(z)$. The training of GANs is carried iteratively until the generator produces a distribution indistinguishable from the target distribution.

A quantum version of GANs was proposed theoretically by Dallaire-Demers and Killoran (2018) and Lloyd and Weedbrook (2018) and further developed for near-term quantum devices by Romero and Aspuru-Guzik (2019), Zeng *et al.* (2019), and Situ *et al.* (2020), where PQCs are used for adversarial learning instead of classical neural networks.

The different adaptions of quantum GANs can be divided into different categories depending on whether the data and networks are classical or quantum (Romero and Aspuru-Guzik, 2019). There have been different studies with hybrid models of GANs using both classical and quantum data and it has been shown that the training of these networks is robust to a moderate level of noise (Anand *et al.*, 2020).

The training of quantum GANs has been demonstrated experimentally on various quantum processing units for a variety of tasks, including quantum state estimation (Hu *et al.*, 2019), image generation (H.-L. Huang *et al.*, 2020; K. Huang *et al.*, 2020), generation of continuous distributions (Anand *et al.*, 2020), and learning distribution (Zoufal, Lucchi, and Woerner, 2019; Nakaji and Yamamoto, 2020b).

3. Reinforcement learning

The general framework of RL involves an agent interacting with an environment attempting to maximize an underlying reward function. The mathematics of RL can be captured using the Markov decision process (MDP) (Sutton and Barto, 2018). A MDP is a 4-tuple (S, A, R, P), where S is the set of all possible valid states, A is the set of all possible actions, R is the reward function, i.e., a map $R: S \times A \times S \to \mathbb{R}$, and P is the transition probability, i.e., a map $P: S \times A \to [0,1]$. Specifically, the transition probability $P(\tilde{s}|s,a)$ represents the probability of the transition to state $\tilde{s}$ given that the present state is s and the action a has been taken. The term "Markov" in MDP means that the transitions are memoryless and depend only on the current state and action. The agents in RL learn via trial and error. For a successful training, a proper balance between the exploration of unknown strategies and the exploitation of prior experience is required.

RL models are trained via agent-environment interactions. At the beginning of time step t, the environment state is s_t. From the set A, the agent selects an action a_t. The transition probability dictates the next state of the environment s_{t+1} and the agent gets the reward r_{t+1} based on the reward function R. The agent-environment interaction yields a series of states and actions of the form $\tau = (s_1, a_1, s_2, a_2, \ldots, s_H, a_H)$. The aforementioned series is called a trajectory and the number of interactions (H) in an episode is called horizon. Suppose that the probability of a trajectory is $P(\tau)$ and the corresponding cumulative reward is $R_{\mathrm{tot}}(\tau)$. The expected reward is then $\sum_\tau P(\tau) R_{\mathrm{tot}}(\tau)$.

By harnessing quantum-mechanical phenomena such as superposition and entanglement, one can expect to achieve speedups in the RL tasks (Dong *et al.*, 2008; Paparo *et al.*, 2014; Dunjko, Taylor, and Briegel, 2016, 2017). The aforementioned intuition has led to recent work toward quantum RL (Dunjko, Taylor, and Briegel, 2017; Cornelissen, 2018)

We discuss the essence of quantum RL by providing a synopsis of the quantum agent-environment (AE) paradigm. For details, see Dunjko, Taylor, and Briegel (2017). In the AE paradigm, agent and environment are modeled via sequences of unitary maps for the agent $\{\mathcal{E}_A^j\}_j$ and the environment $\{\mathcal{E}_E^j\}_j$, respectively. The agent and environment have access to memory registers belonging to Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_E$. The communication register between the agent and the environment belongs to the Hilbert space $\mathcal{H}_C$. The agent maps $\{\mathcal{E}_A^j\}_j$ act on $\mathcal{H}_A \otimes \mathcal{H}_C$ and the environment maps $\{\mathcal{E}_E^j\}_j$ act on $\mathcal{H}_E \otimes \mathcal{H}_C$. The agent and environment interact with each other by applying their maps sequentially. The set of actions and states correspond to an orthonormal set of vectors $\{|a\rangle | a \in A\}$ and $\{|s\rangle | s \in S\}$, respectively. The Hilbert space of the communication register is given by $\mathcal{H}_C = \mathrm{span}(|y\rangle | y \in S \cup A)$. The classical AE paradigm corresponds to the case where the agent and environment maps are classical.

Quantum RL has been studied for algorithms such as the state-action-reward-state-action algorithm or Q-learning (Jerbi *et al.*, 2019), which are some of the elementary RL algorithms (Sutton and Barto, 2018).

In the setup of variational quantum circuits, RL has been explored for small input sizes (Chen, Yang *et al.*, 2020). Chen, Yang *et al.* revealed the possibility of a quadratic advantage in parameter space complexity. Using better encoding schemes, Lockwood and Si (2020b) showed a case of RL with variational quantum circuits for larger input sizes. In a follow-up work, Lockwood and Si (2020a) demonstrated the possibility of dealing with the relatively complicated example of playing Atari games.

RL with quantum annealers was also investigated by Crawford *et al.* (2016). In their framework, they explored RL with QBMs. A detailed study of basic RL protocols with superconducting circuits was provided by Lamata (2017). Some interesting proposals of RL with trapped ions and superconducting circuits were also recently proposed (Cárdenas-López *et al.*, 2018). For quantum eigensolvers, RL has been investigated as well (Albarrán-Arriagada *et al.*, 2020). RL with an optical setup was discussed by Yu *et al.* (2019).

C. Combinatorial optimization

Given a finite set of objects S, combinatorial optimization aims to find the optimal object from the set S. It is a subdiscipline of mathematical optimization theory, with applications in diverse fields such as artificial intelligence, logistics, supply chains, and theoretical computer science. Some typical examples of combinatorial optimization problems are the traveling-salesman problem (Lenstra and Kan, 1975), job-shop scheduling (Manne, 1960), max cut (Festa *et al.*, 2002), and Boolean satisfiability (Tovey, 1984).

To understand combinatorial optimization, we consider the canonical problem of Boolean satisfiability. Boolean variables admit two truth values, TRUE and FALSE. These can be combined together using operators AND or a conjunction (denoted by $\wedge$), NOT or a negation (denoted by $\neg$), and OR or a disjunction (denoted by $\vee$). These combinations are called Boolean expressions. A Boolean expression is said to be satisfiable if it can be TRUE for appropriate assignment of logical values to its constituent Boolean variables. Given a Boolean expression E, the Boolean satisfiability problem (SAT) consists of checking if E is satisfiable. The well-known Cook-Levin theorem showed that SAT is NP complete (Arora and Barak, 2009).

Every combinatorial optimization problem can be expressed as m clauses over n Boolean variables. A Boolean variable is known as the positive literal, while its negation is known as the negative literal. A disjunction of literals is known as the clause or constraint. For every constraint C_α for $\alpha \in \{1, 2, \ldots, m\}$ and every string $z \in \{0,1\}^n$, we define

$$C_\alpha(z) = \begin{cases} 1 & \text{if } z \text{ satisfies } C_\alpha(z), \\ 0 & \text{if } z \text{ does not satisfy.} \end{cases}$$

The goal of a combinatorial optimization problem is to find a string that maximizes the following objective function:

$$C(z) = \sum_{\alpha=1}^{m} C_\alpha(z), \tag{104}$$

which counts the number of satisfied constraints.

Approximate optimization algorithms such as QAOA seek to find a solution z (usually a bit string) with a desired approximation ratio $r^* \leq C(z)/C_{\max}$, where $C_{\max}$ is the maximum value of $C(z)$. Using $C(z)$ and computational basis vectors $|e_i\rangle \in \mathbb{C}^{2^n}$ for $i = 1, \ldots, 2^n$, one can construct the problem Hamiltonian as that in Eq. (18) and thus map the combinatorial optimization problem to a Hamiltonian ground state problem. The NISQ algorithms for combinatorial optimization discussed next are listed in Table III of the Supplemental Material (824).

1. Max-cut problem

Max cut is an important combinatorial optimization problem with applications in diverse fields such as theoretical physics and circuit design. In theoretical physics, the max-cut problem is equivalent to finding the ground state of a spin glass Hamiltonian. Given a graph $G = (V, E)$ with a vertex set V and an edge set E, a cut is a partition of the elements of V into two disjoint subsets. Given a weight function $w: E \to \mathbb{R}^+$ such that the edge $(i, j) \in E$ has a weight E_{ij}, the max-cut problem consists of finding a cut $K \cup \bar{K} = V$ that maximizes

$$\sum_{i\in K, j \in \bar{K}, (i,j) \in E} w_{ij}. \tag{105}$$

For every vertex $v_i \in V$, we associate a variable x_i that can take the values ± 1. Given an arbitrary cut $K \cup \bar{K} = V$, we define $x_i = 1$ if $v_i \in K$ and -1 otherwise. The max-cut problem is then equivalent to the following quadratic program:

$$\max \sum_{(v_i, v_j) \in E} w_{ij} \frac{1 - x_i x_j}{2}, \tag{106}$$

which is subject to $x_i \in \{-1, +1\} \ \forall \ v_i \in V$.

Considering n vertices as n qubits in the computational basis, we can classify qubits by assigning quantum state $|0\rangle$ or $|1\rangle$. For the classical objective function in the optimization program from Eq. (106), we can use the following Hamiltonian as the problem Hamiltonian:

$$H_P = \sum_{(i,j)\in E} \frac{1}{2} (I - \hat{\sigma}_z^i \otimes \hat{\sigma}_z^j) \equiv \sum_{(i,j)\in E} C_{ij}. \tag{107}$$

It has been shown that it is NP hard to achieve an approximation ratio of $r^* \geq 16/17 \approx 0.9412$ for max cut on all graphs (Håstad, 2001). For the QAOA with $p = 1$, it has been shown that for a general graph

$$\langle C_{ij} \rangle = \tfrac{1}{2} + \tfrac{1}{4}(\sin 4\beta \sin\gamma)(\cos^{d_i}\gamma + \cos^{d_j}\gamma) \\ - \tfrac{1}{4}(\sin^2\beta \cos^{d_i + d_j - 2\lambda_{ij}}\gamma)(1 - \cos^{\lambda ij} 2\gamma), \tag{108}$$

where $d_i + 1$ and $d_j + 1$ denote the degrees of vertices i and j, respectively, and λ_{ij} is the number of triangles containing the edge (i, j) in the graph (Wang, Hadfield *et al.*, 2018). Here γ and β refer to the QAOA parameters from Eq. (20). Analytical results for general Ising optimization problems with $p = 1$ have also been found (Ozaeta, van Dam, and McMahon, 2020).

In the case of unweighted 3-regular (u3R) graphs, the previously mentioned result gives the approximation ratio of 0.692, which is consistent with the pioneering result by Farhi, Goldstone, and Gutmann (2014). In comparison, the best classical algorithms to date gives the approximation ratio of $r^* \approx 0.8786$ for general graphs (Goemans and Williamson, 1995), and $r^* \approx 0.9326$ for u3R graphs (Halperin, Livnat, and Zwick, 2004) using semidefinite programming. While QAOA for $p = 1$ does not outperform its classical counterparts for the max-cut problem, QAOA was found to surpass the Goemans-Williamson bound for larger values of p (Crooks, 2018).

QAOA was also applied to the clustering problem of unsupervised learning by mapping it to the max-cut problem (Otterbach *et al.*, 2017). It was shown that by fixing the QAOA parameters and selecting the typical problem instances from a reasonable distribution, the objective function value concentrates; i.e., the objective function value is almost independent of the problem instance (Brandao *et al.*, 2018). This implies that the parameters optimized for one instance can be used for other typical instances, which would drastically reduce the optimization cost. Similar concentration behavior was also reported for the Sherrington-Kirkpatrick model in the infinite size limit (Farhi *et al.*, 2019).

Recently a nonlocal version of QAOA called recursive QAOA (RQAOA) was proposed (Bravyi *et al.*, 2019). It consists of running a QAOA as a subroutine on a specific problem with N qubits and measuring the expectation values of the correlations between all qubit pairs (i, j) with $M_{ij} = \langle \sigma_z^i \sigma_z^j \rangle$. One then picks out the pair of qubits (n, m) that have maximal absolute value of correlation $n, m = \arg\max_{(i,j)} |M_{ij}|$. For $M_{nm} > 0$, the selected qubit pair (n, m) are positively correlated and likely to be in the same state, whereas for $M_{nm} < 0$ they are anticorrelated and likely to be in the opposite state. This correlation is fixed as a constraint on the problem by fixing the state of the qubit $\sigma_z^m = \mathrm{sgn}(M_{nm})\sigma_z^n$. With this constraint, one of the two qubits can be removed, as its state is completely determined by the other, thus reducing the total qubit number by 1. This procedure is repeated for the reduced problem of size $N - 1$ qubits; i.e., one again runs the QAOA subroutine, measures the correlations, and fixes the qubit pairs with maximal correlation. The RQAOA algorithm is run recursively until the size of the problem is reduced to a small number of qubits such that it can be easily solved classically. When RQAOA is run with the QAOA subroutine of depth $p = 1$, it can be efficiently simulated on a classical computer, which can serve as an important benchmark with classical algorithms (Bravyi *et al.*, 2019). Numerical experiments with higher p suggest similar or better performance on combinatorial problems than with other classical algorithms (Bravyi, Kliesch *et al.*, 2020; Egger, Marecek, and Woerner, 2020).

Finally, QAOA with depth $p = 1$ has been investigated in comparison with quantum annealing (Streif and Leib, 2020). In the limit of infinite depth p, QAOA is equivalent to quantum annealing; see Sec. II.B.1 for QAOA and Sec. III.A for quantum annealing. However, QAOA can already outperform quantum annealing at depth $p = 1$. For specific problems, QAOA arrives at the correct solution with unit probability, whereas quantum annealing struggles to find the solution (Streif and Leib, 2020). This shows that QAOA is strictly more powerful than quantum annealing.

2. Other combinatorial optimization problems

While the usage of QAOA on max cut has been studied extensively, QAOA also has applications in other important combinatorial optimization problems such as the max-k vertex cover problem, which seeks the set of k vertices on a graph that maximizes the number of edges incident on the vertices (Cook, Eidenbenz, and Bärtschi, 2019). Other applications of QAOA are for the exact-cover problem with applications to the tail-assignment problem (Bengtsson *et al.*, 2020; Vikstål *et al.*, 2020), lattice protein folding (Fingerhuth, Babej, and Ing, 2018; Robert *et al.*, 2021), the knapsack problem as applied to battery revenue optimization (de la Grand'rive and Hullo, 2019), multicoloring graph problems (Oh *et al.*, 2019), maximum independent set problems with applications to scheduling (Choi, Oh, and Kim, 2020; Saleem, 2020), and the vehicle routing problem (Utkarsh, Behera, and Panigrahi, 2020). Streif *et al.* (2021) described how to utilize QAOA to solve the binary paint shop problem and showed that in the infinite size limit QAOA with constant depth can better classical heuristics on average. An adiabatically assisted approach was suggested by Garcia-Saez and Latorre (2018) to tackle combinatorial optimization problems. Investigations involving a variational Grover search could be helpful to solve combinatorial optimization problems (Morales, Tlyachev, and Biamonte, 2018; Zhang, Rao *et al.*, 2021). Gaussian boson sampling (see Sec. III.B) has been used to assist in a wide variety of combinatorial optimization problems (Arrazola, Bromley, and Rebentrost, 2018; Bromley *et al.*, 2020), most prominently to solve the max-clique algorithm (Arrazola and Bromley, 2018; Banchi, Quesada, and Arrazola, 2020). This has applications in predicting molecular docking configurations (Banchi *et al.*, 2020), computing vibrational spectra of molecules (Huh *et al.*, 2015), and electron-transfer reactions (Jahangiri, Arrazola, and Delgado, 2020). Using NISQ devices, an approach was suggested by Metwalli, Gall, and Van Meter (2020) for the triangle finding problem and its k-clique generalization.

Quantum annealing, which has been the inspiration of QAOA, is a prominent platform that has been applied to various combinatorial optimization problems and its applications, such as protein folding (Perdomo-Ortiz *et al.*, 2012), which was reviewed by Hauke *et al.* (2020). As gate-based devices mature, it will open the possibility for experimental benchmarking of QAOA against state-of-the-art solvers for suitable real-world applications, as performed by Perdomo-Ortiz *et al.* (2019) in the context of quantum annealing machines.

D. Numerical solvers

We now discuss NISQ algorithms used to solve numerical problems such as factoring, singular value decomposition, linear equations, and nonlinear differential equations, which are listed in Table IV of the Supplemental Material (824).

1. Variational quantum factoring

The factoring problem accepts a composite positive integer N as input and returns its prime factors as output. There is no known efficient classical algorithm for prime factorization and the hardness of factoring is used to provide security in the Rivest-Shamir-Adleman (RSA) public-key cryptosystems. Shor's algorithm is a polynomial-time quantum algorithm for the factoring problem (Shor, 1999), which implies that prime factorization is in BQP and hence has been extensively investigated by quantum computing researchers; for details see Anschuetz *et al.* (2019) and references therein. However, the resource estimates for implementing the Shor's algorithm are far beyond the capabilities of the NISQ era. A detailed analysis has shown that factoring a 2048-bit RSA number would necessitate a quantum processor with 10^5 logical qubits and a circuit depth of the order of 10^9 to run for roughly ten days (Van Meter *et al.*, 2010; Jones *et al.*, 2012). On a photonic architecture with 1.9×10^9 photonic modules, factoring a 1024-bit RSA number is expected to require around 2.3 yr (Devitt *et al.*, 2013). To tackle the factoring problem in near-term quantum devices, it is imperative to develop NISQ-era compatible alternatives to Shor's algorithm.

The factoring problem can be mapped to the ground state problem of an Ising Hamiltonian (Burges, 2002; Dattani and Bryans, 2014). To understand this mapping, we consider the factoring of $m = p \times q$. Suppose that the binary representations of m, p, and q are $m = \sum_{k=0}^{n_m-1} 2^i m_k$, $p = \sum_{k=0}^{n_p-1} 2^i p_k$, and $q = \sum_{k=0}^{n_q-1} 2^i q_k$. Here $m_k \in \{0, 1\}$ is the kth bit of m and the total number of bits for m is denoted by n_m. A similar notation has been employed for p and q. Since $m = p \times q$, this induces $n_c = n_p + n_q - 1$ constraints on the individual bits of m, p, and q,

$$\sum_{j=0}^{i} q_i p_{i-j} + \sum_{j=0}^{i} z_{j,i} - m_i - \sum_{j=1}^{n_c} 2^j z_{i,i+j} = 0 \quad (109)$$

for $i \in [0, n_c)$, where the carry bit from position i to position j is represented by $z_{i,j}$. The constraint i in Eq. (109) induces the clause $C_i \equiv \sum_{j=0}^{i} q_i p_{i-j} + \sum_{j=0}^{i} z_{j,i} - m_i - \sum_{j=1}^{n_c} 2^j z_{i,i+j}$ over $\mathbb{Z}$ such that factoring can be modeled as an assignment of binary variables $\{m_i\}$, $\{p_i\}$, and $\{q_i\}$ that solves $\sum_{i=0}^{n_c-1} C_i^2 = 0$.

One can map the binary variables to quantum observables to quantize the clause C_i to $\hat{C}_i$ using the mapping $b_k \to (1/2)(1 - \sigma_{b,k}^z)$ and obtain the Hamiltonian $H_P = \sum_{i=0}^{n_c-1} \hat{C}_i{}^2$, which we refer to as the factoring Hamiltonian. Note that the factoring Hamiltonian is a 4-local Ising Hamiltonian.

By using the aforementioned ideas, one can use NISQ algorithms for the ground state problem to tackle the factoring problem; see Secs. VI.A.3 and VI.C. Anschuetz *et al.* (2019) employed QAOA to find the ground state of the factoring Hamiltonian with the variational quantum factoring (VQF) algorithm. Numerical simulations were provided for numbers as high as 291 311. For a recent experimental realization and a detailed analysis of VQF, see Karamlou *et al.* (2020).

2. Singular value decomposition

Given a matrix $M \in \mathbb{C}^{m\times n}$, the singular value decomposition (SVD) provides a factorization of the form $M = U\Sigma V^\dagger$, where $U \in \mathbb{C}^{m\times m}$ is a unitary matrix, $\Sigma \in \mathbb{R}_+^{m\times n}$ is a rectangular diagonal matrix with non-negative real diagonal entries, and $V \in \mathbb{C}^{n\times n}$ is a unitary matrix. The diagonal entries of Σ are called the singular values of matrix M. The columns of the unitary matrices U and V are called left-singular and right-singular vectors of M. Using Dirac notation, one can write

$$M = \sum_{j=1}^{r} d_j |u_j\rangle\langle v_j|, \tag{110}$$

where $d_j, |u_j\rangle$, and $|v_j\rangle$ are singular values, left-singular vectors, and right-singular vectors. The rank of matrix M is r and is equal to the number of nonzero singular values.

SVD finds applications in calculating the pseudoinverse (Gregorcic, 2001), solving homogeneous linear equations (Klema and Laub, 1980), signal processing (Vandewalle and De Moor, 1991), and recommendation systems (Koren, Bell, and Volinsky, 2009). Moreover, the Schmidt decomposition for studying the entanglement of bipartite quantum states is related to SVD.

