Journal of Applied Physics

RESEARCH ARTICLE | OCTOBER 26 2022

The future of quantum computing with superconducting qubits FREE

Sergey Bravyi; Oliver Dial; Jay M. Gambetta; Darío Gil; Zaira Nazario

Check for updates

Journal of Applied Physics 132, 160902 (2022)
https://doi.org/10.1063/5.0082975

View Online

Export Citation

CrossMark

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

AIP Publishing

The future of quantum computing with superconducting qubits

Cite as: J. Appl. Phys. **132**, 160902 (2022); doi: 10.1063/5.0082975
Submitted: 20 December 2021 · Accepted: 15 September 2022 ·
Published Online: 26 October 2022

Sergey Bravyi, Oliver Dial, Jay M. Gambetta, Darío Gil, and Zaira Nazario[a)]

AFFILIATIONS

IBM Quantum, IBM T.J. Watson Research Center, Yorktown Heights, New York 10598, USA

[a)]**Author to whom correspondence should be addressed:** zaira.nazario@ibm.com

ABSTRACT

For the first time in history, we are seeing a branching point in computing paradigms with the emergence of quantum processing units (QPUs). Extracting the full potential of computation and realizing quantum algorithms with a super-polynomial speedup will most likely require major advances in quantum error correction technology. Meanwhile, achieving a computational advantage in the near term may be possible by combining multiple QPUs through circuit knitting techniques, improving the quality of solutions through error suppression and mitigation, and focusing on heuristic versions of quantum algorithms with asymptotic speedups. For this to happen, the performance of quantum computing hardware needs to improve and software needs to seamlessly integrate quantum and classical processors together to form a new architecture that we are calling quantum-centric supercomputing. In the long term, we see hardware that exploits qubit connectivity in higher than 2D topologies to realize more efficient quantum error correcting codes, modular architectures for scaling QPUs and parallelizing workloads, and software that evolves to make the intricacies of the technology invisible to the users and realize the goal of ubiquitous, frictionless quantum computing.

Published under an exclusive license by AIP Publishing. https://doi.org/10.1063/5.0082975

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

I. INTRODUCTION

The history of computing is that of advances born out of the need to perform ever more sophisticated calculations. Increasingly advanced semiconductor manufacturing processes have resulted in faster and more efficient chips—most recently the 2 nm technology node[1]—and special accelerators like the graphics processing unit (GPU), tensor processing unit (TPU), and artificial intelligence (AI) processors[2] have allowed more efficient computations on larger data sets. These advances share the same model of computation dating back to 1936 with the origins of the Church–Turing thesis. Now for the first time in history, the field of computing has branched with the emergence of quantum computers, which, when scaled, promise to implement computations intractable for conventional computers—from modeling quantum mechanical systems[3] to linear algebra,[4] factoring,[5] search,[6] and more.

Unlocking the full potential of quantum processors requires the implementation of computations with a large number of operations. Since quantum gates are considerably less accurate than their classical counterparts, it is strongly believed that error correction will be necessary to realize long computations with millions or billions of gates. Accordingly, most of quantum computing platforms are designed with the long term goal of realizing error-corrected quantum circuits. As the noise rate decreases below a constant architecture-dependent threshold, an arbitrarily long quantum circuit can be executed reliably by redundantly encoding each qubit and repeatedly measuring parity check operators to detect and correct errors. However, the number of qubits required to realize error-corrected quantum circuits solving classically hard problems exceeds the size of systems available today by several orders of magnitude.

Meanwhile, as the quality and number of qubits in quantum computers continue to grow, we must be able to harvest the computational power of quantum circuits available along the way. For example, a quantum processing unit (QPU) with two-qubit gate fidelity of 99.99% can implement circuits with a few thousand gates to a fair degree of reliability without resorting to error correction. Such circuits are strongly believed to be practically impossible to simulate classically, even with the help of modern supercomputers. This suggests the possibility that the first demonstrations of a computational *quantum advantage*—where a computational task of business or scientific relevance can be performed more efficiently, cost-effectively, or accurately using a quantum computer than with classical computations alone—*may be achieved without or with limited error correction*.

Published under an exclusive license by AIP Publishing

Three central questions need to be answered for this to happen: (1) how to extract useful data from the output of noisy quantum circuits in the weak noise regime, (2) how to design quantum algorithms based on shallow circuits that can potentially solve some classically hard problems, and (3) how to improve the efficiency of quantum error-correction schemes and use error correction more sparingly.

These questions and our approach are discussed in detail in Sec. II. As an illustration, we pick one of the simplest scientifically relevant applications of quantum computers—simulating time evolution of a spin chain Hamiltonian.[7] We discuss state-of-the-art quantum algorithms for this problem and highlight the cost of making time evolution circuits fault-tolerant by encoding each qubit into the surface code,[8,9] considered the best fit for hardware with two-dimensional qubit connectivity. For problem sizes of practical interest, error correction increases the size of quantum circuits by nearly six orders of magnitude, making it prohibitively expensive for near-term QPUs (see Sec. II A).

Question (1) is approached in Secs. II B and II C through quantum error mitigation[10,11] and circuit knitting.[12–15] These techniques extend the size of quantum circuits that can be executed reliably on a given QPU without resorting to error correction. We estimate the overhead introduced by state-of-the-art error mitigation methods and discuss recent ideas on how to combine error correction and mitigation. Circuit knitting techniques exploit structural properties of the simulated system, such as geometric locality, to decompose a large quantum circuit into smaller sub-circuits or combine solutions produced by multiple QPUs.

The classical simulation algorithms used in computational physics or chemistry are often heuristics and work well in practice, even though they do not offer rigorous performance guarantees. Thus, it is natural to ask whether rigorous quantum algorithms designed for simulating time evolution admit less expensive heuristic versions that are more amenable to near-term QPUs. We discuss such algorithms in Sec. II D to address question (2).

To approach question (3), we discuss generalizations of the surface code known as low-density parity check (LDPC) quantum codes.[16,17] These codes can pack many more logical qubits into a given number of physical qubits such that, as the size of quantum circuits grows, only a constant fraction of physical qubits is devoted to error correction (see Sec. II A for details). These more efficient codes need long-range connections between qubits embedded in a two-dimensional grid,[18] but the efficiency benefits are expected to outweigh the long-range connectivity costs.

We then focus on quantum-centric supercomputing, which is a new architecture for realizing error mitigation, circuit knitting, and heuristic quantum algorithms with substantial classical calculations. At the heart of this architecture is classical and quantum integration and modularity. We need classical integration at real-time to enable conditioning quantum circuits on classical computations (dynamic circuits), at near-time to enable error mitigation and eventually error correction, and at compile time to enable circuit knitting and advanced compiling. We need modularity to enable scaling and speeding up workflows by using parallelization. We first start in Sec. III by focusing on superconducting computing hardware and we introduce a series of schemes—which we denote m, l, c, and t couplers—that give us the amount of flexibility needed for realizing LDPC codes, scaling QPUs, and enabling workflows that take advantage of local operations and classical communication (LOCC) and parallelization. In Sec. IV, we discuss the requirements on the quantum stack by defining different layers for integrating classical and quantum computations, which define requirements on latency, parallelization, and the compute instructions. From this, we can define a cluster-like architecture that we call quantum-centric supercomputer. It consists of many quantum computation nodes comprised of classical computers, control electronics, and QPUs. A quantum runtime can be executed on a quantum-centric supercomputer, working in the cloud or other classical computers to run many quantum runtimes in parallel. Here, we propose that a serverless model should be used so that developers can focus on code and do not have to manage the underlying infrastructure. We conclude with a high level view from a developer's/user's lens.

This paper offers a perspective of the future of quantum computing focusing on an examination of what it takes to build and program near-term superconducting quantum computers and demonstrate their utility. Realizing the computational power of these machines requires the concerted efforts of engineers, physicists, computer scientists, and software developers. Hardware advances will raise the bar of quantum computers' size and fidelity. Theory and software advances will lower the bar for implementing algorithms and enable new capabilities. As both bars converge in the next few years, we will start seeing the first practical benefits of quantum computation.

