Quantum Computer Systems for Scientific Discovery

Yuri Alexeev[1], Dave Bacon[2], Kenneth R. Brown[3,4,5], Robert Calderbank[3,6], Lincoln D. Carr[7], Frederic T. Chong[8], Brian DeMarco[9], Dirk Englund[10], Edward Farhi[11,12], Bill Fefferman[8], Alexey V. Gorshkov[13,14], Andrew Houck[15], Jungsang Kim[3,5,16], Shelby Kimmel[17], Michael Lange[18], Seth Lloyd[19], Mikhail D. Lukin[20], Dmitri Maslov[21], Peter Maunz[22], Christopher Monroe*[13,16], John Preskill[23], Martin Roetteler[24], Martin J. Savage[25], and Jeff Thompson[15]

[1] *Argonne National Laboratory, Lemont, IL*
[2] *Google, Inc., Seattle, WA*
[3] *Department of Electrical and Computer Engineering, Duke University, Durham, NC*
[4] *Department of Chemistry, Duke University, Durham, NC*
[5] *Department of Physics, Duke University, Durham, NC*
[6] *Department of Computer Science and Department of Mathematics, Duke University, Durham, NC*
[7] *Department of Physics, Colorado School of Mines, Golden, CO*
[8] *Department of Computer Science, University of Chicago, Chicago, IL*
[9] *Department of Physics and IQUIST, University of Illinois, Urbana-Champaign, IL*
[10] *Department of Electrical Engineering and Computer Science,*
Massachusetts Institute of Technology, Cambridge, MA
[11] *Google, Inc., Venice CA*
[12] *Department of Physics, Massachusetts Institute of Technology, Cambridge, MA*
[13] *Joint Quantum Institute, Joint Center for Quantum Information and Computer Science,*
and Department of Physics, University of Maryland, College Park, MD
[14] *National Institute of Standards and Technology, Gaithersburg, MD*
[15] *Department of Electrical Engineering, Princeton University, Princeton, NJ*
[16] *IonQ, Inc., College Park, MD*
[17] *Department of Computer Science, Middlebury College, Middlebury, VT*
[18] *L3Harris Technologies, Melborune, FL*
[19] *Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA*
[20] *Department of Physics, Harvard University, Cambridge, MA*
[21] *IBM T.J. Watson Research Center, Yorktown Heights, NY*
[22] *Sandia National Laboratories, Albuquerque, NM*
[23] *Institute for Quantum Information and Matter and Walter Burke Institute for Theoretical Physics,*
California Institute of Technology, Pasadena CA
[24] *Microsoft Quantum, Redmond, WA and*
[25] *Institute for Nuclear Theory and Department of Physics, University of Washington, Seattle, WA*

The great promise of quantum computers comes with the dual challenges of building them and finding their useful applications. We argue that these two challenges should be considered together, by co-designing full-stack quantum computer systems along with their applications in order to hasten their development and potential for scientific discovery. In this context, we identify scientific and community needs, opportunities, a sampling of a few use case studies, and significant challenges for the development of quantum computers for science over the next 2–10 years.

This document is written by a community of university, national laboratory, and industrial researchers in the field of Quantum Information Science and Technology, and is based on a summary from a U.S. National Science Foundation workshop on Quantum Computing held on October 21–22, 2019 in Alexandria, VA.

* to whom correspondence should be addressed: monroe@umd.edu

Executive Summary

The quantum computer promises enormous information storage and processing capacity that can eclipse any possible conventional computer [1]. This stems from quantum entangled superpositions of individual quantum systems, usually expressed as a collection of quantum bits or qubits, whose full description requires exponentially many parameters. However, there are two critical challenges in bringing quantum computers to fruition:

Challenge 1: The vast amount of information contained in massively entangled quantum systems is itself not directly accessible, due to the reduction of quantum superpositions upon measurement. Instead, useful quantum computer algorithms guide a quantum state to a much simpler form to produce some type of global property of the information that can be directly measured. However, the full scope of applications that can exploit entangled superpositions in this way and how exactly quantum computers will be used in the future remains unclear.

Challenge 2: Quantum computers are notoriously difficult to build, requiring extreme isolation of a large number of individual qubits, while also allowing exquisite control of their quantum states and high-accuracy measurements. Quantum computer technology is nothing like classical computer hardware and involves unconventional information carriers in exotic environments like high vacuum or very low temperature. Ultimately, large-scale quantum computers will utilize error-correction techniques that are much more complex than their classical counterparts.

This article combines these two challenges by promoting the idea of *co-designing* quantum computers with their scientific applications in a vertically-integrated approach that addresses scientific opportunities at all levels of the quantum computer stack.

Since the birth of quantum computing in the 1990s, there has been enormous progress in the isolation and control of good qubit platforms [2]. Some of these quantum technologies, based on individual atoms controlled by laser beams [3, 4] or superconducting circuitry coupled with microwave fields [5], are now being built into small systems. This has led to a new era of quantum computing, paralleling the transition from transistors to integrated circuits many decades ago, which is expected to lead to significant scientific opportunities. In this position paper, we, therefore, do not focus on the physics or development of qubit technologies at the component level. We also do not speculate on new qubit technologies that may emerge in the coming years. These are important and foundational research activities, but they are also typically divorced from systems-level considerations of operating quantum computers.

Here we concentrate on the near-term prospects and scientific opportunities generated by an integrated consideration of the complete quantum computer stack using existing quantum system technologies. At the bottom of the stack, device-specific qubit control considerations will impact the engineering of native interactions or gate sets, connectivity, and thus net performance on high-level quantum computer applications. In the middle of the stack, compilation of native gates into standard gate sets and higher-level quantum subroutines can be compressed and compiled further using techniques from quantum computer science. Quantum error-correction encoding of qubits, a relatively new field in its own right, will become ever more important as the systems grow in the number of qubits and their circuit depth. At the top of the stack, the execution of quantum circuits can simulate difficult quantum problems, from molecular properties, chemistry, and materials science [6] to nuclear and particle physics models beyond the reach of conventional computers [7]. They may also find use as general optimizers of generic models [8] that could be applied to logistics, economics, and climate science.

We advocate for the continual development and operation of multiple generations of quantum computer **systems** using current technology specifically designed with the above types of opportunities in mind. We propose iterating between building and using the devices, with a full-stack scientific mission that can be fulfilled by a unified effort at universities, at national laboratories, and in industry. As future qubit technologies are developed, we expect they will be integrated into the stack. While there is growing industrial interest in building quantum computers, we note that these efforts may not be focused on co-designing quantum computers that will have sufficient flexibility to address scientific opportunities at the various levels of the quantum computer stack. Over the next 2–10 years, the research and development approach we propose here is expected to generate new science, stimulate the transition of academic and national laboratory programs in quantum computing to industry, and also train future quantum engineers. The overriding high-level aim of this proposed path is to hasten the development of a wide range of concrete scientific applications for quantum computers, with parallel efforts in quantum simulation [9] and quantum communication [10].

Introduction

Quantum computers represent a fundamental departure from the way we process information. At its core, a quantum computer consists of quantum bits (qubits) or equivalent quantum information carriers, that allow the storage and processing of quantum superpositions of data. A single qubit $|x_i\rangle$ is a quantum two-level system that can store a superposition of both $x_i = 0$ and $x_i = 1$. A collection of n qubits can be represented by a quantum state $|\Psi\rangle$ that stores an arbitrary *entangled* superposition

of all n-bit binary numbers,

$$|\Psi\rangle = \sum_{k=0}^{2^n-1} \alpha_k |k\rangle \tag{1}$$

$$= \sum_{x_i \in \{0,1\}} \alpha_{x_{n-1}x_{n-2}...x_0} |x_{n-1}\rangle |x_{n-2}\rangle \ldots |x_0\rangle , \tag{2}$$
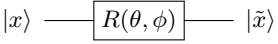
where the 2^n weights or amplitudes of each basis state $|k\rangle$ are given by complex numbers α_k, whose index k can also be expressed as a string of bit values $x_{n-1}x_{n-2}...x_0$. The superposition amplitudes α_k evolve according to the unitary (time-reversible) Schrödinger wave equation that dictates how the amplitudes of the various basis states can be controlled, governed by an underlying Hamiltonian or energy functional. When the qubits are measured, they assume definite values with probabilities $P(k) = |\alpha_k|^2$ given by the corresponding amplitudes of the underlying quantum state. The entanglement of the above general quantum superposition represents a complex web of natural links between qubits, giving rise to a network of correlations while maintaining the character of superposition within individual qubits. Quantum entanglement allows an efficient "wiring" of qubit states without any real wires or physical connections between the qubits, and it has no classical analog.

There are several known quantum algorithms that offer various advantages or speedups over classical computing approaches, some even reducing the complexity class of the problem. These algorithms generally proceed by controlling the quantum interference between the components of the underlying entangled superpositions in such a way that only one or relatively few quantum states have a significant amplitude in the end. A subsequent measurement can, therefore, provide global information on a massive superposition state with significant probability.
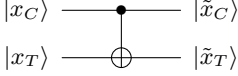
The coherent manipulation of quantum states that defines a quantum algorithm can be expressed through different quantum computational modes with varying degrees of tunability and control. The most expressive quantum computing mode presently known is the universal gate model, similar to universal gate models of classical computation. Here, a quantum algorithm is broken down to a sequence of modular quantum operations or gates between individual qubits. There are many universal quantum gate families operating on single and pairwise qubits [11], akin to the NAND gate family in classical computing. One popular universal quantum gate family is the grouping of two-qubit CNOT gates on every pair of qubits along with rotation gates on every single qubit [1], as displayed in Fig. 1. With universal gates, an arbitrary entangled state and thus any quantum algorithm can be expressed. Alternative modes such as measurement-based or cluster-state quantum computing [12] can be shown to be formally equivalent to the universal gate model. Another interesting quantum model worth mentioning is quantum walks" model [13], which

is a natural mathematical framework to construct quantum gates in qubit systems controlled via pulses. Like the NAND gate in classical CMOS technology, the particular choice of universal gate set or even mode of quantum computing is best determined by the quantum hardware itself and its native interactions and available controls. The structure of the algorithm itself may also impact the optimal choice of gate set or quantum computing mode.