In the quantum information context, the SVD can be used to compute the Schmidt decomposition of bipartite quantum states. For a quantum state $|\psi\rangle \in \mathcal{H}_A \otimes \mathcal{H}_B$, the Schmidt decomposition is given by

$$|\psi\rangle = \sum_i d_i |u_i\rangle|v_i\rangle, \tag{111}$$

where d_i are non-negative real numbers such that $\sum_i d_i^2 = 1$. Moreover, $\{|u_i\rangle\}_i$ and $\{|v_i\rangle\}_i$ correspond to orthonormal basis sets for $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively. The number χ of nonzero d_i is called the Schmidt rank of the quantum state $|\psi\rangle$ and is used to quantify the bipartite entanglement. To calculate the Schmidt decomposition by performing the SVD of a matrix A, one can write the bipartite quantum state as a matrix $|\psi\rangle = \sum_{i,j} A_{ij}|i\rangle|j\rangle$, where $|i\rangle$ and $|j\rangle$ are the computational basis states of each qubit.

Bravo-Prieto, García-Martín, and Latorre (2020) provided a NISQ algorithm to perform SVD of pure bipartite states. Starting with two unitary circuits that act on different bipartitions of the system, they variationally determined the singular values and singular vectors by training the circuits on the exact coincidence of outputs. The central idea of their method is to variationally find circuits that provide the following transformation of the initial quantum state $|\psi\rangle_{AB}$ with Schmidt rank χ:

$$U_A \otimes V_B|\psi\rangle_{AB} = \sum_{i=1}^{\chi} \lambda_i e^{i\gamma_i}|e_i\rangle_A|e_i\rangle_B, \tag{112}$$

where $U_A|v_i\rangle_A = e^{i\alpha_i}|e_i\rangle_A$ and $V_B|v_i\rangle_B = e^{i\beta_i}|e_i\rangle_B$ such that $\alpha_i = \beta_i + \gamma_i \in [0, 2\pi)$ and $\{|e_k\rangle_{A,B}\}_k$ are the computational basis states in $\mathcal{H}_{A,B}$. Using their algorithm, they also suggested the possibility of implementing SWAP gates between parties A and B without the requirement of any gate connecting the two subsystems.

Using variational principles for singular values and the Ky Fan theorem (Fan, 1951), Wang, Song, and Wang (2020) provided an alternative NISQ algorithm for SVD. They provided a proof of principle application of their algorithm in an image compression of handwritten digits. They also discussed the applications of their algorithm in recommendation systems and polar decomposition.

3. Linear system problem

Systems of linear equations play a crucial role in various areas of science, engineering, and finance. When one is given a matrix $A \in \mathbb{C}^{N\times M}$ and $\boldsymbol{b} \in \mathbb{C}^N$, the task of the linear system problem consists of finding $\boldsymbol{x} \in \mathbb{C}^M$ such that

$$A\boldsymbol{x} = \boldsymbol{b}. \tag{113}$$

Depending on the dimensions M and N, the linear system problem takes various forms. If $M = N$ and A is invertible, $\boldsymbol{x} = A^{-1}\boldsymbol{b}$ is unique. If $M \neq N$, the linear system problem can be underdetermined or overdetermined. For simplicity, it is natural to assume the matrix A to be square, i.e., $M = N$. If the matrix A has at most s nonzero elements per row or column, the linear system problem is called s sparse.

The quantum version of the linear system problem, known as the quantum linear system problem, assumes A to be an $N \times N$ Hermitian matrix and $\boldsymbol{b}$ to be a unit vector; i.e., it can be represented as a quantum state $|b\rangle = \sum_{i=1}^N b_i|e_i\rangle$. The quantum linear system problem thus is formulated as

$$A|x\rangle = |b\rangle \to |x\rangle = A^\dagger|b\rangle. \tag{114}$$

The first quantum algorithm proposed for solving the quantum linear system problem was the Harrow-Hassidim-Lloyd (HHL) algorithm (Harrow, Hassidim, and Lloyd, 2009). Apart from the size N of the matrix A and its sparsity s, the two dominant factors that determine the run-time of the algorithm are the condition number κ of the matrix A and the additive error ϵ corresponding to the solution. The condition number is given by the ratio of maximal and minimal singular values of A. The best classical algorithm for the linear system problem is the conjugate gradient method with a run-time complexity $O[Ns\kappa \log(1/\epsilon)]$. On the other hand, the original HHL algorithm for the quantum linear system problem has a run-time complexity $O[\log(N)s^2\kappa^2/\epsilon]$. Further works on the HHL algorithm have improved the scaling of κ to linear (Ambainis, 2012) and the error dependence to poly$[\log(1/\epsilon)]$ (Childs, Kothari, and Somma, 2017). However, the implementation of the HHL algorithm requires a fault-tolerant architecture, and hence its guarantees cannot be leveraged on NISQ devices. The largest quantum linear system problem solved on a gate-based quantum computer was implemented on a NMR processor for $N = 8$ (Wen *et al.*, 2019).

Recently VQA-based implementations of the quantum linear system problem were proposed (Bravo-Prieto *et al.*, 2019; Huang, Bharti, and Rebentrost, 2019; Xu *et al.*, 2019). When one is given a quantum linear system problem with input A and $|b\rangle$, the idea is to find the ground state of the following Hamiltonian:

$$H(u) = A(u)P^{\perp}_{+,b}A(u), \tag{115}$$

where $A(u)$ and $P^{\perp}_{+,b}$ are defined as

$$A(u) \equiv (1-u)\sigma_z \otimes I + u\sigma_x \otimes A, \tag{116}$$

$$P^{\perp}_{+,b} = I - |+,b\rangle\langle b,+|. \tag{117}$$

Both A and $|b\rangle$ are assumed to be constructed efficiently with a quantum circuit, i.e., $A = \sum_{k=1}^{K_A} \beta_k U_k$ and $|b\rangle = U_b|0\rangle$, with $K_A = O[\mathrm{poly}(\log N)]$. The phase in β_k can be absorbed into U_k, and hence one can assume that $\beta_k > 0$. For $u = 1$, the Hamiltonian in Eq. (115) has a unique ground state $|+\rangle|x^\star\rangle = |+\rangle A^{-1}|b\rangle/\|A^{-1}|b\rangle\|_2$ with zero ground state energy. After removing the ancilla, the ground state is proportional to $A^{-1}|b\rangle$. Thus, one can define the following loss function:

$$L_H(|x\rangle) = \langle +,x|H(1)|+,x\rangle. \tag{118}$$

Without the ancilla, Eq. (118) can be written as $L_H(|x\rangle) = \langle x|A^2|x\rangle - \langle x|A|b\rangle\langle b|A|x\rangle$.

Huang, Bharti, and Rebentrost (2019) analyzed the optimization landscape for VQA-based optimization of the loss function of Eq. (118) and showed the presence of barren plateaus that persist independent of the architecture of the quantum circuit for generating $|x(\theta)\rangle$. Even techniques based on adiabatic morphing (Garcia-Saez and Latorre, 2018) fail to evade the effect of the barren plateaus. To circumvent the barren plateau problem, Huang, Bharti, and Rebentrost (2019) proposed a classical-quantum hybrid state [see also Sec. III.E and Eq. (42)] $\boldsymbol{x} = \sum_{i=1}^{r} \alpha_i |\psi_i(\theta_i)\rangle$, where $\alpha_i \in \mathbb{C}$ and $\theta_i \in \mathbb{R}^{k_i}$ for $i \in \{1, 2, \ldots, r\}$. Note that θ_i are the usual variational parameters and α_i are the combination parameters that are stored on a classical computer. The state x is not explicitly created on the quantum processor and may not be normalized. To solve the quantum linear system problem, one minimizes the following loss function:

$$L_R(x) = \|Ax - |b\rangle\|_2^2 = x^\dagger A^\dagger A x - 2\mathrm{Re}\{\langle b|Ax\} + 1. \tag{119}$$

Since optimization with respect to θ_i suffers from the barren plateau problem, one can fix and subsequently drop the variational parameter θ_i.

The optimization landscape is convex in $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_r)$. Starting with $|\psi_1\rangle = |b\rangle$, other quantum states can be generated using the *Ansatz* tree approach of Huang, Bharti, and Rebentrost (2019). It was proved that finding the combination parameters of $|\psi_1\rangle, |\psi_2\rangle, \ldots, |\psi_r\rangle$ to minimize $L_R(\sum_{i=1}^{r} \alpha_i |\psi_i\rangle)$ is BQP complete. Moreover, using $O(K_A^2 r^2/\epsilon)$ measurements one can find a ϵ-suboptimal solution. With this approach, linear systems as high as $2^{300} \times 2^{300}$ can be solved by considering cases that are classically tractable.

4. Nonlinear differential equations

Nonlinear differential equations (NLDEs) are a system of differential equations (DEs) that cannot be expressed as a linear system. The numerical approaches to tackle DEs can be local or global. Local methods employ numerical differentiation techniques (Butcher, 1987) such as Runge-Kutta methods or discretization of the space of variables. Global methods represent the solution via a suitable basis set, and the goal is to find optimal coefficients (Gottlieb and Orszag, 1977). In many cases, as the number of variables or the nonlinearity of the differential equations increase, finding solutions becomes challenging. To achieve higher accuracy, local methods require a fine grid, which requires a high computational cost. In the case of global methods, high accuracy necessitates a large number of elements in the basis set, leading to more extensive resource requirements. To tackle resource challenges, quantum algorithms are proposed.

Linear DEs can be reexpressed as a system of linear equations using the finite-difference method, and one can employ NISQ linear system algorithms to tackle the problem; see Sec. VI.D.3. For a recent theoretical proposal with experimental work on linear differential equations, see Xin *et al.* (2020). However NLDEs defy this approach for large nonlinearities.

A canonical example of a NLDE appearing in quantum theory is the 1D nonlinear Schrödinger equation $[-(1/2)(d^2/dx^2) + V(x) + g|f(x)|^2]f(x) = Ef(x)$. Here E denotes energy, g quantifies nonlinearity, and V is the external potential. Recently NISQ algorithms for NLDEs were proposed. Lubasch *et al.* (2020) used ancillary quantum registers and controlled-multiqubit operations to implement nonlinearities to simulate the nonlinear Schrödinger equation. Haug and Bharti (2020) proposed the nonlinear quantum assisted simulator (NLQAS) to tackle NLDEs without any controlled unitaries. Using NLQAS, they simulated this equation for eight-qubit systems. NLDEs were also studied by Gaitan (2020) for fluid dynamics problems. Using differentiable quantum circuits, Kyriienko, Paine, and Elfving (2020) proposed an interesting approach for solving NLDEs via global methods.

E. Other applications

In this section, we cover other applications for which NISQ algorithms can provide promising improvements. They are listed in Table VI of the Supplemental Material (824).

1. Quantum foundations

One of the first experiments in digital quantum computers were the Bell nonlocality tests known as *Bell inequalities* (Brunner *et al.*, 2014). Those experiments computed a type of Bell inequalities known as Mermin inequalities in up to five superconducting quantum qubits. The experiment consisted of preparing the GHZ state (Greenberger *et al.*, 1990), measuring it in a particular basis, and obtaining the expectation value of the Mermin operator (Alsina and Latorre, 2016). These nonlocality tests can be extended to higher dimensions by controlling quantum levels beyond the qubit subspace. Cervera-Lierta *et al.* (2021) experimentally generated a qutrit GHZ state using a programmable device controlled with QISKIT pulse software (Alexander *et al.*, 2020), which represents the first step toward performing a GHZ test.

In the context of VQA, the nonclassicality in VQEs is examined using contextuality, which is a nonclassical feature of quantum theory (Amaral and Cunha, 2018). Using the notion of "strong contextuality," Kirby and Love, (2019) separated VQE experiments into two categories: contextual and noncontextual. Such foundational works could be utilized to comprehend the possible sources of quantum advantage in NISQ algorithms. Using novel concepts from this field, Kirby, Tranter, and Love (2020) recently proposed contextual subspace VQE.

In another work, the variational consistent history algorithm was suggested for investigating foundational questions (Arrasmith *et al.*, 2019). The consistent history approach has been used to examine topics from quantum cosmology and the quantum-classical transition. In the variationally consistent history algorithm, the quantum computer is used to compute the "decoherence functional," which is challenging to calculate classically. The classical computer is employed to tune the history parameter such that the consistency is improved.

2. Quantum optimal control

Quantum optimal control is a topic of paramount importance in the pursuit to harness the potential of near-term devices. For a given quantum control system and a cost function that measures the quality of control, it aims to find a control that can achieve optimal performance.

Some recent works have investigated quantum optimal control in the NISQ framework. A recent detailed perspective in this direction was given by Magann *et al.* (2021). Li, Yang *et al.* (2017) provided a hybrid quantum-classical approach to quantum optimal control. To remedy some of the difficulties of classical approaches to optimal control related to the scaling of resources, Dive *et al.* (2018) proposed another NISQ framework. Experimental demonstration of quantum control for a 12-qubit system has also been realized (Lu *et al.*, 2017). However, the aforementioned approaches restrict their target states to be sparse matrices. For dense target states, Policharla and Vinjanampathy (2020) recently proposed a NISQ algorithm. Along with their algorithm, they also suggested a few algorithmic primitives to calculate overlap of quantum states and transition matrix elements. Hybrid quantum-classical algorithms have also been implemented for computing quantum optimal control pulses, particularly for controlling molecular systems (Castaldo, Rosa, and Corni, 2020; Magann *et al.*, 2020).

3. Quantum metrology

Quantum metrology harnesses nonclassical features of quantum theory for parameter estimation tasks. A canonical example is estimating the parameter ϕ of a unitary map under the action of a Hamiltonian $\hat{H}$ given by $\hat{\rho}(\phi) = e^{-i\hat{H}\phi}\hat{\rho}_0 e^{+i\hat{H}\phi}$, where the density matrix $\hat{\rho}_0$ refers to the initial state of the system. The goal is to estimate ϕ via measurements on $\hat{\rho}(\phi)$. The quantum Cramér-Rao bound provides the following lower bound to the achievable precision:

$$(\Delta\phi)^2 \geq \frac{1}{nF_Q[\hat{\rho}(\phi)]}. \tag{120}$$

In Eq. (120) n represents number of samples, $F_Q[\hat{\rho}(\phi)]$ is the quantum Fisher information, and $(\Delta\phi)^2$ is the variance of the estimation of ϕ. Common parameters of interest are temperature or the strength of magnetic fields.

Notice that the precision of the estimation procedure increases as the quantum Fisher information increases. Using it as a cost function, recent works have explored quantum metrology to prepare a better probe state in a VQA setup (Kaubruegger *et al.*, 2019; Beckey *et al.*, 2020; Koczor *et al.*, 2020; Ma *et al.*, 2020). In addition, Meyer, Borregaard, and Eisert (2020) provided a toolbox for multiparameter estimation and Haug and Kim (2021a) provided the natural PQC with the lowest possible quantum Cramér-Rao bound for a general class of circuits. For a survey of recent applications of Fisher information for NISQ computing, see Bharti (2021) and Meyer (2021).

4. Fidelity estimation

In Sec. II.A, we discussed how to use the fidelity as an objective function, a quantity that is useful for training particular VQAs. In addition, estimating the fidelity of a quantum state with respect to another state is of general interest in the context of quantum computing. For this reason, algorithms that can estimate fidelity may become useful in the NISQ era.

Given the density matrices of two quantum states ρ_1 and ρ_2, their fidelity is given by

$$F(\rho_1, \rho_2) = \left(\mathrm{Tr}\sqrt{\sqrt{\rho_1}\rho_2\sqrt{\rho_1}}\right)^2. \tag{121}$$

Owing to the large dimensionality of the Hilbert spaces, computing fidelity can be challenging.

Recently the variational quantum fidelity estimation (VQFE) algorithm was proposed to tackle a slightly modified version of the fidelity estimation task that works efficiently when ρ_1 has a low rank. Cerezo, Poremba *et al.* (2020) provided lower and upper bounds on $F(\rho_1, \rho_2)$ via VQFE. The algorithm calculates the fidelity between ρ_1^n, which is a truncated version of ρ_1 obtained by projecting ρ_1 to subspace spanned by the n largest eigenvalue eigenvectors of ρ_1. The bounds improve monotonically with n and are exact for $n = \mathrm{rank}(\rho_1)$. The VQFE algorithm proceeds in three steps: (i) a variational diagonalization of ρ_1, (ii) computing the matrix elements of ρ_2 in the eigenbasis of ρ_1, and (iii) using the matrix elements from (ii) to estimate the fidelity.

5. Quantum error correction

The leading error-correction schemes carry high resource overheads, which renders them impractical for near-term devices (Fowler *et al.*, 2012; Johnson *et al.*, 2017). Moreover, many of the schemes mandate knowledge of the underlying noise model (Fletcher, Shor, and Win, 2008; Kosut, Shabani, and Lidar, 2008; Kosut and Lidar, 2009). For an encoding process $\mathcal{E}$, decoding process $\mathcal{D}$, and noise model $\mathcal{N}$, the quality of a quantum error-correction scheme can be characterized by how close $\mathcal{D}\circ\mathcal{N}\circ\mathcal{E}$ is to the identity. The range of $\mathcal{E}$ is called code space $\mathcal{C}$.

Johnson *et al.* (2017) proposed the quantum variational error corrector (QVECTOR) by defining an objective function over the code space $\mathcal{C}$. They employed two trainable PQCs $V(\boldsymbol{p})$ and $W(\boldsymbol{q})$ for encoding and decoding, respectively, with tunable parameter vectors $\boldsymbol{p}$ and $\boldsymbol{q}$. For a given encoding-decoding pair, characterized by $(\boldsymbol{p},\boldsymbol{q})$, they calculated a quantity called "average code fidelity" with respect to the Haar distribution of states over the code space $\mathcal{C}$. The algorithm is model free, as no assumption of the noise model is made. The goal of the QVECTOR algorithm is to maximize the average code fidelity in a variational setup.

In the context of VQA, error correction was also explored by Xu *et al.* (2019), who encoded the target logical states as a ground state of an appropriate Hamiltonian. Xu *et al.* (2019) employed imaginary time evolution to find the ground state. They implemented a scheme for five- and seven-qubit codes. For a discussion of error correction and quantum fault tolerance, see Sec. VIII.B.

6. Nuclear physics

The standard model of particle physics is the theory that describes the nature of the electromagnetic and nuclear interaction. Its current formulation consists of describing the forces as quantum fields via the quantum field theory (QFT) formalism. Perturbative calculations of QFT provide the dynamics of physical processes at a given energy scale. However, in some cases, as in quantum chromodynamics (QCD), perturbation theory cannot be applied owing to the impossibility of observing free quarks or gluons due to confinement. For this reason, QCD calculations are obtained by means of numerical methods such as Monte Carlo simulations in a discretized version of QFT on a lattice structure (LQFT). The high computational cost of LQFT has resulted in the use of quantum computation and quantum simulation to obtain the desired QCD predictions (Joó *et al.*, 2019).

The Schwinger model describes the dynamics of the QED interaction in one spatial and temporal dimensions. It is used as a toy model to study QCD since it shows fermion confinement but is simple enough to be solved analytically. The experimental quantum simulations of this model were carried out first in trapped ions (Hauke *et al.*, 2013) and later with a superconducting circuit quantum computer (Martinez *et al.*, 2016). The first proposal to use a quantum-classical algorithm to simulate this model was presented by Klco *et al.* (2018), who had the quantum computer simulate the dynamics of the symmetry sectors suggested by a classical computation. Kokail *et al.* (2019) used a VQS in an analog setup to reduce the number of variational parameters and thus reduce the computational cost of the algorithm. Their proposal was experimentally implemented in a trapped-ion analog simulator. A significant reduction of the computational cost of LQFT was proposed by Avkhadiev, Shanahan, and Young (2020) by using a VQA approach to compute the *optimized interpolating operators*, which are approximators of the quantum state wave function.

Adaptations of the UCC quantum chemistry *Ansatz* introduced in Sec. II.B.1 were presented by Dumitrescu *et al.* (2018) and Liu and Xin (2020) for the quantum-variational QCD and by Roggero *et al.* (2020) for neutrino-nucleus scattering. A ten-qubit VQC was used by Wu *et al.* (2020) to study the Higgs boson production processes and decay. Chen, Wei *et al.* (2020) proposed a quantum convolutional neural network model to study basic high-energy processes. Recently PQCs were used to learn the parton distribution function of protons (Pérez-Salinas, Cruz-Martinez *et al.*, 2020).

7. Entanglement properties

Entanglement is a resource for numerous quantum information tasks. A bipartite quantum state $\rho_{AB}\in\mathcal{H}_A\otimes\mathcal{H}_B$ is called separable if it admits the form $\rho_{AB}=\sum_i p_i\rho_i^A\otimes\rho_i^B$, where p_i are non-negative and $\sum_i p_i=1$. If a state is not separable, then it is called entangled. The problem of detecting whether a state is separable or entangled is known as the separability problem and has been shown to be NP hard (Gurvits, 2003).

As mentioned in Sec. VI.D.2, computing the Schmidt rank of ρ_{AB} gives a measure of the bipartite entanglement. Thus, the algorithms that tackle the SVD problem can also be used to extract entanglement properties (Bravo-Prieto, García-Martín, and Latorre, 2020). Wang, Song *et al.* (2020) proposed a NISQ algorithm for the separability problem by providing a variational approach to employ the *positive map criterion*. This criterion establishes that the quantum state ρ_{AB} is separable if and only if, for an arbitrary quantum system R and an arbitrary positive map $\mathcal{N}_{B\to R}$ from B to R, we have $\mathcal{N}_{B\to R}(\rho_{AB})\geq 0$. Wang, Song *et al.* (2020) started with a positive map and decomposed it into a linear combination of NISQ implementable operations. These operations were executed on the target state and the minimal eigenvalue of the final state was variationally estimated. The target state was deemed entangled if the optimized minimal eigenvalue were negative.