II. TOWARD PRACTICALLY USEFUL QUANTUM CIRCUITS

Although, in principle, a quantum computer can reproduce any calculation performed on conventional classical hardware, the vast majority of everyday tasks are not expected to benefit from quantum-mechanical effects. However, using quantum mechanics to store and process information can lead to dramatic speedups for certain carefully selected applications. Of particular interest are tasks that admit a quantum algorithm with the runtime scaling as a small constant power of the problem size n—e.g., as n^2 or n^3—whereas the best known classical algorithm solving the problem has runtime growing faster than any constant power of n—e.g., as 2^n or $2^{\sqrt{n}}$. We define runtime as the number of elementary gates in a circuit (or circuits) implementing the algorithm for a given problem instance. As the problem size n grows, the more favorable scaling of the quantum runtime quickly compensates for a relatively high cost and slowness of quantum gates compared with their classical counterparts. These exponential or, formally speaking, super-polynomial speedups are fascinating from a purely theoretical standpoint and provide a compelling practical reason for advancing quantum technologies.

Known examples of tasks with an exponential quantum speedup include simulation of quantum many-body systems,[19] number theoretic problems such as integer factoring,[5] solving certain types of linear systems,[20] estimation of Betti numbers used in topological data analysis,[21–23] and computing topological invariants of knots and links.[24] (We leave aside speedups obtained in the so-called Quantum RAM model,[25] for although it appears to be more powerful than the standard quantum circuit model, it is

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

unclear whether a Quantum RAM can be efficiently implemented in any real physical system.)

Simulation of quantum many-body systems has received the most attention due to its numerous scientific and industrial applications and for being the original value proposition for quantum computing.[26] The ground state and thermal-equilibrium properties of many-body systems can often be understood, at least qualitatively, using classical heuristic algorithms such as dynamical mean-field theory (DMFT) or perturbative methods. However, understanding their behavior far from equilibrium in the regime governed by coherent dynamics or performing high-precision ground state simulations for strongly interacting electrons—e.g., in the context of quantum chemistry—is a notoriously hard problem for classical computers.

As a simple illustration, consider a spin chain composed of n quantum spins (qubits or qudits) with Hamiltonian

$$H = \sum_{j=1}^{n-1} H_{j,j+1},$$

where $H_{j,j+1}$ is a two-spin nearest-neighbor interaction. The Schrödinger equation

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

governs the coherent time evolution of the system from some fixed initial state $|\psi(0)\rangle$.

Suppose our goal is to compute the expected value of some local observable on the time-evolved state $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$. Such expected values are of great interest for understanding, among other things, thermalization mechanisms in closed quantum systems.[27] Transforming the time-dependent expected values into the frequency domain provides valuable information about the excitation spectrum of the system.[28,29] A slightly modified version of this problem that involves measuring each qubit of $|\psi(t)\rangle$ is known to be BQP-complete,[30–33] meaning that it is essentially as hard as simulating a universal quantum computer.

The known classical algorithms for simulating the coherent time evolution of a quantum spin chain have runtime $\min(2^{O(n)}, 2^{O(vt)})$, where $v \sim \max_j \|H_{j,j+1}\|$ is the Lieb–Robinson velocity, which controls how fast information propagates through the system.[34] For simplicity, we ignore factors polynomial in n and t. The runtime $2^{O(n)}$ can be achieved using a standard state vector simulator while the runtime $2^{O(vt)}$ can be achieved by approximating $|\psi(t)\rangle$ with matrix product states[35,36] or by restricting the dynamics to a light cone.[37] In general, the linear growth of the entanglement entropy with time appears to be an insurmountable obstacle for classical simulation algorithms.

Haah *et al.* recently found a nearly optimal quantum algorithm for simulating the time evolution of spin chain Hamiltonians[38] with runtime $\tilde{O}(nt)$, where $\tilde{O}$ hides factors logarithmic in n, t, and the inverse approximation error. This algorithm works by approximating the time evolution operator e^{-iHt} by a product of simpler unitaries describing forward and backward time evolution of small blocks of spins of length $O(\log nt)$. Assuming that the evolution time t scales as a small constant power of n; e.g., $t \sim n$, this constitutes an exponential quantum speedup.

A natural question is what are the minimum quantum resources; i.e., qubit and gate counts, required to convincingly demonstrate a quantum advantage for simulating coherent dynamics. Childs *et al.* proposed a concrete benchmark problem for this, simulating the time evolution of the spin-1/2 Heisenberg chain with $n = t = 100$ and approximation error 0.001.[7] The Hamiltonian has the form $H = \sum_{j=1}^{n-1} \vec{\sigma}_j \vec{\sigma}_{j+1} + \sum_{j=1}^{n} h_j \sigma_j^z$, where $h_j \in [-1, 1]$ are randomly chosen magnetic fields. Figure 1 shows the gate count estimates for the benchmark problem obtained by Childs *et al.*[39] suggesting that about 10^7 CNOT gates (and a comparable number of single-qubit gates) are needed. This exceeds the size of quantum circuits demonstrated experimentally to date by several orders of magnitude. As we move from simple spin chain models to more practically relevant Hamiltonians, the gate count required to achieve quantum advantage increases dramatically. For example, simulating the active space of molecules involved in catalysis problems may require about 10^{11} Toffoli gates.[40] The only viable path to reliably implementing circuits with 10^7 gates or more on noisy quantum hardware is quantum error correction.

A. Quantum error correction

One reason why conventional classical computers became ubiquitous is their ability to store and process information reliably. Small fluctuations of an electric charge or current in a microchip can be tolerated due to a highly redundant representation of logical 0 and 1 states by a collective state of many electrons. Quantum error-correcting codes provide a similar redundant representation of quantum states that protects them from certain types of errors. A single logical qubit can be encoded into n physical qubits by

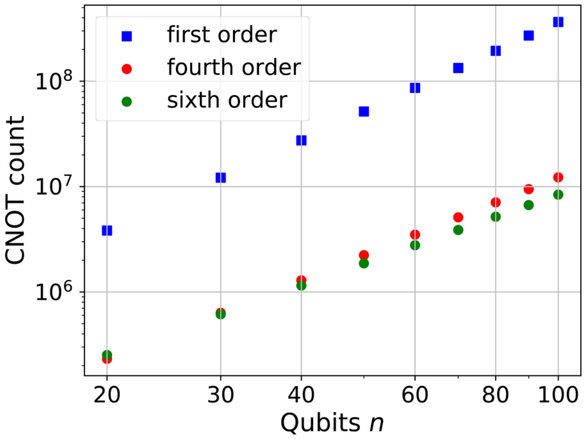

FIG. 1. Estimated number of CNOT gates required to approximate the unitary evolution operator e^{-iHt} for the n-qubit Heisenberg chain with $t = n$ and approximation error 0.001 using randomized kth order product formulas ($k = 1, 4, 6$). The presented data are based on empirical estimates of Ref. 39, see Eq. (70) therein, assuming that exponentiating a single term in the Hamiltonian costs 3 CNOTs.

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

specifying a pair of orthogonal n-qubit states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ called logical-0 and logical-1. A single-qubit state $\alpha|0\rangle + \beta|1\rangle$ is encoded by the logical state $\alpha|\bar{0}\rangle + \beta|\bar{1}\rangle$. A code has distance d if no operation affecting fewer than d qubits can distinguish the logical states $|\bar{0}\rangle$ and $|\bar{1}\rangle$ or map them into each other. More generally, a code may have k logical qubits encoded into n physical qubits and the code distance d quantifies how many physical qubits need to be corrupted before the logical (encoded) state is destroyed. Thus, good codes have a large distance d and a large encoding rate k/n.

Stabilizer-type codes[41,42] are by far the most studied and promising code family. A stabilizer code is defined by a list of commuting multi-qubit Pauli observables called stabilizers such that logical states are $+1$ eigenvectors of each stabilizer. One can view stabilizers as quantum analogs of classical parity checks. Syndrome measurements aim to identify stabilizers whose eigenvalue has flipped due to errors. The eigenvalue of each stabilizer is repeatedly measured and the result—known as the error syndrome—is sent to a classical decoding algorithm. Assuming that the number of faulty qubits and gates is sufficiently small, the error syndrome provides enough information to identify the error (modulo stabilizers). The decoder can then output the operation that needs to be applied to recover the original logical state.

