

Challenges and Opportunities of Near-Term Quantum Computing Systems

This article provides a perspective on quantum computing from a leading industry laboratory, for providing access to users not acquainted with the hardware and to accommodate new applications.

BY ANTONIO D. CÓRCOLES, ABHINAV KANDALA, ALI JAVADI-ABHARI, DOUGLAS T. MCCLURE[✉], ANDREW W. CROSS, *Member IEEE*, KRISTAN TEMME, PAUL D. NATION, MATTHIAS STEFFEN[✉], *Senior Member IEEE*, AND JAY M. GAMBETTA[✉], *Senior Member IEEE*

ABSTRACT | The concept of quantum computing has inspired a whole new generation of scientists, including physicists, engineers, and computer scientists, to fundamentally change the landscape of information technology. With experimental demonstrations stretching back more than two decades, the quantum computing community has achieved a major milestone over the past few years: the ability to build systems that are stretching the limits of what can be classically simulated, and which enable cloud-based research for a wide range of scientists, thus increasing the pool of talent exploring early quantum systems. While such noisy near-term quantum computing systems fall far short of the requirements for fault-tolerant systems, they provide unique test beds for exploring the opportunities for quantum applications. Here, we highlight an IBM-specific perspective of the facets associated with these systems, including quantum software, cloud access, benchmarking quantum systems, error correction and mitigation in such systems, understanding the complexity of quantum circuits, and how early quantum applications can run on near-term quantum computers.

KEYWORDS | Quantum computing; quantum systems; superconducting qubits.

NOMENCLATURE

AI	Artificial intelligence.
CNOT	Controlled-NOT.
NISQ	Noisy intermediate-scale quantum.
QCVV	Quantum characterization, verification, and validation.
QEC	Quantum error correction.
QKD	Quantum key distribution.
QV	Quantum volume.
SDK	Software development kit.
SVM	Support vector machine.
UCC	Unitary coupled cluster.
VQE	Variational quantum eigensolver.

I. INTRODUCTION

Quantum computers can potentially solve problems that are considered intractable on even the fastest classical computers [1]–[6]. They use a fundamentally different paradigm for performing calculations and solving problems compared with standard classical computers by using quantum bits (qubits). The speedup is achieved using quantum physics (including entanglement between qubits) to explore correlations in problems such that the correct answer emerges at the end of computation through constructive interference. Considering that quantum states are written as wave functions, classical interference is a reasonable analog: wave functions are steered to the correct answer through constructive interference (several waves add up, creating another wave with a larger amplitude). Wave functions that do not correspond to the

Manuscript received August 30, 2019; revised November 4, 2019; accepted November 7, 2019. Date of publication December 19, 2019; date of current version July 17, 2020. (Corresponding author: Matthias Steffen.)

The authors are with International Business Machines, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: msteffe@us.ibm.com).

Digital Object Identifier 10.1109/JPROC.2019.2954005

correct answer vanish by destructive interference. This is obviously different from our standard picture of computation, wherein each fundamental information unit is definitely in either the state 0 or 1 (we refer to [7]–[9] as examples for a greater in-depth summary of quantum computing).

There is, however, a catch. The internal states of a quantum computer are fragile and susceptible to noise, introducing errors that lead to incorrect answers. Given the complexity and number of operations that are required for many typical quantum algorithms, it is believed that large-scale practical quantum computing has to incorporate at least some form of QEC [1], [10], which is analogous to classical error correction. Several physical qubits are encoded into a logical qubit such that errors on the physical qubits can be detected and corrected. By applying QEC schemes, the logical qubit error rates can be arbitrarily suppressed, provided physical error rates are below a threshold to enable fault-tolerant quantum computing [11]–[14]. In this context, many quantum codes and techniques have been invented [15]. Because simulating the full dynamics of quantum computers quickly becomes intractable as more qubits are added, QEC codes have been studied assuming simplified noise models, such as the Pauli noise. These simulations, together with assumptions about what is experimentally feasible, provide estimates of what would ultimately be required to operate various quantum algorithms using fully fault-tolerant computation [16]–[19]. Millions of qubits with relatively low physical error rates are predicted to be necessary to solve difficult problems. We will not expand further on fully fault-tolerant quantum computing for the remainder of this article but instead refer the reader to [7], [15], and [20].

This article will focus on quantum computing with devices that are currently available or expected to be available in the near future. Various devices comprising 5–79 qubits have been made available to the public or exist as prototypes in laboratories [21]–[25]. Such devices have been referred to as NISQ [26] systems, i.e., nonfault-tolerant devices comprising tens or hundreds of qubits. They can be classified into two categories: 1) devices constructed for a single demonstration experiment run by the team that created the device and 2) devices built to serve as general-purpose quantum systems for use by others. Designing a general-purpose system requires consideration of many factors not relevant for a one-off demonstration. Five such factors, outlined in the following, are covered in detail in Sections II–VI and represent an IBM perspective on critical near-term system aspects.

A system first needs to be designed to accommodate its intended users, enabling the functionality that they require to do their research and providing adaptability as their needs evolve. The breadth of quantum system users (physicists, computer scientists, engineers, chemists, developers, and others) requires multiple cloud systems and access interfaces for interacting with different systems at varying levels of abstraction. Examples of these include

access levels for pulse and gate control and ultimately for applications and systems comprising different connectivities between the qubits.

Second, while a demonstration can rely on special-purpose code, a system needs a complete SDK providing a set of tools that can be used to develop novel experiments and applications. It is to this end that in an open-source collaboration with the community, we have developed Qiskit [27], which consists of four fundamental elements: Terra [28], Aer [29], Ignis [30], and Aqua [31], each bringing a specific set of features to the user. Terra provides the foundation for composing quantum programs at the level of circuits and pulses, optimizing them for the constraints of a particular device, and managing the execution of batches of experiments on remote-access devices. Aer gives access to high-performance quantum simulators to help us understand the limits of classical processors by demonstrating to what extent they can mimic quantum computers. Ignis offers a set of tools to better characterize errors, improve gates, and compute in the presence of noise. Finally, Aqua is where quantum algorithms are built and ultimately used in the context of applications. It provides translators to map problems from domains, such as chemistry, optimization, finance, and AI onto problems solvable with a quantum computer.

Third, it is important to establish a roadmap for the systems. Much like roadmaps for classical systems, a roadmap for quantum systems provides a community-chosen benchmark to facilitate comparisons across systems and demonstrate progress over time. For quantum computers, many individual metrics are commonly accepted as ingredients for a better quantum system, but, as of today, a single community-wide accepted benchmark does not exist. QV has been proposed as a potential benchmark that incorporates many of the individual metrics (number of qubits, connectivity, gate set performance, and compiler and software stack performance) into a single hardware-agnostic metric [32] (see Section IV). We have shown improved QV over the past two years and strive toward continued improvements.

Fourth, to extend the computational reach of short-depth quantum circuits, error mitigation has been proposed as a technique to increase the accuracy of measured observables [33], [34] (a quantum circuit is an ordered list of quantum logic gates, usually expressed as directed graph, and the depth expresses the length of the critical path). Error mitigation is a term used to express methods by which the impact of error can be reduced (or mitigated) without requiring full fault-tolerant quantum codes. This approach to reducing errors in the absence of full fault tolerance is still very much exploratory, but we view it as an important component of near-term quantum systems.

Fifth, quantum computers without fault tolerance will likely be limited to implementing algorithms with short-depth quantum circuits [35]. In these algorithms, a series of quantum gates are applied, and then the qubits are measured. The outcomes of these circuits are used to compute

an observable or sample from a probability distribution of interest. This limited model is believed to be computationally hard for classical machines [36], and recently, it has been shown to have an unconditional separation between classical and quantum computers [35]. Using this model, researchers have explored applications in quantum machine learning [37] and quantum chemistry [38].

From these five aspects, it is evident that developing a complete, user-friendly, cloud-accessible quantum system necessitates considering a rich landscape of design aspects. Successfully implementing all of the ingredients simultaneously to achieve this goal is no small task. This article reviews these considerations in Sections II–VI. For quantum applications, particular emphasis is placed on describing algorithms for quantum machine learning and quantum chemistry because they are examples of applications that can be mapped to short-depth circuits that are believed to be hard for a classical computer to simulate and are currently areas of great interest. As far as hardware is concerned, we only make brief mention of it in Section VII.

II. CLOUD QUANTUM SYSTEMS AND USER ACCESS LEVELS

Although experimental research in quantum computing has been active for over two decades, it was not until the mid-2010s that it became possible to physically connect a handful of superconducting qubits together to implement small multiqubit tests with sufficient fidelity for meaningful results. In 2016, IBM built a quantum processor composed of five superconducting qubits and integrated it into a system called the IBM Quantum Experience [39], available for use via cloud access. Almost immediately after launch, research articles were published based on results obtained through this cloud access to a quantum device. It demonstrated one key aspect that we believe is important for the future as well: there is already demand from physicists, scientists, developers, and many others to access and test various aspects of quantum computing even if the systems are still small, comprise only a few qubits, and suffer from noise levels worse than the fault-tolerance threshold. Over time, other groups in industry and academia have also started to offer quantum cloud services to varying degrees, either in the form of real system access [23], [24], [40] or simulators [41], [42].

Since the initial release of the five-qubit backend, the IBM Quantum Experience has hosted 18 unique quantum systems made available as backend services either to the public or to members of the IBM Q Network [43]. Over time, user executions have increased to 28 million, culminating in over 200 research articles exploring areas in quantum information science. The proliferation and quality of enabled research serve as an affirmation of the community-wide demand for a variety of quantum systems.

The qubit connectivity maps and the distribution of two-qubit error rates for a few of the available IBM backends

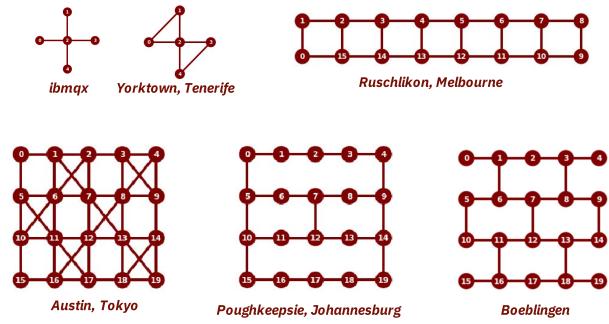


Fig. 1. Examples of several IBM cloud-accessible devices. Dots represent qubits, and lines connect qubit pairs that can be directly entangled. The top left five-qubit device was the first one made available via the IBM Quantum Experience [39], consisting of fixed-frequency superconducting transmon devices. The one to the right of it was made available after including additional entangling gates between two pairs of qubits. A 16-qubit device was made available approximately a year after the first device. The devices in the bottom row show three variations of 20-qubit devices available to members of the IBM Q Network [43].

are shown in Figs. 1 and 2, respectively. The variety of devices allows us to explore user preferences and device performance for different connectivities.