Exploring a strategy similar to the one presented by Bravo-Prieto, García-Martín, and Latorre (2020), Pérez-Salinas, García-Martín *et al.* (2020) proposed a VQA to compute the tangle, which is a measure of tripartite entanglement. VQAs were also employed for extracting the entanglement spectrum of quantum systems by LaRose *et al.* (2019) and Cerezo, Sharma *et al.* (2020).

VII. BENCHMARKING

One of the central questions at the intersection of software and hardware is how to evaluation the performance and capabilities of NISQ devices. Here benchmarking concepts provide various metrics to measure and compare the capabilities of different machines and track their change over time.

A benchmarking protocol can be characterized by its inherent assumptions, resource costs, and information gain. The goal is to build benchmarking protocols that make minimal and practical assumptions, have low resource costs, and have high information gain.

Benchmarking protocols have been developed for NISQ as well as for fault-tolerant devices. For a pedagogical summary, see Eisert *et al.* (2019). In this review, we focus on quantum benchmarking protocols for NISQ devices. Some of the leading NISQ benchmarking schemes are randomized benchmarking, quantum volume, cross-entropy benchmarking, and application-based benchmarks.

A. Randomized benchmarking

The most straightforward way to compare devices is by simply counting qubits. To really compare different qubits, we must also have a sense of how many operations we can do with them before the noise arising from errors drowns out the signal. Randomized benchmarking (RB) is a convenient method for finding average error rates for quantum operations, particularly for single- and two-qubit gates (Magesan, Gambetta, and Emerson, 2011; Magesan, Gambetta, and Emerson, 2012). RB is robust against state preparation and measurement (SPAM) errors and, unlike tomography, admits an efficient and practical implementation.

RB involves the following assumptions: (i) For every gate, the incurred noise is independent of other Clifford gates. (ii) The involved unitaries should constitute a 2-design (see Sec. IV.A) and should not be universal. In other words, no T gates are allowed. (iii) During the experiment, there is no drifting in the noise processes. (iv) One can describe noise processes using completely positive trace-preserving maps.

A RB protocol starts by sampling a sequence of m *Clifford* gates; see Sec. V.B.1. The sequence is applied to the initial state, followed by its inverse. Finally, a two-outcome positive operator-valued measure (POVM) measurement is performed to calculate the fidelity between the initial state and the output state, followed by classical postprocessing. The RB protocol discretizes time so that it is measured in the number of gates and then averages over many sequences of each length m. More formally, a four-step RB protocol consists of the following.

(1) Generate K_m sequences of m quantum operations C_{i_j} with $i \in [1, m]$ and $j \in [1, K_m]$, where i indexes over the sequence of operations and j indexes over the statistical samples. These operations are randomly chosen from the Clifford group, and a $(m+1)$th operation is chosen that cancels the first m operations such that the net operation is the identity. The operations can be chosen from the 2-, 4-, or 2^n-dimensional Clifford group, depending on whether we are benchmarking single-, two-, or n-qubit operations (McKay *et al.*, 2019). These operations come with a certain error, which is modeled with linear operators $\Lambda_{i_j,j}$. The full sequence of m operations is given by

$$S_{K_m} = \bigcirc_{j=1}^{m+1} (\Lambda_{i_j,j} \circ C_{i_j}). \qquad (122)$$

In Eq. (122) $\circ$ denotes composition and $\bigcirc$ represents composition of the terms defined with index j.

(2) For each sequence we find the fidelity with the initial state by measuring $\mathrm{Tr}\{E_\psi S_{K_m}[\rho(\psi)]\}$, where $\rho(\psi)$ is the initial state (with preparation errors) and E_ψ is a POVM measurement operator corresponding to the measurement including noise. Without noise, this would be the projector $E_\psi = |\psi\rangle\langle\psi|$.

(3) Average over the K_m statistical samples to find the sequence fidelity $F(m,\psi) = \mathrm{Tr}\{E_\psi S_m[\rho(\psi)]\}$, where S_m is the mean over the operations S_{K_m}.

(4) Fit the data with the function

$$F_{\mathrm{fit}}(m,\psi) = A_0 p^m + B_0, \qquad (123)$$

where we assume that the errors are independent of gate and time. This is not a fundamental assumption but can be relaxed (Magesan, Gambetta, and Emerson, 2011, 2012). The average gate error here is given by $\epsilon_{\mathrm{RB}} = 1 - p - (1-p)/2^n$ and the constants A_0 and B_0 absorb the SPAM errors.

The operations C_{i_j} are chosen from the Clifford group because these are relatively easy to perform on quantum hardware, and because the final $(m+1)$th operation that undoes the sequence can easily be precomputed on a classical computer. Averaging over the Clifford group (or any other finite group) also has the property that even though the real noise channel would be more complicated than the purely depolarizing one, the average over the group will still give rise to exponential decay.

These gate errors extracted from randomized benchmarking can be used to compare the quality of quantum gates, and to estimate that an algorithm of depth $\sim 1/\epsilon_{\mathrm{RB}}$ gates can be run on the device before the output is only statistical noise. The intuition behind the RB protocol is that a purely depolarizing channel will cause exponential decay of an excited state over time.

Simultaneous randomized benchmarking has been proposed to acquire information about cross talk and undesired coupling between neighboring qubits (Gambetta *et al.*, 2012). RB has also been extended for gate sets that do not form a Clifford group (Gambetta *et al.*, 2012; Carignan-Dugas, Wallman, and Emerson, 2015; Cross *et al.*, 2016; Harper and Flammia, 2017; Brown and Eastin, 2018; França and Hashagen, 2018; Hashagen *et al.*, 2018). In such cases, the expression for $F_{\mathrm{fit}}(m,\psi)$ does not follow Eq. (123) (Helsen *et al.*, 2019). An extension of RB has been suggested to extract the fidelity for a broad category of gate sets, including T gate, using principles from representation theory (Helsen *et al.*, 2019). A practically scalable protocol called cycle benchmarking was developed lately to characterize local and global errors for multiqubit quantum computers (Erhard *et al.*, 2019).

B. Quantum volume

To further refine the concept of the computational power of a quantum computer from just counting qubits and gate errors, the IBM Quantum team introduced the "quantum volume" (Moll *et al.*, 2018; Cross *et al.*, 2019). It is one of the widely accepted metrics for benchmarking NISQ-era quantum computers. As mentioned earlier, one cannot rank quantum computers based on the number of qubits alone. Quantum volume gives a rough estimate of the number of effective qubits that a quantum computer has based on their performance on the "heavy output generation problem." The heavy output generation problem is related to the random circuit sampling task used in Google's quantum supremacy experiment. Quantum volume treats the depth and width of a quantum circuit on the same footing. Hence, its estimation depends on the largest square-sized circuit, which can successfully implement the heavy output generation problem. The quantum computer's performance also depends on its software stack, such as the compiler, and thus quantum volume can increase with improvements in the software stack.

The quantum volume benchmark can be considered analogous to the classical LINPACK benchmark (Dongarra, 1987). Like the LINPACK benchmark, it is architecture agnostic and provides a single real number metric based on the quantum computer's performance for a model problem such as the heavy output generation problem.

More formally, quantum volume can be defined in the following terms. Given an n-qubit quantum computer with the largest achievable *model circuit* depth $d(m)$ for model circuit width $m \in \{1, 2, \dots, n\}$ such that the probability of observing a *heavy output* for a random selection of model circuits is strictly greater than $2/3$, the quantum volume V_Q is defined as (Cross *et al.*, 2019)

$$\log_2 V_Q = \underset{m}{\operatorname{argmax}} \min [m, d(m)]. \tag{124}$$

Intuitively speaking, quantum volume estimates the largest square random quantum circuit for which the quantum computer can successfully implement the so-called heavy output generation problem. To conclude the discussion, it remains to describe the model circuit and the heavy output generation problem.

The model circuit with depth d and width m for estimating quantum volume is given by the d-layered sequence $U = U^{(d)} U^{(d-1)} \cdots U^{(1)}$, where layer t consists of random permutations $\pi_t \in S_m$ applied to qubit labels, followed by the tensor product of Haar-random two-qubit unitaries from SU(4). If the model circuit width m is odd, one of the qubits is left idle in every layer; see Fig. 7 for a pictorial description.

Given a model circuit U with width m, the ideal output distribution over bit strings $x \in \{0, 1\}^m$ is given by $P_U(x) = |\langle x|U|0\rangle|^2$. One can arrange the probabilities for various bit strings in ascending order in a set $\mathbb{P} = \{p_0 \leq p \leq \cdots \leq p_{2^m - 1}\}$. The median of the set $\mathbb{P}$ is given by $p_{\text{med}} = (p_{2^{m-1}} + p_{2^{m-1}-1})/2$. The heavy outputs are defined as $H_U = \{x \in \{0, 1\}^m | p_U(x) > p_{\text{med}}\}$. The goal of the heavy output problem is to sample a set of strings such that at least $2/3$ are heavy output. For an ideal quantum circuit, the expected heavy output probability asymptotically tends to ~0.85. For a completely depolarized device, it is ~0.5.

On the target system, one implements $\tilde{U}$ by using a circuit compiler with a native gate set such that $1 - F_{\text{avg}}(U, \tilde{U}) \leq \epsilon \leq 1$ for an approximation error ϵ, where F_{avg} is the average gate fidelity as defined by Horodecki, Horodecki, and Horodecki (1999). The role of the circuit compiler is crucial in the aforementioned step. Suppose that the observed distribution for the implemented circuit $\tilde{U}$ of the model circuit U is $g_u(x)$. The probability of sampling a heavy output is given by

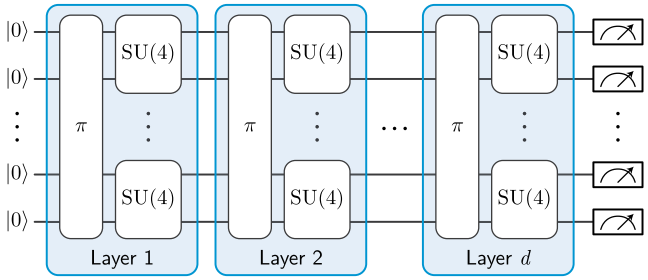

FIG. 7. Model circuit for the quantum volume benchmark. Each layer consists of random permutations of qubit labels, followed by two-qubit Haar-random unitaries. Adapted from Cross *et al.*, 2019.

$$h_U = \sum_{x \in H_U} q_U(x). \tag{125}$$

For a randomly selected circuit of depth d, the probability of sampling a heavy output is given by

$$h_d = \int_U h_U dU. \tag{126}$$

The term $d(m)$ in Eq. (124) is equal to the largest depth d for a model circuit of width $m \in \{1, 2, \dots, n\}$ such that $h_d > 2/3$.

The quantum volume benchmark requires one to simulate the model circuit's heavy output generation problem on a classical computer. Hence, it is not a scalable method as the quantum volume increases. Moreover, the special treatment for square circuits is not entirely justified. Investigations are needed to devise other interesting benchmarks. A benchmark for rectangular circuits has also been proposed in the literature (Blume-Kohout and Young, 2020).

To date Honeywell's system model H1 has achieved $\log_2 V_Q = 9$,[3] and the IBM Quantum device named "IBM Montreal" has demonstrated $\log_2 V_Q = 6$.[4]

C. Cross-entropy benchmarking

The linear cross-entropy benchmarking is a statistical test used by Google in their quantum supremacy experiment (Neill *et al.*, 2018; Arute *et al.*, 2019). It measures how often high-probability bit strings are sampled in an experimental scenario. Suppose that we perform a sampling task and obtain bit strings $\{x_j\}_j$ via measurements on a given m-qubit circuit $\mathcal{C}_E$. The linear cross-entropy benchmarking fidelity is given by

$$\mathcal{F}_{XEB} = 2^m \langle P(x_j) \rangle_j - 1. \tag{127}$$

[3] See https://www.honeywell.com/us/en/news/2021/07/honeywell-sets-another-record-for-quantum-computing-performance.

[4] See https://t.ly/hoPX.

In Eq. (127) the average $\langle\cdot\rangle_j$ is over the experimentally observed bit strings $\{x_j\}_j$ and $P(x_j)$ denotes the probability of observing bit string x_j for the ideal circuit version of $\mathcal{C}_E$. In other words, $P(x_j)$ denotes the ideal probability of the generated sample x_j. Since one cannot have an ideal circuit in practice, $P(x_j)$ are calculated using a classical computer simulation of the ideal circuit. $\mathcal{F}_{XEB}$ compares how often a bit string x_j is observed experimentally with its classically simulated ideal probability. For the ideal case, $\mathcal{F}_{XEB}$ approaches unity for a large number of qubits. On the other hand, it is equal to zero for uniform distribution. As the circuit's noise grows, F_{XEB} decreases and approaches zero. Since the probabilities $P(x_j)$ are calculated via classical simulation, it renders the computation of $\mathcal{F}_{XEB}$ intractable in the supremacy regime. The classical hardness of spoofing linear cross-entropy benchmarking was studied by Aaronson and Gunn (2019), who suggested the absence of any efficient classical algorithms for the aforementioned task.

D. Application benchmarks

While hardware benchmarks such as randomized benchmarking or quantum volume provide valuable insight into the performance of quantum devices, they may not well represent or predict the performance of VQAs which employ structured circuits. Application benchmarks were developed to complement hardware benchmarks and provide a more complete picture of both the performance and the near-term utility of quantum devices. These benchmarks execute experimental demonstrations of VQA instances that can be compared to classically computed exact results. Examples of application benchmarks were given by Dallaire-Demers and Killoran (2018), Benedetti, Garcia-Pintos *et al.* (2019), Arute *et al.* (2020), and Karamlou *et al.* (2020). In particular, Arute *et al.* (2020) demonstrated VQE experiments for the hydrogen chain binding energy and diazene isomerization mechanism with PQCs of up to 12 qubits and 72 two-qubit gates.

As a specific example of an application benchmark, quantum circuits that diagonalize spin Hamiltonians were proposed in recent years (Verstraete, Cirac, and Latorre, 2009; Schmoll and Orús, 2017; Cervera-Lierta, 2018). By comparing the results obtained from the quantum device to the analytical solution, one can discern the performance of the computation for a specific purpose experiment. Small experiments have shown that gate fidelities and decoherence times alone do not provide a complete picture of the noise model (Cervera-Lierta, 2018).

In that vein, Dallaire-Demers *et al.* (2020) proposed a figure of merit called the *effective fermionic length* to quantify the performance of a NISQ device in which the application at hand is estimating the energy density of the one-dimensional Fermi-Hubbard model over increasing chain lengths. Theoretically, as the chain length increases the energy density should approach the infinite chain limit. In practice, the NISQ device will accrue some level of noise and decoherence, which causes the computed energy density to diverge past a certain chain length. The maximum chain length after which noise and decoherence start degrading the algorithm performance reveals the "limit" of the quantum device in carrying out related algorithms. Dallaire-Demers *et al.* (2020) abstracted this idea to redefine an application benchmark as a way to systematically test the limits of quantum processors using exactly solvable VQA instances that can be scaled to larger system sizes such as the chain length given by Dallaire-Demers *et al.* (2020) or the number of preprocessing steps given by Karamlou *et al.* (2020).

Generative models such as the QCBM (see Sec. VI.B.2) can also serve as benchmarks for NISQ devices (Hamilton, Dumitrescu, and Pooser, 2019; Leyton-Ortega, Perdomo-Ortiz, and Perdomo, 2019; Zhu *et al.*, 2019). Here the measurement output of hardware-efficient variational *Ansätze* are used to represent different types of distributions and study the effect of noise and hardware limitations on the result.

In addition to VQAs, one can analyze more fundamental benchmarks, such as the ability of NISQ devices to violate local realism by means of Mermin inequalities (Alsina and Latorre, 2016) or the entanglement power of the devices by trying to construct maximal entangled states (Wang, Li *et al.*, 2018; Cervera-Lierta, Latorre, and Goyeneche, 2019).

VIII. OUTLOOK

In the last decade, quantum computing has experienced notable progress in applications, experimental demonstrations, and theoretical results. The number of papers in quantum computation, particularly in NISQ applications, is increasing at a nearly exponential rate. This community push is explained by many factors, one of which is noteworthy advancements in quantum hardware.

Quantum computing is a relatively young field in science, and as such there is ample room for pioneering research and discoveries. Together with the theoretical, practical, and experimental challenges (several of them covered in this review), this fact has strengthened the motivation for an open-source strategy in the field. Many universities and research centers currently subscribe to an open-access policy that pushes toward the free and open-source publication of all computational tools, data, and programs used in their research. These policies have proven to be valuable for rapid scientific development as well as for democratizing community knowledge. This way of thinking has percolated through academic walls. It has been introduced at several private companies, not just for its advantage but also because it facilitates the continuous healthy flow of quantum computing researchers to themselves (and in some cases resulting in the foundation of start-ups). Consequently, a rich open quantum computing ecosystem is composed of universities, institutes, large corporations, start-ups, and uncountable individual enthusiasts. Another product of the symbiosis between academia and the private sector is *cloud quantum computing*. Companies are offering access to their hardware remotely, in some cases at zero cost, for their small prototypes and simulators. On the one hand, scientists and quantum computing enthusiasts around the world have the opportunity to experience real quantum devices from their homes. On the other hand, this increases the chances of finding real-world applications in quantum computation and solving the current challenges of the field. The proliferation of open-source quantum computing languages, simulators, and tools (detailed in Sec. V.C) have burgeoned many user communities. Various international initiatives have been set up to attract quantum computing talent, and the private sector's involvement is ramping up. Several nonprofit initiatives are also encouraging the use and development of these tools (QOSF, 2020b).[5]

Experimental realizations of quantum computation, although in the early stages, have interested many communities in this quantum information subfield. Healthy competition has also arisen between the classical and quantum computing branches. Classical computational scientists have put their efforts into moving the quantum advantage frontier further, raising the bar to claim that a quantum algorithm shows a significant speedup. In that regard, an offshoot is an effort at *dequantization*, first exhibited in the case of recommendation systems, to devise quantum-inspired classical algorithms that are nearly as fast as their quantum counterparts (Tang, 2019). Such attempts have eliminated examples of speedup for some problems in linear algebra. Thus far dequantized machine learning algorithms have been developed for recommendation systems (Tang, 2019), principal component analysis and supervised clustering (Tang, 2021), stochastic regression (Gilyén, Lloyd, and Tang, 2018), and low-rank linear systems (Chia, Lin, and Wang, 2018; Arrazola *et al.*, 2019).

Since NISQ devices are inherently noisy, analysis similar to that given by Napp *et al.* (2019), Zhou, Stoudenmire, and Waintal (2020), and Zlokapa, Boixo, and Lidar (2020) will be required to find out how much noise a NISQ algorithm can endure until its classical simulation becomes efficient. This is crucial in order to understand the boundary where quantum computers provide an advantage. Investigating the potential of NISQ algorithms using ideas from quantum foundations such as contextuality and entanglement is helpful in that respect (Bharti *et al.*, 2020; Deutsch, 2020). More theoretical results such as the ones presented by Farhi and Harrow (2016), Lloyd (2018), Movassagh (2019), Bravyi, Gosset *et al.* (2020), Biamonte (2021), Bouland *et al.* (2021), and Bravyi, Gosset, and Movassagh (2021) may also prove valuable. In addition, it is imperative to develop strategies that help us bypass complicated measurements involving controlled multiqubit unitaries (Mitarai and Fujii, 2019). For machine learning tasks, ideas similar to those of Harrow (2020) would be valuable.

Another interesting frontier that we believe needs to be investigated in the next few years involves quantum and classical certification schemes for quantum devices and quantum computation (Eisert *et al.*, 2020). The intractability of quantum computation by classical devices poses the challenge of verifying the correct functioning of the quantum devices as well as the correctness of the final output (Eisert *et al.*, 2020). The existence of multiple quantum computing platforms requires new methodologies and figures of merit to benchmark and compare these devices. Other works are being proposed in that regard (Kottmann *et al.*, 2020; Kyaw *et al.*, 2021), as well as in the development of the benchmarking measures discussed in Sec. VII. Ideas from complexity theory (Mahadev, 2018; Metger and Vidick, 2020) and quantum foundations (Bharti, Ray, Varvitsiotis, Cabello, and Kwek, 2019; Bharti, Ray, Varvitsiotis, Warsi *et al.*, 2019) could be valuable in this effort.

Currently there is no known demonstration of industrially relevant quantum advantage. Quantum computing is still in its early days, and a useful quantum computer is still missing. The potential of NISQ devices is not fully understood, and a lot of rigorous research is required to release the power of early quantum computers. However, several experiments overcoming classical computational resources have been performed and many theoretical and practical tools are being used and developed, as explained in Sec. V.

A. NISQ goals

We expect that experimental pursuit in the NISQ era would focus on the design of quantum hardware with a larger number of qubits and gates with lower error rates capable of executing deeper circuits. Along the way, one of the goals is to demonstrate a quantum advantage for practical use cases. If the NISQ paradigm were not powerful enough to exhibit any quantum advantage, theoretical pursuits would be required to understand its limitations. The prime direction of the NISQ and near-term era is to engineer the best possible solution with the limited quantum resources available. The tools and techniques invented during this period could be valuable in the fault-tolerant era as well.