Most of the codes designed for quantum computing are of the LDPC type,[16,43,44] meaning that each stabilizer acts on only a small number of qubits and each qubit participates in a small number of stabilizers, where small means a constant independent of the code size. The main advantage of quantum LDPC codes is that the syndrome measurement can be performed with a simple constant-depth quantum circuit. This ensures that the syndrome information can be collected frequently enough to cope with the accumulation of errors. Furthermore, errors introduced by the syndrome measurement circuit itself are sufficiently benign since the circuit can propagate errors only within a "light cone" of constant size.

A code must satisfy several requirements to have applications in quantum computing. First, it must have a high enough error threshold—the maximum level of hardware noise that it can tolerate. If the error rate is below the threshold, the lifetime of logical qubit(s) can be made arbitrarily long by choosing a large enough code distance. Otherwise, errors can accumulate faster than the code can correct them and logical qubits can become even less reliable than the constituent physical qubits. Second, one needs a fast decoding algorithm to perform error correction in real time as the quantum computation proceeds. This may be challenging since the decoding problem for general stabilizer codes is known to be NP-hard in the worst case.[45–47] Third, one must be able to compute on the logical qubits without compromising the protection offered by the code. In the sub-threshold regime, one must be able to realize arbitrarily precise logical gates from some universal gate set by choosing a large enough code distance.

The 2D surface code[8,9] has so far been considered an uncontested leader in terms of the error threshold—close to 1% for the commonly studied depolarizing noise[48–50]—yet has two important shortcomings. First, allocating a roughly $d \times d$ patch of physical qubits for each logical qubit incurs a large overhead. Unfortunately, it was shown[51] that any 2D stabilizer code has encoding rate $k/n = O(1/d^2)$ which vanishes for large code distance. This means that as one increases the degree of protection offered by the surface code, quantified by the code distance d, its encoding rate approaches zero. That is, as the size of quantum circuits grows, the vast majority of physical qubits are devoted to error correction. This is a known fundamental limitation of all quantum codes that can be realized locally in the 2D geometry.

To make error correction more practical and minimize qubit overhead, codes with a large encoding rate k/n are preferable. For example, quantum LDPC codes can achieve a constant encoding rate independent of the code size.[44] In fact, the encoding rate can be arbitrarily close to one.[16] A recent breakthrough result[52] demonstrated the existence of so-called good quantum LDPC codes that combine a constant encoding rate k/n (which can be arbitrarily close to $1/2$) and a linear distance $d \geq cn$ for some constant $c > 0$. For comparison, the 2D surface code has an asymptotically vanishing encoding rate and has distance at most $\sqrt{n}$. Certain LDPC codes have a favorable property known as single-shot error correction.[53,54] They provide a highly redundant set of low-weight Pauli observables (known as gauge operators) that can be measured to obtain the error syndrome more efficiently. This reduces the number of syndrome measurement cycles per logical gate from $O(d)$ to $O(1)$ and hence enables very fast logical gates. The syndrome measurement circuit for a quantum LDPC code requires a qubit connectivity dictated by the structure of stabilizers, i.e., one must be able to couple qubits that participate in the same stabilizer. Known examples of LDPC codes with a single-shot error correction require 3D or 4D geometry.[53–55]

The second shortcoming of the surface code is the difficulty of implementing a computationally universal set of logical gates.[56] The surface code and its variations such as the honeycomb code[57] or folded surface code[58] offer a low-overhead implementation of logical Clifford gates such as CNOT, Hadamard H, and phase shift S. These gates can be realized by altering the pattern of stabilizers measured at each time step using the code deformation method. However, Clifford gates are not computationally universal on their own. A common strategy for achieving universality is based on the preparation of logical ancillary states $(|\bar{0}\rangle + e^{i\pi/4}|\bar{1}\rangle)/\sqrt{2}$ known as magic states. A magic state is equivalent (modulo Clifford operations) to a single-qubit gate $T = \mathrm{diag}(1, e^{i\pi/4})$. The Clifford+$T$ gate set is universal and has a rich algebraic structure enabling efficient and nearly optimal compiling of quantum algorithms.[59,60] Unfortunately, the overhead for distilling high-fidelity magic states is prohibitively large. O'Gorman and Campbell[61] performed a careful examination of available distillation methods and their overhead, considering the implementation of a logical Clifford+T circuit of size N with an overall fidelity of 90%. Assuming a physical error rate of 10^{-3}, the following fitting formulas were found for the space-time volume [(physical qubits) × (syndrome measurement cycles)] associated with a single logical gate:

Logical gate	Physical space-time volume
CNOT	$1610 + 45(\log_{10} N)^{2.77}$
T-gate	$3.13 + 3220(\log_{10} N)^{3.20}$

The space-time volume roughly quantifies the number of physical gates required to implement a single logical gate.

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

As an example, consider the Heisenberg benchmark problem described above with 100 logical qubits. The desired time evolution operator can be approximated using about 10^7 CNOTs and single-qubit gates (see Fig. 1). However, each single-qubit gate needs to be compiled using the logical-level gate set $\{H, S, T\}$. In total, this requires roughly 10^9 T-gates and a comparable number of Clifford gates.[7] Accordingly, the physical space-time volumes of a single logical CNOT and T-gate are roughly 2×10^4 and 4×10^6, respectively. (In fact, this underestimates the error correction overhead since the Heisenberg benchmark problem requires logical circuit fidelity 0.999 rather than 0.9, as considered in Ref. 61.)

The large overhead associated with logical non-Clifford gates may rule out the near-term implementation of error-corrected quantum circuits, even if fully functioning logical qubits based on the surface code become available soon. There have been several strategies proposed recently for reducing this overhead, including high-yield magic state distillation methods,[62,63] better strategies for preparing "raw" noisy magic states that reduce the required number of distillation rounds,[64] and better surface code implementations of distillation circuits.[65–68] A recent breakthrough result by Benjamin Brown[69] showed how to realize a logical non-Clifford gate CCZ (controlled-controlled-Z) in the 2D surface code architecture without resorting to state distillation. This approach relies on the fact that a 3D version of the surface code enables an easy (transversal) implementation of a logical CCZ[134,135] and a clever embedding of the 3D surface code into a $2 + 1$ dimensional spacetime. It remains to be seen whether this method is competitive compared with magic state distillation.

B. Error mitigation

Although error correction is vital for realizing large-scale quantum algorithms with great computational power, it may be overkill for small or medium size computations. A limited form of correction for shallow quantum circuits can be achieved by combining the outcomes of multiple noisy quantum experiments in a way that cancels the contribution of noise to the quantity of interest.[10,11] These methods, collectively known as error mitigation, are well suited for the QPUs available today because they introduce little to no overhead in terms of the number of qubits and only a minor overhead in terms of extra gates. However, error mitigation comes at the cost of an increased number of circuits (experiments) that need to be executed. In general, this will result in an exponential overhead; however, the base of the exponent can be made close to one with improvements in hardware and control methods, and each experiment can be run in parallel. Furthermore, known error mitigation methods apply only to a restricted class of quantum algorithms that use the output state of a quantum circuit to estimate the expected value of observables.