$$|0\rangle \longrightarrow \cos\frac{\theta}{2}|0\rangle - ie^{+i\phi}\sin\frac{\theta}{2}|1\rangle$$

$$|1\rangle \longrightarrow \cos\frac{\theta}{2}|1\rangle - ie^{-i\phi}\sin\frac{\theta}{2}|0\rangle$$

$$|0\rangle|0\rangle \longrightarrow |0\rangle|0\rangle$$
$$|0\rangle|1\rangle \longrightarrow |0\rangle|1\rangle$$
$$|1\rangle|0\rangle \longrightarrow |1\rangle|1\rangle$$
$$|1\rangle|1\rangle \longrightarrow |1\rangle|0\rangle$$

$$|x\rangle \ \text{—}\ \boxed{R(\theta,\phi)}\ \text{—}\ |\tilde{x}\rangle$$

$$|x_C\rangle \ \text{———•———}\ |\tilde{x}_C\rangle$$
$$|x_T\rangle \ \text{———}\oplus\text{———}\ |\tilde{x}_T\rangle$$

(a) (b)

FIG. 1. The rotation and controlled-NOT (CNOT) gates are an example of a universal quantum gate family when available on all qubits, with explicit evolution (above) and quantum circuit block schematics (below). (a) The single-qubit rotation gate $R(\theta,\phi)$, with two continuous parameters θ and ϕ, evolves input qubit state $|x\rangle$ to output state $|\tilde{x}\rangle$. (b) The CNOT (or reversible XOR) gate on two qubits evolves two (control and target) input qubit states $|x_C\rangle$ and $|x_T\rangle$ to output states $|\tilde{x}_C = x_C\rangle$ and $|\tilde{x}_T = x_C \oplus x_T\rangle$, where $\oplus$ is addition modulo 2, or equivalently the XOR operation.

There are other modes of quantum computation that are not universal, involving subsets of universal gate operations, or certain global gate operations with less control over the entire space of quantum states. These can be useful for specific routines or quantum simulations that may not demand full universality. Although global adiabatic Hamiltonian quantum computing [14] can be made universal in certain cases [15], it is often better implemented as non-universal subroutines for specific state preparation. Quantum annealing models [16, 17] do not appear to be universal, and there is a current debate over the advantage such models can have over classical computation [18]. Gates that explicitly include error, or decoherence processes [19], used to model quantum computer systems interacting with an environment via quantum simulation [9], we consider outside the scope of this discussion.

Given the continuous amplitudes that define their quantum states (Eq. 1), quantum computers have characteristics akin to classical analog computers, where errors can accumulate over time and lead to computational instability. It is thus critical that quantum computers exploit the technique of quantum error correction (QEC) [20, 21], or at least have sufficiently small native errors that allow the system to complete the algorithm [22].

Quantum Computer Stack

Algorithms
Identify problem
Map to qubits and gates

Quantum Software
Express in native gates/connectivity
Compile & compress circuits
Deploy error correction strategy

Control Engineering
Implement Hamiltonian
control with E/M fields
$$i\hbar\frac{\partial|\Psi\rangle}{\partial t} = H(t)|\Psi\rangle$$

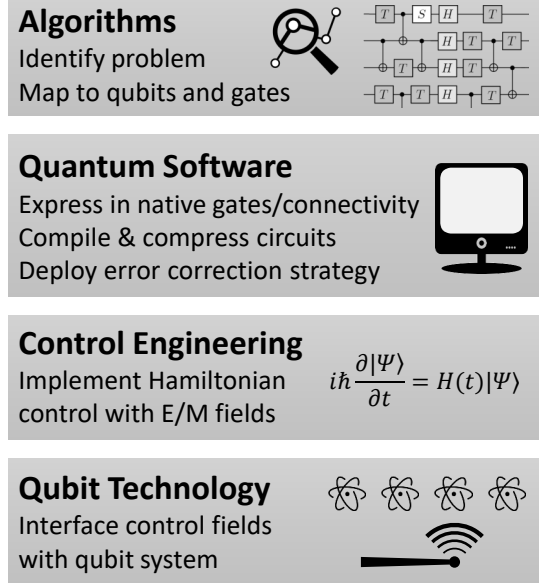**Qubit Technology**
Interface control fields
with qubit system

FIG. 2. Advances in all levels of the quantum computer stack, from algorithms and quantum software down to control engineering and qubit technology, will be required to bring quantum computers to fruition. We expect scientific opportunities at every level and at the interfaces between levels. At the highest levels, quantum computer algorithms are expected to advance many fields of science and technology. At he middle levels (software stack), the compilation and translation of quantum gates will allow for algorithmic compression to accelerate performance, while error-correction techniques will mitigate quantum computing errors. At the lowest levels, new ways to control interactions between qubit technologies may lead to better performance. Future capable qubit technologies will require tight integration with the other layers of the stack to realize their potential.

QEC is an extens
ancilla qubits are
certain ways to sta
surement of a sub
remaining comput
of QEC, but the r
of fault-tolerant C
quantum computa
the number of req
ations. Qubit sys
erties that are nei
QEC methods to
will play a crucia
quantum compute

The general req
ware [25] are tha
quantum informat
Hamiltonian cont
fidelity for the ap
ficient initializatic
conflicting require
ware candidates t
describe those platforms that are currently being built into multi-qubit quantum computer systems and are expected to have the largest impact in the next decade. As we will see below in a definition of levels of the quantum computer stack and a sampling of vertical implementations and applications, the near-term advances in quantum computer science and technology will rely on algorithm designers understanding the intricacies of the quantum hardware, and quantum computer builders understanding the natural structure of algorithms and applications.

The Quantum Computer Stack

Computer architectures are often defined in terms of their various levels of abstraction or "stack," from the user interface and compiler down to the low-level gate operations on the physical hardware itself. The quantum computer stack can be defined similarly, as depicted in Fig. 2. However, the various levels of the quantum computer stack (especially the qubits themselves) are not yet cheap and commoditized like classical computer technology. So it is critical that quantum computers be designed and operated with the entire stack in mind, with a vertical approach of co-designing quantum computer applications to their specific hardware and all levels in between for maximum efficiency. Indeed, early quantum computer system development may parallel current classical application-specific integrated circuits (ASICs) used for specific and intensive computations such as molecular structure or machine learning.

In this section, we list the levels of the quantum computer stack and point to various approaches at each level. The key to co-designing quantum computers is to acknowledge the great opportunities at the interfaces between different levels of the stack, which requires a high level of interdisciplinarity between the physical sciences, engineering, and computer science. In the next section, we illustrate how various levels of the quantum computer stack will be exploited for several use cases.

Quantum Algorithms

Practical interest in quantum computing arose from the discovery that there are certain computational problems that can be solved more efficiently on a quantum computer than on a classical computer, notably number factoring (Shors algorithm [26]) and searching unstructured data (Grovers algorithm [27]). Quantum algorithms typically start at a very high level of description, often as a pseudo code. These algorithms are usually distinguished at this level of abstraction by very coarse scaling of resources such as time and number of qubits, as well as success metrics, such as success probabilities or the distance between a solution and the optimal value. Quantum algorithms can be broadly divided into those with provable success guarantees and those that have no such proofs and must be run with heuristic characteriza-

tions of success.

Once a quantum algorithm has been conceptualized with the promise of outperforming a classical algorithm, it is common to consider whether the algorithm can be run on near-term devices or for future architectures that rely on quantum error correction. A central challenge for the entire field is to determine whether algorithms on current, relatively small quantum processors can outperform classical computers, a goal called quantum advantage. For fault-tolerant quantum algorithms, on the other hand, there is a larger focus on improving the asymptotic performance of the algorithm in the limit of large numbers qubits and gates.

Shor's factoring [26] and Grover's unstructured search algorithms [27] are examples of "textbook" quantum algorithms with provable performance guarantees. These guarantees include a provable quadratic speedup for Grover and a superpolynomial speedup for Shor over known classical algorithms. A handy guide to known quantum algorithms is the quantum algorithm zoo, https://quantumalgorithmzoo.org/. Another important example is the HHL algorithm [28] which is a primitive for solving systems of linear equations. Factoring, unstructured search, and HHL are generally thought to be relevant only for larger fault-tolerant quantum computers.

Another class of quantum algorithms are quantum simulations [9, 29, 30], which use a quantum computer to simulate models of a candidate physical system of interest, such as molecules, materials, or quantum field theories whose models are intractable using classical computers. Quantum simulators often determine the physical properties of a system such as energy levels, phase diagrams, or thermalization times, and can explore both static and dynamic behavior. There is a continuum of quantum simulator types, sorted generally by their degree of system control. Fully universal simulators have arbitrary tunability of the interaction graph and may even be fault-tolerant, allowing the scaling to various versions of the same class of problems. Some quantum simulations do not require the full universal programmability of a quantum computer and are thus easier to realize. Such quantum simulators will likely have the most significant impact on society in the short run. Example simulator algorithms range from molecular structure calculations applied to drug design and delivery or energy-efficient production of fertilizers [31], to new types of models of materials for improving batteries or solar cells well beyond what is accessible with classical computers [32].

Variational quantum algorithms such as the variational quantum eigensolver (VQE) [6, 33, 34] and the quantum approximate optimization algorithm (QAOA) [8] are recent developments. Here, the quantum computer produces a complex entangled quantum state representing the answer to some problem, for example, the ground state of a Hamiltonian model of a molecule. The procedure for generating the quantum state is characterized by a set of classical control parameters that are varied in order to optimize an objective function, such as minimizing the energy of the state. One particular area of active exploration is the use of VQE or QAOA for tasks in machine learning [35] or combinatorial optimization [36, 37], as discussed below. Variational quantum solvers are a workhorse in near-term quantum hardware, partly because they can be relatively insensitive to systematic gate errors. However, these algorithms are usually heuristic: one cannot generally prove that they converge. Instead, they must be tested on real quantum hardware to study and validate their performance and compare to the best classical approaches.