To conduct a successful demonstration of quantum advantage, the right blend of the following three crucial components is required:

(1) *Hardware development*.—The design of quantum computers with more qubits, lesser error rates, longer coherence times, and more connectivity between the qubits will be one of the top priorities in the NISQ era. Intensive research in new qubit developments, quantum optimal control, and material discovery will be indispensable for both universal programmable quantum computers and special-purpose ones. A way to scale up the number of qubits present in a quantum platform is to design a novel qubit that has built-in autonomous quantum error correction down to the hardware level (Paz and Zurek, 1998; Chamberland *et al.*, 2020) or a protected novel qubit (Nataf and Ciuti, 2011; Douçot and Ioffe, 2012; Kyaw, 2019), which is robust against specific noises in the hardware. As a quantum processor's size grows, there is a tremendous need to store quantum information during quantum information processing (Kyaw *et al.*, 2014; Kyaw, Felicetti *et al.*, 2015; Kyaw, Herrera-Marti *et al.*, 2015). Even miniaturizing a microwave circulator onto the superconducting chip (Chapman *et al.*, 2017; Mahoney *et al.*, 2017) can be seen as a means to scale up the quantum platform. However, it has nothing to do with novel qubit design.

(2) *Algorithm design*.—To harness the potential of noisy but powerful quantum devices, we expect breakthroughs on the algorithm frontier. Algorithms with realistic assumptions regarding device capabilities, like the ones mentioned in Sec. V.B, will be favored.

[5] See https://unitary.fund/.

To lessen the effect of noise, progress toward the design of error-mitigation algorithms is expected. Efforts have to be made to develop algorithms that harness the problem's structure in the best possible manner and map it to the given hardware in efficient ways, such as in Sec. III.D. VQA with better expressibility and trainability will also be helpful.

(3) *Application problem.*—We discussed the existing applications of NISQ devices in many areas in Sec. VI. Collaborations between experts with domain knowledge from these fields and quantum algorithm researchers will be increasingly required to develop the field and integrate quantum computation into industrial workflows. New collaborations might reveal difficult problems for classical computers that are well suited for NISQ devices. It is not yet clear which applications will be the first ones to witness a quantum advantage, although there is plenty of speculation and opinions.

B. Long-term goal: Fault-tolerant quantum computing

Noise is regarded as one of the most prominent threats to a quantum computer's practical realization. In 1995, Peter Shor established that by encoding quantum information redundantly using extra qubits one could circumvent the effect of noise (Shor, 1995). The quantum information is spread over multiple physical qubits to generate a logical qubit (Shor, 1995; Calderbank and Shor, 1996; Gottesman, 1997; Knill and Laflamme, 1997). Most of the transformative algorithms, such as Shor's factoring algorithm, the Grover search algorithm, and HHL, require error-corrected qubits for their execution. Soon after the appearance of Shor's error-correcting code, many others were developed. Some of the famous error-correcting codes are stabilizer and topological error-correcting codes (Fowler *et al.*, 2012; Terhal, 2015). While the stabilizer code utilizes extra qubits to protect the logical qubit, topological codes employ a set of qubits positioned on a surface, such as a torus, in a lattice structure.

Over the years, quantum error correction has evolved as a subfield of quantum computation and has transformed from a theoretical pursuit to a practical possibility. The process of detecting and correcting errors can itself be prone to noise. Thus, error correction alone does not guarantee the prospect of storing or processing quantum information for an arbitrarily long period. The aforesaid issue can be tackled by utilizing the *quantum fault-tolerant threshold theorem*. Informally speaking, it is possible to execute arbitrarily large quantum computation by randomly suppressing the quantum error rate, given that the noise in the individual quantum gates is below a certain threshold (Aharonov and Ben-Or, 2008). If one wants to simulate an ideal circuit of size N, the size of the noisy quantum circuit for fault-tolerant quantum computation scales as $O[N(\log N)^c]$ for a given constant c given that the noisy circuit is subjected to stochastic noise strength $p < p_c$ for a certain noise threshold p_c (Terhal, 2015). This theorem rises some practically relevant questions, such as (i) how high is p_c, (ii) what is the value of the constant c, and (iii) what is the value of the multiplicative constant in $O(\cdot)$. These questions determine the practicality of any fault-tolerant quantum computation scheme (Terhal, 2015).

As we look forward, lowering the noise level will be a critical challenge. Although the problem is demanding, significant progress has been made recently at the algorithmic as well as the hardware frontier (Lidar and Brun, 2013; Terhal, 2015; Campagne-Ibarcq *et al.*, 2020; Noh and Chamberland, 2020). Quantum error-correcting codes amenable to architectures with limited qubit connectivity have also been proposed (Chamberland, Zhu *et al.*, 2020). As we transition toward fault-tolerant quantum computing, partial quantum error-correction demonstrations such as the exponential suppression of bit or phase errors (Google, 2021) and approximate quantum error-correction schemes (Leung *et al.*, 1997; Faist *et al.*, 2020) become highly relevant. Recently Monroe's and Brown's groups have confirmed the first-ever fault-tolerant operation on a logical qubit (Egan *et al.*, 2020).

We are at an exciting juncture in the history of computing. Completely new kinds of computers that were once only figments of the imagination are rapidly becoming a reality. The NISQ era offers excellent opportunities to current and future researchers to explore the theoretical limits of these devices and discover practical and exciting applications in the near term. Theoretical investigations and experimental challenges will help us to comprehend quantum devices' power and build better algorithms. The success of the field lies in the hands of the researchers and practitioners of the area, so we encourage everyone with interest to join the effort.

ACKNOWLEDGMENTS

We thank Michael Biercuk, Naresh Boddu, Zhenyu Cai, Edward Farhi, Sam Gutmann, Rahul Jain, Dax Koh, Alejandro Perdomo-Ortiz, and Mark Steudtner for the interesting discussions, feedback, and comments. A. A.-G. acknowledges the generous support of Google, Inc., in the form of a Google Focused Award. This work was supported by the U.S. Department of Energy under Award No. DESC0019374 and the U.S. Office of Naval Research (Grant No. ONS506661). A. A.-G. also acknowledges support from the Canada Industrial Research Chairs Program and the Canada 150 Research Chairs Program. T. H. is supported by a Samsung GRC project and the UK Hub in Quantum Computing and Simulation, part of the UK National Quantum Technologies Programme, with funding from UKRI EPSRC Grant No. EP/T001062/1. L.-C. K and K. B acknowledge the financial support from the National Research Foundation and the Ministry of Education, Singapore.

K. B., A. C.-L., and T. H. K. contributed equally to this work.

REFERENCES

Aaronson, S., 2020, SIAM J. Comput. **49**, STOC18-368.

Aaronson, S., and A. Arkhipov, 2011, in *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing, San Jose, 2011* (Association for Computing Machinery, New York), p. 333.

Aaronson, S., and D. Gottesman, 2004, Phys. Rev. A **70**, 052328.

Aaronson, S., and S. Gunn, 2019, arXiv:1910.12085.

Abbas, A., D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, 2021, Nat. Comput. Sci. **1**, 403.

Acín, A., *et al.*, 2018, New J. Phys. **20**, 080201.

Ackley, D. H., G. E. Hinton, and T. J. Sejnowski, 1985, Cogn. Sci. **9**, 147.

Adams, C. S., J. D. Pritchard, and J. P. Shaffer, 2020, J. Phys. B **53**, 012002.

Aharonov, D., and M. Ben-Or, 2008, SIAM J. Comput. **38**, 1207.

Aharonov, D., W. Van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, 2008, SIAM Rev. **50**, 755.

Aïmeur, E., G. Brassard, and S. Gambs, 2013, Mach. Learn. **90**, 261.

Akshay, V., H. Philathong, M. E. S. Morales, and J. D. Biamonte, 2020, Phys. Rev. Lett. **124**, 090504.

Akshay, V., H. Philathong, I. Zacharov, and J. Biamonte, 2020, arXiv:2007.09148.

Albarrán-Arriagada, F., J. Retamal, E. Solano, and L. Lamata, 2020, Mach. Learn. Sci. Technol. **1**, 015002.

Albash, T., and D. A. Lidar, 2018, Rev. Mod. Phys. **90**, 015002.

Albash, T., T. F. Rønnow, M. Troyer, and D. A. Lidar, 2015, Eur. Phys. J. Special Topics **224**, 111.

Alcazar, J., V. Leyton-Ortega, and A. Perdomo-Ortiz, 2020, Mach. Learn. Sci. Technol. **1**, 035003.

Aleiner, I., *et al.*, 2020, arXiv:2012.00921.

Alexander, T., N. Kanazawa, D. J. Egger, L. Capelluto, C. J. Wood, A. Javadi-Abhari, and D. C. McKay, 2020, Quantum Sci. Technol. **5**, 044006.

Alsina, D., and J. I. Latorre, 2016, Phys. Rev. A **94**, 012314.

Amaral, B., and M. T. Cunha, 2018, *On Graph Approaches to Contextuality and Their Role in Quantum Theory* (Springer, New York).

Amari, S.-I., 1998, Neural Comput. **10**, 251.

Ambainis, A., 2012, https://hal.archives-ouvertes.fr/hal-00678197/.

Amico, L., *et al.*, 2021, AVS Quantum Sci. **3**, 039201.

Amin, M. H., E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, 2018, Phys. Rev. X **8**, 021050.

Amy, M., P. Azimzadeh, and M. Mosca, 2018, Quantum Sci. Technol. **4**, 015002.

Amy, M., D. Maslov, M. Mosca, and M. Roetteler, 2013, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **32**, 818.

Amy, M., and M. Mosca, 2019, IEEE Trans. Inf. Theory **65**, 4771.

Anand, A., M. Degroote, and A. Aspuru-Guzik, 2020, arXiv:2012.00101.

Anand, A., J. Romero, M. Degroote, and A. Aspuru-Guzik, 2020, arXiv:2006.01976.

Anschuetz, E., J. Olson, A. Aspuru-Guzik, and Y. Cao, 2019, in *Proceedings of the International Workshop on Quantum Technology and Optimization Problems, Munich, 2019*, edited by S. Feld and C. Linnhoff-Popien (Springer, New York), p. 74.

Anschuetz, E. R., and Y. Cao, 2019, arXiv:1903.01359.

Argüello-Luengo, J., A. González-Tudela, T. Shi, P. Zoller, and J. I. Cirac, 2019, Nature (London) **574**, 215.

Arimitsu, K., Y. O. Nakagawa, S. Koh, W. Mizukami, Q. Gao, and T. Kobayashi, 2021, arXiv:2107.12705.

Arora, S., and B. Barak, 2009, *Computational Complexity: A Modern Approach* (Cambridge University Press, Cambridge, England).

Arrasmith, A., M. Cerezo, P. Czarnik, L. Cincio, and P. J. Coles, 2020, arXiv:2011.12245.

Arrasmith, A., L. Cincio, R. D. Somma, and P. J. Coles, 2020, arXiv:2004.06252.

Arrasmith, A., L. Cincio, A. T. Sornborger, W. H. Zurek, and P. J. Coles, 2019, Nat. Commun. **10**, 1.

Arrazola, J., *et al.*, 2021, Nature (London) **591**, 54.

Arrazola, J. M., and T. R. Bromley, 2018, Phys. Rev. Lett. **121**, 030503.

Arrazola, J. M., T. R. Bromley, and P. Rebentrost, 2018, Phys. Rev. A **98**, 012322.

Arrazola, J. M., A. Delgado, B. R. Bardhan, and S. Lloyd, 2019, arXiv:1905.10415.

Arunachalam, S., and R. de Wolf, 2017, arXiv:1701.06806.

Arute, F., *et al.*, 2019, Nature (London) **574**, 505.

Arute, F., *et al.*, 2020, Science **369**, 1084.

Aspuru-Guzik, A., A. D. Dutoi, P. J. Love, and M. Head-Gordon, 2005, Science **309**, 1704.

Avkhadiev, A., P. Shanahan, and R. Young, 2020, Phys. Rev. Lett. **124**, 080501.

Ayanzadeh, R., S. Mousavi, M. Halem, and T. Finin, 2019, arXiv:1901.00088.

Azses, D., R. Haenel, Y. Naveh, R. Raussendorf, E. Sela, and E. G. Dalla Torre, 2020, Phys. Rev. Lett. **125**, 120502.

Babbush, R., P. J. Love, and A. Aspuru-Guzik, 2014, Sci. Rep. **4**, 6603.

Babbush, R., A. Perdomo-Ortiz, B. O'Gorman, W. Macready, and A. Aspuru-Guzik, 2012, in *Advances in Chemical Physics*, Vol. 155, edited by S. A. Rice and A. R. Dinner (John Wiley & Sons, New York), p. 201.

Babej, T., *et al.*, 2018, arXiv:1811.00713.

Ball, R., 2005, Phys. Rev. Lett. **95**, 176407.

Banchi, L., and G. E. Crooks, 2020, arXiv:2005.10299.

Banchi, L., M. Fingerhuth, T. Babej, C. Ing, and J. M. Arrazola, 2020, Sci. Adv. **6**, eaax1950.

Banchi, L., N. Quesada, and J. M. Arrazola, 2020, arXiv:2004.04770.

Bang, J., and S. Yoo, 2014, J. Korean Phys. Soc. **65**, 2001.

Barak, B., and K. Marwaha, 2021, arXiv:2106.05900.

Barak, B., A. Moitra, R. O'Donnell, P. Raghavendra, O. Regev, D. Steurer, L. Trevisan, A. Vijayaraghavan, D. Witmer, and J. Wright, 2015, arXiv:1505.03424.

Barenco, A., C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, 1995, Phys. Rev. A **52**, 3457.

Barison, S., D. E. Galli, and M. Motta, 2020, arXiv:2011.08137.

Barison, S., F. Vicentini, and G. Carleo, 2021, Quantum **5**, 512.

Barkoutsos, P. K., G. Nannicini, A. Robert, I. Tavernelli, and S. Woerner, 2020, Quantum **4**, 256.

Barron, G. S., B. T. Gard, O. J. Altman, N. J. Mayhall, E. Barnes, and S. E. Economou, 2020, arXiv:2003.00171.

Bassman, L., *et al.*, 2020, Phys. Rev. B **101**, 184305.

Bastidas, V., T. Haug, C. Gravel, L.-C. Kwek, W. Munro, and K. Nemoto, 2020, arXiv:2009.00823.

Bastidas, V. M., T. H. Kyaw, J. Tangpanitanon, G. Romero, L.-C. Kwek, and D. G. Angelakis, 2018, New J. Phys. **20**, 093004.

Batista, C. D., and G. Ortiz, 2004, Adv. Phys. **53**, 1.

Beckey, J. L., M. Cerezo, A. Sone, and P. J. Coles, 2020, arXiv:2010.10488.

Benedetti, M., M. Fiorentini, and M. Lubasch, 2020, arXiv:2009.12361.

Benedetti, M., D. Garcia-Pintos, O. Perdomo, V. Leyton-Ortega, Y. Nam, and A. Perdomo-Ortiz, 2019, npj Quantum Inf. **5**, 45.

Benedetti, M., E. Lloyd, S. Sack, and M. Fiorentini, 2019, Quantum Sci. Technol. **4**, 043001.

Benedetti, M., J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, 2016, Phys. Rev. A **94**, 022308.

Benedetti, M., J. Realpe-Gómez, R. Biswas, and A. Perdomo-Ortiz, 2017, Phys. Rev. X **7**, 041052.

Benedetti, M., J. Realpe-Gómez, and A. Perdomo-Ortiz, 2018, Quantum Sci. Technol. **3**, 034007.

Bengtsson, A., *et al.*, 2020, Phys. Rev. Applied **14**, 034010.

Benioff, P., 1980, J. Stat. Phys. **22**, 563.

Bharti, K., 2020, arXiv:2009.11001.

Bharti, K., 2021, Quantum Views **5**, 61.

Bharti, K., and T. Haug, 2021a, Phys. Rev. A **104**, L050401.

Bharti, K., and T. Haug, 2021b, Phys. Rev. A **104**, 042418.

Bharti, K., T. Haug, V. Vedral, and L.-C. Kwek, 2019, arXiv:1912.10783.

Bharti, K., T. Haug, V. Vedral, and L.-C. Kwek, 2020, AVS Quantum Sci. **2**, 034101.

Bharti, K., T. Haug, V. Vedral, and L.-C. Kwek, 2021, arXiv:2106.03891.

Bharti, K., M. Ray, A. Varvitsiotis, A. Cabello, and L.-C. Kwek, 2019, arXiv:1911.09448.

Bharti, K., M. Ray, A. Varvitsiotis, N. A. Warsi, A. Cabello, and L.-C. Kwek, 2019, Phys. Rev. Lett. **122**, 250403.

Biamonte, J., 2021, Phys. Rev. A **103**, L030401.

Biamonte, J., P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, 2017, Nature (London) **549**, 195.

Bittel, L., and M. Kliesch, 2021, Phys. Rev. Lett. **127**, 120502.

Björklund, A., B. Gupt, and N. Quesada, 2019, ACM J. Exp. Algorithmics **24**, 1.

Blaauboer, M., and R. De Visser, 2008, J. Phys. A **41**, 395307.

Blank, C., D. K. Park, J.-K. K. Rhee, and F. Petruccione, 2020, npj Quantum Inf. **6**, 41.

Blatt, R., and C. F. Roos, 2012, Nat. Phys. **8**, 277.

Bloch, I., J. Dalibard, and S. Nascimbene, 2012, Nat. Phys. **8**, 267.

Blume-Kohout, R., and K. C. Young, 2020, Quantum **4**, 362.

Bondarenko, D., and P. Feldmann, 2020, Phys. Rev. Lett. **124**, 130502.

Bonet-Monroig, X., R. Babbush, and T. E. O'Brien, 2020, Phys. Rev. X **10**, 031064.

Bonet-Monroig, X., R. Sagastizabal, M. Singh, and T. O'Brien, 2018, Phys. Rev. A **98**, 062339.

Botea, A., A. Kishimoto, and R. Marinescu, 2018, in *Proceedings of the 11th Annual Symposium on Combinatorial Search (SoCS), Stockholm, 2018* (AAAI Press, Palo Alto, CA).

Bouland, A., B. Fefferman, Z. Landau, and Y. Liu, 2021, arXiv:2102.01738.

Bouland, A., and M. Ozols, 2018, in *Proceedings of the 13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC), Sydney, 2018*, Vol. 111, edited by J. Stacey (Schloss Dagstuhl, Dagstuhl, Germany), p. 6:1.

Bouland, A., W. van Dam, H. Joorati, I. Kerenidis, and A. Prakash, 2020, arXiv:2011.06492.

Brady, L. T., C. L. Baldwin, A. Bapat, Y. Kharkov, and A. V. Gorshkov, 2020, arXiv:2003.08952.

Brady, L. T., and W. van Dam, 2016, Phys. Rev. A **94**, 032309.

Brandao, F. G. S. L., M. Broughton, E. Farhi, S. Gutmann, and H. Neven, 2018, arXiv:1812.04170.

Bravo-Prieto, C., 2020, arXiv:2010.06599.

Bravo-Prieto, C., D. García-Martín, and J. I. Latorre, 2020, Phys. Rev. A **101**, 062310.

Bravo-Prieto, C., R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, 2019, arXiv:1909.05820.

Bravo-Prieto, C., J. Lumbreras-Zarapico, L. Tagliacozzo, and J. I. Latorre, 2020, Quantum **4**, 272.

Bravyi, S., D. P. DiVincenzo, D. Loss, and B. M. Terhal, 2008, Phys. Rev. Lett. **101**, 070503.

Bravyi, S., J. M. Gambetta, A. Mezzacapo, and K. Temme, 2017, arXiv:1701.08213.

Bravyi, S., D. Gosset, R. Koenig, and M. Tomamichel, 2020, Nat. Phys. **16**, 1040.

Bravyi, S., D. Gosset, and R. Movassagh, 2021, Nat. Phys. **17**, 337.

Bravyi, S., A. Kliesch, R. Koenig, and E. Tang, 2019, arXiv:1910.08980.

Bravyi, S., A. Kliesch, R. Koenig, and E. Tang, 2020, arXiv:2011.13420.

Bravyi, S. B., and A. Y. Kitaev, 2002, Ann. Phys. (Amsterdam) **298**, 210.

Brennen, G. K., 2003, arXiv:quant-ph/0305094.

Breuer, H.-P. and F. Petruccione, 2002, *The Theory of Open Quantum Systems* (Oxford University Press on Demand, New York).

Briegel, H. J., D. E. Browne, W. Dür, R. Raussendorf, and M. Van den Nest, 2009, Nat. Phys. **5**, 19.

Bromley, T. R., J. M. Arrazola, S. Jahangiri, J. Izaac, N. Quesada, A. D. Gran, M. Schuld, J. Swinarton, Z. Zabaneh, and N. Killoran, 2020, Quantum Sci. Technol. **5**, 034010.

Brown, W. G., and B. Eastin, 2018, Phys. Rev. A **97**, 062323.

Brunner, N., D. Cavalcanti, S. Pironio, V. Scarani, and S. Wehner, 2014, Rev. Mod. Phys. **86**, 419.

Burges, C. J., 2002, Microsoft Research Technical Report No. MSR-TR-2002-83.