Probabilistic error cancellation (PEC)[10,70] aims to approximate an ideal quantum circuit via a weighted sum of noisy circuits that can be implemented on a given quantum computer. The weights assigned to each noisy circuit can be computed analytically if the noise in the system is sufficiently well characterized or learned by mitigating errors on a training set of circuits that can be efficiently simulated classically.[71] We expect that the adoption of PEC will grow due to the recent theoretical and experimental advances in quantum noise metrology.[72–74] For example, Ref. 74 shows how to model the action of noise associated with a single layer of two-qubit gates by a Markovian dynamics with correlated Pauli errors. This model can be described by a collection of single-qubit and two-qubit Pauli errors $P_1, \ldots, P_m$ and the associated error rate parameters $\lambda_1, \ldots, \lambda_m \geq 0$ such that the combined noise channel acting on a quantum register has the form $\Lambda(\rho) = \exp[\mathcal{L}](\rho)$, where $\mathcal{L}$ is a Lindblad generator, $\mathcal{L}(\rho) = \sum_{i=1}^m \lambda_i (P_i \rho P_i^\dagger - \rho)$. The unknown error rates λ_i can be learned to within several digits of precision by repeating the chosen layer of gates many times and measuring the decay of suitable observables.[74] The error mitigation overhead (as measured by the number of circuit repetitions) per layer of gates scales as γ^2, where

$$\gamma = \exp\left(2\sum_{i=1}^{m} \lambda_i\right).$$

For a circuit composed of $d > 1$ layers, the error rates λ_i may be layer-dependent and have to be learned separately for each layer. As observed in Ref. 74, this model can approximate the actual hardware noise very well using only $m = O(n)$ elementary Pauli errors P_i supported on edges of the qubit connectivity graph, where n is the total number of qubits in the circuit. In general, the runtime for getting a noise-free estimate will depend on the circuit implemented and the noise model used.

A large class of quantum algorithms that can benefit from PEC is based on the so-called hardware-efficient circuits.[75] A depth-d hardware-efficient circuit consists of d layers of two-qubit gates such that all gates within the same layer are non-overlapping and couple qubits that are nearest-neighbors in the QPU connectivity graph. Denoting the average error rate per qubit $\bar{\lambda} = (1/n)\sum_{i=1}^m \lambda_i$, averaged over all d layers, the overall PEC overhead scales as $(\bar{\gamma})^{dn}$, where $\bar{\gamma} = \exp(4\bar{\lambda})$. This allows a simple formula for estimating the runtime, J, for a noise-free estimate from a quantum circuit of depth d and width n to be

$$\mathrm{J} = d(\bar{\gamma})^{dn}\beta, \tag{1}$$

where β is the average time to run a single layer of the circuit. One can view β as a measure of the "speed" and $\bar{\gamma}$ as a hardware-dependent parameter that quantifies the average "quality" of gates across the entire QPU.

For a spin chain of $n = 100$ qubits, the size of our benchmark problem, Fig. 2 shows the number of circuit instances that need to be sampled to perform PEC for 100 and 1000 Trotter steps using the decomposition in Fig. 3(a) of Ref. 74. Current hardware runs of up to 10^8 circuits daily (red dashed line) and error rates of 10^{-3} have been demonstrated. Hence, we anticipate that with a couple orders of magnitude improvement this becomes possible. Furthermore, this runtime can be further reduced with the quantum-centric supercomputing architecture that allows parallelized execution of quantum circuits.

We can also measure the quantity of interest at several different values of the noise rate and perform an extrapolation to the zero-noise limit.[10,11,76] This method cancels the leading-order noise

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

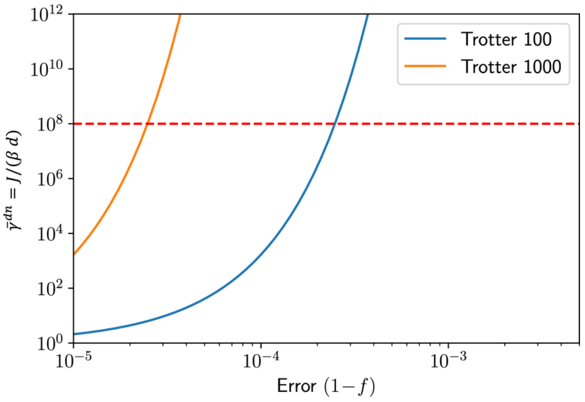

FIG. 2. Runtime scaling (number of circuit instances) needed to error mitigate 100 and 1000 Trotter steps in a circuit of 100 qubits and layers of non-overlapping two-qubit gates, each gate affected by a local depolarizing two-qubit error. The red dotted line identifies 100 million circuits, the daily limit assuming a repetition rate of 1 ms. So far the only architectures that have achieved this speed are solid-state based. The number of circuit instances dramatically decreases with slight improvements in the error rate of the physical gates.

contribution as long as the noise is weak and Markovian. Unlike PEC, this method is biased and heuristic but may require fewer circuits for the reconstruction. This method was recently demonstrated[77] to scale up to 27 qubits and still reconstruct observables. Whether this method can be combined with PEC, which gives an unbiased estimation, remains an open question.

More general (non-Markovian) noise can be mitigated using the virtual distillation technique.[78,79] It works by combining two copies of a noisy output state ρ in a way that enables measurements of observables on a state $\rho^2/\mathrm{Tr}(\rho^2)$. Assuming that ρ has most of its weight on the ideal output state, virtual distillation can quadratically suppress the contributions of errors. However, this method introduces at least a factor of two overhead in the number of qubits and gates. A review of the existing error mitigation proposals can be found in Endo *et al.*[80]

We anticipate error mitigation to continue to be relevant when error-corrected QPUs with a hundred or more logical qubits become available. As discussed in Sec. II A, the first generation of error-corrected quantum chips based on 2D stabilizer codes may not be able to execute universal computations. Such QPUs are likely to offer only high-fidelity Clifford gates such as the Hadamard or CNOT, which can all be efficiently simulated on a classical computer. Meanwhile, logical non-Clifford gates such as the T-gate may remain out of reach due to the need to perform magic state distillation. This leads to the interesting possibility of combining error correction and mitigation. A concrete proposal by Piveteau *et al.*[81] leverages the ability to realize noisy logical T-gates with fidelity comparable to or exceeding that of physical (unencoded) gates. Applying error mitigation protocols at the logical level to cancel errors introduced by noisy T-gates enables one to simulate universal logical circuits without resorting to state distillation. This may considerably reduce the hardware requirements for achieving a quantum advantage. However, error mitigation comes at the cost of an increased number of circuit executions. Assuming a physical gate fidelity of 99.9% and a budget of 1, 000 circuit executions, Piveteau *et al.*[81] estimate that logical Clifford+T circuits with about 2,000 T-gates can be realized reliably. This is far beyond the limit of existing classical algorithms that can simulate Clifford +T circuits with about 50 T-gates.[82,83] Similar ideas for combining error correction and mitigation are discussed in Refs. 84 and 85.

C. Circuit knitting

We can extend the scope of near-term hardware to compensate for other shortcomings such as a limited number of qubits or qubit connectivity by using circuit knitting techniques. This refers to the process of simulating small quantum circuits on a quantum computer and stitching their results into an estimation of the outcome of a larger quantum circuit. As was the case with error mitigation, known circuit knitting techniques apply to a restricted class of quantum algorithms that aim to estimate the expected value of observables.

The most well-known example is circuit cutting.[12–15] In this method, a large quantum circuit is approximated by a weighted sum of circuits consisting of small isolated sub-circuits. Each sub-circuit can be executed separately on a small QPU. The overhead introduced by this method (as measured by the number of circuit repetitions) scales exponentially with the number of two-qubit gates or qubit wires that need to be cut in order to achieve the desired partition of the circuit. Surprisingly, it was recently shown that the circuit cutting overhead can be substantially reduced by running the isolated sub-circuits in parallel using non-interacting

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

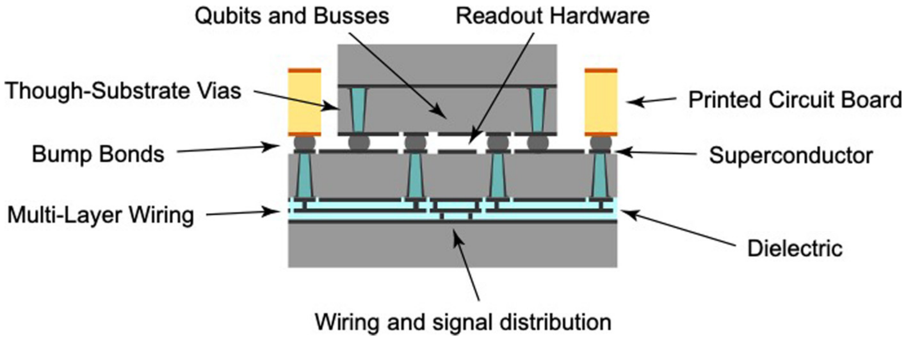

FIG. 3. An example of a scheme that allows breaking the plane for signal delivery compatible with the integration of hundreds of qubits. It is composed of technologies adapted from conventional CMOS processing.