In these present-day systems, imperfections and connectivity can have a large impact on the performance of different algorithms. More connectivity allows users to explore circuits that entangle the qubits in fewer steps but often at the price of hurting gate fidelities or inducing spectator errors [44], [45], i.e., errors that can occur on qubits that are still passively connected but otherwise not directly involved in a particular quantum operation.

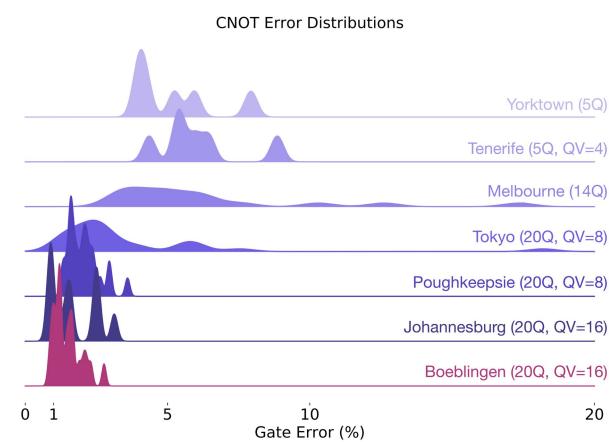


Fig. 2. CNOT gate error distributions for a variety of IBM devices. Beginning with the earlier devices (top rows), the average error rates remained quite large but have improved with continuing research. The bottom row represents the device shown in Fig. 4. The error reductions are the result of improved gate tune-up procedures and increasing coherence times [46], [47], as well as a better understanding of spectator qubit errors.

As we progress through this period of near-term quantum systems, we must evolve toward co-design of the quantum circuits users want to implement, as well as the connectivity that is physically built into the systems.

Furthermore, user feedback showed a clear interest in more than one level of access to a quantum device. We consider three definable fundamental user classes for various levels of cloud access (see Fig. 3): the quantum physicist, the quantum information scientist, and the quantum developer.

The quantum physicist possesses a deep understanding of the underlying device physics and would like to explore more practical technical details, such as optimal control techniques, novel pulse-shaping approaches, techniques to quantify the underlying system Hamiltonian, and error mitigation methods. These users want more of the nitty-gritty details and the ability to examine device-level properties of the system, e.g., control over the frequency, timing, pulse shapes, and measurement integration kernels that are sent to the experiment. To properly meet this level of user access, we have defined the OpenPulse framework [48], along with a corresponding set of tools in Qiskit described later in this review. Succinctly, OpenPulse provides the bare metal access level for users.

The quantum information scientist has a deep understanding of quantum circuits and wants to explore how these circuits run on near-term devices. A circuit can be implemented in multiple ways, in terms of the fundamental gates. Finding the optimal solution is a computationally difficult task and is, therefore, an important research topic. These users are also interested in exploring error-correction primitives, such as parity checks and conditional operations, which depend on these multiqubit measurements to investigate how entropy is taken from the system. For this level, we have defined OpenQASM [49] and the corresponding tools in Qiskit.

The third user class, the quantum developer, wants to see how quantum applications work on quantum computers. They want to run circuits based on an application and receive the outcome as quickly as possible. They are not necessarily interested in how the circuit is implemented; they are focused on the results of the quantum computation and how it can be used in an application of interest.

In parallel, it is also important to provide to each of these users the data appropriate for their respective level. The physicist needs access to device-level specifications, while the quantum information scientist needs the error rates for the calibrated quantum gates and operations. Device specifications are the fundamental properties of the device (e.g., coherence times, qubit frequencies, and crosstalk), while error rates depend on the pulses that represent the gates and include metrics such as average single-qubit errors, two-qubit gate errors, spectator errors, assignment (or readout) errors, and readout crosstalk errors. Cloud-enabled devices typically list most (if not all) of these metrics; see the example in Fig. 4 for the backend properties needed by a user of the quantum infor-

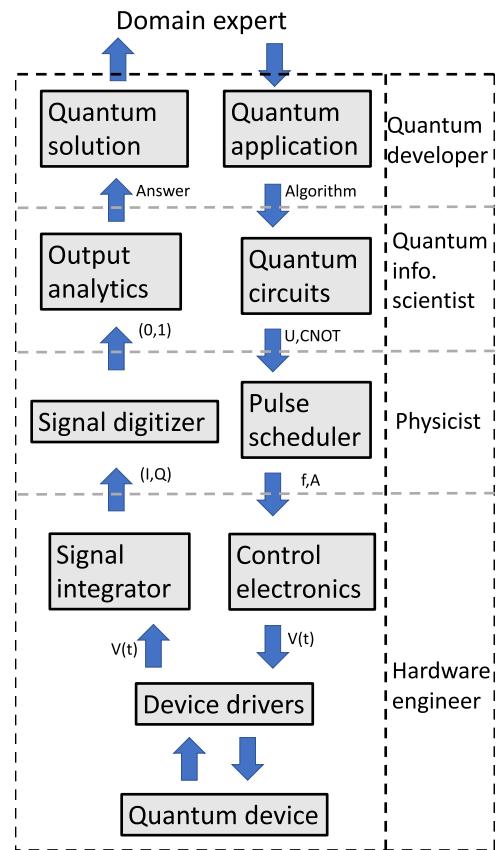


Fig. 3. Schematic overview of a cloud-enabled quantum computer and user access levels. At the high end of the stack, quantum developers create quantum applications executed in the form of algorithms that are translated into quantum circuits at the next level. Quantum information scientists have access to implement specific quantum circuits of interest, which are translated into sequences of quantum operations at the next lower level. Here, we envision that physicists can implement specific pulse-level experiments, such as optimal control or gate-level research. These instructions are sent to the quantum hardware via a set of control electronics in the form of frequencies and drive signal amplitudes. The quantum hardware is accessible by hardware engineers. At the conclusion of an experiment, the hardware passes the readout signal in the form of readout voltages to a signal integrator. The integrator signal is represented as the I and Q quadratures of the readout signal, which get digitized at the physicist access level to a logic 0 or 1. The logical bitstream is passed to output analytics for the quantum information scientists to analyze results of the implemented quantum circuits. Finally, the answer to a full quantum application is sent back to the quantum developer. A domain expert is the expected end user of a fully developed stack. This representative stack demonstrates how different levels of access are possible.

mation scientist persona. By tracking the metrics, we can gauge the importance of any particular metric (or a combination thereof) in order to improve the overall quality of experiments.

Of course, not yet discussed is the hardware engineer who resides at the bottom of the stack. The hardware engineer designs, tests, and interfaces with the quantum

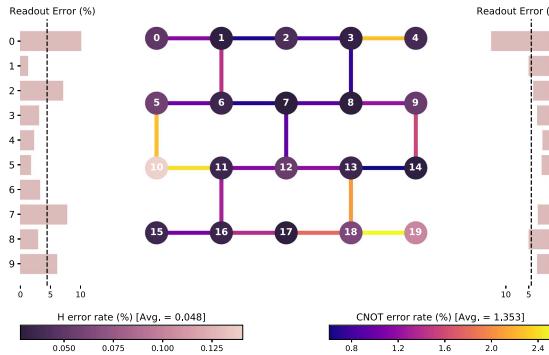


Fig. 4. Error map for the Boeblingen 20-qubit device at IBM. The colors of the qubits represent the measured Hadamard error rate, which is the error rate of a single qubit $\pi/2$ rotation. Z-rotations are implemented in software [50]. For this device, an average Hadamard error of 0.048% is measured. The color of the lines represents the measured two-qubit CNOT error rate between given pairs of qubits. For this device, an average CNOT error of approximately 1.3% is measured. The vertical bars to the left and right of the device plot the readout error rate, with an average readout error of approximately 4.7%.

device. The interface is both physical (installing the device) and virtual (coding and testing control electronics along with associated device drivers). While the hardware engineer is, of course, a critical user class, we envision this role as a more separate functional role to provide the lowest level of required infrastructure for all the users higher in the stack.

III. QISKit AND COMPIRATION

An unprecedented acceleration of research and development in quantum computing has occurred in recent years, chiefly enabled by wide public access to cloud quantum computers. The software stack plays a key role in taking advantage of these systems and enabling quantum information science as a whole [27], [51]–[54]. In this section, we discuss Qiskit, a software suite for near-term quantum computing.

We will pay special attention to the compiler as an indispensable part of any quantum computing system. Our description is focused on compilation strategies tailored to near-term noisy systems. Compiling for fault-tolerant machines is a vast area of research in itself, and we refer the interested reader to [55]–[59] for further reading.

A. Qiskit Architecture

Fig. 5 shows the overall architecture of Qiskit. As an SDK and research tool, Qiskit comprises elements that we deem important in the journey toward quantum advantage. For the sake of completeness, we review them again. The first element, Terra [28], provides the foundations and language to describe quantum computations at different abstraction levels (circuits or pulse schedules) and to compile and optimize them for specific machines. The second

element, Aer [29], provides scalable and realistic simulations of quantum systems and is invaluable to understanding the complexity of different computations, as well as how they behave under certain noise assumptions. The third element, Ignis [30], provides tools to characterize quantum devices and mitigates the effects of noise on them. The fourth element, Aqua [31], is a library of quantum algorithms and translators from near-term application domains (such as chemistry and AI) to quantum circuits.

Since quantum software is so new, much is unknown about how the software stack should be configured for each particular setting. In addition, active research is in progress in all aforementioned areas. For this reason, Qiskit has a highly modular architecture, easily extensible at all levels. This includes adding new circuit optimization passes, new noise models for simulation, new algorithms, and new noise characterization and mitigation methods.

The Qiskit compiler is primarily composed of two parts: the transpiler and the scheduler. The transpiler is a circuit-rewriting toolchain, designed to optimize circuits, both in the abstract and for particular backends. The scheduler converts circuits written for a given device into the sequence of pulses executed on that device.

In the transpiler, multiple circuit analysis and transformation “passes” can be strung together to yield a custom circuit optimization pipeline. A typical sequence might include unrolling the circuit gates to a particular native gate set, allocating ancilla qubits, swapping qubits so that entangling interactions match the device topology, merging consecutive gates into simpler ones, analyzing commutation relations and canceling nonadjacent gates, analyzing the circuit depth, and repeating a couple of optimization passes until the circuit depth reaches a fixed point. A “pass manager” sequences the user’s desired passes, keeps internal context about the progress, and ensures that the control flow of passes is implemented correctly. Similarly, the scheduler contains passes that convert a circuit to a sequence of pulses with specific timing—for example, using an as-soon-as-possible or as-late-as-possible scheduling method—and can optimize them further using methods such as dynamical decoupling [60], [61] or optimal control [62].

B. Compiling for Near-Term Machines

Near-term quantum hardware is severely limited in what it can compute. Errors can build up rapidly during the execution of a program and can render a computation useless. In contrast to classical compilers, for which the goal is to transform a program to run faster, the primary goal of a compiler for near-term quantum computers is to combat these errors. Therefore, a good quantum compiler must ensure that an input program is translated into the most efficient equivalent of itself, squeezing the most out of the available hardware.