Quantum algorithms are typically expressed at a high level with the need to estimate actual resources for implementation. This often starts with a resource estimate for fault-tolerant error-corrected quantum computers [38], where the quantum information is encoded into highly entangled states with additional qubits in order to protect the system from noise. Fault-tolerant quantum computing is a method for reliably processing this encoded information even when physical gates and qubit measurements are imperfect. The catch is that quantum error correction has a high overhead cost in the number of required additional qubits and gates. How high a cost depends on both the quality of the hardware and the algorithm under study. A recent estimate is that running Shors algorithm to factor a 2048-bit number using gates with a 10^{-3} error rate could be achieved with currently known methods using 20 million physical qubits [39]. As the hardware improves, the overhead cost of fault-tolerant processing will drop significantly; nevertheless, fully fault-tolerant scalable quantum computing is presently a distant goal. When estimating resources for fault-tolerant implementations of quantum algorithms, a discrete set of available quantum gates is assumed, which derive from the details of the particular error-correcting code used. There are many different techniques for trading off space (qubit number) for time (circuit depth), resembling conventional computer architecture challenges. It is expected that optimal error correction encoding will depend critically upon specific attributes of the underlying quantum computing architecture, such as qubit coherence, quantum gate fidelity, and qubit connectivity and parallelism.

Estimating resources for quantum algorithms using realistic quantum computing architectures is an important near-term challenge. Here, the focus is generally on reducing the gate count and quantum circuit depth to avoid errors from qubit decoherence or slow drifts in the qubit control system. Different types of quantum hardware support different gate sets and connectivity, and native operations are often more flexible than fault-tolerant gate sets for certain algorithms. This optimizing of specific

algorithms to specific hardware is the highest and most important level of quantum computer co-design.

Quantum Software

A quantum computer will consist of hardware and software. Key components of the software stack include compilers, simulators, verifiers, and benchmarking protocols, as well as the operating system. Compilers — interpreted to include synthesizers, optimizers, transpilers, and the placement and scheduling of operations — play an important role in mapping abstract quantum algorithms onto efficient pulse sequences via a series of progressive decompositions from higher to lower levels of abstraction. The problem of optimal compilation is provably intractable [40], suggesting a need for continuous improvement via sustained research and development. Since optimal synthesis cannot be guaranteed, heuristic approaches to quantum resource optimization (such as gate counts and depth of the quantum circuit) frequently become the only feasible option. Classical compilers cannot be easily applied in the quantum computing domain, so quantum compilers must generally be developed from scratch. Classical simulators are a very important component of the quantum computer stack. There is a range of approaches, from simulating partial or entire state vector evolution during the computation to full unitary simulation (including by the subgroups of the group of all unitaries), with or without noise. Simulators are needed to verify quantum computations, model noise, develop and test quantum algorithms, prototype experiments, and establish computational evidence of a possible advantage of the given quantum computation. Classical simulators generally require exponential resources (otherwise, the need for a quantum computer is obviated) and thus are only useful for simulating small quantum processors with less than 100 qubits, even using high-performance supercomputers. Simulators used to verify the equivalence of quantum circuits or test output samples of a given implementation of a quantum algorithm can be thought of as verifiers.

Benchmarking protocols are needed to test components as well as entire quantum computer systems. Quantum algorithm design, resource trade-offs (space vs. gate count vs. depth vs. connectivity vs. fidelity, etc.) [41], hardware/software co-design, efficient architectures, and circuit complexity are examples of important areas of study that directly advance the power of software.

The quantum operating system (QOS) is the core software that manages the critical resources for the quantum computer system. Similar to the OS for classical computers, the QOS will consist of a kernel that manages all hardware resources to ensure that the quantum computer hardware runs without critical errors, a task manager that prioritizes and executes user-provided programs using the hardware resources, and the peripheral manager that handles peripheral resources such as user/cloud and network interfaces. Given the nature of qubit control in near-term devices which requires careful calibration of the qubit properties and controller outputs, the kernel will consist of automated calibration procedures to ensure high fidelity logic gate operation is possible in the qubit system of choice.

Control Engineering

Advancing the control functions of most quantum computing implementations is largely considered an engineering and economic problem. Current implementations comprise racks of test equipment to drive the qubit gate operations, calibrate the qubit transitions and related control, and calibrate the measurement equipment. While appropriate for the early stages of R&D laboratory development, this configuration will limit scalability, systems integration, and applicability for fielded applications and mobile platforms, and affordability and attractiveness for future applications.

The quantum gate operations for most qubit technologies require precise synthesis of analog control pulses that implement the gates. These take the form of modulated electromagnetic waves at relevant carrier frequencies, which are typically in the microwave or optical domain. Depending on the architecture of the quantum computer, a very large number of such control channels might be necessary for a given system. While the advances in communication technologies can be leveraged, these have to be adapted for quantum applications. A significant level of flexibility and programmability to generate the required pulses with adequate fidelity must be designed into the control system for quantum computers.

The advancement of controls in quantum computing will ultimately require high-speed and application-specific optimized controls and processing. This situation mirrors the explosive advancement of the telecommunications industry in its implementation of 100 to 400+ Gb/s coherent digital optics formats and integrated RF and microwave signal processing for mobile applications. In the short run, however, the challenge we face is defining the control functions and features relevant for the target qubit applications and deriving the required performance specifications. This work is necessary before a dedicated, integrated, and scalable implementation, such as ASIC development, becomes viable. Near-term applications in quantum computing and full system design activities are critical in identifying and defining these needs. There are strong opportunities to engage engineering communities—in academia, national laboratories, and industry—with expertise ranging from computer architecture to chip design to make substantial advances on this front. To foster such efforts, it will be necessary to encourage co-design approaches and to identify common engineering needs and standards.

Generating the types of signals needed can benefit significantly from digital RF techniques that have seen dramatic advances in the last decade. We envision commercially available chipsets based on field-programmable

gate arrays (FPGAs) and ASICs that incorporate "System On a Chip" (SoC) technology, where processing power is integrated with programmable digital circuits and analog input and output functions.

Besides the signal generation necessary for implementing the gates, other control needs include passive and active (servo) stability, maintenance of system operation environment (vacuum, temperature, etc.), and managing the start-up, calibration, and operational sequences. Calibration and drift control [42] are important to both atomic and superconducting systems, though in somewhat different ways. In fixed-frequency superconducting systems, maintaining fidelities above 99% requires periodic calibration of RF pulse amplitudes; for tunable transmons, low frequency tuning of magnetic flux is required to maintain operation at qubit sweet spots. For atomic qubits, calibrating local trapping potentials and slow drifts in laser intensities delivered to the qubits is necessary. These calibration procedures, which can be optimized, automated, and built into the operating mode of the quantum computer system with the help of software controls, are candidates for implementation in SoC technologies. The development of optimal operating procedures for drift control and calibration processes will require innovation at a higher level of the stack. For example, dynamically understanding how control parameter drifts can be tracked and compensated, and when recalibration is needed (if not done on a fixed schedule) will require high-level integration.

Specific performance requirements will help drive progress, by co-design of engineering capabilities and quantum control needs. For example, there is a need for electronic control systems encompassing: (i) analog outputs with faster than 1 Gsamples/s; (ii) synchronized and coherent output with over 100 channels and extensible to above 1000 channels; (iii) outputs switchable among multiple pre-determined states; and (iv) proportional-integrative-derivative (PID) feedback control on each channel with at least kHz bandwidth. Common needs for optical control systems include (i) phase and/or amplitude modulation with a bandwidth of $\approx 100\,\mathrm{MHz}$; (ii) over 100 channels and extensibility to above 1000 channels; (iii) precision better than 12 bits (phase or amplitude); and (iv) operating wavelengths to match qubit splittings.

An essential consideration of the control engineering for high-performance quantum computers is noise. The noise in a quantum system has two distinct sources: one is the intrinsic noise in the qubits arising from their coupling to the environment, known as decoherence, and the other is the control errors. Control errors can be either systematic in nature, such as drift or cross-talk, or stochastic, such as thermal and shot noise on the control sources. The key is to design the controller in a way such that the impact of stochastic noise on the qubits is less than the intrinsic noise of the qubits, and the system-

atic noise is fully characterized and mitigated. Possible mitigation approaches include better hardware design, control loops, and quantum control techniques.

A key element in optimizing control is the characterization of the quantum system. For small systems, full characterization is possible, but as the system size grows, this becomes impractical. For single and two-qubit gates methods like randomized benchmarking [43–45] and gate-set tomography [46, 47] can be used to determine the quality of gates with a tradeoff between information and speed of the benchmark. For larger systems, system level benchmarks have been developed based on random circuits [41, 48]. It is not clear what the best approach for characterizing quantum systems whose scale prevents direct simulation. Research into characterization of quantum systems is an important component for being able to optimize system level control.

Qubit Technology Platforms

We view the various quantum computer technology platforms in terms of their ability to be integrated into a multi-qubit system architecture. To date, only a few qubit technologies have been assembled and engineered in this way, including superconducting circuits [48, 49], trapped atomic ions [50–52], and neutral atoms [53, 54]. While there are many other promising qubit technologies, such as spins in silicon, quantum dots, or topological quantum systems, none of these technologies have been developed beyond the component level. The research and development of new qubit technologies should continue aggressively in materials/fabrication laboratories and facilities. However, their maturity as good qubits may not be hastened by integrating them with the modular full-stack quantum systems development proposed here, so we do not focus this roadmap on new qubit development. In any case, once alternative qubit technologies reach maturity, we expect their integration will benefit from the full-stack quantum computer approach considered here.

It is generally believed that fully fault-tolerant qubits will not likely be available soon. Therefore, specific qubit technologies and their native decoherence and noise mechanisms will play a crucial role in the development of near-term quantum computer systems. There are several systems-level attributes that arise when considering multi-qubit systems as opposed to single- or dual-qubit systems. Each of these critical attributes should be optimized and improved in future system generations:

- **Native quantum gate expression.** Not only must the available physical interactions allow universal control, but high levels of gate expression will be critical to the efficient compilation and compression of algorithms so that the algorithm can be completed before noise and decoherence take hold. This includes developing overcomplete gate libraries, as well as enabling single instruction, multiple data (SIMD) instructions such as those given

by global entangling and multi-qubit control gates [55].

- **Quantum gate speed.** Faster gates are always desired and may even be necessary for algorithms that require extreme repetition, such as variational optimizers [6, 8, 34] or sampling circuits [56]. However, faster gates may also degrade their fidelity and crosstalk, and in these cases, the speed to complete the higher level algorithmic solution should take precedence.

- **Specific qubit noise and crosstalk properties.** Qubit noise properties should be detailed and constantly monitored, for there are many error mitigation techniques for specific or biased error processes that can improve algorithmic success in the software layer. Quantum gate crosstalk is usually unavoidable in a large collection of qubits, and apart from passive isolation of gate operations based on better engineering and control, there are software solutions that exploit the coherent nature of such crosstalk and allow for its cancellation by design.