Published under an exclusive license by AIP Publishing

QPUs that can only exchange classical data.[86] This approach requires hardware capable of implementing dynamic circuits (Dynamic circuits are computational circuits that combine quantum and classical operations, using the outcome of classical computations to adapt subsequent quantum operations. For more information, see Sec. IV, where the control electronics is extended to include independent QPUs.

A second example is entanglement forging,[87] where either an entangled variational state is decomposed into a weighted sum of product states or the entanglement between a pair of qubit registers is converted into time-like correlations within a single register.[88] The overhead of this method typically scales exponentially with the amount of entanglement across the chosen partition of the system.

A third example, closely related to circuit knitting, uses embedding methods to decompose the simulation of a large quantum many-body system into smaller subsystems that can be simulated individually on a QPU. The interactions between subsystems are accounted for by introducing an effective bath that could be either a classical environment or another small quantum system. The decomposition of the original system and the optimization of the bath parameters are performed on a classical computer that can exchange classical data with the QPU. Well-known examples of quantum embedding methods that build on their classical counterparts are dynamical mean-field theory,[89–91] density-matrix embedding,[92–94] and density-functional embedding.[95]

D. Heuristic quantum algorithms

Heuristic quantum algorithms can be employed near-term to solve classical optimization,[96] machine learning,[97] and quantum simulation[98] problems. These fall into two categories—algorithms that use kernel methods[97] and variational quantum algorithms (VQA). Quantum kernel methods have also been found that lead to provable speedups[99] and expand to a class of kernels for data with group structure.[100] For VQA, the basic proposal is appealingly simple: an experimentally controlled trial state is used as variational wavefunction to minimize the expected energy of a given quantum Hamiltonian or a classical cost function encoding the problem of interest. The trial state is usually defined as the output state of a shallow quantum circuit. Rotation angles that define individual gates serve as variational parameters. These parameters are adjusted via a classical feedback loop to optimize the chosen cost function.

At present, there is no mathematical proof that VQA can outperform classical algorithms in any task. In fact, it is known that VQA based on sufficiently shallow (constant depth) variational circuits with 2D or 3D qubit connectivity can be efficiently simulated on a classical computer.[101,102] This rules out a quantum advantage. Meanwhile, the performance of VQA based on deep variational circuits is severely degraded by noise.[103] However, as the error rates of QPUs decrease, we should be able to execute VQA in the intermediate regime where quantum circuits are already hard to simulate classically but the effect of noise can still be mitigated.

As a concrete example, let us discuss possible applications of VQA to the problem of simulating coherent time evolution of quantum spin chains. It is commonly believed[38] that approximating the time evolution operator e^{-iHt} for a Hamiltonian H describing a 1D chain of n qubits requires a quantum circuit of size scaling at least linearly with the space-time volume nt. Meanwhile, the best known rigorous quantum algorithms based on product formulas[39] or Lieb–Robinson bounds[38] require circuits of size $O(n^2t)$ or $O(nt \cdot \mathrm{polylog}(nt))$, respectively. A natural question is whether VQA can reduce the circuit size to what is believed to be optimal, that is, linear in nt. If this is indeed the case, a QPU with gate fidelity around 99.99% may be able to solve the classically hard problem instances described above with space-time volume $nt \sim 10^4$ using existing error mitigation techniques.

Variational quantum time evolution (VarQTE) algorithms, pioneered by Li and Benjamin,[11] could be an alternative to simulate the time evolution of these classically hard instances given near-term noisy QPUs. These algorithms aim to approximate the time-evolved state $|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle$ by a time-dependent variational ansatz $|\phi(\theta)\rangle = U(\theta)|0^n\rangle$, where $U(\theta)$ is a parameterized quantum circuit with a fixed layout of gates and $\theta = \theta(t)$ is a vector of variational parameters. The initial state $|\psi(0)\rangle$ is assumed to be sufficiently simple so that the variational ansatz for $|\psi(0)\rangle$ is easy to find. The goal is to find a function $\theta(t)$ such that the variational state $|\phi(\theta(t))\rangle$ approximates the time evolved state $|\psi(t)\rangle$ for all t in the chosen time interval. As shown in Ref. 11, the desired function $\theta(t)$ can be efficiently computed using the stationary-action principle with a Lagrangian $L(t) = \langle\phi(\theta(t))|d/dt + iH|\phi(\theta(t))\rangle$. This yields a first-order differential equation[98] $\sum_q M_{p,q}\dot{\theta}_q = V_p$ where

$$M_{p,q} = \mathrm{Im}(\langle\partial_p\phi(\theta)|\partial_q\phi(\theta)\rangle),$$

$$V_p = -\mathrm{Re}(\langle\partial_p\phi(\theta)|H|\phi(\theta)\rangle),$$

and $\partial_p \equiv \frac{\partial}{\partial\theta_p}$. As shown in Ref. 11, the entries of M and V can be efficiently estimated on a quantum computer. A comprehensive review of VarQTE algorithms can be found in Refs. 98 and 104.

The fact that VarQTE algorithms are heuristics and therefore lack rigorous performance guarantees raises the question of how to validate them. This becomes particularly important for large problem sizes where verifying a solution of the problem on a classical computer becomes impractical. Reference 105 recently developed a version of VarQTE based on McLachlan's variational principle that comes with efficiently computable bounds on the distance between the exact time-evolved state $|\psi(t)\rangle$ and the approximate variational state found by the VarQTE algorithms. Thus, although VarQTE lacks a rigorous justification, one may be able to obtain *a posteriori* bounds on its approximation error for some specific problems of practical interest.

E. Summary

To summarize, the Heisenberg chain example illustrates what we believe are general guidelines for designing near-term quantum algorithms.

First, our best chance of attaining a quantum advantage is by focusing on problems that admit an exponential (superpolynomial) quantum speedup. Even though a quantum algorithm that achieves such speedup with formal proof may be out of reach for near-term hardware, its mere existence serves as compelling

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

evidence that quantum-mechanical effects such as interference or entanglement are beneficial for solving the chosen problem.

Second, the only known way to realize large-scale quantum algorithms relies on quantum error-correcting codes. The existing techniques based on the surface code are not satisfactory due their poor encoding rate and high cost of logical non-Clifford gates. Addressing these shortcomings may require advances in quantum coding theory such as developing high-threshold fault-tolerant protocols based on quantum LDPC codes and improving the qubit connectivity of QPUs beyond the 2D lattice. Supplementing error correction with cheaper alternatives such as error mitigation and circuit knitting may provide a more scalable way of implementing high-fidelity quantum circuits.

Third, near-term quantum advantage should be possible by exploring less expensive, possibly heuristic versions of the algorithm considered. Those heuristic quantum algorithms lack rigorous performance guarantees, but they may be able to certify the quality of a solution *a posteriori* and offer a way to tackle problems that cannot be simulated classically.

We believe these general guidelines define the future of quantum computing theory and will guide us to important demonstrations of its benefits for the solution of scientifically important problems in the next few years.

III. THE PATH TO LARGE QUANTUM SYSTEMS

The perspective above leads to a challenge in quantum hardware. We believe there will be near-term advantage using a mixture of error mitigation, circuit knitting and heuristic algorithms. On a longer time frame, partially error-corrected systems will become critical to running more advanced applications and further down the line, fault-tolerant systems running on not-as-yet fully explored LDPC codes with non-local checks will be key. The first steps for all of these approaches are the same: we need hardware with more qubits capable of higher fidelity operations. We need tight integration of fast classical computation to handle the high run-rates of circuits needed for error mitigation and circuit knitting, and the classical overhead of the error correction algorithm afterwards. This drives us to identify a hardware path that starts with the early heuristic small quantum circuits and grows until reaching an error-corrected computer.