Some steps in the compilation process are necessary to run the program in the first place. For example,

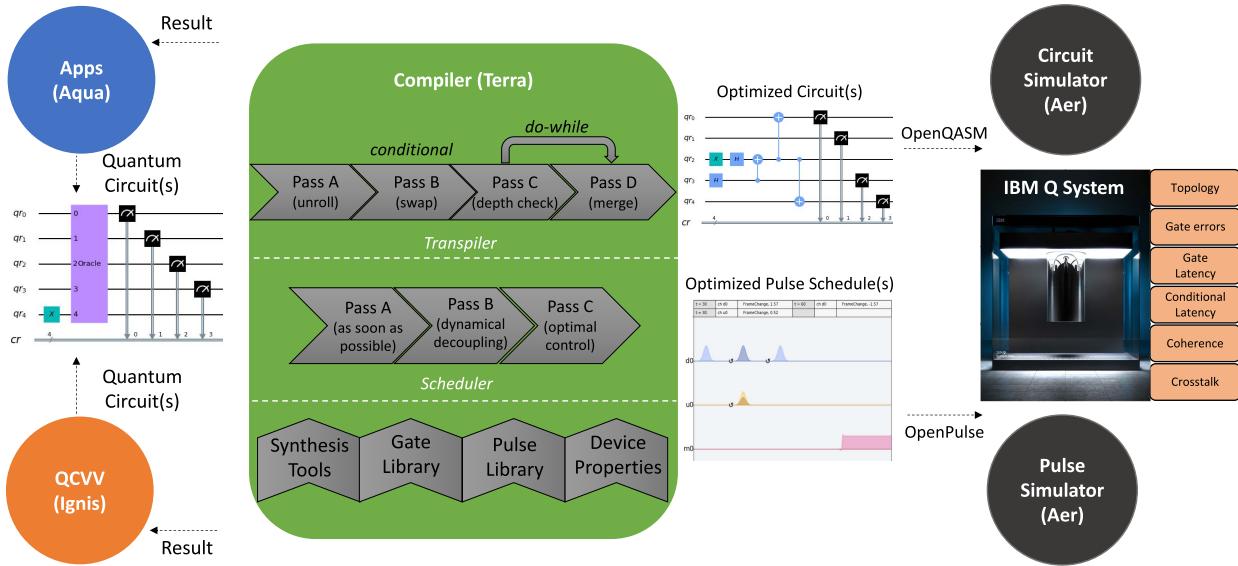


Fig. 5. Architecture of Qiskit. Aqua and Ignis produce circuits for different tasks (algorithms and applications, or device QCVV, respectively). The IBM Q systems and Aer simulators are backends that execute quantum circuits or pulse schedules. The Terra compiler is the bridge that translates and optimizes for a given backend and comprises modular pass-based circuit optimizers (Transpiler) and pulse optimizers (Scheduler). Some example passes are shown. Efficient high-level synthesis methods, access to a library of precomputed gate and pulse equivalents, and information about device constraints and properties all increase compilation quality.

high-level program routines, such as an abstract unitary evolution, must first be synthesized into a quantum circuit [63]–[66]. A circuit must be transformed to conform to the hard constraints of a device, such as which qubits can interact with one another, or which gates are natively supported [67]–[71]. Finally, circuits must be translated into pulses that control the qubits [46], [50].

Beyond this, an optimizing compiler should focus on the soft constraints given by the physics of the device and optimize within that space. For near-term quantum computers, seemingly small optimizations, such as reducing the two-qubit entangling gate (e.g., CNOT) count by 15%, can yield dramatic improvements in the final fidelity of computation. The compilation problem, in general, is NP-hard [67]. Finding optimal layouts of program qubits on the device, or finding optimal swapping routes between the hardware qubits, can be done by solving subgraph isomorphism and token-swapping problems, respectively. We may be able to find optimal solutions for small systems, but soon we need to devise effective heuristics.

An optimizing compiler must generally be aware of the set of constraints and parameters within which it is trying to optimize. To first order, these can be generic truths, such as the fact that two-qubit gates have higher errors than single-qubit gates or that qubits lose their information if the program length (i.e., quantum circuit depth) is too long. Given these constraints, general optimization objectives are defined for a quantum compiler, such as minimizing the circuit depth or the number of entangling gates. This has been the traditional approach to circuit optimization for more than a decade [72]–[75].

While effective, circuit depth and gate count are only pseudoobjectives to simplify reasoning about the quality of a compiler's optimizations. In reality, what matters is the fidelity of computation when running on actual quantum hardware. Every quantum device is different, and thus, benchmarking and characterizing the system are critical to successful compilation. As an example, it is often taken for granted that lower circuit depth is better. This has resulted in trying to parallelize gates as much as possible [76]–[78], which may yield bad results on a high-crosstalk system. Conversely, randomized compiling [79] prolongs circuit depth by inserting extra gates, yet the effect of these gates is to randomize and mitigate coherent errors, achieving a better overall fidelity.

The key takeaway is that compilers for noisy quantum computers excel when more information is made available to them from the device. With cloud-access quantum computers, the field of quantum computer science is moving toward evaluating the effect of compilation strategies on real hardware, rather than based on objectives that may not be comprehensive. In the IBM Q ecosystem, device properties are shared openly and can be benchmarked, resulting in a flurry of recent compiler innovations [80]–[84]. Pertinent hardware characteristics include, but are not limited to, qubit topology, native gate sets, gate error rates, latencies of gates, readouts and feed-forward, qubit lifetimes (decoherence and relaxation), and crosstalk errors.

A key observation in compiling for noisy quantum computers is that since errors always exist, it may not always be worth performing a numerically exact compilation.

Alternatively, approximate compilation aims to approximate a unitary by some numerically close alternative, in order to potentially save significant resources [32], [85]. If the reduction in error due to the shorter alternative is more than the loss of precision in the approximation, then this tradeoff is worthwhile. Evaluating this tradeoff is again dependent on the exact characteristics of the device.

Finally, verification of the compiler becomes a serious challenge even in the near term, as verification of general circuit transformation on circuits of roughly 50 or more qubits is impractical. Consequently, expansive testing of smaller cases or formal verification methods [86], [87] will be essential.

Quantum compilers have benefited from decades of classical compiler design, yet the new domain creates new challenges and opportunities. For example, commutation relationships among quantum gates provide additional flexibility, compared to a classical program for instructions to be reordered, merged, or canceled [76]–[78], [88]. In contrast to the classical computing, in which a program can be compiled once and reused thereafter, quantum programs must often be recompiled, as device properties change over time. Finally, traditional abstraction boundaries for separating logical computation from physical implementation may need to be blurred on near-term devices, as they may sacrifice some efficiency in favor of abstraction [89]. Designing an industrial-scale compiler suitable for the coming generation of quantum computers remains an exciting and hard task.

IV. BENCHMARKING NEAR-TERM DEVICES

Benchmarking quantum systems will be a necessity to measure progress. Assuming several physical realizations of quantum computers will emerge over time, there must be a way to quantify their respective performance much like classical benchmarks. While benchmarking appears to be an obvious requirement, it is far less obvious how to devise a rigorous set of metrics applicable to quantum computers.

Important factors for formulating an appropriate quantum benchmark include the following.

- 1) *Number of Qubits*: More qubits are required to solve increasingly difficult problems, and thus, everything else being equal, the more qubits a quantum computer has, the more computational power it has. Systems with several tens of qubits can be simulated on a classical computer, and therefore, having a few qubits will not be beneficial in the long run.
- 2) *Connectivity*: How qubits are connected to one another matters. At one extreme, qubits connected on a line would require a significant overhead for any randomly selected gate between any random pair of qubits. At the other extreme, if all qubits are connected to each other, there is no additional overhead for a randomly selected gate between any random qubit pair. However, at the hardware level,

connectivity matters a lot, and it greatly influences metrics such as crosstalk and fidelity. It is important to strike a balance between connectivity and overhead for a given application.

- 3) *Error Rates*: Quantum operations that feature lower errors rates are generally better. A critical component associated with error rates is spectator errors (errors on qubits that are not participating in the applied quantum gate). The spectator errors can significantly degrade or even dominate overall circuit performance. For example, while a two-qubit gate might lead to very low errors on the two qubits involved, it is possible that another qubit might undergo significant errors due to the application of the two-qubit gate.
- 4) *Gate Set*: The choice and performance of the underlying gate set are important. A large set of gates reduces the overhead to synthesize arbitrary gates or move quantum information but also requires far more complexity from a calibration and stability standpoint.
- 5) *Compilers and Software Stack Performance*: Compilers are critical for optimal translation of circuits to the underlying hardware. A compiler needs to consider and optimize over the device connectivity and potentially even variations of gate fidelities and spectator errors across the device.

IBM has devised a benchmark called QV that balances all of the abovementioned ingredients [32]. We believe that this system-agnostic metric provides a way to compare devices across different physical implementations and incorporates qualities, such as low error rates, that are ultimately necessary for a practical quantum computer.

The QV measures the largest model circuits the quantum computer can successfully run. A model circuit consists of random two-qubit gates acting on random pairs of qubits and has as many parallelized layers of these gates as it has qubits. The model circuits are compiled to the particular quantum system. A given run is considered successful if the observed measurement outcome is in the upper half of the ideal output probability distribution. To claim that the QV exceeds some value for some system, we are required to succeed for more than two-thirds of the runs on a given number of qubits.

We wish to particularly highlight the role of circuit compilation in the QV because a full quantum system is the combination of the qubits, gates, control electronics, and the software stack that optimizes for those components. The QV provides a way to benchmark the whole quantum computing system, including the optimizing components of the software stack.

Choices for universal benchmarks will evolve as the community continues to learn more about near-term devices [90]–[92], but the QV is an accessible and measurable quantity that tracks progress on current devices. We have released an open-source library for measuring QV in Qiskit. The largest QV measured thus far is 16, measured first on the 20-qubit Johannesburg system [32], and more recently on Boeblingen (see Fig. 4).

V. ERROR MITIGATION AND CORRECTION

For decades, researchers have understood that decoherence would limit the duration of useful quantum computation [93] and have devised many techniques for overcoming the noise. Today, we are keenly aware of the impact of decoherence and control error on the size and accuracy of quantum computations. The way forward is necessarily a mixture of approaches. Foremost, we must understand and reduce the fundamental error sources in our hardware, control systems, and environment. Beyond this, we must correct the remaining errors and/or mitigate their effect on the quantum computer's accuracy in a resource-efficient way. These are among the central research challenges for the foreseeable future.

It is by now well known that the principles of QEC allow errors to be dramatically and efficiently suppressed in theory [1], [10]–[14] so that the computational time can extend well beyond the coherence time. This requires physical error rates to be low enough and noise to be sufficiently uncorrelated. When that happens, fault-tolerant gates can successfully limit the spread of errors, and QEC procedures can remove entropy faster than it accumulates. If error rates are only modestly below the threshold error rate, the additional space and time to implement fault-tolerant gates can be prohibitively large. Furthermore, topological codes [94], which are among the most well suited to planar quantum computing architectures, are expected to correct errors very well but protect qubits by encoding each of them into a large number of physical qubits.