- **Qubit connectivity.** The ability to implement quantum gate operations across the qubit collection is critical to algorithmic success. While full connectivity is obviously optimal, this may not only lead to higher levels of crosstalk, but ultimately resolving the many available connection channels may significantly decrease the gate and algorithmic speed. Such a trade-off will depend on details of the algorithm, and a good software layer will optimize this trade-off. A connection graph that is software-reconfigurable will also be useful.

- **High level qubit modularity.** For very large-scale qubit systems, a modular architecture may be necessary [3, 5, 57]. Just as for multi-core classical CPU systems, the ability to operate separated groups of qubits with quantum communication channels between the modules will allow the continual scaling of quantum computers. Modularity necessarily limits the connectivity between qubits, but importantly allows a hierarchy of quantum gate technologies to allow indefinite scaling, in principle.

Increasing the number of qubits from hundreds to thousands will be challenging because current systems cannot easily be increased in size via brute force. Instead, a new way of thinking on how to reduce the number of external controls of the system will be needed to achieve a large number of qubits. This could be approached by further integrating control into the core parts of the system or by multiplexing a smaller number of external control signals to a larger number of qubits. In addition, modularizing subsystems to be produced at scale and integrating these into a networked quantum computer may well turn out to be the optimal way to achieve the necessary system size [10, 57]. Many challenges and possible solutions will only become visible once we start to design and engineer systems of such a size, which will, in turn, be motivated by scientific applications.

Quantum Computer Case Studies

Below we briefly illustrate the use of the quantum computer full stack in several case studies, from optimization problems and programmable simulations of quantum problems to quantum error-correcting codes and "textbook algorithms. These examples are not meant to be exhaustive, but may indicate how any future quantum application might be realized and co-designed. Mapping these problems onto particular quantum computing modes and specific hardware platforms illustrates the critical translation from cost functions or hard problems to native interactions between qubit systems. A full stack systems approach to any quantum computation is also expected to inform other scientific applications, even those not yet discovered. In the below use cases, we highlight particular computing modes and qubit technologies that are available now, are expected to scale up significantly in the next 2-10 years, and appear well-matched to the application.

I. Gate-Based Quantum Simulation

The brute-force classical simulation of n interacting qubits requires a solution to 2^n complex differential equations, limiting classical computers to simulate arbitrary dynamics on no more than about 50 qubits. One of the most promising near-term applications of quantum computers is thus the simulation of difficult quantum Hamiltonian models [29] such as frustrated magnetism, superconductivity, and topological dynamics in condensed matter physics or quantum field theories in nuclear and high energy physics. There are many specialized approaches to quantum simulation [9] that may have limited tunability but may be easier to implement because of symmetries or natural interactions in the experimental system. However, the flexibility of universal gate-based quantum simulators may allow the full power of quantum computers to solve entire classes of models without necessarily specializing in particular cases.

General Hamiltonian simulation evolves an initial state according to the given Hamiltonian model of the system under study. The goal is to minimize the gate count as a function of system size, evolution time, desired precision, and other parameters. There is a wide range of polynomial-time quantum algorithms that solve this problem. They roughly fall into two types: product formulas [30, 58–61], and linear combinations of unitaries [62, 63]. Apart from theoretical studies aiming to discover new efficient quantum algorithms and improve asymptotic upper bounds, there are also resource count and empirical performance studies [61, 64, 65] as well as

approaches that take advantage of the spatial locality of the Hamiltonian to reduce the gate count [66, 67].

Recent results have provided strong theoretical evidence that sampling from the distribution obtained by evolving some initial state with a Hamiltonian and then measuring cannot be solved by any polynomial time classical algorithm [68]. This suggests that the simulation of quantum dynamics is a problem well-suited for a solution by a quantum computer. The major open problem is to understand how robust this exponential speedup is to experimentally relevant noise.

Here we consider universal gate-based approaches to quantum simulation, in its original spirit of attacking a general class of quantum problems [29]. We assume that a model quantum Hamiltonian is expressed in terms of controllable interactions or gates on a collection of qubits, allowing the simulation of arbitrary quantum processes [30]. Such universal quantum simulations can be used to find equilibrium properties of arbitrary Hamiltonians but also allow the more difficult problem of evolving them in time for simulations of quantum dynamics and nonequilibrium processes in physical systems. Below we consider two gate-based quantum simulation applications: variational simulations of Hamiltonian ground states and the simulation of quantum field theories.

(a) Variational Estimation of Ground States

For many problems in quantum chemistry and materials science, it is of great interest to understand the structure of the system electronic ground state or thermal state under a given Hamiltonian. While preparing the ground state of a general Hamiltonian (even if it is spatially local) is believed to be hard even on a quantum computer [69, 70], there are algorithms for preparing ground states in special cases [71, 72]. Quantum Approximate Optimization Algorithms (QAOA) [8], discussed further in the next section, may also be used to heuristically approach Hamiltonian ground states [73].

Here we concentrate on the implementation of the VQE algorithm [6] of an electronic Hamiltonian using a gate-based approach. The general procedure of VQE involves two steps: The first (preparation) step maps the problem Hamiltonian to a collection of qubits. In cases such as electronic structure in molecules or materials, this step typically involves a transformation from the native fermionic electron operators to spin or qubit operators through the Jordan-Wigner or Bravyi-Kitaev transformation [74]. The binary occupancy of a fermionic lattice site is replaced by an effective qubit through this substitution. This transformation results in nonlocal "string" operators between the effective qubits [6, 74], which typically represent the most expensive part of the quantum circuit and are best expressed with highly connected qubit graphs [75]. The second (operational) step directly evaluates the Hamiltonian expectation with respect to an initial quantum state parameterized in terms of variables that are classically optimized to minimize the Hamiltonian function, as depicted in Fig. 3.
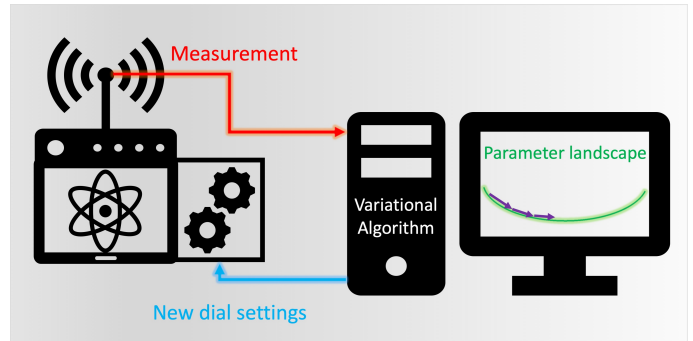

FIG. 3. Schematic of the measurement and feedback optimization in the variational quantum eigensolver algorithm (from Ref. [34]).

(b) Simulating Dynamics in Quantum Field Theories

An important class of gate-based quantum simulation is the modeling of classically intractable dynamics of quantum field theories (QFTs) at the heart of many phenomena in condensed matter, high-energy, and nuclear physics. Quantum simulations of QFT dynamics could shed light on many important scientific problems, including phase transitions in the early universe, the response of new exotic materials, explosive astrophysical events and the inner structure of neutron stars, studies of topological features in quantum systems, and high-energy collisions used to search for new fundamental physics.

For scalar quantum field theories, such as those used to describe electron densities in materials or spontaneous symmetry breaking responsible for generating the masses of subatomic particles, the resource estimates for mapping the problem onto qubit registers and the entangling gates have been performed [76–78]. Early estimates for a one-dimensional (1D) lattice scalar field theory shows that some small-scale problems can be mapped onto quantum computers that might become available within the next decade. Non-Abelian gauge field theories require even more control, although the first SU(2) calculations has been performed on IBM's QExperience [79]. Resource estimates for qubits and entangling gates are currently being determined for both SU(2) and SU(3) models. The resource requirement estimates are complicated by the need to explicitly enforce Gauss's law. Quantum chromodynamic (QCD) theory with six flavors of quarks — each with four Dirac degrees of freedom and three colors — requires a much larger number of qubits per site. Low-dimensional quantum field theories that share features of the Standard Model are beginning to be explored with available quantum hardware [80–85].

Gate-based universal quantum simulators may be preferable over analog simulation methods for QFT calculations in the longer term, given the need for quantifying uncertainties. However, there is significant value in

performing simulations on near-term devices until error-corrected quantum computers become available for simulation. Computations performed on present-day superconducting devices are limited by the qubit coherence time, and the time-dependent behavior of the communication fabric. Measurement errors can be mitigated for modest-sized Hilbert spaces, but this may become challenging at larger scales due to the required classical computing resources.

Given the current status of the hardware capabilities and the resource requirements for QFT methods, co-design is expected to play a key role in closing the gap. Starting in the 1980s, co-design played an essential role in developing high performance computing (HPC) capabilities for high-energy physics theories, such as calculating lattice QCD models. An example is the co-design team of Columbia–Brookhaven–IBM that led the design and development of customized computational engines for QCD. In their approach, RAM was integrated onto the floating point unit, with a four-dimensional toroidal communication fabric. This effort led to IBMs BlueGene/L,P,Q series of supercomputers. We anticipate that similar co-design efforts will be required for the development of quantum devices for lattice QFT, e.g., for an efficient evaluation of the sequence of Trotter steps required in time evolution.

This task requires a close collaboration among a number of experts in a range of areas. First, particle, nuclear, and condensed matter theorists with expertise in QFT and phenomenology are required to prioritize target observables. These problems must be mapped onto target hardware, initial states prepared and evolved forward in time, which requires efforts from quantum computer scientists, developers, and engineers with detailed knowledge and hands-on access to the devices. Circuit optimization will be key in the near term to accomplish the goals of the project.

Over the course of the next decade, we expect new quantum algorithms to be developed to tackle problems including real-time dynamics, scattering and inelastic processes in low-dimensional gauge theories, such as the 1D Schwinger model and Z_2 models, interacting scalar field theories and SU(N) gauge theories, and subsequently extended to higher dimensions. Early implementations on advanced quantum systems are expected to lead to new physical insights in reliable calculations of dynamic SU(2) and QCD processes, including at finite density, into new and exotic materials, and into the design of quantum memories for quantum devices.