Butcher, J. C., 1987, *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods* (Wiley Inter-Science, New York).

Bylaska, E. J., D. Song, N. P. Bauman, K. Kowalski, D. Claudino, and T. S. Humble, 2020, arXiv:2009.00080.

Cade, C., L. Mineh, A. Montanaro, and S. Stanisic, 2020, Phys. Rev. B **102**, 235122.

Cai, Z., 2020, Phys. Rev. Applied **14**, 014059.

Cai, Z., 2021a, npj Quantum Inf. **7**, 1.

Cai, Z., 2021b, arXiv:2101.03151.

Calderbank, A. R., and P. W. Shor, 1996, Phys. Rev. A **54**, 1098.

Campagne-Ibarcq, P., *et al.*, 2020, Nature (London) **584**, 368.

Cao, Y., *et al.*, 2019, Chem. Rev. **119**, 10856.

Cárdenas-López, F. A., L. Lamata, J. C. Retamal, and E. Solano, 2018, PLoS One **13**, e0200455.

Carignan-Dugas, A., J. J. Wallman, and J. Emerson, 2015, Phys. Rev. A **92**, 060302.

Carleo, G., I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, 2019, Rev. Mod. Phys. **91**, 045002.

Carleo, G., and M. Troyer, 2017, Science **355**, 602.

Carvalho, A. R., H. Ball, M. J. Biercuk, M. R. Hush, and F. Thomsen, 2020, arXiv:2010.08057.

Castaldo, D., M. Rosa, and S. Corni, 2020, arXiv:2007.00368.

Céleri, L. C., D. Huerga, F. Albarrán-Arriagada, E. Solano, and M. Sanz, 2021, arXiv:2103.15689.

Cerezo, M., A. Poremba, L. Cincio, and P. J. Coles, 2020, Quantum **4**, 248.

Cerezo, M., K. Sharma, A. Arrasmith, and P. J. Coles, 2020, arXiv:2004.01372.

Cerezo, M., A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, 2021, Nat. Commun. **12**, 1791.

Cervera-Lierta, A., 2018, Quantum **2**, 114.

Cervera-Lierta, A., J. S. Kottmann, and A. Aspuru-Guzik, 2021, PRX Quantum **2**, 020329.

Cervera-Lierta, A., M. Krenn, A. Aspuru-Guzik, and A. Galda, 2021, arXiv:2104.05627.

Cervera-Lierta, A., J. I. Latorre, and D. Goyeneche, 2019, Phys. Rev. A **100**, 022342.

Chamberland, C., G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, 2020, Phys. Rev. X **10**, 011022.

Chamberland, C., *et al.*, 2020, arXiv:2012.04108.

Chang, D. E., V. Vuletić, and M. D. Lukin, 2014, Nat. Photonics **8**, 685.

Chapman, B. J., E. I. Rosenthal, J. Kerckhoff, B. A. Moores, L. R. Vale, J. Mates, G. C. Hilton, K. Lalumiere, A. Blais, and K. Lehnert, 2017, Phys. Rev. X **7**, 041043.

Chen, J., H. I. Nurdin, and N. Yamamoto, 2020, arXiv:2001.09498.

Chen, S., W. Yu, P. Zeng, and S. T. Flammia, 2020, arXiv:2011.09636.

Chen, S. Y.-C., C. H. H. Yang, J. Qi, P. Y. Chen, X. Ma, and H. S. Goan, 2020, IEEE Access **8**, 141007.

Chen, S. Y.-C., T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, 2020, arXiv:2012.12177.

Cheng, S., J. Chen, and L. Wang, 2018, Entropy **20**, 583.

Chia, N.-H., H.-H. Lin, and C. Wang, 2018, arXiv:1811.04852.

Chien, R. W., and J. D. Whitfield, 2020, arXiv:2009.11860.

Childs, A. M., R. Kothari, and R. D. Somma, 2017, SIAM J. Comput. **46**, 1920.

Chivilikhin, D., A. Samarin, V. Ulyantsev, I. Iorsh, A. R. Oganov, and O. Kyriienko, 2020, arXiv:2007.04424.

Choi, J., S. Oh, and J. Kim, 2020, arXiv:2004.11229.

Choi, J.-y., S. Hild, J. Zeiher, P. Schauß, A. Rubio-Abadal, T. Yefsah, V. Khemani, D. A. Huse, I. Bloch, and C. Gross, 2016, Science **352**, 1547.

Choi, V., 2008, Quantum Inf. Process. **7**, 193.

Choi, V., 2011, Quantum Inf. Process. **10**, 343.

Chong, F. T., D. Franklin, and M. Martonosi, 2017, Nature (London) **549**, 180.

Choo, K., C. W. von Keyserlingk, N. Regnault, and T. Neupert, 2018, Phys. Rev. Lett. **121**, 086808.

Choquette, A., A. Di Paolo, P. K. Barkoutsos, D. Sénéchal, I. Tavernelli, and A. Blais, 2020, arXiv:2008.01098.

Chowdhury, A. N., G. H. Low, and N. Wiebe, 2020, arXiv:2002.00055.

Ciliberto, C., M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, 2018, Proc. R. Soc. A **474**, 20170551.

Cirstoiu, C., Z. Holmes, J. Iosue, L. Cincio, P. J. Coles, and A. Sornborger, 2020, npj Quantum Inf. **6**, 82.

Cohen, J., A. Khan, and C. Alexander, 2020, arXiv:2007.01430.

Colless, J. I., V. V. Ramasesh, D. Dahlen, M. S. Blok, M. Kimchi-Schwartz, J. McClean, J. Carter, W. De Jong, and I. Siddiqi, 2018, Phys. Rev. X **8**, 011021.

Commeau, B., M. Cerezo, Z. Holmes, L. Cincio, P. J. Coles, and A. Sornborger, 2020, arXiv:2009.02559.

Consiglio, M., W. J. Chetcuti, C. Bravo-Prieto, S. Ramos-Calderer, A. Minguzzi, J. I. Latorre, L. Amico, and T. J. Apollaro, 2021, arXiv:2106.15552.

Cook, J., S. Eidenbenz, and A. Bärtschi, 2019, arXiv:1910.13483.

Córcoles, A. D., E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow, 2015, Nat. Commun. **6**, 6979.

Cornelissen, A., 2018, master's thesis (Technische Universiteit Delft).

Cortes, C., and V. Vapnik, 1995, Mach. Learn. **20**, 273.

Cowtan, A., S. Dilkes, R. Duncan, A. Krajenbrink, W. Simmons, and S. Sivarajah, 2019, arXiv:1902.08091.

Cowtan, A., S. Dilkes, R. Duncan, W. Simmons, and S. Sivarajah, 2019, arXiv:1906.01734.

Coyle, B., M. Henderson, J. C. J. Le, N. Kumar, M. Paini, and E. Kashefi, 2021, Quantum Sci. Technol. **6**, 024013.

Coyle, B., D. Mills, V. Danos, and E. Kashefi, 2020, npj Quantum Inf. **6**, 60.

Crawford, D., A. Levit, N. Ghadermarzy, J. S. Oberoi, and P. Ronagh, 2016, arXiv:1612.05695.

Crawford, D., A. Levit, N. Ghadermarzy, J. S. Oberoi, and P. Ronagh, 2019, arXiv:1612.05695.

Crawford, O., B. van Straaten, D. Wang, T. Parks, E. Campbell, and S. Brierley, 2019, arXiv:1908.06942.

Crooks, G. E., 2018, arXiv:1811.08419.

Cross, A. W., L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, 2019, Phys. Rev. A **100**, 032328.

Cross, A. W., L. S. Bishop, J. A. Smolin, and J. M. Gambetta, 2017, arXiv:1707.03429.

Cross, A. W., E. Magesan, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, 2016, npj Quantum Inf. **2**, 16012.

Czarnik, P., A. Arrasmith, P. J. Coles, and L. Cincio, 2020, arXiv:2005.10189.

Dallaire-Demers, P.-L., and N. Killoran, 2018, Phys. Rev. A **98**, 012324.

Dallaire-Demers, P.-L., J. Romero, L. Veis, S. Sim, and A. Aspuru-Guzik, 2019, Quantum Sci. Technol. **4**, 045005.

Dallaire-Demers, P.-L., M. Stęchły, J. F. Gonthier, N. T. Bashige, J. Romero, and Y. Cao, 2020, arXiv:2003.01862.

Dalla Torre, E. G., S. Diehl, M. D. Lukin, S. Sachdev, and P. Strack, 2013, Phys. Rev. A **87**, 023831.

Dalzell, A. M., A. W. Harrow, D. E. Koh, and R. L. La Placa, 2020, Quantum **4**, 264.

Das, S., A. J. Wildridge, S. B. Vaidya, and A. Jung, 2019, arXiv:1903.08879.

Dattani, N. S., and N. Bryans, 2014, arXiv:1411.6758.

Dawson, C. M., and M. A. Nielsen, 2006, Quantum Inf. Comput. **6**, 81.

De, A., 2015, arXiv:1509.07905.

Deb, K., S. Agrawal, A. Pratab, and T. Meyarivan, 2000, in *Proceedings of the Sixth International Conference on Parallel Problem Solving from Nature, Leiden, Netherlands, 2000*, edited by M. Schoenauer and K. Deb (Springer-Verlag, Berlin), p. 849.

de Beaudrap, N., X. Bian, and Q. Wang, 2019, arXiv:1911.09039.

de Brugière, T. G., M. Baboulin, B. Valiron, S. Martiel, and C. Allouche, 2020, in *Reversible Computation*, Lecture Notes in Computer Science Vol. 12227, edited by I. Lanese and M. Rawski (Springer, New York), p. 189.

de Griend, A. M.-v., and R. Duncan, 2020, arXiv:2004.06052.

de Guise, H., O. Di Matteo, and L. L. Sánchez-Soto, 2018, Phys. Rev. A **97**, 022328.

de la Grand'rive, P. D., and J.-F. Hullo, 2019, arXiv:1908.02210.

Denchev, V. S., S. Boixo, S. V. Isakov, N. Ding, R. Babbush, V. Smelyanskiy, J. Martinis, and H. Neven, 2016, Phys. Rev. X **6**, 031015.

Derby, C., and J. Klassen, 2020, arXiv:2003.06939.

Deutsch, D., 1985, Proc. R. Soc. A **400**, 97.

Deutsch, I. H., 2020, PRX Quantum **1**, 020101.

Devitt, S. J., A. M. Stephens, W. J. Munro, and K. Nemoto, 2013, Nat. Commun. **4**, 2524.

Dive, B., A. Pitchford, F. Mintert, and D. Burgarth, 2018, Quantum **2**, 80.

Dodd, J. L., M. A. Nielsen, M. J. Bremner, and R. T. Thew, 2002, Phys. Rev. A **65**, 040301.

Dong, D., C. Chen, H. Li, and T.-J. Tarn, 2008, IEEE Trans. Syst. Man Cybern. B (Cybern.) **38**, 1207.

Dongarra, J. J., 1987, in *Supercomputing*, Lecture Notes in Computer Science Vol. 297, edited by E. N. Houstis, T. S. Papatheodorou, and C. D. Polychronopoulos (Springer, New York), p. 456.

Douçot, B., and L. Ioffe, 2012, Rep. Prog. Phys. **75**, 072001.

Du, Y., M.-H. Hsieh, T. Liu, and D. Tao, 2020, Phys. Rev. Research **2**, 033125.

Dumitrescu, E. F., A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, 2018, Phys. Rev. Lett. **120**, 210501.

Duncan, R., A. Kissinger, S. Perdrix, and J. Van De Wetering, 2020, Quantum **4**, 279.

Dunjko, V., and H. J. Briegel, 2018, Rep. Prog. Phys. **81**, 074001.

Dunjko, V., J. M. Taylor, and H. J. Briegel, 2016, Phys. Rev. Lett. **117**, 130501.

Dunjko, V., J. M. Taylor, and H. J. Briegel, 2017, in *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics, Banff, Alberta, Canada, 2017* (IEEE, New York), p. 282.

Dür, W., M. Skotiniotis, F. Fröwis, and B. Kraus, 2014, Phys. Rev. Lett. **112**, 080801.

Dury, B., and O. Di Matteo, 2020, arXiv:2009.00140.

Ebadi, S., *et al.*, 2020, arXiv:2012.12281.

Edmunds, C., C. Hempel, R. Harris, V. Frey, T. Stace, and M. Biercuk, 2020, Phys. Rev. Research **2**, 013156.

Egan, L., *et al.*, 2020, arXiv:2009.11482.

Egger, D. J., J. Marecek, and S. Woerner, 2020, arXiv:2009.10095.

Eisert, J., D. Hangleiter, N. Walk, I. Roth, D. Markham, R. Parekh, U. Chabaud, and E. Kashefi, 2019, arXiv:1910.06343.

Eisert, J., D. Hangleiter, N. Walk, I. Roth, D. Markham, R. Parekh, U. Chabaud, and E. Kashefi, 2020, Nat. Rev. Phys. **2**, 382.

Elben, A., B. Vermersch, C. F. Roos, and P. Zoller, 2019, Phys. Rev. A **99**, 052323.

Elben, A., B. Vermersch, R. van Bijnen, C. Kokail, T. Brydges, C. Maier, M. K. Joshi, R. Blatt, C. F. Roos, and P. Zoller, 2020, Phys. Rev. Lett. **124**, 010504.

Endo, S., S. C. Benjamin, and Y. Li, 2018, Phys. Rev. X **8**, 031027.

Endo, S., Z. Cai, S. C. Benjamin, and X. Yuan, 2020, arXiv:2011.01382.

Endo, S., I. Kurata, and Y. O. Nakagawa, 2020, Phys. Rev. Research **2**, 033281.

Endo, S., J. Sun, Y. Li, S. C. Benjamin, and X. Yuan, 2020, Phys. Rev. Lett. **125**, 010501.

Erhard, A., J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, 2019, Nat. Commun. **10**, 5347.

Evangelista, F. A., G. K.-L. Chan, and G. E. Scuseria, 2019, J. Chem. Phys. **151**, 244112.

Faist, P., S. Nezami, V. V. Albert, G. Salton, F. Pastawski, P. Hayden, and J. Preskill, 2020, Phys. Rev. X **10**, 041018.

Fan, K., 1951, Proc. Natl. Acad. Sci. U.S.A. **37**, 760.

Farhi, E., D. Gamarnik, and S. Gutmann, 2020a, arXiv:2004.09002.

Farhi, E., D. Gamarnik, and S. Gutmann, 2020b, arXiv:2005.08747.

Farhi, E., J. Goldstone, and S. Gutmann, 2002, arXiv:0201031.

Farhi, E., J. Goldstone, and S. Gutmann, 2014, arXiv:1411.4028.

Farhi, E., J. Goldstone, S. Gutmann, and M. Sipser, 2000, arXiv:quant-ph/0001106.

Farhi, E., J. Goldstone, S. Gutmann, and L. Zhou, 2019, arXiv:1910.08187.

Farhi, E., and A. W. Harrow, 2016, arXiv:1602.07674.

Farhi, E., and H. Neven, 2018, arXiv:1802.06002.

Ferguson, R. R., L. Dellantonio, K. Jansen, A. A. Balushi, W. Dür, and C. A. Muschik, 2020, arXiv:2010.13940.

Fermann, J. T., and E. F. Valeev, 2020, arXiv:2007.12057.

Festa, P., P. M. Pardalos, M. G. Resende, and C. C. Ribeiro, 2002, Optim. Methods Software **17**, 1033.

Feynman, R. P., 1982, Int. J. Theor. Phys. **21**, 467.

Fingerhuth, M., T. Babej, and C. Ing, 2018, arXiv:1810.13411.

Finnila, A. B., M. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, 1994, Chem. Phys. Lett. **219**, 343.

Fitzpatrick, M., N. M. Sundaresan, A. C. Li, J. Koch, and A. A. Houck, 2017, Phys. Rev. X **7**, 011016.

Fleming, C., and B. Hu, 2012, Ann. Phys. (Amsterdam) **327**, 1238.

Fletcher, A. S., P. W. Shor, and M. Z. Win, 2008, IEEE Trans. Inf. Theory **54**, 5705.

Fletcher, R., 2000, *Mathematics of Computation* (John Wiley & Sons, Chichester, England).

Fowler, A. G., M. Mariantoni, J. M. Martinis, and A. N. Cleland, 2012, Phys. Rev. A **86**, 032324.

Foxen, B., *et al.*, 2020, Phys. Rev. Lett. **125**, 120504.

França, D. S., and A. Hashagen, 2018, J. Phys. A **51**, 395302.

Fujii, K., and K. Nakajima, 2017, Phys. Rev. Applied **8**, 024030.

Gaitan, F., 2020, npj Quantum Inf. **6**, 61.

Galicia, A., B. Ramon, E. Solano, and M. Sanz, 2020, Phys. Rev. Research **2**, 033103.

Gambetta, J. M., F. Motzoi, S. Merkel, and F. K. Wilhelm, 2011, Phys. Rev. A **83**, 012308.

Gambetta, J. M., *et al.*, 2012, Phys. Rev. Lett. **109**, 240504.

Ganzhorn, M., *et al.*, 2019, Phys. Rev. Applied **11**, 044092.

Gao, X., E. R. Anschuetz, S.-T. Wang, J. I. Cirac, and M. D. Lukin, 2021, arXiv:2101.08354.

García-Álvarez, L., J. Casanova, A. Mezzacapo, I. L. Egusquiza, L. Lamata, G. Romero, and E. Solano, 2015, Phys. Rev. Lett. **114**, 070502.

García-Molina, P., A. Martin, and M. Sanz, 2021, arXiv:2107.12969.

Garcia-Saez, A., and J. Latorre, 2018, arXiv:1806.02287.

Garcia-Saez, A., and J. Riu, 2019, arXiv:1911.09682.

Gard, B. T., L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, 2020, npj Quantum Inf. **6**, 10.

Garhwal, S., M. Ghorani, and A. Ahmad, 2021, Arch. Comput. Methods Eng. **28**, 289.

Gay, S. J., 2006, Math. Struct. Comput. Sci. **16**, 581.

Georgescu, I. M., S. Ashhab, and F. Nori, 2014, Rev. Mod. Phys. **86**, 153.

Gheorghiu, V., S. M. Li, M. Mosca, and P. Mukhopadhyay, 2020, arXiv:2011.12191.

Ghosh, S., A. Opala, M. Matuszewski, T. Paterek, and T. C. Liew, 2019, npj Quantum Inf. **5**, 35.

Gilyén, A., S. Lloyd, and E. Tang, 2018, arXiv:1811.04909.

Giovannetti, V., S. Lloyd, and L. Maccone, 2008, Phys. Rev. Lett. **100**, 160501.

Giurgica-Tiron, T., Y. Hindy, R. LaRose, A. Mari, and W. J. Zeng, 2020, arXiv:2005.10921.

Goemans, M. X., and D. P. Williamson, 1995, J. ACM **42**, 1115.

Gokhale, P., O. Angiuli, Y. Ding, K. Gui, T. Tomesh, M. Suchara, M. Martonosi, and F. T. Chong, 2019, arXiv:1907.13623.

Gokhale, P., and F. T. Chong, 2019, arXiv:1908.11857.

Goldman, N., and J. Dalibard, 2014, Phys. Rev. X **4**, 031027.

Gonzalez-Raya, T., R. Asensio-Perea, A. Martin, L. C. Céleri, M. Sanz, P. Lougovski, and E. F. Dumitrescu, 2021, PRX Quantum **2**, 020328.

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, 2014, in *Proceedings of the 27th International Conference on Advances in Neural Information Processing Systems, Montreal, 2014* (Curran, Red Hook, NY), p. 2672.

Google, Q. A., 2021, Nature (London) **595**, 383.

Gosset, D., V. Kliuchnikov, M. Mosca, and V. Russo, 2013, arXiv:1308.4134.

Gottesman, D., 1997, arXiv:quant-ph/9705052.

Gottlieb, D., and S. A. Orszag, 1977, *Numerical Analysis of Spectral Methods: Theory and Applications* (SIAM, Philadelphia).

Gould, S. H., 2012, *Variational Methods for Eigenvalue Problems: An Introduction to the Methods of Rayleigh, Ritz, Weinstein, and Aronszajn* (Courier Corporation, North Chelmsford, MA).

Grant, E., L. Wossnig, M. Ostaszewski, and M. Benedetti, 2019, Quantum **3**, 214.

Greenbaum, D., 2015, arXiv:1509.02921.

Greenberger, D. M., M. A. Horne, A. Shimony, and A. Zeilinger, 1990, Am. J. Phys. **58**, 1131.

Gregorcic, G., 2001, Matrix **2**, 2.

Grimsley, H. R., D. Claudino, S. E. Economou, E. Barnes, and N. J. Mayhall, 2019, J. Chem. Theory Comput. **16**, 1.

Grimsley, H. R., S. E. Economou, E. Barnes, and N. J. Mayhall, 2019, Nat. Commun. **10**, 3007.

Gross, C., and I. Bloch, 2017, Science **357**, 995.

Gurvits, L., 2003, in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, San Diego, 2003* (Association for Computing Machinery, New York), p. 10.

Hadfield, C., S. Bravyi, R. Raymond, and A. Mezzacapo, 2020, arXiv:2006.15788.

Hadfield, S., Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, 2019, Algorithms Mol. Biol. **12**, 34.