A. Cycles of learning

The first step in this path is to build systems able to demonstrate near-term advantage with error mitigation and limited forms of error correction. Just a few years ago, QPU sizes were limited by control electronics cost and availability, I/O space, quality of control software, and a problem referred to as "breaking the plane,"[106] i.e., routing microwave control and readout lines to qubits in the center of dense arrays. Today, solutions to these direct barriers to scaling have been demonstrated, which has allowed us to lift qubit counts beyond 100—above the threshold where quantum systems become intractably difficult to simulate classically and examples of quantum advantage become possible. The next major milestones are (1) increasing the fidelity of QPUs enough to allow exploration of quantum circuits for near-term quantum advantage with limited error correction and (2) improving qubit connectivity beyond 2D—either through modified gates, sparse connections with non-trivial topologies, and/or increasing the number of layers for quantum signals in 3D integration—to enable the longer term exploration of efficient non-2D LDPC error-correction codes. These developments are both required for our longer term vision, but can be pursued in parallel.

Work on improving the quality of quantum systems by improving gate fidelities involves many cycles of learning, trying coupling schemes, process changes, and innovations in controlling coupling and cross-talk. Scaling this work to large QPUs capable of demonstrating quantum advantage, and ultimately to the extreme system scales we anticipate in the distant future, involves integrating different technologies with enough reliability and skill to make size be limited by cost and need, not by technological capability. This adds challenges in reliability, predictability, and manufacturability of QPUs while continuing to incorporate improved technologies into these complex systems. Meanwhile, the increased development, fabrication, and test times for larger systems creates a lag in cycles of innovation that must be overcome.

The manufacturing cycle time increases with QPU sophistication. Many simple transmon QPUs require just a single level of lithography and can be easily fabricated in a day or two. Even the 5- and 16-qubit QPUs that were IBM's initial external cloud quantum systems involved only two lithography steps and took a week to fabricate. Compare this to more advanced packaging schemes like those at MIT Lincoln Laboratory[107–109] or IBM's newer "Eagle" QPUs (Fig. 3), which involve dozens of lithography steps and slow process steps, and take months to build at a research-style facility with one-of-a-kind tools. This increased cycle time makes it harder to reach the fidelities and coherence times needed as well as debug the manufacturing and assembly for reliable QPU yield.

Reliability in semiconductor manufacturing is not a new problem. In general, among the unique component challenges faced in building a scaled machine, the conventional semiconductor technologies integrated on chip are the most well studied. Incorporating them in superconducting technologies is more a matter of ensuring that the associated processes are compatible with each other than inventing new approaches. However, the rapid growth of volume we anticipate being needed is a major challenge.

Many failure modes in superconducting quantum systems are not detectable until the QPUs are cooled to their operating temperature, sub-100 mK. This is a severe bottleneck that renders in-line test (where a device sub-component is tested for key metrics before the QPU build finishes) and process feed-forward (where future process steps are modified to correct for small deviations in early steps and stabilize total device performance) difficult or impossible. There are exceptions where it is possible to tightly correlate an easy measurement at room temperature with ultimate QPU performance: for example, resistance measurements of Josephson junctions can accurately predict their critical currents and hence, the frequency of qubits made with them—a key parameter in fixed frequency systems. We can take advantage of these statistical correlations wherever they exist for rapid progress in parts of our process[110] or in post-process tuning.[111] However, reliably establishing these correlations requires measuring hundreds or thousands of devices, a nontrivial feat.

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

Absent these correlations, we can use simplified test vehicles; for example, rather than using the entire complicated signal delivery stack when trying to improve qubit coherence, we can use a simplified device designed to obtain good statistics and fast processing.[112] Still, identifying specific steps leading to increased coherence is nontrivial. It is rarely possible to change just one parameter in materials processing. Changing a metal in a qubit may also change etch parameters, chemicals compatible with the metal for subsequent processing, and even allowed temperature ranges.[113] Once an improved process is found, it is hard to identify exactly which steps were critical vs simply expedient.

We must gather sufficient statistics when performing materials research for the results be meaningful and provide enough certainty.[114] We should carefully document process splits wherever relevant, and we should publish changes in materials processes that lead to neutral or even negative results, not just just publish highly successful work.

Similar difficulties occur in non-material based research on devices. Some gates work well between pairs of qubits yet exhibit strong couplings that make them unsuitable for larger QPUs or compromise single-qubit performance. Three- and four-qubit experiments are no longer challenging from a technical or budgetary perspective. To be relevant to larger QPUs, research needs to move away from two-qubit demos, especially hero experiments between a single pair of qubits in which many critical defects can be masked by luck.

A mixture of long cycle-time complex devices and short cycle-time test vehicles for sub-process development and quantum operations is key to continuing improvements in the quality of QPUs and provides a recipe for continued R&D contributions as the largest QPUs begin to exceed the capabilities of smaller groups and labs. Nonetheless, reductions in long cycle times are needed. Some of this will come naturally—first-of-a-kind processes and QPUs usually take longer as they tend to include extra steps, inspections, and in-line tests that, while suggested by general best practices, may not be necessary. While counterproductive from a cost viewpoint, building the "same" QPU repeatedly to iron out manufacturing problems and speed up cycles of innovation will likely be a successful strategy for the largest QPUs with the most complex fabrication flows.

B. Supporting hardware

Scaling to larger systems also involves scaling classical control hardware and the input/output (I/O) chain in and out of the cryostat. This I/O chain, while still needing substantial customization for the exact QPU being controlled, consists of high volumes of somewhat more conventional devices; for example, isolators, amplifiers, scaled signal delivery systems, and more exotic replacements such as non-ferrite isolators and quantum limited amplifiers that may offer performance, cost, or size improvements. These components have enormous potential for being shared between various groups pursuing quantum computing, and in some instances can be purchased commercially already. However, assembling these systems at the scale required today, let alone a few years time, requires a high volume cryogenic test capability that does not currently exist in the quantum ecosystem, creating a short-term need for vertically integrated manufacturing of quantum systems. The challenge here is establishing a vendor and test ecosystem capable of scaled, low-cost production—a challenge made difficult by the fact that the demand is somewhat speculative.

There are also one-off components per system; for example, each quantum computer we deploy only requires a single dilution refrigerator, or in many cases a fraction thereof. The dilution refrigerator manufacturer effectively acts as a systems integrator for cryocoolers, wiring solutions, pumping systems, and even some active electronics. Maintaining the flexibility we need to change quickly as the systems scale will be most easily attainable if we can standardize many of these interfaces so that, for example, moving to a more scalable cooling technology at 4K doesn't require redesigning the entire refrigeration infrastructure.

Currently, each group building large QPUs has their own bespoke control hardware. Given the radically different control paradigms and requirements,[115–118] it is unlikely that the analog front-ends of these systems could ever be shared. However, there is a common need for sequencing logic (branching, local and nonlocal conditionals, looping) at low-cost and low-power for all types of quantum computers, not just solid-state. These will likely need to be built into a custom processor—an Application Specific Integrated Circuit or ASIC—as we scale to thousands of qubits and beyond. On top of this, the software that translates a quantum circuit into the low-level representation of this control hardware is becoming increasingly complex and expensive to produce. Reducing cost favors a common control platform with customized analog front ends. Open-specification control protocols like OpenQASM3[119] are already paving the way for this transformation.

C. Classical parallelization of quantum processors

Reaching near-term quantum advantage will require taking advantage of techniques like circuit knitting and error mitigation that effectively stretch the capabilities of QPUs—trading off additional circuit executions to emulate more qubits or higher fidelities. These problems can be pleasingly parallel, where individual circuits can execute totally independently on multiple QPUs, or may benefit from the ability to perform classical communication between these circuits that span multiple QPUs. Introduction of control hardware that is able to run multiple QPUs as if they were a single QPU with shared classical logic, or split a single QPU into multiple virtual QPUs to allow classical parallelization of quantum workloads [Fig. 4(a)] is an important near-term technology for stretching this advantage to the limit. Longer term, these technologies will play a critical enabling role as we begin to build quantum systems that span multiple chips and multiple cryostats, i.e., modular quantum systems.

D. Modularity

The introduction of modular quantum systems will be key to bootstrapping ourselves from near-term quantum advantage toward long-term error-corrected quantum systems. These are systems with repeating unit cells that can be replaced if defective, with quantum links between the chips to entangle unit cells or perform remote gates. This approach simplifies QPU design and test, and allows us to scale quantum systems at will.