In the near term, we are unlikely to have both sufficiently low error rates and sufficiently many qubits to implement a fault-tolerant quantum computer. Nevertheless, these near-term systems present an early opportunity to research error mitigation and error correction in real noise environments. On one hand, QEC experiments spur development, confirm predictions, and expose facts about detecting realistic errors. On the other hand, error mitigation experiments have low overhead and significantly improve computational results today, so they are eminently practical. Error mitigation can improve estimates of expectation values, which can be important in explorations of quantum advantage, for example, as eigenvalues of the molecular Hamiltonians or kernels in classification problems addressed by quantum machine learning algorithms. However, unlike QEC that removes entropy, error mitigation cannot extend the computation far beyond the coherence time.

We now focus our attention on error mitigation schemes that are more recent and less well known than error correction schemes. To date, there have been two general-purpose error mitigation schemes developed. The first, zero-noise extrapolation, was developed independently in the works of [33] and [34], and the second, probabilistic error cancellation, was introduced in [33]. In zero-noise extrapolation, the output from a circuit of interest

is remeasured under different amplified noise strengths. The measured expectation values from these noisy runs can then be recombined to extrapolate to an estimate of the expectation value at the zero-noise limit that is more accurate than the best individual run. Using measurements at an increasing number of noise strengths, increasingly higher order noise contributions to the zero-noise estimate are suppressed by extrapolation [95]. Temme *et al.* [33] showed that such noise amplification could be achieved by stretching the time evolution of the quantum state, under the influence of the time-dependent drives that constitute the quantum circuit. Under the assumption of time-invariant noise, the stretch factor for the time evolution is equivalent to the noise amplification factor. Beyond this assumption, no further characterizations of the noise models are required, making this extremely attractive for experimental implementations. This method was demonstrated and integrated into a variational algorithm in the experiment of [38], using superconducting qubits and all-microwave gates. It was also employed to improve the performance of a binary classifier realized on the same device [37].

A second, general-purpose error mitigation scheme, also proposed in [33], is termed probabilistic error cancellation or quasi-probability decomposition. In this method, every well-characterized noise channel in a quantum circuit is acted upon by its inverse. While implementing the inverse noise channel is in itself an unphysical task, it was shown that an “average” error-mitigated estimate of the outcome can instead be obtained by sampling from an ensemble of noisy circuits with probabilities related to the coefficients of the inverse noise map. The variance of the error-mitigated estimate is related to the number of noisy circuits sampled and measured. In contrast to the zero-noise extrapolation technique, a key experimental challenge here lies in the characterization of noisy gates employed in the quantum circuit. For up to two-qubit experiments, this method was recently realized for superconducting qubit [96] and trapped-ion architectures [97], both employing gate set tomography for noise characterization.

In addition to these techniques, other methods have been proposed that are more problem-specific. The quantum subspace expansion method [98], [99] involves the measurement of additional excitation operators for variational ground states and, in addition to providing excited state energies, also mitigates on energy estimates. Other recent approaches to error mitigation for fermionic problems rely on the conservation of “known quantities,” such as particle number [100]–[102]. Such symmetries can be enforced by using ancillary qubits to perform stabilizer checks.

Error mitigation is still in its infancy but has shown some promising first steps. As we look forward, we hope that access to near-term systems will enable new error mitigation and correction techniques at the intersection of theory and practice. Ultimately, we believe that QEC

and fault-tolerant design will still be necessary. Therefore, continued experiments such as demonstrations of the Bell state parity measurements [103], stabilizer measurements [45], error detecting codes [104], [105], and other codes [106] are critical for understanding how to protect encoded quantum information in the long term [107]. We anticipate the theory and practice of error correction and mitigation to continue to develop together in the future and that new ideas will emerge, particularly with respect to many well-known types of errors, such as correlated errors, leakage, and fluctuations of these errors.

VI. QUANTUM APPLICATIONS ON NEAR-TERM QUANTUM SYSTEMS

The relevance of a quantum computer is derived from the algorithms that can be performed on it. For some problems, such as factoring integers [2] or simulating quantum mechanics [108], quantum algorithms have theoretical guarantees to drastically outperform any known classical algorithm. It is important to state that not every problem that is challenging for a classical computer will benefit from a quantum speedup. This means that the applications for a quantum computer need to be identified individually, and a specific quantum algorithm must be developed for them. Up to this point, the set of algorithms that can be shown to outperform classical computers [109] all depend on an architecture that is fully fault tolerant. The quantum hardware that is currently available is not yet at a stage to run fault-tolerant computations. Nevertheless, making current hardware available to the research community allows for the investigation of quantum algorithms that have the potential to run on near-term quantum devices. Due to device imperfections and decoherence, we expect that such algorithms will be comprised of shallow-depth quantum circuits. To tackle a complex computational task, some of the computation that does not benefit from a quantum speedup can be outsourced to a classical computer. Examples for such quantum-classical hybrid schemes are variational algorithms for quantum many-body systems [38], [110], [111] and machine learning [37]. Such shallow-depth variational hybrid algorithms can be understood from the following picture: the classical computer tries to find the best quantum circuit, limited in size to a depth determined by the noise, to perform a particular computational task. The task could be the preparation of an approximation to the ground state of a Hamiltonian or the construction of a classifier in machine learning. This simple scheme has opened up a pathway to trying heuristic algorithms that do not come with any performance guarantees on current quantum hardware.

The development of classical algorithms has greatly profited from the wide availability of computational hardware. Many heuristic algorithms were found by trial and error and come without performance guarantees. It is, therefore, reasonable to follow an experimental route in the search for applications that could benefit from a quantum computer. However, there is an important

difference between the development of classical algorithms and quantum algorithms. Not all quantum circuits can lead to quantum advantage. If the quantum algorithm can be efficiently simulated on classical hardware [112]–[118], it cannot provide a computational advantage. The advantage of quantum computers is based on the complexity of the algorithm and not on the quantum computer's ability to perform fast operations. It is, therefore, paramount to ensure that the quantum algorithm is based on a circuit that cannot be efficiently simulated on a classical computer.

To ensure the classical hardness of simulation is of particular importance when performing algorithms on a small number of qubits that are subject to noise since a quantum advantage may not be immediately apparent. The first fundamental question that arises is whether, and under which circumstances, a shallow-depth quantum circuit can provide a computational advantage. This question was recently addressed and answered [35] by demonstrating an unconditional separation in computational power between shallow quantum and classical circuits. An unconditional separation means that the complexity of the classical problem is understood and the separation to quantum can be proven. Further results [119]–[121] that are based on computational complexity assumptions show the existence of elementary quantum circuits that are likely difficult to simulate on a classical computer. While these results are encouraging, we need to continue researching quantum circuit complexity to point toward meaningful quantum applications that offer a speedup over classical approaches.

A more systematic path toward quantum applications for near-term quantum devices that exhibit a reliable advantage is based on the complexity-theoretic hardness of quantum circuits. In this approach, it is the quantum circuit that determines the application, placing the formal complexity result at the beginning of the development.

A. Quantum Machine Learning

One example of a quantum-classical hybrid algorithm that relies on quantum circuits believed to scale inefficiently for classical methods has been presented in [37]. In this work, the authors describe and implement two methods of binary classification using supervised training. These classification algorithms are related to standard SVMs. The idea in this work is to implement a nonlinear feature map that brings the data to classify into a space in which it can be linearly separated. The key aspect exploited by a quantum processor is that the feature map is implemented as a quantum circuit, mapping the initial data to the high-dimensional quantum state space, so it can be separated by linear binary classifier data. The use of a quantum feature map has also been proposed in [122]. For this algorithm to provide a quantum advantage, the quantum circuit must have transition amplitudes that cannot be estimated classically to an additive sampling

error. The feature map circuit used in [37] can be related to a hardness result derived in [123], which guarantees an exponential separation in query complexity to the best classical algorithm.

Havlíček *et al.* [37] explore two methods to construct a binary classifier based on the hard feature map circuit. In the first method, the feature map circuit is directly followed by a variational circuit. The circuit can be used as a classifier that implements a binary measurement on the quantum feature space. This variational algorithm is, therefore, directly related to a classical SVM. The second method directly exploits the connection to classical SVMs by estimating the kernel matrix directly on the quantum computer and then using a conventional SVM. The hardware implementation of these two methods showed that even on a modest quantum processor, some sort of error mitigation [38] was needed. We have discussed a few error mitigation proposals in Section V.

A key observation of this proposal is the existence of quantum circuits that give rise to feature maps that are hard to evaluate classically, relative to complexity-theoretic assumptions. However, to obtain a quantum advantage for a practically relevant machine learning problem, a hard feature map circuit is a necessary condition. To make this sufficient, more circuits need to be explored that can be tied to complex real-world classification problems.

B. Quantum Chemistry

A second example uses quantum-classical hybrid algorithms with short-depth quantum circuits for quantum chemistry. The VQE [110] has been implemented on a number of different quantum hardware platforms [38], [110], [111], [124]–[127]. Here, the central objective is to obtain a good estimate of the ground state energy for chemistry or the general many-body Hamiltonian. Although typically fermionic Hamiltonians are considered, these Hamiltonians can be readily mapped to qubit/spin degrees of freedom by a common procedure [128]. To obtain ground state energy estimates for a target Hamiltonian, variational trial states are prepared on a quantum computer by using shallow circuits with free parameters that are experimentally adjustable. The quantum computer is used to estimate the mean energy of the target Hamiltonian by measuring the operators of the Hamiltonian directly on the quantum computer with respect to the trial state. The energy associated with each trial state is then fed to a classical computer, which runs an optimization routine that supplies a new set of parameters. The goal of the optimization procedure is to prepare a new trial state that will tend to lower the energy. This process is then iterated until some convergence condition is met.

In this approach to near-term quantum algorithms, the quantum computer’s utility lies in the preparation of trial states and the measurement of associated expectation values. Depending on the trial state, these tasks may be hard on a classical computer, e.g., [130] and [131]. Preparing and measuring trial states on a quantum computer can

provide additive error approximations to expectation values. However, as highlighted previously, the algorithm can only yield a quantum advantage if the circuits employed for trial state preparation are difficult to simulate classically. Most VQE implementations to date have focused on small (<10 qubit) molecular Hamiltonians in quantum chemistry. These implementations have employed circuits that implement “hardware-efficient” trial states [38], [111] or a UCC ansatz [131]. While the UCC approach offers a structured ansatz that maintains physical symmetries, the hardware-efficient circuits employ interactions that are native to the quantum hardware. Other important considerations for the practical implementation of VQE for quantum chemistry are qubit-efficient fermionic mapping schemes [128], the robustness of classical optimizers to hardware noise, and, most importantly, the effect of decoherence [33], [34], [38] and the measurement cost for the molecular Hamiltonians [132], [133].