Hadfield, S., Z. Wang, E. G. Rieffel, B. O'Gorman, D. Venturelli, and R. Biswas, 2017, in *Proceedings of the Second International Workshop on Post Moores Era Supercomputing, Denver, 2017* (Association for Computing Machinery, New York), pp. 15–21.

Häffner, H., C. F. Roos, and R. Blatt, 2008, Phys. Rep. **469**, 155.

Hallberg, K. A., 2006, Adv. Phys. **55**, 477.

Halperin, E., D. Livnat, and U. Zwick, 2004, J. Algorithms **53**, 169.

Hamamura, I., and T. Imamichi, 2020, npj Quantum Inf. **6**, 56.

Hamilton, C. S., R. Kruse, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, 2017, Phys. Rev. Lett. **119**, 170501.

Hamilton, K. E., E. F. Dumitrescu, and R. C. Pooser, 2019, Phys. Rev. A **99**, 062323.

Hammersley, J., 2013, *Monte Carlo Methods* (Springer Science +Business Media, New York).

Hanks, M., M. P. Estarellas, W. J. Munro, and K. Nemoto, 2020, Phys. Rev. X **10**, 041030.

Hansen, N., S. D. Müller, and P. Koumoutsakos, 2003, Evol. Comput. **11**, 1.

Harper, R., and S. T. Flammia, 2017, Quantum Sci. Technol. **2**, 015008.

Harrigan, M. P., *et al.*, 2021, Nat. Phys. **17**, 332.

Harrow, A., and J. Napp, 2019, arXiv:1901.05374.

Harrow, A. W., 2020, arXiv:2004.00026.

Harrow, A. W., A. Hassidim, and S. Lloyd, 2009, Phys. Rev. Lett. **103**, 150502.

Härtle, R., C. Benesch, and M. Thoss, 2008, Phys. Rev. B **77**, 205314.

Hartmann, M. J., 2016, J. Opt. **18**, 104005.

Hashagen, A., S. Flammia, D. Gross, and J. Wallman, 2018, Quantum **2**, 85.

Håstad, J., 2001, J. ACM **48**, 798.

Hastings, M. B., 2019, arXiv:1905.07047.

Hastings, M. B., 2020, arXiv:2005.03791.

Haug, T., and K. Bharti, 2020, arXiv:2011.14737.

Haug, T., K. Bharti, and M. Kim, 2021, PRX Quantum **2**, 040309.

Haug, T., R. Dumke, L.-C. Kwek, C. Miniatura, and L. Amico, 2021, Phys. Rev. Research **3**, 013034.

Haug, T., and M. S. Kim, 2021a, arXiv:2107.14063.

Haug, T., and M. S. Kim, 2021b, arXiv:2104.14543.

Haug, T., C. N. Self, and M. S. Kim, 2021, arXiv:2108.01039.

Hauke, P., H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, 2020, Rep. Prog. Phys. **83**, 054401.

Hauke, P., D. Marcos, M. Dalmonte, and P. Zoller, 2013, Phys. Rev. X **3**, 041018.

Havlíček, V., A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, 2019, Nature (London) **567**, 209.

Havlíček, V. c. v., M. Troyer, and J. D. Whitfield, 2017, Phys. Rev. A **95**, 032332.

Headley, D., T. Müller, A. Martin, E. Solano, M. Sanz, and F. K. Wilhelm, 2020, arXiv:2002.12215.

Heim, B., M. Soeken, S. Marshall, C. Granade, M. Roetteler, A. Geller, M. Troyer, and K. Svore, 2020, Nat. Rev. Phys. **2**, 709.

Helsen, J., X. Xue, L. M. Vandersypen, and S. Wehner, 2019, npj Quantum Inf. **5**, 1.

Helstrom, C. W., 1969, J. Stat. Phys. **1**, 231.

Hen, I., and M. S. Sarandy, 2016, Phys. Rev. A **93**, 062312.

Hen, I., and F. M. Spedalieri, 2016, Phys. Rev. Applied **5**, 034007.

Hensgens, T., T. Fujita, L. Janssen, X. Li, C. Van Diepen, C. Reichl, W. Wegscheider, S. D. Sarma, and L. M. Vandersypen, 2017, Nature (London) **548**, 70.

Herbst, M. F., 2018, Ph.D. thesis (Ruprecht-Karls-Universität Heidelberg).

Heya, K., K. M. Nakanishi, K. Mitarai, and K. Fujii, 2019, arXiv:1904.08566.

Heyfron, L. E., and E. T. Campbell, 2018, Quantum Sci. Technol. **4**, 015004.

Higgott, O., D. Wang, and S. Brierley, 2019, Quantum **3**, 156.

Ho, W. W., and T. H. Hsieh, 2019, SciPost Phys. **6**, 29.

Ho, W. W., C. Jonay, and T. H. Hsieh, 2019, Phys. Rev. A **99**, 052332.

Hofmann, T., B. Schölkopf, and A. J. Smola, 2008, Ann. Stat. **36**, 1171.

Holmes, Z., A. Arrasmith, B. Yan, P. J. Coles, A. Albrecht, and A. T. Sornborger, 2020, arXiv:2009.14808.

Holmes, Z., K. Sharma, M. Cerezo, and P. J. Coles, 2021, arXiv:2101.02138.

Horodecki, M., P. Horodecki, and R. Horodecki, 1999, Phys. Rev. A **60**, 1888.

Houck, A. A., H. E. Türeci, and J. Koch, 2012, Nat. Phys. **8**, 292.

Hu, L., *et al.*, 2019, Sci. Adv. **5**, eaav2761.

Hu, Z., R. Xia, and S. Kais, 2020, Sci. Rep. **10**, 1.

Huang, C.-J., H. Ma, Q. Yin, J.-F. Tang, D. Dong, C. Chen, G.-Y. Xiang, C.-F. Li, and G.-C. Guo, 2020, Phys. Rev. A **102**, 032412.

Huang, H.-L., *et al.*, 2020, arXiv:2010.06201.

Huang, H.-Y., K. Bharti, and P. Rebentrost, 2019, arXiv:1909.07344.

Huang, H.-Y., M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, 2021, Nat. Commun. **12**, 1.

Huang, H.-Y., R. Kueng, and J. Preskill, 2020, Nat. Phys. **16**, 1050.

Huang, K., *et al.*, 2020, arXiv:2009.12827.

Huang, L., and L. Wang, 2017, Phys. Rev. B **95**, 035105.

Huembeli, P., and A. Dauphin, 2021, Quantum Sci. Technol. **6**, 025011.

Huggins, W. J., J. Lee, U. Baek, B. O'Gorman, and K. B. Whaley, 2020, New J. Phys. **22**, 073009.

Huggins, W. J., J. McClean, N. Rubin, Z. Jiang, N. Wiebe, K. B. Whaley, and R. Babbush, 2019, arXiv:1907.13117.

Hughes, A., V. Schäfer, K. Thirumalai, D. Nadlinger, S. Woodrow, D. Lucas, and C. Ballance, 2020, Phys. Rev. Lett. **125**, 080504.

Huh, J., G. G. Guerreschi, B. Peropadre, J. R. McClean, and A. Aspuru-Guzik, 2015, Nat. Photonics **9**, 615.

Huh, J., S. Mostame, T. Fujita, M.-H. Yung, and A. Aspuru-Guzik, 2014, New J. Phys. **16**, 123008.

Huyer, W., and A. Neumaier, 2008, ACM Trans. Math. Softw. **35**, 1.

Inagaki, T., *et al.*, 2016, Science **354**, 603.

Itoko, T., R. Raymond, T. Imamichi, and A. Matsuo, 2020, Integration **70**, 43.

Itoko, T., R. Raymond, T. Imamichi, A. Matsuo, and A. W. Cross, 2019, in *Proceedings of the 24th Asia and South Pacific Design Automation Conference, Tokyo, 2019* (Association for Computing Machinery, New York), p. 191.

Izmaylov, A. F., M. Díaz-Tinoco, and R. A. Lang, 2020, Phys. Chem. Chem. Phys. **22**, 12980.

Izmaylov, A. F., T.-C. Yen, R. A. Lang, and V. Verteletskyi, 2020, J. Chem. Theory Comput. **16**, 190.

Izmaylov, A. F., T.-C. Yen, and I. G. Ryabinkin, 2019, Chem. Sci. **10**, 3746.

Jaderberg, B., A. Agarwal, K. Leonhardt, M. Kiffner, and D. Jaksch, 2020, Quantum Sci. Technol. **5**, 034015.

Jahangiri, S., J. M. Arrazola, and A. Delgado, 2020, arXiv:2012.09231.

Jahangiri, S., J. M. Arrazola, N. Quesada, and N. Killoran, 2020, Phys. Rev. E **101**, 022134.

Jensen, P. W. K., L. B. Kristensen, J. S. Kottmann, and A. Aspuru-Guzik, 2020, Quantum Sci. Technol. **6**, 015004.

Jerbi, S., L. M. Trenkwalder, H. P. Nautrup, H. J. Briegel, and V. Dunjko, 2019, arXiv:1910.12760.

Jiang, Z., A. Kalev, W. Mruczkiewicz, and H. Neven, 2020, Quantum **4**, 276.

Johnson, P. D., J. Romero, J. Olson, Y. Cao, and A. Aspuru-Guzik, 2017, arXiv:1711.02249.

Jones, N. C., R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, and Y. Yamamoto, 2012, Phys. Rev. X **2**, 031007.

Jones, T., 2020, arXiv:2011.02991.

Jones, T., S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin, 2019, Phys. Rev. A **99**, 062304.

Joó, B., C. Jung, N. H. Christ, W. Detmold, R. G. Edwards, M. Savage, and P. Shanahan, 2019, Eur. Phys. J. A **55**, 199.

Jordan, P., and O. Klein, 1927, Z. Phys. **45**, 751.

Jordan, P., and E. P. Wigner, 1928, Z. Phys. **47**, 631.

Jørgensen, P., 2012, *Second Quantization-Based Methods in Quantum Chemistry* (Elsevier, New York).

Joshi, M. K., A. Elben, B. Vermersch, T. Brydges, C. Maier, P. Zoller, R. Blatt, and C. F. Roos, 2020, Phys. Rev. Lett. **124**, 240505.

Kadowaki, T., and H. Nishimori, 1998, Phys. Rev. E **58**, 5355.

Kaiser, D., 2011, *How the Hippies Saved Physics: Science, Counterculture, and the Quantum Revival* (W. W. Norton & Company, New York).

Kako, S., T. Leleu, Y. Inui, F. Khoyratee, S. Reifenstein, and Y. Yamamoto, 2020, Adv. Quantum Technol. **3**, 2000045.

Kandala, A., A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, 2017, Nature (London) **549**, 242.

Kandala, A., K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, 2019, Nature (London) **567**, 491.

Karamlou, A. H., W. A. Simon, A. Katabarwa, T. L. Scholten, B. Peropadre, and Y. Cao, 2020, arXiv:2012.07825.

Kassal, I., and A. Aspuru-Guzik, 2009, J. Chem. Phys. **131**, 224102.

Kaubruegger, R., P. Silvi, C. Kokail, R. van Bijnen, A. M. Rey, J. Ye, A. M. Kaufman, and P. Zoller, 2019, Phys. Rev. Lett. **123**, 260505.

Kempe, J., A. Kitaev, and O. Regev, 2006, SIAM J. Comput. **35**, 1070.

Khairy, S., R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, 2019, arXiv:1911.04574.

Khammassi, N., G. G. Guerreschi, I. Ashraf, J. W. Hogaboam, C. G. Almudever, and K. Bertels, 2018, arXiv:1805.09607.

Khatri, S., R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, 2019, Quantum **3**, 140.

Khodjasteh, K., and L. Viola, 2009, Phys. Rev. Lett. **102**, 080501.

Kieferová, M., and N. Wiebe, 2017, Phys. Rev. A **96**, 062327.

Killoran, N., J. Izaac, N. Quesada, V. Bergholm, M. Amy, and C. Weedbrook, 2019, Quantum **3**, 129.

Kirby, W. M., and P. J. Love, 2019, Phys. Rev. Lett. **123**, 200501.

Kirby, W. M., A. Tranter, and P. J. Love, 2020, arXiv:2011.10027.

Kissinger, A., and A. M.-v. de Griend, 2019, arXiv:1904.00633.

Kissinger, A., and J. van de Wetering, 2019, arXiv:1903.10477.

Kissinger, A., and J. van de Wetering, 2020, Phys. Rev. A **102**, 022406.

Kitaev, A. Y., 2003, Ann. Phys. (Amsterdam) **303**, 2.

Kivlichan, I. D., J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K.-L. Chan, and R. Babbush, 2018, Phys. Rev. Lett. **120**, 110501.

Kivlichan, I. D., N. Wiebe, R. Babbush, and A. Aspuru-Guzik, 2017, J. Phys. A **50**, 305301.

Kjaergaard, M., M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, 2020, Annu. Rev. Condens. Matter Phys. **11**, 369.

Klco, N., E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage, 2018, Phys. Rev. A **98**, 032331.

Klema, V., and A. Laub, 1980, IEEE Trans. Autom. Control **25**, 164.

Klymko, C., B. D. Sullivan, and T. S. Humble, 2014, Quantum Inf. Process. **13**, 709.

Knill, E., and R. Laflamme, 1997, Phys. Rev. A **55**, 900.

Knill, E., R. Laflamme, and W. H. Zurek, 1998, Science **279**, 342.

Koczor, B., and S. C. Benjamin, 2019, arXiv:1912.08660.

Koczor, B., and S. C. Benjamin, 2020, arXiv:2008.13774.

Koczor, B., S. Endo, T. Jones, Y. Matsuzaki, and S. C. Benjamin, 2020, New J. Phys. **22**, 083038.

Koh, D. E., and S. Grewal, 2020, arXiv:2011.11580.

Kokail, C., *et al.*, 2019, Nature (London) **569**, 355.

Koppenhöfer, M., C. Bruder, and A. Roulet, 2020, Phys. Rev. Research **2**, 023026.

Koren, Y., R. Bell, and C. Volinsky, 2009, Computer **42**, 30.

Kosut, R. L., and D. A. Lidar, 2009, Quantum Inf. Process. **8**, 443.

Kosut, R. L., A. Shabani, and D. A. Lidar, 2008, Phys. Rev. Lett. **100**, 020502.

Kottmann, J. S., 2018, Ph.D. thesis (Humboldt-Universität zu Berlin).

Kottmann, J. S., A. Anand, and A. Aspuru-Guzik, 2021, Chem. Sci. **12**, 3497.

Kottmann, J. S., and F. A. Bischoff, 2017, J. Chem. Theory Comput. **13**, 5945.

Kottmann, J. S., M. Krenn, T. H. Kyaw, S. Alperin-Lea, and A. Aspuru-Guzik, 2020, arXiv:2006.03075.

Kottmann, J. S., P. Schleich, T. Tamayo-Mendoza, and A. Aspuru-Guzik, 2021, J. Phys. Chem. Lett. **12**, 663.

Krantz, P., M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, 2019, Appl. Phys. Rev. **6**, 021318.

Krenn, M., M. Erhard, and A. Zeilinger, 2020, Nat. Rev. Phys. **2**, 649.

Krenn, M., J. Kottmann, N. Tischler, and A. Aspuru-Guzik, 2020, arXiv:2005.06443.

Kreula, J., S. R. Clark, and D. Jaksch, 2016, Sci. Rep. **6**, 32940.

Krinner, S., P. Kurpiers, B. Royer, P. Magnard, I. Tsitsilin, J.-C. Besse, A. Remm, A. Blais, and A. Wallraff, 2020, Phys. Rev. Applied **14**, 044039.

Kruse, R., C. S. Hamilton, L. Sansoni, S. Barkhofen, C. Silberhorn, and I. Jex, 2019, Phys. Rev. A **100**, 032326.

Kubo, K., Y. O. Nakagawa, S. Endo, and S. Nagayama, 2020, arXiv:2012.04429.

Kunitsa, A. A., and S. Hirata, 2020, Phys. Rev. E **101**, 013311.

Kusumoto, T., K. Mitarai, K. Fujii, M. Kitagawa, and M. Negoro, 2019, arXiv:1911.12021.

Kwon, H., and J. Bae, 2021, IEEE Trans. Comput. **70**, 1401.

Kyaw, T., S. Felicetti, G. Romero, E. Solano, and L. C. Kwek, 2014, Proc. SPIE Int. Soc. Opt. Eng. **9225**, 92250B.

Kyaw, T. H., 2019, *Towards a Scalable Quantum Computing Platform in the Ultrastrong Coupling Regime* (Springer, New York).

Kyaw, T. H., S. Allende, L.-C. Kwek, and G. Romero, 2017, Quantum Sci. Technol. **2**, 025007.

Kyaw, T. H., V. M. Bastidas, J. Tangpanitanon, G. Romero, and L.-C. Kwek, 2020, Phys. Rev. A **101**, 012111.

Kyaw, T. H., S. Felicetti, G. Romero, E. Solano, and L.-C. Kwek, 2015, Sci. Rep. **5**, 8621.

Kyaw, T. H., D. A. Herrera-Marti, E. Solano, G. Romero, and L.-C. Kwek, 2015, Phys. Rev. B **91**, 064503.

Kyaw, T. H., Y. Li, and L.-C. Kwek, 2014, Phys. Rev. Lett. **113**, 180501.

Kyaw, T. H., T. Menke, S. Sim, N. P. Sawaya, W. D. Oliver, G. G. Guerreschi, and A. Aspuru-Guzik, 2021, Phys. Rev. Applied **16**, 044042.

Kyriienko, O., A. E. Paine, and V. E. Elfving, 2020, arXiv:2011.10395.

Kyriienko, O., and A. S. Sørensen, 2018, Phys. Rev. Applied **9**, 064029.

Lacroix, N., *et al.*, 2020, PRX Quantum **1**, 110304.

Lamata, L., 2017, Sci. Rep. **7**, 1.

Lamata, L., U. Alvarez-Rodriguez, J. D. Martín-Guerrero, M. Sanz, and E. Solano, 2018, Quantum Sci. Technol. **4**, 014007.

Lamata, L., A. Parra-Rodriguez, M. Sanz, and E. Solano, 2018, Adv. Phys. X **3**, 1457981.

Landsman, K. A., C. Figgatt, T. Schuster, N. M. Linke, B. Yoshida, N. Y. Yao, and C. Monroe, 2019, Nature (London) **567**, 61.

Lang, R. A., I. G. Ryabinkin, and A. F. Izmaylov, 2020, arXiv:2002.05701.

LaRose, R., A. Tikku, É. O'Neel-Judy, L. Cincio, and P. J. Coles, 2019, npj Quantum Inf. **5**, 57.

Las Heras, U., U. Alvarez-Rodriguez, E. Solano, and M. Sanz, 2016a, Phys. Rev. Lett. **116**, 230504.

Las Heras, U., U. Alvarez-Rodriguez, E. Solano, and M. Sanz, 2016b, Phys. Rev. Lett. **116**, 230504.

Lau, J. W. Z., K. Bharti, T. Haug, and L. C. Kwek, 2021, arXiv:2101.07677.

Lau, J. W. Z., T. Haug, L. C. Kwek, and K. Bharti, 2021, arXiv:2103.05500.

Lavrijsen, W., A. Tudor, J. Müller, C. Iancu, and W. de Jong, 2020, arXiv:2004.03004.

Lavrijsen, W., *et al.*, 2020, computer code SCIKIT-QUANT, https://github.com/scikit-quant/scikit-quant.

Lechner, W., 2020, IEEE Trans. Quantum Eng. **1**, 1.

Lechner, W., P. Hauke, and P. Zoller, 2015, Sci. Adv. **1**, e1500838.

Lee, C.-K., J. W. Z. Lau, L. Shi, and L. C. Kwek, 2021, arXiv:2101.06879.

Lee, J., W. J. Huggins, M. Head-Gordon, and K. B. Whaley, 2019, J. Chem. Theory Comput. **15**, 311.

Lenstra, J. K., and A. R. Kan, 1975, J. Oper. Res. Soc. Am. **26**, 717.

Leung, D. W., M. A. Nielsen, I. L. Chuang, and Y. Yamamoto, 1997, Phys. Rev. A **56**, 2567.

Leyton-Ortega, V., A. Perdomo-Ortiz, and O. Perdomo, 2019, arXiv:1901.08047.

Li, A. C., F. Petruccione, and J. Koch, 2016, Phys. Rev. X **6**, 021037.

Li, G., Y. Ding, and Y. Xie, 2019, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, Providence, 2019*, edited by I. Bahar, M. Herlihy, E. Witchel, and A. R. Lebeck (Association for Computing Machinery, New York), p. 1001.

Li, J., X. Yang, X. Peng, and C.-P. Sun, 2017, Phys. Rev. Lett. **118**, 150503.

Li, L., M. Fan, M. Coram, P. Riley, and S. Leichenauer, 2020, Phys. Rev. Research **2**, 023074.

Li, R., U. Alvarez-Rodriguez, L. Lamata, and E. Solano, 2017, Quantum Meas. Quantum Metrol. **4**, 1.

Li, R. Y., R. Di Felice, R. Rohs, and D. A. Lidar, 2018, npj Quantum Inf. **4**, 1.

Li, W., and D.-L. Deng, 2021, arXiv:2108.13421.

Li, Y., and S. C. Benjamin, 2017, Phys. Rev. X **7**, 021050.

Lidar, D. A., 2014 Adv. Chem. Phys. **154**, 295.

Lidar, D. A., and T. A. Brun, 2013, *Quantum Error Correction* (Cambridge University Press, Cambridge, England).