(c) Physical Platform

For the above gate-based quantum simulations, we focus on the trapped ion quantum computer architecture [75, 86] owing to its high gate fidelities, long-range connectivity and flexible gate expression. Other architectures such as neutral atoms [87]), and superconducting qubits [88–90] can also be considered as potential leading candidates. The noise in trapped ion experiments appears to scale sublinearly [91], indicating that the dominant source of error may be due to a coherent source, which can be suppressed through calibration procedures and pulse-shaping techniques. This allows for larger quantum circuit depths and permits an extension of the practical lifespan of computational experiments, compared to physical platforms where the error is dominated by native qubit decoherence.

More flexible and expressive gate sets allow a given unitary operation to be compiled down to a shorter depth quantum circuit, thus increasing performance. Trapped ion quantum computing allows parallel non-overlapping gates [92], native long-range interactions (all-to-all connectivity that obviates SWAP operations), small-angle two-qubit rotations, and global gates that act on multiple qubits [93]. The above features are instrumental in pushing the ability to implement deep algorithms for gate-based quantum simulations and other applications in practice.

A good example of this reduction in circuit complexity is a key subroutine in a variational quantum algorithm that takes small steps through the Hilbert space, as shown in Fig. 4. Here, rotation operations with a small angle $d\theta$ sandwiched by the CNOT gates can be replaced by full rotation operations and entangling Ising XX gates evolving with a small angle. Because the single-qubit rotations are generally much higher quality than the entangling gates in ion traps, translating the small evolution time from the rotation gates to the entangling gates leads to significantly lower errors [65, 75].
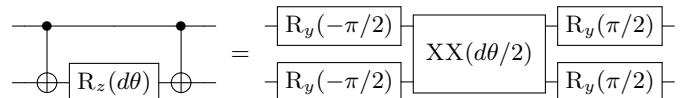

FIG. 4. In the variational quantum eigensolver algorithm commonly applied to chemistry simulations, small changes in the quantum state induced by controlled rotation involve CNOT gates between qubits (left). However, with the Ising or XX gate native to ion trap quantum computers [91], these small controlled rotations can be expressed as a small-angle Ising gate accompanied by single-qubit rotation operations as written. This expression greatly reduces noise [75]. The rotation operations are $R_x(\theta) = R(\theta, \phi = 0)$, $R_y(\theta) = R(\theta, \phi = \pi/2)$ and $R_z(\theta) = R_x(\pi/2)R_y(\theta)R_x(-\pi/2)$.

Many gate-based quantum simulations have been performed with ion trap quantum computers. These include the simulation of many-body spin dynamics [94], real-time dynamics of a lattice gauge theory (the Schwinger model) [80], and simulations of Fermi-Hubbard ground states [95]. To make efficient use of quantum resources, these demonstrations map the original problem to a spin model by exploiting the native long-range interactions, which can be directly and efficiently implemented on an

ion trap architecture. The second class of gate-based simulations based on the VQE algorithm has been performed with trapped ion quantum computers to calculate the ground state of molecules, from H_2 and LiH [86] to H_2O [75]. Related experiments calculated the binding energy of the deuteron (bound proton and neutron) in both superconducting [96] and ion trap systems [97].

II. Combinatorial Optimization with QAOA

Combinatorial optimization describes a broad class of problems aimed at minimizing a cost function over a combinatorially large set of possible solutions. Examples include problems such as route finding (e.g., the traveling salesman and other graph optimization problems), cost minimization, and portfolio optimization. Optimization applications may ultimately become the most general use case for quantum computers, as they appear in nearly all areas of the natural sciences, engineering, and social sciences.

Here we concentrate on the use of QAOA [8] to solve a particular graph optimization problem. While QAOA can be implemented on a universal gate-based quantum computer, the simple form of its dynamics suggests that QAOA could be implemented on near-term quantum information processors using simpler and more direct techniques than are required for building universal systems. In what follows, we discuss the implementation of the QAOA on the platform of optically-addressed Rydberg atoms. This platform offers different forms of interactions and controls and so is suitable for various classes of geometries for the graph of interactions in the problem Hamiltonian. This point is crucial: even though any NP-complete problem can be encoded in the ground state of a programmable Ising model, and even with nearest-neighbor interactions on a two-dimensional lattice, the extra overhead required to map any given problem onto such an architecture can be large, which can strongly limit the class of problems that can be encoded with a given pattern of interactions. The long-range and non-local interactions available in the Rydberg atom system can greatly expand the range of problems that can be addressed using a device that implements quantum approximate optimization. Identifying optimal embeddings of the problem to be solved into particular architectures of connections is a widely studied problem in quantum software.

(a) Maximum Independent Set Problem

To illustrate these considerations, we focus on a specific graph optimization problem known as the Maximum Independent Set (MIS), a well known problem involving the search for the maximum set of vertices in a graph that share no connection [98]. Finding the MIS is a canonical optimization problem in the NP-hard complexity class. The most advanced classical algorithms designed to solve such problems exactly often display exponential scaling for vertex sizes above 100. In particular, there are known instances of graphs with N above 300 for which classical

supercomputers cannot find the MIS. The MIS problem has practical applications in areas such as signal routing in ad-hoc wireless networks such as 5G mobile networks, finance, social network analysis, and machine learning.

The MIS problem can be encoded into the Hamiltonian

$$H_{\mathrm{MIS}} = -\Delta \sum_{v \in V} n_v + U \sum_{(v,w) \in E} n_v n_w, \qquad (3)$$

where each vertex in the set of all vertices V is represented by a qubit, $n_v = |1\rangle_v \langle 1|$, E denotes the edges connecting vertices, and U is the energy cost to excite connected vertices. When $U \gg 0$, the ground state of the Hamiltonian encodes the solution to the MIS problem specified by the vertices and edges V, E.
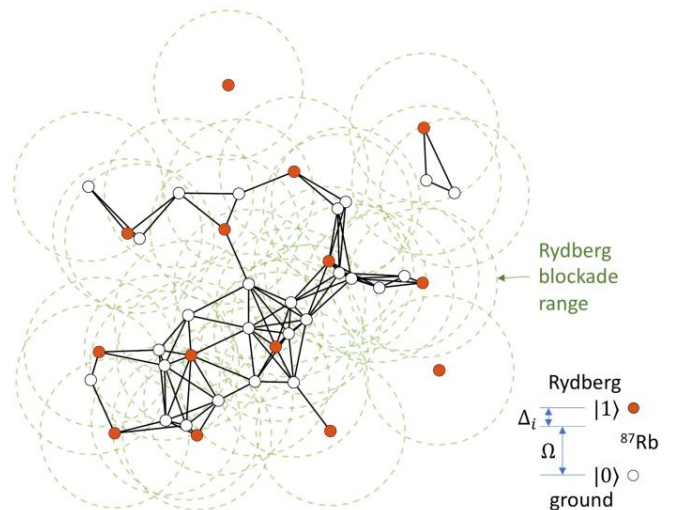

FIG. 5. Maximum Independent Set (MIS) problem on a random Unit Disk Graph, which is natively represented in an array of atoms with a Rydberg blockade interaction. The MIS of this instance is labeled in red. With variable weights on each vertex, the problem can be generalized to the maximum-weight independent set problem. The legend shows the encoding of qubits in the ground and excited Rydberg state of a neutral atom. The detuning Δ_i of a driving field with resonant Rabi frequency Ω controls the Rydberg blockade radius, which is denoted by the green dashed circles surrounding each atom and is assumed uniform ($\Delta_i = \Delta$).

(b) Physical Platform

The QAOA for the MIS problems can be efficiently encoded in the Rydberg atom platform. For example, on a subset of graphs, the Unit Disk Graphs (UDGs), the MIS problem can be natively represented with neutral trapped atoms interacting via Rydberg states, without encoding overhead. As shown in Fig. 5, every vertex maps directly onto one atom [99], and the edge cost U comes from the Rydberg blockade constraint. Here, the strong Rydberg–Rydberg interaction impresses a large energy penalty for having two atoms both in excited Rydberg states, represented by the last term of Eq. 3.

Crucially, the specific MIS problem can be directly encoded in the positions of the atoms (see Fig. 5), which can be arbitrarily specified in two or three dimensions using reconfigurable optical tweezer arrays based on existing control hardware such as liquid crystal spatial light modulators or digital micromirror devices [100–102]. The driving laser pulses that execute the QAOA algorithm can then be global, significantly reducing the control complexity. More general MIS problems can be approached with more sophisticated control and local addressing capabilities. For example, local light-shifts can be used to realize variable weights Δ_i on each vertex by addressing each atom with a unique driving field frequency and thereby generalizing this situation to the maximum-weight independent set (MWIS) problem [98]. Locally programmed interactions using site-resolved excitations to Rydberg states can add additional edges beyond the unit disk graphs, allowing more general instances of MIS problem to be efficiently encoded.

Current demonstrations of optical tweezer arrays for neutral atoms have realized coherent evolution in 51-atom chains [53], and scaling to larger arrays of 100–1000 atoms is within the reach of current technology. This opens the door to testing for quantum advantage enabled by QAOA for practically relevant problems. Trapped ion systems [103] and coherent superconducting circuits [90] also represent natural platforms for performing QAOA. Existing superconducting systems support programmable Ising models with local connectivity over 50 superconducting qubits [48], while programmable spin models featuring long-range interactions have been implemented in systems exceeding 50 trapped ion qubits [104]. Such systems can also be employed for realizing and testing QAOA on problems of increasing complexity.

Beyond realizing large-scale combinatorial optimization experimentally, there are several open challenges and opportunities in this area. One is to identify how subsets of NP-complete problems such as MIS can be mapped onto other problem classes with near-term devices. For example, how generally applicable would a fast sampling algorithm for the unit-disk MIS problem be? Another is to understand the optimal parameters and Hamiltonian H_0 for implementing QAOA for a particular type of problem. A final opportunity is to extend QAOA beyond solving optimization problems, to the rapid preparation of entangled states for other applications [73].

III. Quantum Error Correction and Architectures

Current quantum algorithms are limited by the number of qubits and the quality of gate operations. The physical nature of current qubit technologies and their lack of modularization can limit high level systems engineering and computer architecture approaches to scaling. In this use case, we present a vision for an interdisciplinary architectural goal of the "virtualization" of qubits, or their abstract representation, through quan-

tum error corrected memories [38]. This activity requires computer architects to design the virtualization strategy, systems engineers to define and design key modules, hardware engineers to improve the classical control and reliability of the system, classical and quantum information theorists to optimize error correction protocols for the platform, and physicists to design new qubits and gate protocols. The team requires academics to explore the design space, industrial partners to develop equipment and scale promising ideas, and national laboratories to aid in the characterization of materials and testing of devices.