One specific example is the combination of the previously mentioned error mitigation techniques and quantum chemistry. For VQE, the quantum subspace expansion technique also provides access to error-mitigated energy estimates as well as to the excited state energies [98], [99]. As discussed previously, more general-purpose schemes, such as a zero-noise extrapolation technique and quasi-probability method, have been proposed to access noise-free estimates of expectation values. Even without significant improvements to coherence times and gate fidelities or any additional qubit overhead, the experimental implementation of the zero-noise extrapolation [38] showed an otherwise inaccessible level of accuracy in the ground state estimates of H₂ and LiH, as shown in Fig. 6.

VII. HARDWARE

To this point, not many hardware details have been discussed. Hardware is of course required to have a system at all, but much of the system-level work can be done in a manner that is somewhat hardware-agnostic, the further away the user operates in the stack, and emphasizes the need for some modularity of the associated system components. This is particularly important as the underlying hardware may change over time, potentially significantly.

A detailed discussion of specific challenges associated with our quantum hardware of choice (superconducting qubits) is beyond the scope of this article. However, we briefly describe the devices that are publicly accessible or were used in the described experiments (e.g., the quantum chemistry [38], [111] and machine-learning experiments [37]). These devices are single-junction transmon [134] qubits coupled to each other according to the specified connectivity. Single-qubit gates are implemented using microwave pulses and benchmarked using the standard randomized benchmarking protocols (see, for example, [135]). Two-qubit gates are implemented using the cross-resonance gate [136], [137]. Qubits are measured using dispersive readout techniques [138]. For a

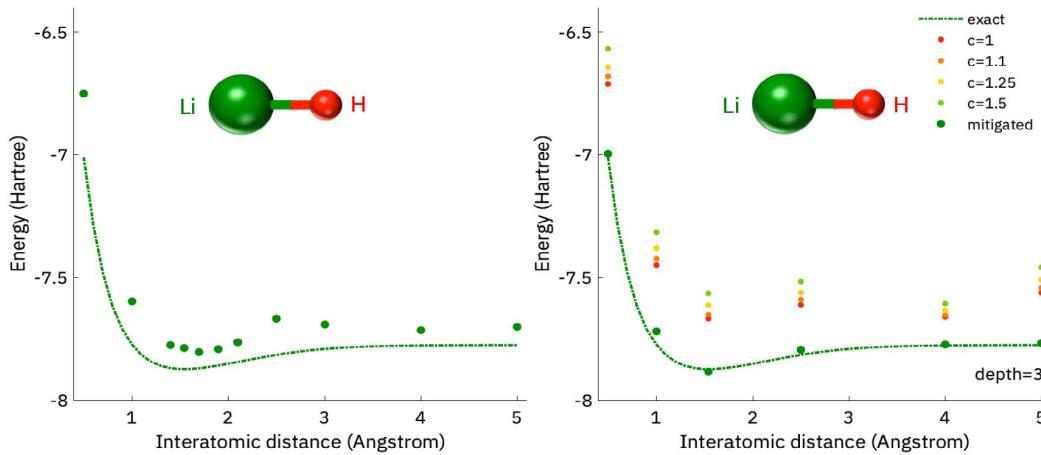


Fig. 6. Molecular simulations of LiH using four qubits from [38] and [112]. The experimental data (black circles) are compared with the exact energy curve (green dotted line). Left: results from [111]. Right: data from [38], which uses error mitigation. The error-mitigated estimates (black circles) in right are obtained by extrapolating the results from experiments of varying noise (colored circles) and display far superior accuracies, without significant hardware improvements to the processors used for these computations. Figures modified from [38] and [112].

broad review of superconducting circuits, we refer the reader to [7], [8], [140], and [141] for in-depth reviews of superconducting qubits that provide extensive information about types of superconducting qubits, single- and two-qubit operations, and readout, as well as some examples of anticipated engineering challenges.

Moving forward, the engineering challenges for any quantum computer are enormous, not just for superconducting circuits. Depending on the technology, various hurdles must be overcome to implement increasing numbers of qubits. Within superconducting qubits, it is widely acknowledged that microwave engineering plays a crucial role (e.g., [141]). As chip sizes increase beyond the wavelength of interest (qubit frequencies are typically in the GHz regime), it is critical to account for parasitic microwave modes [142], microwave crosstalk, and microwave filtering [143] in order to ensure that the qubits are not unintentionally coupled to spurious modes or each other, to ensure the qubit coherence times are not impacted, and to allow for fast readout [138] all while still retaining precise microwave control (or flux biasing in the case for flux-biased operation). In addition, advanced packaging is generally considered a requirement to allow vertical signal delivery to the qubits. Therefore, there is much interest in fabrication integration, including superconducting bump bonds, thru-silicon vias, and chip stacking. A recent comprehensive article describes 3-D integration for superconducting qubits, including many of the aforementioned technologies [144]. Finally, moving beyond the qubit chip, the infrastructure to support large qubit systems needs to be developed [145]. This includes transitioning away from standard microwave coaxial lines in favor of flex coax lines [146] and low-cost solutions for microwave (or flux bias) control either at room temperature or at cryogenic temperatures [147], [148].

Finally, overall system reliability also plays a crucial role. We stress the importance of engineering systems capable of operating over prolonged periods of time. Rudimentary demonstrations are notoriously unstable and often barely capable of gathering sufficient data for a scientific publication; they are essentially physics experiments. Both external influences and internal device noise can cause parameters to quickly drift on experimental timescales, potentially rendering a device unsuitable for use in a quantum system due to the prohibitive amount of device calibrations required.

Needless to say, tackling all of the engineering challenges is a substantial undertaking and should not be underestimated. We anticipate that substantial research needs to be directed toward solving these issues while, at the same time, fostering essential basic research contributions from the broader community. We believe fundamental contributions will be critical for the success of long-term quantum computing.

VIII. CONCLUSION

This article described the various challenges and opportunities associated with near-term quantum systems, highlighting the necessary components to bring practical quantum computers closer to reality. A unique interplay between hardware, hardware access, software, benchmarking, applications, and error mitigation techniques is required to develop a quantum system capable of one day executing practical calculations or simulations.

We have also laid out our software approach, which is heavily user-oriented. We feel strongly that a healthy user base will be a guiding force, helping to shape the technical direction for future quantum devices. We have presented the available toolset (i.e., various access levels,

SDKs, Qiskit, and so on), and we observe appreciable demand for such tools. Integrating those tools with stable hardware is a significant effort, but we feel it is worth the challenge, as we are exploring and developing systems that have not yet been built!

It is duly acknowledged, however, that the full potential of near-term quantum systems is presently unknown. While no fundamental roadblocks have yet materialized, it is possible that the application range lags behind some of the more enthusiastic expectations. We are optimistic that the near-term quantum systems will be capable of at least shedding new light on unexplored physics lying just out of reach for modern simulation tools or other derivative applications related to, for example, quantum sensing. There are many areas to study and explore, and by giving the right access and tools to researchers, we hope to accelerate the pace of discovery. ■

REFERENCES

- [1] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*. New York, NY, USA: Cambridge Univ. Press, 2000.
- [2] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.
- [3] L. K. Grover, “Quantum mechanics helps in searching for a needle in a haystack,” *Phys. Rev. Lett.*, vol. 79, no. 2, p. 325, Jul. 1997.
- [4] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, 1997.
- [5] M. H. Yung, J. D. Whitfield, S. Boixo, D. G. Tempel, and A. Aspuru-Guzik, “Introduction to quantum algorithms for physics and chemistry,” *Adv. Chem. Phys.*, vol. 154, pp. 67–106, Feb. 2014.
- [6] S. Jordan, *Quantum Algorithm Zoo*. [Online]. Available: <https://math.nist.gov/quantum/zoo/>
- [7] J. Gambetta, J. M. Chow, and M. Steffen, “Building logical qubits in a superconducting quantum computing system,” *NPJ Quantum Inf.*, vol. 3, no. 2, pp. 1–7, 2017.
- [8] M. Devoret and R. J. Schoelkopf, “Superconducting circuits for quantum information: An outlook,” *Science*, vol. 339, pp. 1169–1174, Mar. 2013.
- [9] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O’Brien, “Quantum computers,” *Nature*, vol. 464, no. 7285, pp. 45–53, Mar. 2010.
- [10] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Phys. Rev. A, Gen. Phys.*, vol. 52, no. 4, p. R2493, 1995.
- [11] D. Aharonov and M. Ben-Or, “Fault-tolerant quantum computation with constant error rate,” in *Proc. 29th Annu. ACM Symp. Theory Comput. (STOC)*, 1997, pp. 176–188.
- [12] P. W. Shor, “Fault-tolerant quantum computation,” in *Proc. 37th Symp. Found. Comput.*, Oct. 1996, pp. 56–65.
- [13] J. Preskill, “Fault-tolerant quantum computation,” 1997, *arXiv:quant-ph/9712048v1*. [Online]. Available: <https://arxiv.org/abs/quant-ph/9712048v1>
- [14] A. Y. Kitaev, “Fault-tolerant quantum computation by anyons,” *Ann. Phys.*, vol. 303, no. 1, pp. 2–30, Jan. 2003.
- [15] E. T. Campbell, B. M. Terhal, and C. Vuillot, “Roads towards fault-tolerant universal quantum computation,” *Nature*, vol. 549, pp. 172–179, Sep. 2017.
- [16] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, “Surface codes: Towards practical large-scale quantum computation,” *Phys. Rev. A, Gen. Phys.*, vol. 86, Sep. 2012, Art. no. 032324.
- [17] J. O’Gorman and E. T. Campbell, “Quantum computation with realistic magic-state factories,” *Phys. Rev. A, Gen. Phys.*, vol. 95, Mar. 2017, Art. no. 032338.
- [18] D. Litinski, “A game of surface codes: Large-scale quantum computing with lattice surgery,” *Quantum*, vol. 3, p. 128, Mar. 2019.
- [19] C. Gidney and M. Ekerå, “How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits,” 2019, *arXiv:1905.09749*. [Online]. Available: <https://arxiv.org/abs/1905.09749>
- [20] B. M. Terhal, “Quantum error correction for quantum memories,” *Rev. Mod. Phys.*, vol. 87, no. 2, p. 307, 2015.
- [21] [Online]. Available: <https://www.research.ibm.com/ibm-q/technology/devices/>
- [22] [Online]. Available: <https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html>
- [23] [Online]. Available: <https://www.rigetti.com>
- [24] [Online]. Available: <https://ionq.com/>
- [25] F. Arute *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, 2019, doi: [10.1038/s41586-019-1666-5](https://doi.org/10.1038/s41586-019-1666-5).
- [26] J. Preskill, “Quantum computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug. 2018.
- [27] G. Aleksandrowicz *et al.*, “Qiskit: An open-source framework for quantum computing,” *Tech. Rep.*, 2019, doi: [10.5281/zenodo.2562111](https://doi.org/10.5281/zenodo.2562111).
- [28] IBM. (2017). *Qiskit Terra*. Accessed: Aug. 1, 2019. [Online]. Available: <https://github.com/Qiskit/qiskit-terra>
- [29] IBM. (2019). *Qiskit Aer*. Accessed: Aug. 1, 2019. [Online]. Available: <https://github.com/Qiskit/qiskit-aer>
- [30] IBM. (2019). *Qiskit Ignis*. Accessed: Aug. 1, 2019. [Online]. Available: <https://github.com/Qiskit/qiskit-ignis>
- [31] IBM. (2019). *Qiskit Aqua*. Accessed: Aug. 1, 2019. [Online]. Available: <https://github.com/Qiskit/qiskit-aqua>
- [32] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta, “Validating quantum computers using randomized model circuits,” *Phys. Rev. A, Gen. Phys.*, vol. 100, Sep. 2019, Art. no. 032328.
- [33] K. Temme, S. Bravyi, and J. M. Gambetta, “Error mitigation for short-depth quantum circuits,” *Phys. Rev. Lett.*, vol. 119, Nov. 2017, Art. no. 180509.
- [34] Y. Li and S. C. Benjamin, “Efficient variational quantum simulator incorporating active error minimization,” *Phys. Rev. X*, vol. 7, no. 2, 2017, Art. no. 021050.
- [35] S. Bravyi, D. Gosset, and R. Koenig, “Quantum advantage with shallow circuits,” *Science*, vol. 362, no. 6412, pp. 308–311, 2018.
- [36] B. M. Terhal and D. P DiVincenzo, “Adaptive quantum computation, constant depth quantum circuits and arthur-merlin games,” *Quantum Inf. Comput.*, vol. 4, pp. 134–145, Jun. 2004.
- [37] V. Havlíček *et al.*, “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, pp. 209–212, Mar. 2019.
- [38] A. Kandala, K. Temme, A. D. Corcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, “Extending the computational reach of a noisy superconducting quantum processor,” *Nature*, vol. 567, p. 491, May 2019.
- [39] [Online]. Available: <https://www-03.ibm.com/press/us/en/pressrelease/49661.wss>
- [40] [Online]. Available: <https://www.quantumcomputer.ac.cn>
- [41] [Online]. Available: <https://www.microsoft.com/en-us/quantum/development-kit>
- [42] [Online]. Available: <https://qutech.nl/quantum-inspire/>
- [43] [Online]. Available: <https://www.research.ibm.com/ibm-q/network/overview/>
- [44] D. C. McKay, S. Sheldon, J. A. Smolin, J. M. Chow, and J. M. Gambetta, “Three-qubit randomized benchmarking,” *Phys. Rev. Lett.*, vol. 122, May 2019, Art. no. 200502.
- [45] M. Takita *et al.*, “Demonstration of weight-four parity measurements in the surface code architecture,” *Phys. Rev. Lett.*, vol. 117, no. 21, 2016, Art. no. 210505.
- [46] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, “Procedure for systematically tuning up cross-talk in the cross-resonance gate,” *Phys. Rev. A, Gen. Phys.*, vol. 93, no. 6, 2016, Art. no. 060302.
- [47] J. M. Gambetta *et al.*, “Investigating surface loss effects in superconducting transmon qubits,” *IEEE Trans. Appl. Supercond.*, vol. 27, no. 1, Jan. 2017, Art. no. 1700205.
- [48] D. C. McKay *et al.*, “Qiskit backend specifications for openqasm and openpulse experiments,” 2018, *arXiv:1809.03452*. [Online]. Available: <https://arxiv.org/abs/1809.03452>
- [49] A. W. Cross, L. S. Bishop, J. A. Smolin, and J. M. Gambetta, “Open quantum assembly language,” 2017, *arXiv:1707.03429*. [Online]. Available: <https://arxiv.org/abs/1707.03429>
- [50] D. C. McKay, C. J. Wood, S. Sheldon, J. M. Chow, and J. M. Gambetta, “Efficient Z gates for quantum computing,” *Phys. Rev. A, Gen. Phys.*, vol. 96, Aug. 2017, Art. no. 022330.
- [51] Rigetti. (2018). *PyQuil*. Accessed: Aug. 1, 2018. [Online]. Available: <https://github.com/rigetticomputing/pyquil>
- [52] Google. (2018). *Cirq*. Accessed: Nov. 29, 2018. [Online]. Available: <https://github.com/google/cirq>