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

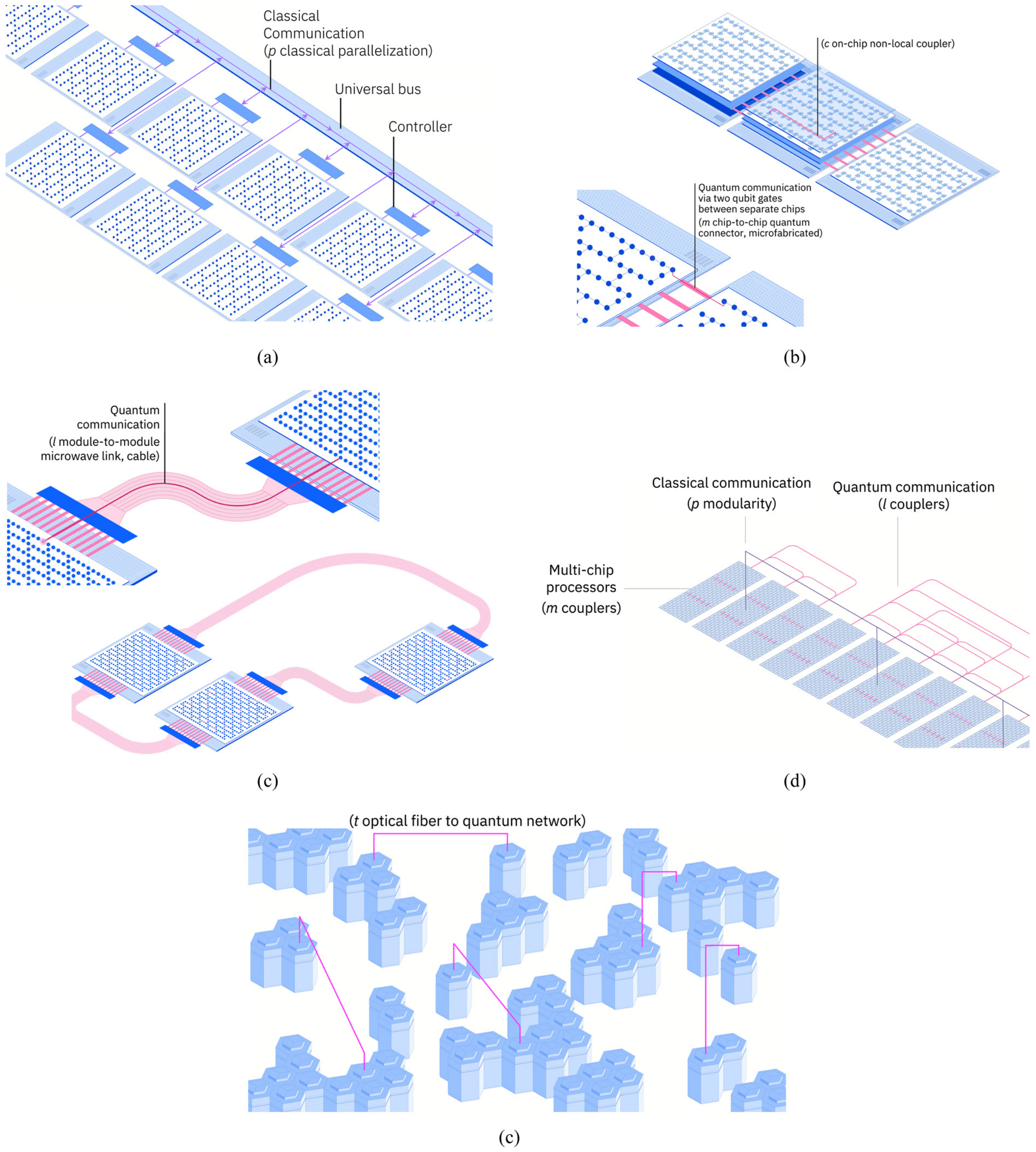

FIG. 4. Beyond classical parallelization of QPUs, shown in (a), long-range quantum connections carry a high penalty in gate speed and fidelity. As shown in (b)–(e), a high fidelity, large quantum system will likely involve three levels of modularity—a very short-range modularity *m* that allows breaking a QPU into multiple chips with minimal cost in gate speed and fidelity, a longer range connection *l* for use within a single cryogenic environment to both get around I/O bottlenecks and allow non-trivial topologies or routing, and a very long-range optical "quantum network" *t* to allow nearby QPUs to work together as a single quantum computational node (QCN). We will also need on-chip non-local couplers *c* as shown in (b) for the exploration of LDPC codes. In this figure, pink lines represent quantum communication and purple lines represent classical communication. (a) *p* type modularity for classical parallelization of QPUs, (b) Dense modularity *m* and on-chip non-local couplers *c* for LDPC codes for creating a single QPU from multiple chips, (c) Long-range *l* type modularity to enable quantum parallelization of multiple QPUs, (d) *l*, *m*, *p* schemes can be combined to extend the scale of hardware to thousands of qubits, (f) *t* type modularity involves microwave-to-optical transduction to link QPUs in different dilution refrigerators.

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

In the near term, given limited or no error correction, the unit cells will require high-bandwidth and high fidelity links to connect them—there is not enough time to use complex protocols such as entanglement distillation. The simplest proposals to accomplish this extend quantum busses off chip, allowing the same gates between distant chips as on a single processor.[120,121] This "dense modularity," which we denote m, effectively extends the chip size. This requires linking adjacent chips with ultra low loss, low cross-talk lines that are short enough to be effectively single-mode—the distance between chips has to be of the order of the distance between qubits on a single chip. Several technologies from classical computational hardware may be adaptable to this problem but adding the flexibility to replace individual units will require other alternatives.[122]

The high density of qubits in this "dense modularity" creates a spatial bottleneck for classical I/O and cooling. Proposals to ameliorate this near term include the development of high-density connectors and cables to route classical signals on and off the chip,[123,124] and the addition of time- and frequency-domain multiplexing of controls. A longer term approach to address this is to improve qubit connectivity through the use of a modified gate performed over a long conventional cable,[125–127] called l modularity. Beyond allowing us to escape control and cooling bottlenecks, these long-range couplers enable the realization of non-2D topologies, thereby not only reducing the average distance between qubits but also opening the door to the exploration of more efficient non-2D LDPC error correction codes.[128] Developing these long-range couplers thus not only allows us to scale our near-term systems, but begins to form the basis for how to build quantum systems with multiple QPUs.

The technologies that enable both dense modularity and long-range couplers, once developed and optimized, will ultimately be ported back into the qubit chip to enable non-local, non-2D connectivity. These on-chip non-local c couplers will ultimately allow implementation of high-rate LDPC codes, bringing our long-term visions to completion.

Finally, connecting multiple quantum computers in an *ad hoc* way will allow us to create larger systems as needed. In this "quantum networking" approach, the signals are typically envisioned to leave the dilution refrigerator, enabled by long-term technological advancements in microwave-to-optical transduction using photonic t links between different fridges.

With these four forms of modularity, we can redefine "scale" for a quantum system by

$$\mathrm{n} = ([(q\,m)\,l]t)p,$$

where n is the number of qubits in the entire modular and parallelized quantum system. The system is comprised of QPUs made from m chips, each QPU having $q \times m$ qubits. The QPUs can be connected with $l\,t$ quantum channels (quantum parallelization), with l of them being microwave connections and t optical connections. Finally, to enable things like circuit cutting and speeding up error mitigation, each of these multi-chip QPUs can support classical communication, allowing p classical parallelizations.

A practical quantum computer will likely feature all five types of modularity–classical parallelization, dense chip-to-chip extension of 2D lattices of qubits (m), sparse connections with non-trivial topology within a dilution refrigerator (l), non-local on-chip couplings for error correction (c), and long-range fridge-to-fridge quantum networking (t) (Table I and Fig. 4). The optimal characteristic size of each level of modularity is an open question. The individual "chip-to-chip" modules will still be made as large as possible, maximizing fidelity and connection bandwidth. Performing calculations on a system like this with multiple tiers of connectivity is still a matter of research and development.[129,130]

TABLE I. Types of modularity in a long-term scalable quantum system.