Lim, K. H., T. Haug, L. C. Kwek, and K. Bharti, 2021, arXiv:2104.01931.

Liu, D. C., and J. Nocedal, 1989, Math. Program. **45**, 503.

Liu, J., Z. Li, and J. Yang, 2020, arXiv:2012.07047.

Liu, J., Y. Qi, Z. Y. Meng, and L. Fu, 2017, Phys. Rev. B **95**, 041101.

Liu, J., and Y. Xin, 2020, J. High Energy Phys. 12, 011.

Liu, J.-G., and L. Wang, 2018, Phys. Rev. A **98**, 062324.

Liu, J.-G., Y.-H. Zhang, Y. Wan, and L. Wang, 2019, Phys. Rev. Research **1**, 023025.

Liu, Z., L.-M. Duan, and D.-L. Deng, 2020, arXiv:2008.05488.

Lloyd, S., 1996, Science **273**, 1073.

Lloyd, S., 2018, arXiv:1812.11075.

Lloyd, S., M. Mohseni, and P. Rebentrost, 2013, arXiv:1307.0411.

Lloyd, S., M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, 2020, arXiv:2001.03622.

Lloyd, S., and C. Weedbrook, 2018, Phys. Rev. Lett. **121**, 040502.

Lockwood, O., and M. Si, 2020a, in *Proceedings of the Thirty-Fourth Conference on Neural Information Processing Systems (NeurIPS), 2020*, edited by L. Bertinetto, J. F. Henriques, S. Albanie, M. Paganini, and G. Varol, https://proceedings.mlr.press/v148/lockwood21a.html.

Lockwood, O., and M. Si, 2020b, in *Proceedings of the 16th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, 2020*, Vol. 16, edited by L. Lelis and D. Thue (AAAI Press, Palo Alto, CA), p. 245.

Lowe, A., M. H. Gordon, P. Czarnik, A. Arrasmith, P. J. Coles, and L. Cincio, 2020, arXiv:2011.01157.

Lu, D., *et al.*, 2017, npj Quantum Inf. **3**, 45.

Lubasch, M., J. Joo, P. Moinier, M. Kiffner, and D. Jaksch, 2020, Phys. Rev. A **101**, 010301.

Lucas, A., 2014, Front. Phys. **2**, 5.

Lyu, C., V. Montenegro, and A. Bayat, 2020, Quantum **4**, 324.

Ma, Z., P. Gokhale, T.-X. Zheng, S. Zhou, X. Yu, L. Jiang, P. Maurer, and F. T. Chong, 2020, arXiv:2010.08702.

Magann, A. B., C. Arenz, M. D. Grace, T.-S. Ho, R. L. Kosut, J. R. McClean, H. A. Rabitz, and M. Sarovar, 2021, PRX Quantum **2**, 010101.

Magann, A. B., M. D. Grace, H. A. Rabitz, and M. Sarovar, 2020, arXiv:2002.12497.

Magesan, E., J. M. Gambetta, and J. Emerson, 2011, Phys. Rev. Lett. **106**, 180504.

Magesan, E., J. M. Gambetta, and J. Emerson, 2012, Phys. Rev. A **85**, 042311.

Mahadev, U., 2018, in *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS), Paris, 2018*, edited by M. Thorup (IEEE, New York), p. 259.

Mahoney, A., J. Colless, S. Pauka, J. Hornibrook, J. Watson, G. Gardner, M. Manfra, A. Doherty, and D. Reilly, 2017, Phys. Rev. X **7**, 011007.

Manin, Y., 1980, *Computable and Non-Computable* (Sovetskoe Radio, Moscow).

Manne, A. S., 1960, Oper. Res. **8**, 219.

Marandi, A., Z. Wang, K. Takata, R. L. Byer, and Y. Yamamoto, 2014, Nat. Photonics **8**, 937.

Mari, A., T. R. Bromley, and N. Killoran, 2020, arXiv:2008.06517.

Marrero, C. O., M. Kieferová, and N. Wiebe, 2020, arXiv:2010.15968.

Martin, A., L. Lamata, E. Solano, and M. Sanz, 2020, Phys. Rev. Research **2**, 013012.

Martinez, E. A., *et al.*, 2016, Nature (London) **534**, 516.

Marzec, M., 2016, in *Handbook of High-Frequency Trading and Modeling in Finance*, edited by I. Florescu, M. C. Mariani, H. E. Stanley, and F. G. Viens (Wiley, New York), p. 73.

Massey, P., J. A. Clark, and S. Stepney, 2004, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), Seattle, 2004*, edited by K. Deb *et al.* (Springer, New York), p. 569.

Massey, P., J. A. Clark, and S. Stepney, 2006, Evol. Comput. **14**, 21.

McArdle, S., S. Endo, A. Aspuru-Guzik, S. C. Benjamin, and X. Yuan, 2020, Rev. Mod. Phys. **92**, 015003.

McArdle, S., T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, 2019, npj Quantum Inf. **5**, 75.

McArdle, S., A. Mayorov, X. Shan, S. Benjamin, and X. Yuan, 2019, Chem. Sci. **10**, 5725.

McArdle, S., X. Yuan, and S. Benjamin, 2019, Phys. Rev. Lett. **122**, 180501.

McClean, J. R., S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, 2018, Nat. Commun. **9**, 4812.

McClean, J. R., M. P. Harrigan, M. Mohseni, N. C. Rubin, Z. Jiang, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, 2020, arXiv:2008.08615.

McClean, J. R., Z. Jiang, N. C. Rubin, R. Babbush, and H. Neven, 2020, Nat. Commun. **11**, 636.

McClean, J. R., M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, 2017, Phys. Rev. A **95**, 042308.

McClean, J. R., J. Romero, R. Babbush, and A. Aspuru-Guzik, 2016, New J. Phys. **18**, 023023.

McKay, D. C., S. Sheldon, J. A. Smolin, J. M. Chow, and J. M. Gambetta, 2019, Phys. Rev. Lett. **122**, 200502.

McMahon, P. L., *et al.*, 2016, Science **354**, 614.

Mei, F., Q. Guo, Y.-F. Yu, L. Xiao, S.-L. Zhu, and S. Jia, 2020, Phys. Rev. Lett. **125**, 160503.

Meitei, O. R., B. T. Gard, G. S. Barron, D. P. Pappas, S. E. Economou, E. Barnes, and N. J. Mayhall, 2020, arXiv:2008.04302.

Mengoni, R., and A. Di Pierro, 2019, Quantum Mach. Intell. **1**, 65.

Menke, T., F. Häse, S. Gustavsson, A. J. Kerman, W. D. Oliver, and A. Aspuru-Guzik, 2019, arXiv:1912.03322.

Merkel, S. T., J. M. Gambetta, J. A. Smolin, S. Poletto, A. D. Córcoles, B. R. Johnson, C. A. Ryan, and M. Steffen, 2013, Phys. Rev. A **87**, 062119.

Metger, T., and T. Vidick, 2020, arXiv:2001.09161.

Metwalli, S. A., F. L. Gall, and R. Van Meter, 2020, arXiv:2008.12525.

Meyer, D. A., and N. R. Wallach, 2002, J. Math. Phys. (N.Y.) **43**, 4273.

Meyer, J. J., 2021, Quantum **5**, 539.

Meyer, J. J., J. Borregaard, and J. Eisert, 2020, arXiv:2006.06303.

Mezzacapo, A., U. Las Heras, J. Pedernales, L. DiCarlo, E. Solano, and L. Lamata, 2014, Sci. Rep. **4**, 1.

Mi, X., *et al.*, 2021, arXiv:2101.08870.

Milne, A. R., C. L. Edmunds, C. Hempel, F. Roy, S. Mavadia, and M. J. Biercuk, 2020, Phys. Rev. Applied **13**, 024022.

Miquel, C., J. P. Paz, M. Saraceno, E. Knill, R. Laflamme, and C. Negrevergne, 2002, Nature (London) **418**, 59.

Mitarai, K., and K. Fujii, 2019, Phys. Rev. Research **1**, 013006.

Mitarai, K., M. Negoro, M. Kitagawa, and K. Fujii, 2018, Phys. Rev. A **98**, 032309.

Mitarai, K., T. Yan, and K. Fujii, 2019, Phys. Rev. Applied **11**, 044087.

Mitchell, T. M., 1997, *Machine Learning* (McGraw-Hill, New York).

Moll, N., *et al.*, 2018, Quantum Sci. Technol. **3**, 030503.

Monroe, C., *et al.*, 2019, arXiv:1912.07845.

Montanaro, A., 2016, npj Quantum Inf. **2**, 15023.

Morales, M. E. S., J. Biamonte, and Z. Zimborás, 2019, arXiv:1909.03123.

Morales, M. E. S., T. Tlyachev, and J. Biamonte, 2018, Phys. Rev. A **98**, 062333.

Mott, A., J. Job, J.-R. Vlimant, D. Lidar, and M. Spiropulu, 2017, Nature (London) **550**, 375.

Motzoi, F., J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm, 2009, Phys. Rev. Lett. **103**, 110501.

Movassagh, R., 2019, arXiv:1909.06210.

Nakaji, K., and N. Yamamoto, 2020a, arXiv:2005.12537.

Nakaji, K., and N. Yamamoto, 2020b, arXiv:2010.13727.

Nakajima, K., K. Fujii, M. Negoro, K. Mitarai, and M. Kitagawa, 2019, Phys. Rev. Applied **11**, 034021.

Nakanishi, K. M., K. Fujii, and S. Todo, 2020, Phys. Rev. Research **2**, 043158.

Nakanishi, K. M., K. Mitarai, and K. Fujii, 2019, Phys. Rev. Research **1**, 033062.

Nam, Y., *et al.*, 2020, npj Quantum Inf. **6**, 33.

Napp, J., R. L. La Placa, A. M. Dalzell, F. G. Brandao, and A. W. Harrow, 2019, arXiv:2001.00021.

Nash, B., V. Gheorghiu, and M. Mosca, 2020, Quantum Sci. Technol. **5**, 025010.

Nataf, P., and C. Ciuti, 2011, Phys. Rev. Lett. **107**, 190402.

Navarrete-Benlloch, C., 2015, arXiv:1504.05266.

Negoro, M., K. Mitarai, K. Fujii, K. Nakajima, and M. Kitagawa, 2018, arXiv:1806.10910.

Neill, C., *et al.*, 2018, Science **360**, 195.

Nghiem, N. A., S. Y.-C. Chen, and T.-C. Wei, 2020, arXiv:2010.13186.

Nguyen, T., A. Santana, T. Kharazi, D. Claudino, H. Finkel, and A. McCaskey, 2020, arXiv:2010.03935.

Nishio, S., Y. Pan, T. Satoh, H. Amano, and R. V. Meter, 2020, ACM J. Emerging Technol. Comput. **16**, 1.

Niu, S., A. Suau, G. Staffelbach, and A. Todri-Sanial, 2020, IEEE Trans. Quantum Eng. **1**,

Noh, K., and C. Chamberland, 2020, Phys. Rev. A **101**, 012316.

Nokkala, J., R. Martínez-Peña, G. L. Giorgi, V. Parigi, M. C. Soriano, and R. Zambrini, 2020, arXiv:2006.04821.

O'Brien, T. E., B. Senjean, R. Sagastizabal, X. Bonet-Monroig, A. Dutkiewicz, F. Buda, L. DiCarlo, and L. Visscher, 2019, npj Quantum Inf. **5**, 1.

O'Gorman, B., R. Babbush, A. Perdomo-Ortiz, A. Aspuru-Guzik, and V. Smelyanskiy, 2015, Eur. Phys. J. Special Topics **224**, 163.

Oh, Y.-H., H. Mohammadbagherpoor, P. Dreher, A. Singh, X. Yu, and A. J. Rindos, 2019, arXiv:1911.00595.

Oka, T., and S. Kitamura, 2019, Annu. Rev. Condens. Matter Phys. **10**, 387.

O'Malley, P. J., *et al.*, 2016, Phys. Rev. X **6**, 031007.

Orús, R., 2014, Ann. Phys. (Amsterdam) **349**, 117.

Orús, R., S. Mugel, and E. Lizaso, 2019, Rev. Phys. **4**, 100028.

Ostaszewski, M., E. Grant, and M. Benedetti, 2019, arXiv:1905.09692.

Otten, M., C. L. Cortes, and S. K. Gray, 2019, arXiv:1910.06284.

Otten, M., and S. K. Gray, 2019, npj Quantum Inf. **5**, 11.

Otterbach, J. S., *et al.*, 2017, arXiv:1712.05771.

Ozaeta, A., W. van Dam, and P. L. McMahon, 2020, arXiv:2012.03421.

Pagano, G., *et al.*, 2020, Proc. Natl. Acad. Sci. U.S.A. **117**, 25396.

Paparo, G. D., V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, 2014, Phys. Rev. X **4**, 031002.

Park, D. K., C. Blank, and F. Petruccione, 2020, Phys. Lett. A **384**, 126422.

Park, D. K., F. Petruccione, and J.-K. K. Rhee, 2019, Sci. Rep. **9**, 1.

Park, D. K., I. Sinayskiy, M. Fingerhuth, F. Petruccione, and J.-K. K. Rhee, 2019, New J. Phys. **21**, 083024.

Parra-Rodriguez, A., P. Lougovski, L. Lamata, E. Solano, and M. Sanz, 2020, Phys. Rev. A **101**, 022305.

Parrish, R. M., E. G. Hohenstein, P. L. McMahon, and T. J. Martínez, 2019, Phys. Rev. Lett. **122**, 230401.

Parrish, R. M., J. T. Iosue, A. Ozaeta, and P. L. McMahon, 2019, arXiv:1904.03206.

Parrish, R. M., and P. L. McMahon, 2019, arXiv:1909.08925.

Paz, J. P., and W. H. Zurek, 1998, Proc. R. Soc. A **454**, 355.

Pepper, A., N. Tischler, and G. J. Pryde, 2019, Phys. Rev. Lett. **122**, 060501.

Perdomo, A., C. Truncik, I. Tubert-Brohman, G. Rose, and A. Aspuru-Guzik, 2008, Phys. Rev. A **78**, 012320.

Perdomo-Ortiz, A., M. Benedetti, J. Realpe-Gómez, and R. Biswas, 2018, Quantum Sci. Technol. **3**, 030502.

Perdomo-Ortiz, A., N. Dickson, M. Drew-Brook, G. Rose, and A. Aspuru-Guzik, 2012, Sci. Rep. **2**, 571.

Perdomo-Ortiz, A., J. Fluegemann, S. Narasimhan, R. Biswas, and V. N. Smelyanskiy, 2015, Eur. Phys. J. Special Topics **224**, 131.

Perdomo-Ortiz, A., *et al.*, 2019, Phys. Rev. Applied **12**, 014004.

Pérez-Salinas, A., A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, 2020, Quantum **4**, 226.

Pérez-Salinas, A., J. Cruz-Martinez, A. A. Alhajri, and S. Carrazza, 2020, arXiv:2011.13934.

Pérez-Salinas, A., D. García-Martín, C. Bravo-Prieto, and J. I. Latorre, 2020, Entropy **22**, 436.

Peruzzo, A., J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, 2014, Nat. Commun. **5**, 4213.

Pesah, A., M. Cerezo, S. Wang, T. Volkoff, A. T. Sornborger, and P. J. Coles, 2021, Phys. Rev. X **11**, 041011.

Peters, E., J. Caldeira, A. Ho, S. Leichenauer, M. Mohseni, H. Neven, P. Spentzouris, D. Strain, and G. N. Perdue, 2021, arXiv:2101.09581.

Peterson, E. C., G. E. Crooks, and R. S. Smith, 2020, Quantum **4**, 247.

Pino, J. M., *et al.*, 2020, arXiv:2003.01293.

Pirhooshyaran, M., and T. Terlaky, 2020, arXiv:2012.04046.

Platt, J., 1998, https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/.

Pokharel, B., N. Anand, B. Fortman, and D. A. Lidar, 2018, Phys. Rev. Lett. **121**, 220502.

Policharla, G.-V., and S. Vinjanampathy, 2020, arXiv:2011.13777.

Potoček, V., A. P. Reynolds, A. Fedrizzi, and D. W. Corne, 2018, arXiv:1812.04458.

Poulin, D., M. B. Hastings, D. Wecker, N. Wiebe, A. C. Doherty, and M. Troyer, 2014, arXiv:1406.4920.

Powell, M. J., 2009, University of Cambridge NA Report No. NA2009/06, p. 26.

Pozzi, M. G., S. J. Herbert, A. Sengupta, and R. D. Mullins, 2020, arXiv:2007.15957.

Preskill, J., 2018, Quantum **2**, 79.

Qi, H., D. J. Brod, N. Quesada, and R. García-Patrón, 2020, Phys. Rev. Lett. **124**, 100502.

QOSF, 2020a, https://github.com/qosf/awesome-quantum-software.

QOSF, 2020b, https://qosf.org/.

Quantum, G. A., *et al.*, 2020, Science **369**, 1084.

Quesada, N., J. M. Arrazola, and N. Killoran, 2018, Phys. Rev. A **98**, 062322.

Rath, A., R. van Bijnen, A. Elben, P. Zoller, and B. Vermersch, 2021, arXiv:2102.13524.

Raussendorf, R., and J. Harrington, 2007, Phys. Rev. Lett. **98**, 190504.

Rechenberg, I., 1978, in *Simulationsmethoden in der Medizin und Biologie* (Springer, New York), pp. 83–114.

Reiher, M., N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, 2017, Proc. Natl. Acad. Sci. U.S.A. **114**, 7555.

Richter, J., and A. Pal, 2020, arXiv:2012.02795.

Rivas, Á., S. F. Huelga, and M. B. Plenio, 2010, Phys. Rev. Lett. **105**, 050403.

Robert, A., P. K. Barkoutsos, S. Woerner, and I. Tavernelli, 2021, npj Quantum Inf. **7**, 38.

Roggero, A., A. C. Li, J. Carlson, R. Gupta, and G. N. Perdue, 2020, Phys. Rev. D **101**, 074038.

Rohwedder, T., 2010, Ph.D. thesis (Technische Universität Berlin).

Romero, J., and A. Aspuru-Guzik, 2019, arXiv:1901.00848.

Romero, J., R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, 2018, Quantum Sci. Technol. **4**, 014008.

Romero, J., J. P. Olson, and A. Aspuru-Guzik, 2017, Quantum Sci. Technol. **2**, 045001.

Roos, B. O., P. R. Taylor, and P. E. Sigbahn, 1980, Chem. Phys. **48**, 157.

Rosenberg, G., P. Haghnegahdar, P. Goddard, P. Carr, K. Wu, and M. L. De Prado, 2016, IEEE J. Sel. Top. Signal Process. **10**, 1053.

Roushan, P., *et al.*, 2017, Science **358**, 1175.

Rudolph, M. S., N. T. Bashige, A. Katabarwa, S. Johr, B. Peropadre, and A. Perdomo-Ortiz, 2020, arXiv:2012.03924.

Ryabinkin, I. G., S. N. Genin, and A. F. Izmaylov, 2019, J. Chem. Theory Comput. **15**, 249.

Ryabinkin, I. G., A. F. Izmaylov, and S. N. Genin, 2021, Quantum Sci. Technol. **6**, 024012.

Ryabinkin, I. G., R. A. Lang, S. N. Genin, and A. F. Izmaylov, 2020, J. Chem. Theory Comput. **16**, 1055.

Ryabinkin, I. G., T.-C. Yen, S. N. Genin, and A. F. Izmaylov, 2018, J. Chem. Theory Comput. **14**, 6317.

Sack, S. H., and M. Serbyn, 2021, arXiv:2101.05742.

Sagastizabal, R., *et al.*, 2019a, Phys. Rev. A **100**, 010302.

Sagastizabal, R., *et al.*, 2019b, Phys. Rev. A **100**, 010302.

Saleem, Z. H., 2020, Int. J. Quantum. Inf. **18**, 2050011.

Santagati, R., *et al.*, 2018, Sci. Adv. **4**, eaap9646.

Santos, L. F., and L. Viola, 2005, Phys. Rev. A **72**, 062303.

Sauvage, F., S. Sim, A. A. Kunitsa, W. A. Simon, M. Mauri, and A. Perdomo-Ortiz, 2021, arXiv:2103.08572.

Sawaya, N. P., and J. Huh, 2019, J. Phys. Chem. Lett. **10**, 3586.

Sawaya, N. P., T. Menke, T. H. Kyaw, S. Johri, A. Aspuru-Guzik, and G. G. Guerreschi, 2020, npj Quantum Inf. **6**, 49.

Schmoll, P., and R. Orús, 2017, Phys. Rev. B **95**, 045112.

Schölkopf, B., R. Herbrich, and A. J. Smola, 2001, in *Computational Learning Theory*, edited by D. Helmbold and B. Williamson (Springer, New York), p. 416.

Schuld, M., 2021, arXiv:2101.11020.

Schuld, M., V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, 2019, Phys. Rev. A **99**, 032331.

Schuld, M., A. Bocharov, K. M. Svore, and N. Wiebe, 2020, Phys. Rev. A **101**, 032308.

Schuld, M., K. Brádler, R. Israel, D. Su, and B. Gupt, 2020, Phys. Rev. A **101**, 032314.

Schuld, M., M. Fingerhuth, and F. Petruccione, 2017, Europhys. Lett. **119**, 60002.

Schuld, M., and N. Killoran, 2019, Phys. Rev. Lett. **122**, 040504.