On a broader scale, research in the quantum realm continues to capture the imaginations of many researchers. Active research areas not discussed in this article include fault-tolerant quantum computing [7], [15], [20] (e.g., both theory and experiment for small and large demonstrations of fault-tolerant operations), detailed superconducting qubit hardware considerations [7], [8], [139], [140] (e.g., implementations of qubit gates, coherence times, qubit packaging, and qubit measurement) or other quantum hardware approaches (references within, for example, [7]), quantum information [1], [149] (including its relationship with complexity theory [150]), quantum communication [151], [152] (e.g., QKD), quantum sensing [153], and postquantum cryptography [154]. Taken together, we feel confident that quantum science as a whole will shape our future technology one day, likely in ways we cannot currently foresee. ■

- quantumlib/Cirq
- [53] D. S. Steiger, T. Häner, and M. Troyer, "ProjectQ: An open source software framework for quantum computing," *Quantum*, vol. 2, p. 49, Jan. 2018, doi: [10.22331/q-2018-01-31-49](https://doi.org/10.22331/q-2018-01-31-49).
- [54] A. Javadi-Abhari *et al.*, "ScaffCC: A framework for compilation and analysis of quantum computing programs," in *Proc. 11th ACM Conf. Comput. Frontiers*, 2014, p. 1.
- [55] C. M. Dawson and M. A. Nielsen, "The Solovay-Kitaev algorithm," 2005, *arXiv:quant-ph/0505030*. [Online]. Available: <https://arxiv.org/abs/quant-ph/0505030>
- [56] N. J. Ross and P. Selinger, "Optimal ancilla-free clifford+T approximation of z-rotations," 2014, *arXiv:1403.2975*. [Online]. Available: <https://arxiv.org/abs/1403.2975>
- [57] M. Amy, D. Maslov, and M. Mosca, "Polynomial-time T-depth optimization of Clifford+T circuits via modular partitioning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 33, no. 10, pp. 1476–1489, Sep. 2014.
- [58] A. Bocharov, M. Roetteler, and K. M. Svore, "Efficient synthesis of universal repeat-until-success quantum circuits," *Phys. Rev. Lett.*, vol. 114, no. 8, 2015, Art. no. 080502.
- [59] A. Javadi-Abhari *et al.*, "Optimized surface code communication in superconducting quantum computers," in *Proc. 50th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Oct. 2017, pp. 692–705.
- [60] L. Cywiński, R. M. Lutchyn, C. P. Nave, and S. D. Sarma, "How to enhance dephasing time in superconducting qubits," *Phys. Rev. B, Condens. Matter*, vol. 77, no. 17, 2008, Art. no. 174509.
- [61] L. Viola, E. Knill, and S. Lloyd, "Dynamical decoupling of open quantum systems," *Phys. Rev. Lett.*, vol. 82, no. 12, p. 2417, 1999.
- [62] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. Glaser, "Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms," *J. Magn. Reson.*, vol. 172, no. 2, pp. 296–305, 2005.
- [63] R. Iten, R. Colbeck, I. Kulukjian, J. Home, and M. Christandl, "Quantum circuits for isometries," *Phys. Rev. A, Gen. Phys.*, vol. 93, no. 3, Mar. 2016, Art. no. 032318.
- [64] V. V. Shende, S. S. Bullock, and I. L. Markov, "Synthesis of quantum-logic circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 6, pp. 1000–1010, Jun. 2006.
- [65] F. Vatan and C. Williams, "Optimal quantum circuits for general two-qubit gates," *Phys. Rev. A, Gen. Phys.*, vol. 69, no. 3, 2004, Art. no. 032315.
- [66] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 32, no. 6, pp. 818–830, Jun. 2013.
- [67] D. Maslov, S. M. Falconer, and M. Mosca, "Quantum circuit placement," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 4, pp. 752–763, Apr. 2008.
- [68] M. Y. Siraichi, V. F. D. Santos, S. Collange, and F. M. Q. Pereira, "Qubit allocation," in *Proc. Int. Symp. Code Gener. Optim.*, 2018, pp. 113–125.
- [69] D. Venturelli, M. Do, E. Rieffel, and J. Frank, "Compiling quantum circuits to realistic hardware architectures using temporal planners," *Quantum Sci. Technol.*, vol. 3, no. 2, 2018, Art. no. 025004.
- [70] A. Zulehner and R. Wille, "Compiling SU(4) quantum circuits to IBM QX architectures," in *Proc. 24th Asia South Pacific Design Autom. Conf.*, 2019, pp. 185–190.
- [71] A. M. Childs, E. Schoute, and C. M. Unsal, "Circuit transformations for quantum architectures," 2019, *arXiv:1902.09102*. [Online]. Available: <https://arxiv.org/abs/1902.09102>
- [72] R. Duncan, A. Kissinger, S. Pedrix, and J. van de Wetering, "Graph-theoretic simplification of quantum circuits with the ZX-calculus," 2019, *arXiv:1902.03178*. [Online]. Available: <https://arxiv.org/abs/1902.03178>
- [73] M. Amy, P. Azimzadeh, and M. Mosca, "On the controlled-NOT complexity of controlled-NOT-phase circuits," *Quantum Sci. Technol.*, vol. 4, no. 1, 2018, Art. no. 015002.
- [74] Y. Nam, N. J. Ross, Y. Su, A. M. Childs, and D. Maslov, "Automated optimization of large quantum circuits with continuous parameters," *npj Quantum Inf.*, vol. 4, no. 1, p. 23, 2018.
- [75] D. Maslov, G. W. Dueck, D. M. Miller, and N. Camille, "Quantum circuit simplification and level compaction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 3, pp. 436–444, Mar. 2008.
- [76] T. S. Metodi, D. D. Thaker, A. W. Cross, F. T. Chong, and I. L. Chuang, "Scheduling physical operations in a quantum information processor," *Proc. SPIE*, vol. 6244, May 2006, Art. no. 62440T.
- [77] D. Maslov, "Basic circuit compilation techniques for an ion-trap quantum machine," *New J. Phys.*, vol. 19, no. 2, 2017, Art. no. 023035.
- [78] G. G. Guerreschi and J. Park, "Two-step approach to scheduling quantum circuits," *Quantum Sci. Tech.*, vol. 3, no. 4, 2018, Art. no. 045003.
- [79] J. J. Wallman and J. Emerson, "Noise tailoring for scalable quantum computation via randomized compiling," *Phys. Rev. A, Gen. Phys.*, vol. 94, no. 5, 2016, Art. no. 052325.
- [80] S. S. Tannu and M. K. Qureshi, "A case for variability-aware policies for NISQ-era quantum computers," 2018, *arXiv:1805.10224*. [Online]. Available: <https://arxiv.org/abs/1805.10224>
- [81] P. Murali, J. M. Baker, A. Javadi-Abhari, F. T. Chong, and M. Martonosi, "Noise-adaptive compiler mappings for noisy intermediate-scale quantum computers," in *Proc. 24th Int. Conf. Architectural Support Program. Lang. Oper. Syst.*, 2019, pp. 1015–1029.
- [82] P. Murali, N. M. Linke, M. Martonosi, A. Javadi-Abhari, N. H. Nguyen, and C. H. Alderete, "Full-stack real-system quantum computer studies: Architectural comparisons and design insights," in *Proc. 46th Int. Symp. Comput. Archit. (ISCA)*, 2019, pp. 527–540. [Online]. Available: <http://doi.acm.org/10.1145/3307650.3322273>
- [83] S. Nishio, Y. Pan, T. Satoh, H. Amano, and R. V. Meter, "Extracting success from IBM's 20-qubit machines using error-aware compilation," 2019, *arXiv:1903.10963*. [Online]. Available: <https://arxiv.org/abs/1903.10963>
- [84] W. Finigan, M. Cubeddu, T. Lively, J. Flick, and P. Narang, "Qubit allocation for noisy intermediate-scale quantum computers," 2018, *arXiv:1810.08291*. [Online]. Available: <https://arxiv.org/abs/1810.08291>
- [85] E. C. Peterson, G. E. Crooks, and R. S. Smith, "Fixed-depth two-qubit circuits and the monodromy polytope," 2019, *arXiv:1904.10541*. [Online]. Available: <https://arxiv.org/abs/1904.10541>
- [86] M. Amy, M. Roetteler, and K. M. Svore, "Verified compilation of space-efficient reversible circuits," in *Proc. Int. Conf. Comput. Aided Verification*, Cham, Switzerland: Springer, 2017, pp. 3–21.
- [87] Y. Shi *et al.*, "Contract-based verification of a realistic quantum compiler," 2019, *arXiv:1908.08963*. [Online]. Available: <https://arxiv.org/abs/1908.08963>
- [88] T. Itoko, R. Raymond, T. Imamichi, and A. Matsuo, "Optimization of quantum circuit mapping using gate transformation and commutation," 2019, *arXiv:1907.02686*. [Online]. Available: <https://arxiv.org/abs/1907.02686>
- [89] Y. Shi *et al.*, "Optimized compilation of aggregated instructions for realistic quantum computers," in *Proc. 24th Int. Conf. Architectural Support Programming Lang. Oper. Syst.*, 2019, pp. 1031–1044.
- [90] R. Blume-Kohout and K. Young, "A volumetric framework for quantum computer benchmarks," 2019, *arXiv:1904.05546*. [Online]. Available: <https://arxiv.org/abs/1904.05546>
- [91] A. Erhard *et al.*, "Characterizing large-scale quantum computers via cycle benchmarking," 2019, *arXiv:1902.08543*. [Online]. Available: <https://arxiv.org/abs/1902.08543>
- [92] K. E. Hamilton, E. F. Dumitrescu, and R. C. Pooser, "Generative model benchmarks for superconducting qubits," *Phys. Rev. A, Gen. Phys.*, vol. 99, no. 6, 2019, Art. no. 062323.
- [93] W. G. Unruh, "Maintaining coherence in quantum computers," *Phys. Rev. A, Gen. Phys.*, vol. 51, no. 2, p. 992, 1995.
- [94] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *J. Math. Phys.*, vol. 43, no. 9, pp. 4452–4505, 2002.
- [95] L. F. Richardson, "IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam," *Philos. Trans. Roy. Soc. London A, Containing Papers Math. Phys. Character.*, vol. 210, nos. 459–470, pp. 307–357, 1911.
- [96] C. Song, J. Cui, H. Wang, J. Hao, H. Feng, and Y. Li, "Quantum computation with universal error mitigation on superconducting quantum processor," 2018, *arXiv:1812.10903*. [Online]. Available: <https://arxiv.org/abs/1812.10903>
- [97] S. Zhang *et al.*, "Error-mitigated quantum gates exceeding physical fidelities in a trapped-ion system," 2019, *arXiv:1905.10135*. [Online]. Available: <https://arxiv.org/abs/1905.10135>
- [98] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, "Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states," *Phys. Rev. A, Gen. Phys.*, vol. 95, Apr. 2017, Art. no. 042308. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.95.042308>
- [99] J. I. Colless *et al.*, "Computation of molecular spectra on a quantum processor with an error-resilient algorithm," *Phys. Rev. X*, vol. 8, Feb. 2018, Art. no. 011021.
- [100] S. McArdle, X. Yuan, and S. Benjamin, "Error-mitigated digital quantum simulation," *Phys. Rev. Lett.*, vol. 122, May 2019, Art. no. 180501. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.122.180501>
- [101] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O'Brien, "Low-cost error mitigation by symmetry verification," *Phys. Rev. A, Gen. Phys.*, vol. 98, Dec. 2018, Art. no. 062339. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.98.062339>
- [102] R. Sagastizabal *et al.*, "Experimental error mitigation via symmetry verification in a variational quantum eigensolver," *Phys. Rev. A, Gen. Phys.*, vol. 100, Jul. 2019, Art. no. 010302. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevA.100.010302>
- [103] A. Córcoles *et al.*, "Demonstration of a quantum error detection code using a square lattice of four superconducting qubits," *Nature Commun.*, vol. 6, Apr. 2015, Art. no. 6979.
- [104] N. M. Linke *et al.*, "Fault-tolerant quantum error detection," *Sci. Adv.*, vol. 3, no. 10, 2017, Art. no. e1701074.
- [105] M. Takita, A. W. Cross, A. D. Córcoles, J. M. Chow, and J. M. Gambetta, "Experimental demonstration of fault-tolerant state preparation with superconducting qubits," *Phys. Rev. Lett.*, vol. 119, Oct. 2017, Art. no. 180501.
- [106] M. Gong *et al.*, "Experimental verification of five-qubit quantum error correction with superconducting qubits," 2019, *arXiv:1907.04507*. [Online]. Available: <https://arxiv.org/abs/1907.04507>
- [107] C. Chamberland, G. Zhu, T. Yoder, J. Hertzberg, and A. W. Cross, "Topological and subsystem codes on low-degree graphs with flag qubits," 2019, *arXiv:1907.09528*. [Online]. Available: <https://arxiv.org/abs/1907.09528>
- [108] C. Zalka, "Simulating quantum systems on a quantum computer," *Proc. Roy. Soc. London A, Math., Phys. Eng. Sci.*, vol. 454, no. 1969, pp. 313–322, 1998.
- [109] *Quantum Algorithm Zoo*. Accessed: Sep. 24, 2019. [Online]. Available: <https://quantumalgorithmzoo.org/>
- [110] A. Peruzzo *et al.*, "A variational eigenvalue solver