(a) Qubit Virtualization

The architecture we envision separates more volatile qubits involved in operations from highly stable qubits stored in memory. This availability of high-quality memory can simplify device design at the cost of requiring more serial operations. We can study the performance of algorithms implemented on this architecture as the ratio of active and memory qubits changes, and estimate the required fidelity of operations and memory for achieving key scientific goals such as simulation of molecules, materials, and high-energy physics. Furthermore, virtualization provides an abstraction to enable computer sciences to use different devices without the need to understand the physical layers.

In the near term, virtualization can be performed using a heterogenous quantum architecture where certain qubits are designed for fast operations, and other qubits are designed for long lifetimes. The cost and benefits of transferring quantum information to and from memory can already be tested and optimized at this stage. In the long term, we envision that the two qubits could correspond to two quantum error correction codes: for example, a color code for efficient operation [105] and a finite-rate code for efficient data storage [106]. The first step in this direction is implementing a fault-tolerant quantum memory with a quantum error correction code.

For universal quantum computing, a broader control space is needed, but the control problem becomes simpler if our goal is to implement a fault-tolerant quantum memory. By optimizing the use of these simple controls, we should be able to study how the performance of the memory improves as it scales up under realistic conditions.

As the memory is built, we can employ engineering principles to improve the lifetime of the quantum states. A natural strategy is to develop models that track the most likely errors at regular time intervals. Adopting the language of wireless communications, we might refer to this time interval as the channel coherence time and to this process as the channel estimation phase. To optimize for the next time interval, we might develop strategies to efficiently synthesize a quantum error correction code that possesses logical operators that are resilient to those errors identified by the previous channel estimation

phase, thereby relaxing full fault-tolerance. Note that the few well-known quantum error correction codes will always be feasible solutions, and our strategy can only do better.

(b) Virtualization at the Hardware Level

In this use case, we lay out a vertically integrated project to develop virtualized qubits based on physical superconducting qubits combined into fault-tolerant qubits. We emphasize that both the specific choices of the top architecture and the physical platform are only examples for concreteness.

Superconducting qubits occupy a large design space with potential tradeoffs among gate speed, coherence time, and readout fidelity [107]. Recent work has examined transferring quantum information from transmon qubits that can be controlled and processed to cavity qubits with long memory times [108]. This design space allows us to test the ideas of modularization and virtualization with devices that can be built immediately.

Consider, for example, a system of two coupled transmons and two multi-mode cavities. The cavities can store N qubits of information that can be readily swapped to each transmon, and two-qubit quantum gates can act on the pair of transmons. Computer scientists can study extended models of the system to examine tradeoffs (with approximately fixed physics parameters) as the number of qubits and cavities increases. Meanwhile, physicists can continue to develop better methods for transferring information between cavities and transmons [109]. This approach could be combined with bosonic codes that can further extend the memory times of cavity qubits [110].

A heterogeneous design for superconducting quantum circuits, combining a variety of qubits and devices, could be more powerful than any single-qubit approach. One could consider a system that combines transmons, cavity-encoded qubits [110], "$0 - \pi$" qubits [111], and other designs [107], each with its own role in the system. We envision that this heterogeneous design incorporating noise suppression at the device level will be fed into a quantum error-correcting code arrayed on a two-dimensional lattice, which provides further protection at the software level. For error correction via quantum coding to work effectively, improvements will be needed in both the underlying physics and the engineering of the control systems.

Separate from the challenge of memory and more difficult is the implementation of universal fault-tolerant operations. We expect that this is a place where hardware dependent solutions can yield a significant advantage. For superconductors, a promising area of research is to develop fault-tolerant gates in the context of the bosonic codes. An alternate approach is to design a circuit that implements the light-weight universal gate sets on small block codes. Realizing two universal fault-tolerant qubits is a grand challenge that will stretch our ability to make large and reliable systems.

Architectural requirements for abstract quantum systems place constraints and design requirements on engineered quantum systems. The construction of systems that achieve these goals requires tight collaboration between computer scientists, engineers, and physicists.

IV. Standard Quantum Algorithms

Historically, Shor's [26] and Grover's [27] algorithms, which are sometimes referred to as "textbook algorithms," were the first algorithms of practical relevance where a quantum computer offered a dramatic speedup over the best known classical approaches. Here we consider co-design problems and full-stack development opportunities that arise from mapping these textbook algorithms to quantum computing platforms.

At the top of the stack, challenges that both algorithms face include implementing classical oracles with quantum gates. In Grover's algorithm, the oracle can be implemented using Bennett's pebbling game [112]. However, it is a non-trivial task to find an efficient reversible circuit, since the most efficient implementation on a quantum computer may not follow the structure suggested by a given classical algorithmic description, even when the latter is efficient. There are also intriguing questions in terms of how we can use classical resources to aid in quantum algorithms. For example, in Shor's algorithm we can leverage classical optimizations such as windowed arithmetic [113], or trade off quantum circuit complexity with classical post-processing complexity [39, 114].

One of the key challenges in implementing textbook algorithms in physical systems is to optimize these algorithms for a given qubit connectivity and native gate set. While these technology-dependent factors will not affect asymptotic scaling, they could greatly influence whether these algorithms can be implemented on near-term devices. For instance, high connectivity between qubits can provide significant advantages in algorithm implementation [91]. Also, in this vein, work has been done to implement Shor's algorithm with a constrained geometry (1D nearest neighbor [115, 116]), but there are many open questions that involve collaborations across the stack.

Developing implementations of Shor's algorithm and Grover's algorithm will provide exciting avenues for improved error correction, detection, and mitigation. While error correction codes are often designed to correct specific types of errors for particular physical systems, considering error correction for textbook algorithms provides a basis for designing error correction for both application and hardware. Because the output of these textbook algorithms is easily verifiable, they are good testbeds for error mitigation and characterization. For example, simple error correction/mitigation circuits, including randomized compiling [117, 118], could be implemented in the context of small implementations of Grovers algorithm and Shors algorithm to better understand the challenges in integrating these protocols into more complex ones. While this approach has been used in the quantum an-

nealing community [119], it could be fruitful to explore in more detail for textbook gate-based algorithms.

Grovers algorithm seems to break down in the presence of a particular type of error [120, 121] that appears to be unrealistic in actual physical systems. For other algorithms, realistic errors do not appear to be as detrimental [122]. Testing textbook algorithms on different architectures with and without error mitigation would provide us with a way to explore the space of errors relative to a specific algorithm, and give insight into which realistic errors are critical. This would inform error mitigation (not necessarily correction) techniques at both the code and hardware level, tailored to specific algorithms. This way of viewing error correction calls for a full integration of experts at the hardware, software, and algorithm design levels.

We also expect that the work on these textbook algorithms will lead to improved modularity in the quantum computing stack. In software design, modularity refers to the idea of decomposing a large program into smaller functional units that are independent and whose implementations can be interchanged. Modular design is a scalable technique that allows the development of complex algorithms while focusing on small modules, each containing a specific and well isolated functionality. These modules can exist both at the software level as well as at the hardware level. As an example in classical computing, the increased use of machine learning algorithms and cryptocurrency mining has led to a repurposing of GPUs.

In the design of full implementations of quantum algorithms such as Shors and Grovers, modular design can be applied by using library functions that encapsulate circuits for which an optimized implementation was derived earlier. Examples include quantum Fourier transforms [1, 123, 124], multiple-control gates [125, 126], libraries for integer arithmetic and finite fields, and many domain-specific applications such as chemistry, optimization, and finance [127–129]. All major quantum computing programming languages are open source, including Googles Cirq, IBMs QisKit, Microsofts Quantum Development Kit, and Rigettis Quil, which facilitates the development and contribution of such libraries. Highly optimized libraries that are adapted to specific target architectures, as well as compilers that can leverage such libraries and further optimize code, are great opportunities for large-scale collaborative efforts between academia, industry, and national labs.

Like libraries, programming patterns provide opportunities for modularity. Programming patterns capture a recurring quantum circuit design solution that is applicable in a broad range of situations. Typically, a pattern consists of a skeleton circuit with subroutines that can be instantiated independently. Examples of patterns are various forms of Quantum Phase Estimation (QPE) [130–133], amplitude amplification [27], period finding [134],

hidden subgroup problems [135], hidden shift problems [136], and quantum walks [137].

Finally, we expect that implementing textbook algorithms will become important benchmarks for the quantum computing stack as a whole and also at the level of individual components. The need for such benchmarks is evident in the recent proliferation of a variety of benchmarks that test various aspects and components of quantum systems and entire systems, such as quantum volume [138], two-qubit fidelity, cross-entropy [95, 139, 140], probability of success, reversible computing [141], and the active IEEE working group project [142]. It is not likely that any single benchmark will characterize all relevant aspects of a quantum computer system. However, implementing textbook algorithms provides an easy-to-verify test of the full quantum system from hardware to software, as all aspects of the system must work in concert to produce the desired output.

Outlook and Paths Forward

The field of quantum computing is now in an era of quantum systems development, where full stack consideration is poised to accelerate progress in building, using, and optimizing quantum computers. The birth and development of quantum computers have taken place in a scientific atmosphere, and we believe that the largest opportunities in quantum computing applications in the coming decade will continue in the realm of scientific discovery. This stems from the physics of good qubit systems, and the electrical and systems engineering of controlling them to computer science approaches in optimizing the expression of algorithms, finally to be used for nearly all areas of science that will reap the benefits of a new type of computing tool.

Community building and engagement

Quantum computing pulls the rug out from underneath the principles of conventional computing, and its progress will be accelerated by the active engagement of a broad society of users. One mechanism is to underwrite various forms of challenge competitions, which have been successful in stimulating interest and engagement in conventional computing platforms while reaching a large group of coders (e.g., Google Jams, Facebook, and Topcoder competitions). In the quantum computing world, such challenges have already been successfully held, such as the IBM layout and computation challenges, the Microsoft Q# coding challenges, and the meQuanics quantum circuit minimization challenge. Supporting more challenges like these might help to engage a future quantum workforce around concrete and small scale problems as well as stimulate a cultural shift toward quantum computational thinking. These events can also exploit the growing quantum cloud presence by running routines on real hardware and their virtual simulators.