Schuld, M., and F. Petruccione, 2018, *Supervised Learning with Quantum Computers* (Springer, New York).

Schuld, M., I. Sinayskiy, and F. Petruccione, 2014, Quantum Inf. Process. **13**, 2567.

Schuld, M., I. Sinayskiy, and F. Petruccione, 2016, Phys. Rev. A **94**, 022342.

Schuld, M., R. Sweke, and J. J. Meyer, 2020, arXiv:2008.08605.

Schwefel, H.-P., 1977, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie: Mit einer vergleichenden Einführung in die Hill-Climbing-und Zufallsstrategie*, Vol. 1 (Springer, New York).

Scott, A. J., 2004, Phys. Rev. A **69**, 052330.

Seeley, J. T., M. J. Richard, and P. J. Love, 2012, J. Chem. Phys. **137**, 224109.

Seki, K., and S. Yunoki, 2021, PRX Quantum **2**, 010333.

Serafini, A., 2017, *Quantum Continuous Variables: A Primer of Theoretical Methods* (CRC Press, Boca Raton).

Setia, K., S. Bravyi, A. Mezzacapo, and J. D. Whitfield, 2019, Phys. Rev. Research **1**, 033033.

Setia, K., R. Chen, J. E. Rice, A. Mezzacapo, M. Pistoia, and J. D. Whitfield, 2020, J. Chem. Theory Comput. **16**, 6091.

Setia, K., and J. D. Whitfield, 2018, J. Chem. Phys. **148**, 164104.

Sharma, K., M. Cerezo, L. Cincio, and P. J. Coles, 2020, arXiv:2005.12458.

Shin, S. W., G. Smith, J. A. Smolin, and U. Vazirani, 2014, arXiv:1401.7087.

Shingu, Y., Y. Seki, S. Watabe, S. Endo, Y. Matsuzaki, S. Kawabata, T. Nikuni, and H. Hakoshima, 2020, arXiv:2007.00876.

Shor, P. W., 1994, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, 1994* (IEEE, New York), p. 124.

Shor, P. W., 1995, Phys. Rev. A **52**, R2493.

Shor, P. W., 1999, SIAM Rev. **41**, 303.

Sieberer, L. M., M. Buchhold, and S. Diehl, 2016, Rep. Prog. Phys. **79**, 096001.

Sim, S., P. D. Johnson, and A. Aspuru-Guzik, 2019, Adv. Quantum Technol. **2**, 1900070.

Sim, S., J. Romero, J. F. Gonthier, and A. A. Kunitsa, 2020, arXiv:2010.00629.

Siraichi, M. Y., V. F. d. Santos, S. Collange, and F. M. Q. Pereira, 2018, in *Proceedings of the 16th Annual International Symposium on Code Generation and Optimization (CGO), Vienna, 2018* (Association for Computing Machinery, New York), pp. 113–125.

Situ, H., Z. He, Y. Wang, L. Li, and S. Zheng, 2020, Inf. Sci. **538**, 193.

Skolik, A., J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib, 2020, arXiv:2006.14904.

Smith, A., B. Jobst, A. G. Green, and F. Pollmann, 2019, arXiv:1910.05351.

Smith, A., M. Kim, F. Pollmann, and J. Knolle, 2019, npj Quantum Inf. **5**, 106.

Smith, R. S., M. J. Curtis, and W. J. Zeng, 2016, arXiv:1608.03355.

Somma, R., G. Ortiz, E. Knill, and J. Gubernatis, 2003, arXiv:quant-ph/0304063.

Song, C., J. Cui, H. Wang, J. Hao, H. Feng, and Y. Li, 2019, Sci. Adv. **5**, eaaw5686.

Spagnolo, N., E. Maiorino, C. Vitelli, M. Bentivegna, A. Crespi, R. Ramponi, P. Mataloni, R. Osellame, and F. Sciarrino, 2017, Sci. Rep. **7**, 14316.

Spall, J. C., 1992, IEEE Trans. Autom. Control **37**, 332.

Stair, N. H., R. Huang, and F. A. Evangelista, 2020, J. Chem. Theory Comput. **16**, 2236.

Stefanucci, G., and R. Van Leeuwen, 2013, *Nonequilibrium Many-Body Theory of Quantum Systems: A Modern Introduction* (Cambridge University Press, Cambridge, England).

Steudtner, M., and S. Wehner, 2018, New J. Phys. **20**, 063010.

Steudtner, M., and S. Wehner, 2019, Phys. Rev. A **99**, 022308.

Stokes, J., J. Izaac, N. Killoran, and G. Carleo, 2020, Quantum **4**, 269.

Streif, M., and M. Leib, 2020, Phys. Rev. A **102**, 042416.

Streif, M., S. Yarkoni, A. Skolik, F. Neukart, and M. Leib, 2021, Phys. Rev. A **104**, 012403.

Strikis, A., D. Qin, Y. Chen, S. C. Benjamin, and Y. Li, 2020, arXiv:2005.07601.

Struchalin, G., Y. A. Zagorovskii, E. Kovlakov, S. Straupe, and S. Kulik, 2021, PRX Quantum **2**, 010307.

Suchsland, P., F. Tacchino, M. H. Fischer, T. Neupert, P. K. Barkoutsos, and I. Tavernelli, 2020, arXiv:2008.10914.

Sun, J., X. Yuan, T. Tsunoda, V. Vedral, S. C. Bejamin, and S. Endo, 2020, arXiv:2001.04891.

Surján, P. R., 2012, *Second Quantized Approach to Quantum Chemistry: An Elementary Introduction* (Springer Science+Business Media, New York).

Sutton, R. S., and A. G. Barto, 2018, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA).

Suzuki, M., 1976, Commun. Math. Phys. **51**, 183.

Sweke, R., F. Wilde, J. J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, 2020, Quantum **4**, 314.

Tacchino, F., P. Barkoutsos, C. Macchiavello, I. Tavernelli, D. Gerace, and D. Bajoni, 2020, Quantum Sci. Technol. **5**, 044010.

Tacchino, F., A. Chiesa, S. Carretta, and D. Gerace, 2020, Adv. Quantum Technol. **3**, 1900052.

Tacchino, F., C. Macchiavello, D. Gerace, and D. Bajoni, 2019, npj Quantum Inf. **5**, 26.

Takeshita, T., N. C. Rubin, Z. Jiang, E. Lee, R. Babbush, and J. R. McClean, 2020, Phys. Rev. X **10**, 011004.

Tan, B., and J. Cong, 2020, in *Proceedings of the 39th International Conference on Computer-Aided Design, 2020* (Association for Computing Machinery, New York), p. 1.

Tan, J., T. H. Kyaw, and Y. Yeo, 2010, Phys. Rev. A **81**, 062119.

Tang, E., 2019, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, Phoenix, 2019* (Association for Computing Machinery, New York), p. 217.

Tang, E., 2021, Phys. Rev. Lett. **127**, 060503.

Tang, H. L., E. Barnes, H. R. Grimsley, N. J. Mayhall, and S. E. Economou, 2019, arXiv:1911.10205.

Taube, A. G., and R. J. Bartlett, 2006, Int. J. Quantum Chem. **106**, 3393.

Temme, K., S. Bravyi, and J. M. Gambetta, 2017, Phys. Rev. Lett. **119**, 180509.

Terhal, B. M., 2015, Rev. Mod. Phys. **87**, 307.

Tilly, J., G. Jones, H. Chen, L. Wossnig, and E. Grant, 2020, Phys. Rev. A **102**, 062425.

Torlai, G., G. Mazzola, G. Carleo, and A. Mezzacapo, 2020, Phys. Rev. Research **2**, 022060.

Torlai, G., G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, 2018, Nat. Phys. **14**, 447.

Torrontegui, E., and J. J. García-Ripoll, 2019, Europhys. Lett. **125**, 30004.

Tóth, G., and I. Apellaniz, 2014, J. Phys. A **47**, 424006.

Tovey, C. A., 1984, Discrete Appl. Math. **8**, 85.

Trabesinger, A., 2012, Nat. Phys. **8**, 263.

Tranter, A., P. J. Love, F. Mintert, and P. V. Coveney, 2018, J. Chem. Theory Comput. **14**, 5617.

Tranter, A., S. Sofia, J. Seeley, M. Kaicher, J. McClean, R. Babbush, P. V. Coveney, F. Mintert, F. Wilhelm, and P. J. Love, 2015, Int. J. Quantum Chem. **115**, 1431.

Utkarsh, B. K. Behera, and P. K. Panigrahi, 2020, arXiv:2002.01351.

Utsunomiya, S., K. Takata, and Y. Yamamoto, 2011, Opt. Express **19**, 18091.

Vallury, H. J., M. A. Jones, C. D. Hill, and L. C. Hollenberg, 2020, Quantum **4**, 373.

Vandewalle, J., and B. De Moor, 1991, in *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms*, edited by G. H. Golub and P. Van Dooren, NATO Advanced Study Institutes, Ser. F, Vol. 70 (Springer, New York).

van de Wetering, J., 2020, arXiv:2012.13966.

van Enk, S. J., and C. W. J. Beenakker, 2012, Phys. Rev. Lett. **108**, 110503.

Van Meter, R., T. D. Ladd, A. G. Fowler, and Y. Yamamoto, 2010, Int. J. Quantum. Inf. **08**, 295.

Veis, L., J. Višňák, H. Nishizawa, H. Nakai, and J. Pittner, 2016, Int. J. Quantum Chem. **116**, 1328.

Venturelli, D., and A. Kondratyev, 2019, Quantum Mach. Intell. **1**, 17.

Verdon, G., M. Broughton, and J. Biamonte, 2017, arXiv:1712.05304.

Verdon, G., M. Broughton, J. R. McClean, K. J. Sung, R. Babbush, Z. Jiang, H. Neven, and M. Mohseni, 2019, arXiv:1907.05415.

Verdon, G., J. Marks, S. Nanda, S. Leichenauer, and J. Hidary, 2019, arXiv:1910.02071.

Verstraete, F., and J. I. Cirac, 2005, J. Stat. Mech. P09012.

Verstraete, F., J. I. Cirac, and J. I. Latorre, 2009, Phys. Rev. A **79**, 032316.

Verstraete, F., J. J. Garcia-Ripoll, and J. I. Cirac, 2004, Phys. Rev. Lett. **93**, 207204.

Verstraete, F., V. Murg, and J. I. Cirac, 2008, Adv. Phys. **57**, 143.

Verteletskyi, V., T.-C. Yen, and A. F. Izmaylov, 2020, J. Chem. Phys. **152**, 124114.

Vidal, G., 2003, Phys. Rev. Lett. **91**, 147902.

Vidal, G., 2004, Phys. Rev. Lett. **93**, 040502.

Vidal, J. G., and D. O. Theis, 2019, arXiv:1901.11434.

Vikstål, P., M. Grönkvist, M. Svensson, M. Andersson, G. Johansson, and G. Ferrini, 2020, Phys. Rev. Applied **14**, 034009.

Vinjanampathy, S., and J. Anders, 2016, Contemp. Phys. **57**, 545.

Viola, L., and E. Knill, 2005, Phys. Rev. Lett. **94**, 060502.

Viola, L., E. Knill, and S. Lloyd, 1999, Phys. Rev. Lett. **82**, 2417.

Volkoff, T., and P. J. Coles, 2021, Quantum Sci. Technol. **6**, 025008.

von Burg, V., G. H. Low, T. Häner, D. S. Steiger, M. Reiher, M. Roetteler, and M. Troyer, 2020, arXiv:2007.14460.

Vovrosh, J., K. E. Khosla, S. Greenaway, C. Self, M. Kim, and J. Knolle, 2021, arXiv:2101.01690.

Wallman, J. J., and J. Emerson, 2016, Phys. Rev. A **94**, 052325.

Wan, K. H., O. Dahlsten, H. Kristjánsson, R. Gardner, and M. Kim, 2017, npj Quantum Inf. **3**, 36.

Wan, Y., R. Jördens, S. D. Erickson, J. J. Wu, R. Bowler, T. R. Tan, P.-Y. Hou, D. J. Wineland, A. C. Wilson, and D. Leibfried, 2020, Adv. Quantum Technol. **3**, 2000028.

Wang, D., O. Higgott, and S. Brierley, 2019, Phys. Rev. Lett. **122**, 140504.

Wang, K., Z. Song, X. Zhao, Z. Wang, and X. Wang, 2020, arXiv:2012.14311.

Wang, L., 2016, Phys. Rev. B **94**, 195105.

Wang, S., E. Fontana, M. Cerezo, K. Sharma, A. Sone, L. Cincio, and P. J. Coles, 2020, arXiv:2007.14384.

Wang, X., Z. Song, and Y. Wang, 2020, arXiv:2006.02336.

Wang, Y., Y. Li, Z.-q. Yin, and B. Zeng, 2018, npj Quantum Inf. **4**, 1.

Wang, Z., S. Hadfield, Z. Jiang, and E. G. Rieffel, 2018, Phys. Rev. A **97**, 022304.

Wang, Z., A. Marandi, K. Wen, R. L. Byer, and Y. Yamamoto, 2013, Phys. Rev. A **88**, 063853.

Watkins, C. J., and P. Dayan, 1992, Mach. Learn. **8**, 279.

Watts, P., M. O'Connor, and J. Vala, 2013, Entropy **15**, 1963.

Wauters, M. M., G. B. Mbeng, and G. E. Santoro, 2020, arXiv:2003.07419.

Wauters, M. M., E. Panizon, G. B. Mbeng, and G. E. Santoro, 2020, Phys. Rev. Research **2**, 033446.

Wecker, D., M. B. Hastings, and M. Troyer, 2015, Phys. Rev. A **92**, 042303.

Wecker, D., M. B. Hastings, and M. Troyer, 2016, Phys. Rev. A **94**, 022309.

Wen, J., X. Kong, S. Wei, B. Wang, T. Xin, and G. Long, 2019, Phys. Rev. A **99**, 012320.

White, S. R., 1992, Phys. Rev. Lett. **69**, 2863.

Wierichs, D., C. Gogolin, and M. Kastoryano, 2020, arXiv:2004.14666.

Wierstra, D., T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, 2014, J. Mach. Learn. Res. **15**, 949, https://www.jmlr.org/papers/volume15/wierstra14a/wierstra14a.pdf.

Wille, R., L. Burgholzer, and A. Zulehner, 2019, 2019 in *Proceedings of the 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, 2019* (Association for Computing Machinery, New York), p. 1.

Williams, C. P., and A. G. Gray, 1999, in *Quantum Computing and Quantum Communications*, edited by C. P. Williams, Lecture Notes in Computer Science Vol. 1509 (Springer, New York), p. 113.

Wilson, M., R. Stromswold, F. Wudarski, S. Hadfield, N. M. Tubman, and E. G. Rieffel, 2021, Quantum Mach. Intell. **3**, 13.

Woitzik, A. J. C., P. K. Barkoutsos, F. Wudarski, A. Buchleitner, and I. Tavernelli, 2020, Phys. Rev. A **102**, 042402.

Wright, K., *et al.*, 2019, Nat. Commun. **10**, 1.

Wright, L. G., and P. L. McMahon, 2019, arXiv:1908.01364.

Wu, S. L., *et al.*, 2020, arXiv:2012.11560.

Xiang, D., X. Wang, C. Jia, T. Lee, and X. Guo, 2016, Chem. Rev. **116**, 4318.

Xin, T., S. Wei, J. Cui, J. Xiao, I. n. Arrazola, L. Lamata, X. Kong, D. Lu, E. Solano, and G. Long, 2020, Phys. Rev. A **101**, 032307.

Xu, X., J. Sun, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, 2019, arXiv:1909.03898.

Yalouz, S., B. Senjean, J. Günther, F. Buda, T. E. O'Brien, and L. Visscher, 2021, Quantum Sci. Technol. **6**, 024004.

Yamamoto, N., 2019, arXiv:1909.05074.

Yamamoto, Y., T. Leleu, S. Ganguli, and H. Mabuchi, 2020, Appl. Phys. Lett. **117**, 160501.

Yang, Z.-C., A. Rahmani, A. Shabani, H. Neven, and C. Chamon, 2017, Phys. Rev. X **7**, 021027.

Yao, J., M. Bukov, and L. Lin, 2020, in *Proceedings of the First Mathematical and Scientific Machine Learning Conference, Princeton, NJ, 2020*, Vol. 107, edited by J. Lu and R. Ward, p. 605, http://proceedings.mlr.press/v107/yao20a.html.

Yao, J., P. Köttering, H. Gundlach, L. Lin, and M. Bukov, 2020, arXiv:2012.06701.

Yao, N. Y., L. Jiang, A. V. Gorshkov, P. C. Maurer, G. Giedke, J. I. Cirac, and M. D. Lukin, 2012, Nat. Commun. **3**, 800.

Yao, Y.-X., N. Gomes, F. Zhang, T. Iadecola, C.-Z. Wang, K.-M. Ho, and P. P. Orth, 2020, arXiv:2011.00622.

Yen, T.-C., and A. F. Izmaylov, 2020, arXiv:2007.01234.

Yen, T.-C., V. Verteletskyi, and A. F. Izmaylov, 2020, J. Chem. Theory Comput. **16**, 2400.

Yordanov, Y. S., D. R. Arvidsson-Shukur, and C. H. Barnes, 2020, Phys. Rev. A **102**, 062612.

Yoshioka, N., Y. O. Nakagawa, K. Mitarai, and K. Fujii, 2020, Phys. Rev. Research **2**, 043289.

Yserentant, H., 2010, *Regularity and Approximability of Electronic Wave Functions*, Lecture Notes in Mathematics (Springer, Berlin).

Yu, J., J. Retamal, M. Sanz, E. Solano, and F. Albarrán-Arriagada, 2021, arXiv:2103.15696.

Yu, S., *et al.*, 2019, Adv. Quantum Technol. **2**, 1800074.

Yuan, X., S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, 2019, Quantum **3**, 191.

Yung, M.-H., J. Casanova, A. Mezzacapo, J. McClean, L. Lamata, A. Aspuru-Guzik, and E. Solano, 2014, Sci. Rep. **4**, 3589.

Zanardi, P., and M. Rasetti, 1999, Phys. Lett. A **264**, 94.

Zeng, J., Y. Wu, J.-G. Liu, L. Wang, and J. Hu, 2019, Phys. Rev. A **99**, 052306.

Zhang, C., Y. Chen, Y. Jin, W. Ahn, Y. Zhang, and E. Z. Zhang, 2020, arXiv:2009.02346.

Zhang, J., T. H. Kyaw, S. Filipp, L.-C. Kwek, E. Sjöqvist, and D. Tong, 2021, arXiv:2110.03602.

Zhang, J., T. H. Kyaw, D. Tong, E. Sjöqvist, and L.-C. Kwek, 2016, Sci. Rep. **5**, 18414.

Zhang, K., P. Rao, K. Yu, H. Lim, and V. Korepin, 2021, arXiv:2102.01783.

Zhang, S., Y. Lu, K. Zhang, W. Chen, Y. Li, J.-N. Zhang, and K. Kim, 2020, Nat. Commun. **11**, 587.

Zhang, T., J. Sun, X.-X. Fang, X. Zhang, X. Yuan, and H. Lu, 2021, arXiv:2106.10190.

Zhang, Y.-H., P.-L. Zheng, Y. Zhang, and D.-L. Deng, 2020, Phys. Rev. Lett. **125**, 170501.

Zhang, Z.-J., T. H. Kyaw, J. Kottmann, M. Degroote, and A. Aspuru-Guzik, 2021, Quantum Sci. Technol. **6**, 035001.

Zhang, Z.-J., J. Sun, X. Yuan, and M.-H. Yung, 2020, arXiv:2011.05283.

Zhao, A., A. Tranter, W. M. Kirby, S. F. Ung, A. Miyake, and P. J. Love, 2020, Phys. Rev. A **101**, 062322.

Zhao, T., G. Carleo, J. Stokes, and S. Veerapaneni, 2021, Mach. Learn. Sci. Technol. **2**, 02LT01.

Zhong, H.-S., *et al.*, 2020, Science **370**, 1460.

Zhou, L., S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, 2020, Phys. Rev. X **10**, 021067.

Zhou, Y., E. M. Stoudenmire, and X. Waintal, 2020, arXiv:2002.07730.

Zhu, D., *et al.*, 2019, Sci. Adv. **5**, eaaw9918.

Zlokapa, A., S. Boixo, and D. Lidar, 2020, arXiv:2005.02464.

Zoufal, C., A. Lucchi, and S. Woerner, 2019, npj Quantum Inf. **5**, 103.

Zoufal, C., A. Lucchi, and S. Woerner, 2020, arXiv:2006.06004.

Zulehner, A., S. Gasser, and R. Wille, 2017, in *Proceedings of the Ninth International Conference on Reversible Computation, Kolkata, India, 2017*, edited by I. Phillips and H. Rahaman (Springer, New York), p. 185.

Zulehner, A., A. Paler, and R. Wille, 2019, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **38**, 1226.

Zulehner, A., and R. Wille, 2019, in *Proceedings of the 24th Asia and South Pacific Design Automation Conference, Tokyo, 2019* (Association for Computing Machinery, New York), p. 185.

See Supplemental Material at http://link.aps.org/supplemental/10.1103/RevModPhys.94.015004 for the tables of NISQ applications, software packages, and external libraries, as well as details on classical optimization strategies, unitary t-designs, and NISQ applications for finance.