- on a photonic quantum processor,” *Nature Commun.*, vol. 5, Jul. 2014, Art. no. 4213.
- [111] A. Kandala *et al.*, “Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets,” *Nature*, vol. 549, no. 7671, p. 242, 2017.
- [112] E. Tang, “A quantum-inspired classical algorithm for recommendation systems,” in *Proc. STOC*, 2019, pp. 217–228.
- [113] A. Gilley, S. Lloyd, and E. Tang, “Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension,” 2019, *arXiv:1811.04909*. [Online]. Available: <https://arxiv.org/abs/1811.04909>
- [114] S. Bravyi and D. Gosset, “Improved classical simulation of quantum circuits dominated by clifford gates,” *Phys. Rev. Lett.*, vol. 116, no. 25, 2016, Art. no. 250501.
- [115] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, “Simulation of quantum circuits by low-rank stabilizer decompositions,” 2018, *arxiv:1808.00128*. [Online]. Available: <https://arxiv.org/abs/1808.00128>
- [116] S. Aaronson and D. Gottesman, “Improved simulation of stabilizer circuits,” *Phys. Rev. A, Gen. Phys.*, vol. 70, no. 5, 2004, Art. no. 052328.
- [117] A. Razborov, “An upper bound on the threshold quantum decoherence rate,” *Quantum Inf. Comput.*, vol. 4, no. 3, pp. 222–228, 2004.
- [118] M. Bremner, A. Montanaro, and D. J. Shepherd, “Achieving quantum supremacy with sparse and noisy commuting quantum computations,” *Quantum*, vol. 1, p. 8, Apr. 2017.
- [119] D. Shepherd and M. J. Bremner, “Instantaneous quantum computation,” *Proc. Roy. Soc., vol. 465*, pp. 1413–1439, Jan. 2009.
- [120] M. Bremner, A. Montanaro, and D. J. Shepherd, “Average-case complexity versus approximate simulation of commuting quantum computations,” *Phys. Rev. Lett.*, vol. 117, no. 8, 2016, Art. no. 080501.
- [121] A. Boulard, J. F. Fitzsimons, and D. E. Koh, “Complexity classification of conjugated Clifford circuits,” in *Proc. 33rd Comput. Complex. Conf. (CCC)*, 2018, pp. 21:1–21:25. [Online]. Available: <http://drops.dagstuhl.de/opus/volltexte/2018/8867>
- [122] M. Schuld and N. Killoran, “Quantum machine learning in feature Hilbert spaces,” *Phys. Rev. Lett.*, vol. 122, Feb. 2019, Art. no. 040504. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.122.040504>
- [123] S. Aaronson and A. Ambainis, “Forrelation: A problem that optimally separates quantum from classical computing,” *SIAM J. Comput.*, vol. 47, no. 3, pp. 982–1038, 2018, doi: 10.1137/15M1050902.
- [124] Y. Wang *et al.*, “Quantum simulation of helium hydride cation in a solid-state spin register,” *ACS Nano*, vol. 9, no. 8, pp. 7769–7774, 2015.
- [125] P. J. J. O’Malley *et al.*, “Scalable quantum simulation of molecular energies,” *Phys. Rev. X*, vol. 6, no. 3, 2016, Art. no. 031007.
- [126] Y. Shen, X. Zhang, S. Zhang, J.-N. Zhang, M.-H. Yung, and K. Kim, “Quantum implementation of the unitary coupled cluster for simulating molecular electronic structure,” *Phys. Rev. A, Gen. Phys.*, vol. 95, Feb. 2017, Art. no. 020501.
- [127] C. Hempel *et al.*, “Quantum chemistry calculations on a trapped-ion quantum simulator,” *Phys. Rev. X*, vol. 8, Jul. 2018, Art. no. 031022.
- [128] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme, “Tapering off qubits to simulate fermionic Hamiltonians,” 2017, *arXiv:1701.08213*. [Online]. Available: <https://arxiv.org/abs/1701.08213>
- [129] N. Schuch, M. M. Wolf, F. Verstraete, and J. I. Cirac, “Computational complexity of projected entangled pair states,” *Phys. Rev. Lett.*, vol. 98, no. 14, 2007, Art. no. 140506.
- [130] M. Schwarz, K. Temme, F. Verstraete, D. Perez-Garcia, and T. S. Cubitt, “Preparing topological projected entangled pair states on a quantum computer,” *Phys. Rev. A, Gen. Phys.*, vol. 88, no. 3, 2013, Art. no. 032321.
- [131] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, “Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz,” *Quantum Sci. Technol.*, vol. 4, no. 1, 2018, Art. no. 014008.
- [132] P. Gokhale and F. T. Chong, “ $O(N^3)$ measurement cost for variational quantum eigensolver on molecular Hamiltonians,” 2019, *arXiv:1908.11857*. [Online]. Available: <https://arxiv.org/abs/1908.11857>
- [133] A. F. Izmaylov, T.-C. Yen, R. A. Lang, and V. Verteletskyi, “Unitary partitioning approach to the measurement problem in the variational quantum eigensolver method,” 2019, *arXiv:1907.09040*. [Online]. Available: <https://arxiv.org/abs/1907.09040>
- [134] J. Koch *et al.*, “Charge-insensitive qubit design derived from the Cooper pair box,” *Phys. Rev. A, Gen. Phys.*, vol. 76, no. 4, Oct. 2007, Art. no. 042319.
- [135] E. Magesan, J. M. Gambetta, and J. Emerson, “Scalable and robust randomized benchmarking of quantum processes,” *Phys. Rev. Lett.*, vol. 106, May 2011, Art. no. 180504. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevLett.106.180504>
- [136] G. S. Paraoanu, “Microwave-induced coupling of superconducting qubits,” *Phys. Rev. B, Condens. Matter*, vol. 74, Oct. 2006, Art. no. 140504. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevB.74.140504>
- [137] C. Rigetti and M. Devoret, “Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies,” *Phys. Rev. B, Condens. Matter*, vol. 81, Apr. 2010, Art. no. 134507. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevB.81.134507>
- [138] A. Blais, R.-S. Huang, A. Wallraff, S. M. Girvin, and R. J. Schoelkopf, “Cavity quantum electrodynamics for superconducting electrical circuits: An architecture for quantum computation,” *Phys. Rev. A, Gen. Phys.*, vol. 69, no. 6, 2004, Art. no. 062320.
- [139] P. Krantz, M. Kjaergaard, F. Yan, T. Orlando, S. Gustavsson, and W. D. Oliver, “A quantum engineer’s guide to superconducting qubits,” *Appl. Phys. Rev.*, vol. 6, no. 2, 2019, Art. no. 021318.
- [140] G. Wendin, “Quantum information processing with superconducting circuits: A review,” *Rep. Prog. Phys.*, vol. 80, no. 10, 2017, Art. no. 106001.
- [141] B. Lienhard *et al.*, “Microwave packaging for superconducting qubits,” 2005, *arXiv:1906.05425*. [Online]. Available: <https://arxiv.org/abs/1906.05425>
- [142] C. E. Murray and D. W. Abraham, “Predicting substrate resonance mode frequency shifts using conductive, through-substrate vias,” *Appl. Phys. Lett.*, vol. 108, no. 8, 2016, Art. no. 084101, doi: 10.1063/1.4942453.
- [143] M. D. Reed *et al.*, “Fast reset and suppressing spontaneous emission of a superconducting qubit,” *Appl. Phys. Lett.*, vol. 96, no. 20, 2010, Art. no. 203110.
- [144] D. Rosenberg *et al.*, “3D integration and packaging for solid-state qubits,” 2019, *arXiv:1906.11146*. [Online]. Available: <https://arxiv.org/abs/1906.11146>
- [145] S. Krinner *et al.*, “Engineering cryogenic setups for 100-qubit scale superconducting circuit systems,” 2018, *arXiv:1806.07862*. [Online]. Available: <https://arxiv.org/abs/1806.07862>
- [146] D. B. Tuckerman *et al.*, “Flexible superconducting nb transmission lines on thin film polyimide for quantum computing applications,” *Superconductor Sci. Technol.*, vol. 29, no. 8, 2016, Art. no. 084007.
- [147] J. C. Bardin *et al.*, “A 28 nm bulk-CMOS 4-to-8 GHz <2 mw cryogenic pulse modulator for scalable quantum computing,” 2019, *arXiv:1902.10864*. [Online]. Available: <https://arxiv.org/abs/1902.10864>
- [148] M. Mehrpoor *et al.*, “Benefits and challenges of designing cryogenic CMOS RF circuits for quantum computers,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.
- [149] M. M. Wilde, *Quantum Information Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2013.
- [150] J. Watrous, *Quantum Computational Complexity*. New York, NY, USA: Springer, 2009, pp. 7174–7201.