Universities and national laboratories across the country are setting up quantum information science and tech-

nology centers to grow their faculty, unify with their researchers and students, and engage with an increasingly fascinated and enthusiastic general public. These centers, often regional in nature, might play an important role in the future, growing to become large research hubs and advancing state-of-the-art quantum information science. However, they currently play a limited role in consolidating resources, coordinating efforts, and developing collaborative research programs between industry, academia, and national laboratories. All efforts discussed in this roadmap to accelerate the progress in quantum computing should coordinate with these centers of excellence, stimulating the exchange of people and ideas, and growing their potential.

Quantum Computing Laboratory User Facilities

The grand challenges of quantum computing, from finding useful quantum applications to building the machines themselves, are well-motivated by the vast potential scientific and technological opportunities that lie ahead. It is the consensus of the quantum computing community as represented by our workshop that scientific quantum computing user facilities (QCLabs), bringing all of the science, computer science, and engineering of the quantum computer stack together in one place, may best address these challenges and opportunities.

Each QCLab could have its own type of qubit system, scaling architecture plans, or use-case family, for instance. But the QCLabs would also provide the capabilities to realize these complex systems and enable collaboration to feature quantum computer co-design up and down the quantum stack, continually iterating on device design, software optimization, and use cases. These facilities would feature a deep bench of scientists and engineers permanently on site (faculty, staff engineers, etc.), but would also support visitors from all levels: from theorists and algorithmic designers, computer scientists, electrical and computer engineers, to physicists, chemists, and materials scientists. Each level would be expected to contribute to not just the use of the devices, but also in the continual building of next generations of quantum computers at the QCLab.

QCLabs are expected to be highly leveraged by industrial efforts in quantum computing. There are many industrial teams now building ever more powerful quantum computing systems, with many of these efforts providing cloud access to their systems. We expect the availability of various types of industrial quantum computers to grow rapidly in the coming years. These services will have varying characteristics such as qubit platform, qubit number, gate depth, and level of connectivity and expression, that can be exploited by the scientific community. However, these cloud services will not likely allow users to dig deep into the stack to potentially optimize the low level control of qubits in order to achieve a particular scientific application, and such systems will not likely be built for scientific research goals having no obvious commercial use. Moreover, commercial quantum computer cloud providers may not want much flexibility in their system design, as this could degrade high level performance for a widely available and reliable cloud service. We envision that the QCLabs would leverage industrial cloud quantum computers in order to assist and benchmark aspects of the QCLab activity itself. The industry should be very interested in this interaction, as QCLabs may help find quantum killer applications of the future.

Acknowledgements and Notes

---

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2011).

[2] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, Nature **464**, 45 (2010).

[3] C. Monroe and J. Kim, Science **339**, 1164 (2013).

[4] M. Saffman, National Science Review **6**, 24 (2018).

[5] M. H. Devoret and R. J. Schoelkopf, Science **339**, 1169 (2013).

[6] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, New Journal of Physics **18**, 023023 (2016).

[7] J. Preskill, in *Proceedings of The 36th Annual International Symposium on Lattice Field Theory PoS (LATTICE2018)* (SISSA Medialab, 2019), vol. 334, p. 024.

[8] E. Farhi, J. Goldstone, and S. Gutmann, arXiv:1411.4028 (2014).

[9] E. Altman, K. R. Brown, G. Carleo, L. D. Carr, E. Demler, C. Chin, B. DeMarco, S. E. Economou, M. Eriksson, K.-M. C. Fu, et al., arXiv: 1912.06938 (2019).

[10] D. Awschalom, K. K. Berggren, H. Bernien, S. Bhave, L. D. Carr, P. Davids, S. E. Economou, D. Englund, A. Faraon, M. Fejer, et al., arXiv:1912.06642 (2019).

[11] D. P. DiVincenzo, Phys. Rev. A **51**, 1015 (1995).

[12] R. Raussendorf, D. E. Browne, and H. J. Briegel, Phys. Rev. A **68**, 022312 (2003).

[13] D. Solenov, arXiv preprint arXiv:1906.11701 (2019).

[14] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, arXiv:0001106 (2000), arXiv: quant-ph/0001106.

[15] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, in *45th Annual IEEE Symposium on Foundations of Computer Science* (2004), pp. 42–51.

[16] T. Kadowaki and H. Nishimori, Phys. Rev. E **58**, 5355 (1998).

[17] A. Das and B. K. Chakrabarti, Rev. Mod. Phys. **80**, 1061 (2008).

[18] T. F. Rønnow, Z. Wang, J. Job, S. Boixo, S. V. Isakov, D. Wecker, J. M. Martinis, D. A. Lidar, and M. Troyer, Science **345**, 420 (2014).

[19] Y. Lin, J. Gaebler, F. Reiter, T. R. Tan, R. Bowler, A. Sørensen, D. Leibfried, and D. J. Wineland, Nature **504**, 415 (2013).

[20] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).

[21] A. Steane, IEEE Transactions on Information Theory **45**, 2492 (1999).

[22] J. Preskill, Quantum **2**, 79 (2018).

[23] J. Preskill, in *Introduction to Quantum Computation and Information* (World Scientific, 1998), pp. 213–269.

[24] A. M. Steane, Phys. Rev. A **68**, 042322 (2003).

[25] D. P. DiVincenzo, Fortschritte der Physik **48**, 771 (2000).

[26] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).

[27] L. K. Grover, in *Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, USA, 1996), STOC '96, pp. 212–219.

[28] A. W. Harrow, A. Hassidim, and S. Lloyd, Phys. Rev. Lett. **103**, 150502 (2009).

[29] R. Feynman, International Journal of Theoretical Physics **21**, 467 (1982).

[30] S. Lloyd, Science **273**, 1073 (1996).

[31] M. Reiher, N. Wiebe, K. M. Svore, D. Wecker, and M. Troyer, Proc. Nat. Acad. Sci. **114**, 7555 (2017).

[32] J. E. Saal, S. Kirklin, M. Aykol, B. Meredig, and C. Wolverton, Jom **65**, 1501 (2013).

[33] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. OBrien, Nature Communications **5**, 4213 (2014).

[34] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Quantum **3**, 191 (2019).

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

[36] Z.-C. Yang, A. Rahmani, A. Shabani, H. Neven, and C. Chamon, Phys. Rev. X **7**, 021027 (2017).

[37] Y.-H. Oh, H. Mohammadbagherpoor, P. Dreher, A. Singh, X. Yu, and A. J. Rindos, arXiv:1911.00595 (2019).

[38] B. M. Terhal, Rev. Mod. Phys. **87**, 307 (2015).

[39] C. Gidney and M. Eker, arXiv:1905.09749 (2019).

[40] D. Janzing, P. Wocjan, and T. Beth, arXiv:0305050 (2003).

[41] L. S. Bishop, S. Bravyi, A. Cross, J. M. Gambetta, and J. Smolin, Quantum Volume. Technical Report (2017).

[42] T. Proctor, M. Revelle, E. Nielsen, K. Rudinger, D. Lobser, P. Maunz, R. Blume-Kohout, and K. Young, arXiv:1907.13608 (2019).

[43] J. Emerson, R. Alicki, and K. Życzkowski, Journal of Optics B: Quantum and Semiclassical Optics **7**, S347 (2005).

[44] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland, Phys. Rev. A **77**, 012307 (2008), URL https://link.aps.org/doi/10.1103/PhysRevA.77.012307.

[45] C. Dankert, R. Cleve, J. Emerson, and E. Livine, Phys. Rev. A **80**, 012304 (2009), URL https://link.aps.org/doi/10.1103/PhysRevA.80.012304.

[46] S. T. Merkel, J. M. Gambetta, J. A. Smolin, S. Poletto, A. D. Córcoles, B. R. Johnson, C. A. Ryan, and M. Steffen, Phys. Rev. A **87**, 062119 (2013), URL https://link.aps.org/doi/10.1103/PhysRevA.87.062119.

[47] R. Blume-Kohout, J. K. Gamble, E. Nielsen, J. Mizrahi, J. D. Sterk, and P. Maunz, arXiv:1310.4492 (2013).

[48] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, et al., Nature **574**, 505 (2019).

[49] V. Havlicek, A. D. Corcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Nature **567**, 209 (2019).

[50] A. Erhard, J. J. Wallman, L. Postler, M. Meth, R. Stricker, E. A. Martinez, P. Schindler, T. Monz, J. Emerson, and R. Blatt, Nature Communications **10**, 5347 (2019).

[51] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins, et al., Nature Communications **10**, 5464 (2019).

[52] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z. X. Gong, and C. Monroe, Nature **551**, 601 (2017).

[53] H. Bernien, S. Schwartz, A. Keesling, H. Levine, A. Omran, H. Pichler, S. Choi, A. S. Zibrov, M. Endres, M. Greiner, et al., Nature **551**, 579 (2017).

[54] H. Levine, A. Keesling, G. Semeghini, A. Omran, T. T. Wang, S. Ebadi, H. Bernien, M. Greiner, V. Vuletić, H. Pichler, et al., Phys. Rev. Lett. **123**, 170503 (2019).

[55] D. Maslov and Y. Nam, New Journal of Physics **20**, 033018 (2018).

[56] S. Aaronson and A. Arkhipov, Theory of Computing **9**, 143 (2013).

[57] C. Monroe, R. J. Schoelkopf, and M. D. Lukin, Scientific American p. 50 (2016).

[58] M. Suzuki, Journal of Mathematical Physics **26**, 601 (1985).

[59] J. Huyghebaert and H. D. Raedt, Journal of Physics A: Mathematical and General **23**, 5777 (1990).

[60] D. W. Berry, G. Ahokas, R. Cleve, and B. C. Sanders, Communications in Mathematical Physics **270**, 359 (2007).

[61] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Proc. Nat. Acad. Sci. **115**, 9456 (2018).

[62] D. W. Berry, A. M. Childs, R. Cleve, R. Kothari, and R. D. Somma, Phys. Rev. Lett. **114**, 090502 (2015).

[63] G. H. Low and I. L. Chuang, Phys. Rev. Lett. **118**, 010501 (2017).

[64] R. Babbush, C. Gidney, D. W. Berry, N. Wiebe, J. McClean, A. Paler, A. Fowler, and H. Neven, Phys. Rev. X **8**, 041015 (2018).

[65] Y. Nam and D. Maslov, Nature Quantum Information **5**, 44 (2019).

[66] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, arXiv:1801.03922 (2018).