- [151] C. H. Bennett and G. Brassard, “Quantum cryptography: Public key distribution and coin tossing,” *Theor. Comput. Sci.*, vol. 560, no. 12, pp. 7–11, 2014.
- [152] N. Gisin and R. Thew, “Quantum communication,” *Nature Photon.*, vol. 1, no. 3, pp. 165–171, 2007.
- [153] C. L. Degen, F. Reinhard, and P. Cappellaro, “Quantum sensing,” *Rev. Mod. Phys.*, vol. 89, Jul. 2017, Art. no. 035002.
- [154] D. Bernstein and T. Lange, “Post-quantum cryptography,” *Nature*, vol. 549, no. 7671, pp. 188–194, 2017.

ABOUT THE AUTHORS

Antonio D. Córcoles received the Ph.D. degree in physics from the Royal Holloway, University of London, Egham, U.K., under the supervision of Prof. J. Saunders.

He then worked as a Postdoctoral Research Associate at the Cavendish Laboratory, Cambridge, U.K., with Prof. Sir Michael Pepper on hyperfine interactions in quantum wires. He joined IBM Watson, Yorktown Heights, NY, USA, as a Postdoctoral Researcher, in 2010, where he has been a Research Staff Member since 2014, working on the design, experimental realization, calibration, characterization, and benchmarking of superconducting quantum computing systems.

Abhinav Kandala received the bachelor’s degree in engineering physics from IIT Bombay, Mumbai, India, and the Ph.D. degree in physics from The Pennsylvania State University, State College, PA, USA. His graduate research was focused on a class of materials known as topological insulators, studying their interplay with magnetism to engineer and detect dissipation-less electrical transport in these material systems.

He joined the Quantum Computing Group, IBM Research, Yorktown Heights, NY, USA, as a Postdoctoral Researcher, where he is currently a Research Staff Member with the IBM Thomas J. Watson Research Center. His research has focused on the coherence and control of superconducting qubits, multiqubit device characterization, and applications of near-term quantum computers.

Ali Javadi-Abhari received the Ph.D. degree from Princeton University, Princeton, NJ, USA, where he investigated the resource requirements of fully fault-tolerant quantum computations and the benefits of specializing quantum computations to hardware. His work led to the Scaffold software tools for quantum program analysis and optimization.

He is currently a Research Staff Member with IBM Watson, Yorktown Heights, NY, USA. He is the Technical Lead for IBM's open-source quantum software, Qiskit. His research focuses on the software stack for near-term quantum machines and optimizations to enable useful computations on noisy devices.

From 2012 to 2014, he was an Erwin Schrödinger Postdoctoral Fellow at the Massachusetts Institute of Technology, Cambridge, MA, USA. From March 2014 to September 2015, he held the IQIM Postdoctoral Fellowship of the Gordon and Betty Moore Foundation at the California Institute of Technology, Pasadena, CA, USA. He joined IBM Research, Yorktown Heights, NY, USA, in the fall of 2015, where he is currently a Research Staff Member with the Theory of Quantum Computation and Information Group, IBM Thomas J. Watson Research Center. His research currently centers around quantum algorithms and noise in complex quantum systems with a focus on the applications of noisy intermediate-scale quantum devices.

Douglas T. McClure received the Ph.D. degree in experimental physics from Harvard University, Cambridge, MA, USA, where under the guidance of Prof. C. Marcus, he studied the physics of fractional quantum Hall states with potential applications in topological quantum computing.

Since 2012, he has been with IBM Research, Yorktown Heights, NY, USA, where he has focused on several aspects of building quantum computers based on superconducting qubits: improving quantum measurements, qubit coherence times, and most recently the performance and reliability of large multiquantum devices and is currently a Research Staff Member and the Manager of Quantum System Deployment.

Andrew W. Cross (Member, IEEE) received the B.S. degree in electrical engineering, minoring in mathematics, physics, and music, from Case Western Reserve University, Cleveland, OH, USA, and the S.M. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2005 and 2008, respectively. His doctoral thesis focused on quantum error-correcting codes and fault-tolerant architectures for quantum computers.

He has been working in the quantum computing field since 2003. He joined IBM Research, Yorktown Heights, NY, USA, in 2013, where he is currently a Research Staff Member and the Manager of the Theory of Quantum Computing and Information Group. He has been recognized by IBM for his contributions to the development of the IBM Q Experience and for authoring early versions of the Qiskit open-source software for quantum computing. He has authored over 30 peer-reviewed articles. His research interests include quantum computing, quantum error correction, fault tolerance, and verification, validation, and control of superconducting qubits.

Kristan Temme received the Diploma degree in physics from the University of Heidelberg, Heidelberg, Germany, in 2007, and the Ph.D. degree in physics from the University of Vienna, Vienna, Austria, in 2011. During this time, he studied the quantum Markov processes and complex quantum many-body systems.

Paul D. Nation received the Ph.D. degree from Dartmouth College, Hanover, NH, USA, in 2010.

He was a Postdoctoral Researcher at RIKEN, Wakō, Japan, and an Assistant Professor of physics at Korea University, Seoul, South Korea. He joined IBM Watson, Yorktown Heights, NY, USA. He is a Theoretical Physicist focusing on numerical methods for open quantum systems and high-fidelity quantum operations in superconducting quantum systems.

Matthias Steffen (Senior Member, IEEE) received the B.S. degree in physics from Emory University, Atlanta, GA, USA, in 1998, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2000 and 2003, respectively.

He has been working in the field of quantum computing since 1998 and focused on a variety of approaches toward building a quantum computer. His Ph.D. thesis focused on testing small prototype quantum computers using nuclear spins in a liquid solution, while his postdoctoral work and work at IBM centered on advancing superconducting quantum bits. He joined IBM Watson, Yorktown Heights, NY, USA, in 2006 and managed the Experimental Quantum Computing Group from 2010 to 2014, after which he became the Chief Quantum Architect. He has authored or coauthored over 40 peer-reviewed articles in the field of quantum computing.

Dr. Steffen was appointed as an APS Fellow in 2013 for his contributions to the quantum computing field. In 2017, he was named IBM Fellow for his work in quantum computing.

Jay M. Gambetta (Senior Member, IEEE) received the B.S. and Ph.D. degrees in physics from Griffith University in 1999 and 2004, respectively.

He was a Postdoctoral Fellow with Yale University, New Haven, CT, USA, from 2004 to 2007, where he started his research in quantum information processing with superconducting circuits. He then held positions as a Postdoctoral Fellow from 2007 to 2009, a CIFAR Junior Fellow from 2009 to 2010, and a Research Assistant Professor from 2010 to 2011 with the Institute for Quantum Computing, Waterloo, ON, Canada. He joined IBM Research, Yorktown Heights, NY, USA, in 2011, as a Research Staff Member and became the Manager of the Quantum Theory Group in 2013, where he is currently the Vice President (VP) of quantum computing.

Dr. Gambetta was named a fellow of the American Physical Society in 2014 and was named an IBM Fellow for his contributions to quantum computing in 2018.