[67] M. C. Tran, A. Y. Guo, Y. Su, J. R. Garrison, Z. El-dredge, M. Foss-Feig, A. M. Childs, and A. V. Gorshkov, Phys. Rev. X **9**, 031006 (2019).

[68] S. Aaronson and A. Arkhipov, in *Proceedings of the forty-third annual ACM symposium on Theory of computing* (ACM, 2011), pp. 333–342.

[69] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (American Mathematical Scoiety, Providence, 1999).

[70] R. Oliveira and B. M. Terhal, Quant. Inf. Comp. **8**, 0900 (2008).

[71] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, Phys. Rev. A **92**, 062318 (2015).

[72] Z. Jiang, K. J. Sung, K. Kechedzhi, V. N. Smelyanskiy, and S. Boixo, Phys. Rev. Applied **9**, 044036 (2018).

[73] W. W. Ho and T. H. Hsieh, SciPost Phys. **6**, 29 (2019).

[74] A. Tranter, P. J. Love, F. Mintert, and P. V. Coveney, J. Chem. Theory Comput. **14**, 5617 (2018).

[75] Y. Nam et al., arXiv:1902.10171 (2019).

[76] S. P. Jordan, K. S. M. Lee, and J. Preskill, Quant. Inf. Comput. **14** (2014).

[77] S. P. Jordan, K. S. M. Lee, and J. Preskill, Science **336**, 1130 (2012).

[78] N. Klco and M. J. Savage, Phys. Rev. A **99** (2019).

[79] N. Klco, J. R. Stryker, and M. J. Savage, arXiv:1908.06935 (2019).

[80] E. A. Martinez, C. A. Muschik, P. Schindler, D. Nigg, A. Erhard, M. Heyl, P. Hauke, M. Dalmonte, T. Monz, P. Zoller, et al., Nature **534**, 516 (2016).

[81] N. Klco, E. F. Dumitrescu, A. J. McCaskey, T. D. Morris, R. C. Pooser, M. Sanz, E. Solano, P. Lougovski, and M. J. Savage, Phys. Rev. A **98**, 032331 (2018).

[82] H.-H. Lu et al., Phys. Rev. A **100**, 012320 (2019).

[83] K. Yeter-Aydeniz, E. F. Dumitrescu, A. J. McCaskey, R. S. Bennink, R. C. Pooser, and G. Siopsis, Phys. Rev. **A99**, 032306 (2019).

[84] N. Klco and M. J. Savage, arXiv:1904.10440 (2019).

[85] Z. Davoudi, M. Hafezi, C. Monroe, G. Pagano, A. Seif, and A. Shaw, arXiv:1908.03210 (2019).

[86] C. Hempel, C. Maier, J. Romero, J. McClean, T. Monz, H. Shen, P. Jurcevic, B. P. Lanyon, P. Love, R. Babbush, et al., Phys. Rev. X **8**, 031022 (2018).

[87] M. C. Bauls et al., arXiv:1911.00003 (2019).

[88] P. J. J. O'Malley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, et al., Phys. Rev. X **6**, 031007 (2016).

[89] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature **549**, 242 (2017).

[90] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi, Phys. Rev. X **8**, 011021 (2018).

[91] N. M. Linke, D. Maslov, M. Rötteler, S. Debnath, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, Proc. Nat. Acad. Sci. **114**, 3305 (2017).

[92] C. Figgatt, A. Ostrander, N. M. Linke, K. A. Landsman, D. Zhu, D. Maslov, and C. Monroe, Nature **572**, 368 (2019).

[93] Y. Lu, S. Zhang, K. Zhang, W. Chen, Y. Shen, J. Zhang, J.-N. Zhang, and K. Kim, Nature **572**, 363 (2019).

[94] B. P. Lanyon, C. Hempel, D. Nigg, M. Mller, M. Gerritsma, F. Zhringer, P. Schindler, J. T. Barreiro, M. Rambach, G. Kirchmair, et al., Science **334**, 57 (2011).

[95] N. M. Linke, S. Johri, C. Figgatt, K. A. Landsman, A. Y. Matsuura, and C. Monroe, Phys. Rev. A **98**, 052334 (2018).

[96] E. F. Dumitrescu, A. J. McCaskey, G. Hagen, G. R. Jansen, T. D. Morris, T. Papenbrock, R. C. Pooser, D. J. Dean, and P. Lougovski, Phys. Rev. Lett. **120**, 210501 (2018).

[97] O. Shehab, K. Landsman, Y. Nam, D. Zhu, N. M. Linke, M. Keesan, R. C. Pooser, and C. Monroe, Phys. Rev. A **100**, 062319 (2019).

[98] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman & Co., USA, 1979).

[99] H. Pichler, S.-T. Wang, L. Zhou, S. Choi, and M. D. Lukin, arXiv:1809.04954 (2018).

[100] D. Barredo, S. de Léséluc, V. Lienhard, T. Lahaye, and A. Browaeys, Science **354**, 1021 (2016).

[101] D. Barredo, V. Lienhard, S. De Léséleuc, T. Lahaye, and A. Browaeys, Nature (2018).

[102] A. Kumar, T.-Y. Wu, F. Giraldo, and D. S. Weiss, Nature (2018).

[103] G. Pagano, A. Bapat, P. Becker, K. S. Collins, A. De, P. W. Hess, H. B. Kaplan, A. Kyprianidis, W. L. Tan, C. Baldwin, et al., arXiv:1906.02700v1 (2019).

[104] J. Zhang, G. Pagano, P. W. Hess, A. Kyprianidis, P. Becker, H. Kaplan, A. V. Gorshkov, Z. X. Gong, and C. Monroe, Nature **551**, 601 (2017).

[105] A. Kubica and M. E. Beverland, Phys. Rev. A **91**, 032330 (2015).

[106] A. A. Kovalev and L. P. Pryadko, Phys. Rev. A **88**, 012311 (2013).

[107] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, Applied Physics Reviews **6**, 021318 (2019).

[108] R. K. Naik, N. Leung, S. Chakram, P. Groszkowski, Y. Lu, N. Earnest, D. C. McKay, J. Koch, and D. I. Schuster, Nat. Commun. **8**, 1904 (2017).

[109] Y. Y. Gao, B. J. Lester, K. S. Chou, L. Frunzio, M. H. Devoret, L. Jiang, S. M. Girvin, and R. J. Schoelkopf, Nature **566**, 509 (2019).

[110] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. M. Girvin, L. Jiang, et al., Nature **536**, 441 (2016).

[111] A. Gyenis, P. S. Mundada, A. Di Paolo, T. M. Hazard, X. You, D. I. Schuster, J. Koch, A. Blais, and A. A. Houck, arXiv:1910.07542 (2019).

[112] C. H. Bennett, SIAM Journal on Computing **18**, 766 (1989).

[113] C. Gidney, arXiv:1905.07682 (2006).

[114] M. Eker and J. Hstad, in *Post-Quantum Cryptography*, edited by T. Lange and T. Takagi (Springer International Publishing, Cham, 2017), Lecture Notes in Computer Science.

[115] A. G. Fowler, S. J. Devitt, and L. C. L. Hollenberg, Quantum Info. Comput. **4**, 237 (2004).

[116] S. A. Kutin, in *Asian conference on Quantum Information Science* (2007), pp. 12–13.

[117] J. J. Wallman and J. Emerson, Phys. Rev. A **94**, 052325 (2016).

[118] E. Campbell, Phys. Rev. Lett. **123**, 070503 (2019).

[119] A. Pearson, A. Mishra, I. Hen, and D. A. Lidar, NPJ Quantum Information **5**, 1 (2019).

[120] O. Regev and L. Schiff, in *Automata, Languages and Programming*, edited by L. Aceto, I. Damgrd, L. A. Goldberg, M. M. Halldrsson, A. Inglfsdttir, and I. Walukiewicz (Springer, Berlin, Heidelberg, 2008), Lecture Notes in Computer Science, pp. 773–781.

[121] K. Temme, Phys. Rev. A **90**, 022310 (2014).

[122] D. Rist, M. P. d. Silva, C. A. Ryan, A. W. Cross, A. D. Crcoles, J. A. Smolin, J. M. Gambetta, J. M. Chow, and B. R. Johnson, Nature Quantum Information **3**, 1 (2017).

[123] D. Coppersmith, arXiv:0201067 (2002).

[124] Y. Nam, Y. Su, and D. Maslov, arXiv:1803.04933 (2018).

[125] C. Jones, Phys. Rev. A **87**, 022328 (2013).

[126] D. Maslov, Phys. Rev. A **93**, 022311 (2016).

[127] J. R. McClean et al., arXiv:1710.07629 (2019), URL www.openfermion.org.

[128] Qiskit Aqua: Algorithms for quantum computing applications, IBM (2018), URL https://qiskit.org/aqua.

[129] G. H. Low, N. P. Bauman, C. E. Granade, B. Peng, N. Wiebe, E. J. Bylaska, D. Wecker, S. Krishnamoorthy, M. Roetteler, K. Kowalski, et al., arXiv:1904.01131 (2019).

[130] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Proc. R. Soc. Lond. A **454**, 339 (1998).

[131] S. Kimmel, G. H. Low, and T. J. Yoder, Phys. Rev. A **92**, 062315 (2015).

[132] K. Rudinger, S. Kimmel, D. Lobser, and P. Maunz, Phys. Rev. Lett. **118**, 190502 (2017).

[133] N. Wiebe and C. Granade, Phys. Rev. Lett. **117**, 010503 (2016).

[134] D. R. Simon, SIAM J. Comput. **26**, 1474 (1997).

[135] M. Ettinger, P. Hyer, and E. Knill, Information Processing Letters **91**, 43 (2004).

[136] W. van Dam, S. Hallgren, and L. Ip, SIAM Journal on Computing **36** (2002).

[137] S. E. Venegas-Andraca, Quantum Information Processing **11**, 1015 (2012).

[138] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, Phys. Rev. A **100**, 032328 (2019).

[139] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, Nature Physics **14**, 595 (2018).

[140] T. Brydges, A. Elben, P. Jurcevic, B. Vermersch, C. Maier, B. P. Lanyon, P. Zoller, R. Blatt, and C. F. Roos, Science **364**, 260 (2019).

[141] D. Maslov, Online document, Available: http://www.cs.uvic.ca/~dmaslov (2005).

[142] IEEE P7130 - Standard for Quantum Technologies Definitions, URL https://standards.ieee.org/project/7130.html.