Type	Description	Use
p	Real-time classical communication	Classical parallelization of QPUs
m	Short range, high speed, chip-to-chip	Extend effective size of QPUs
l	Meter-range, microwave, cryogenic	Escape I/O bottlenecks, enabling multi-QPUs
c	On-chip non-local couplers	Non-planar error-correcting code
t	Optical, room-temperature links	*Ad hoc* quantum networking

Modularity needs to happen not just at the scale of the QPU, but at all levels of the system. Modular classical control systems allow for easy subsystem testing, replacement, and assembly. It's much easier to build a test infrastructure for a large number of small modules each year than a single, re-workable monolith. The same can be said of refrigeration, with the added benefit that shipping and deploying monolithic large refrigeration systems is impractical. A large number of our current failure points come in I/O and signal delivery, so modular solutions where sub-assemblies can be swapped out are essential. The challenge here is moving the replaceable unit from a single unit (a cable) to a larger unit (a flexible ribbon cable or other cable assembly).

While the jury is still out on module size and other hardware details, what is certain is that the utility of any quantum computer is determined by its ability to solve useful problems with a quantum advantage while its adoption relies on the former plus our ability to separate its use from the intricacies of its hardware and physics-level operation. Ultimately, the power provided by the hardware is accessed through software that must enable flexible, easy, intuitive programming of the machines.

IV. THE QUANTUM STACK

For quantum computing to succeed in changing what it means to compute, we need to change the architecture of computing. Quantum computing is not going to replace classical computing but rather become an essential part of it. We see the future of computing being a quantum-centric supercomputer where QPUs, CPUs, and GPUs all work together to accelerate computations. In integrating classical and quantum computation, it is important to identify (1) latency, (2) parallelism (both quantum and classical), and (3) what instructions should be run on quantum vs classical processors. These points define different layers of classical and quantum integration.

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

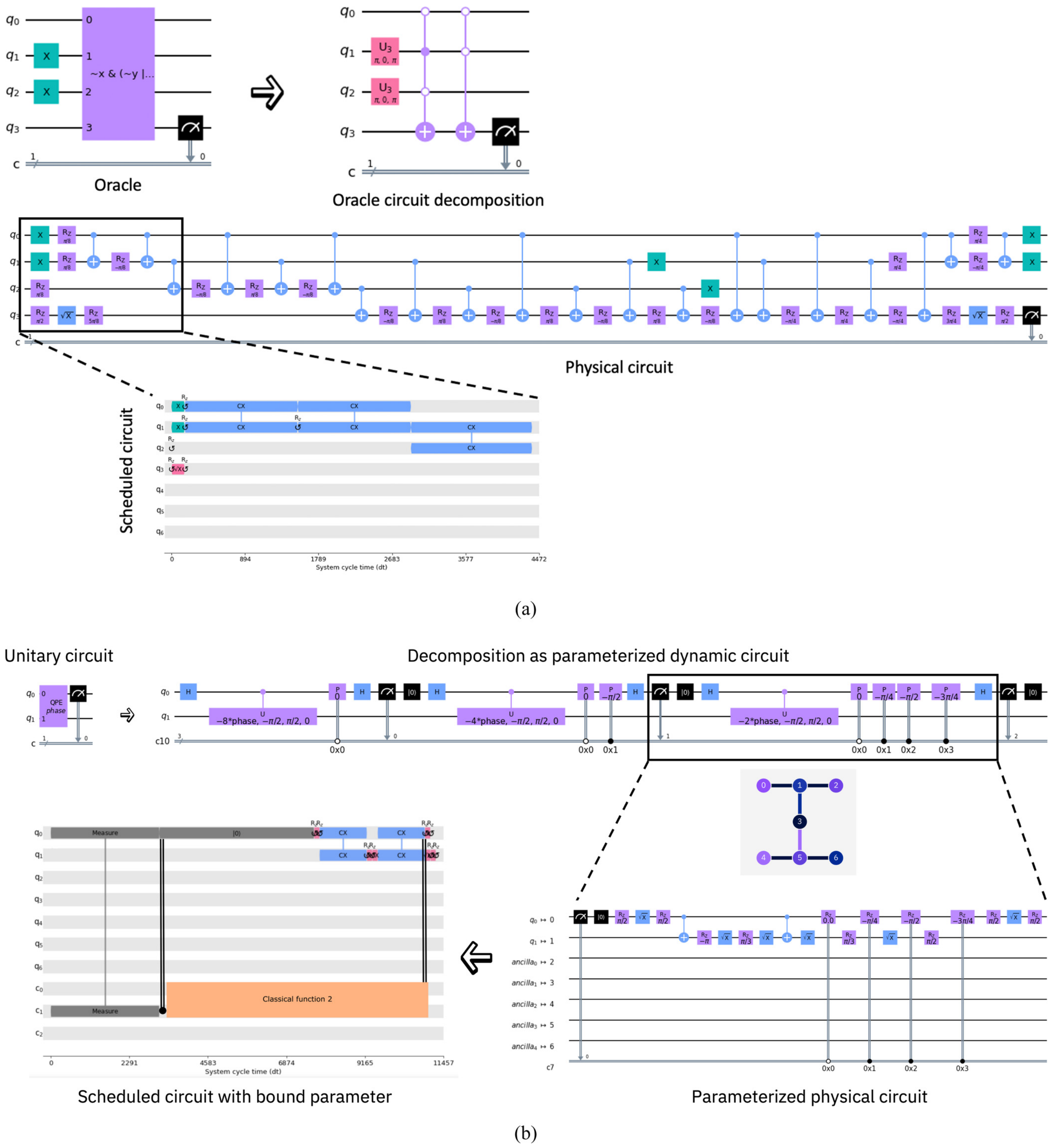

FIG. 5. Circuits can be represented at various levels. Unitary blocks represent circuits from libraries. These can be decomposed into parameterized circuits using the universal set of gates. Parameterized physical circuits use the physical gates supported by the hardware, while scheduled circuits specify timing, calibrations, and pulse shapes.

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

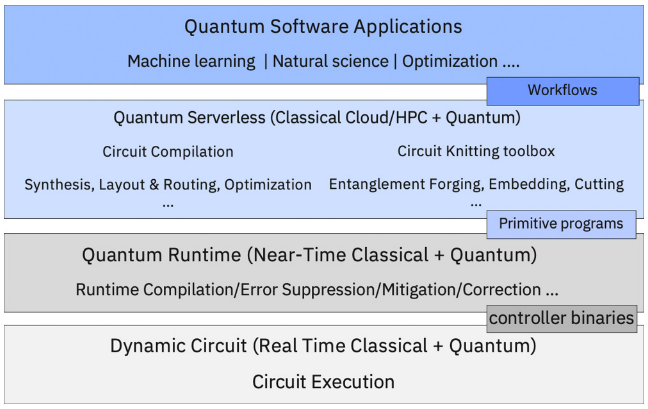
Quantum Software Applications
Machine learning | Natural science | Optimization ....
Workflows
Quantum Serverless (Classical Cloud/HPC + Quantum)
Circuit Compilation
Synthesis, Layout & Routing, Optimization
...
Circuit Knitting toolbox
Entanglement Forging, Embedding, Cutting
...
Primitive programs
Quantum Runtime (Near-Time Classical + Quantum)
Runtime Compilation/Error Suppression/Mitigation/Correction ...
controller binaries
Dynamic Circuit (Real Time Classical + Quantum)
Circuit Execution

FIG. 6. The quantum software stack is comprised of four layers, each targeting the most efficient execution of jobs at different levels of detail. The bottom layer focuses on the execution of quantum circuits. Above it, the quantum runtime efficiently integrates classical and quantum computations, executes primitive programs, and implements error mitigation or correction. The next layer up (quantum serverless) provides the seamless programming environment that delivers integrated classical and quantum computations through the cloud without burdening developers with infrastructure management. Finally, the top layer allows users to define workflows and develop software applications.

Before we go into the stack, we need to redefine a quantum circuit. Here we define a quantum circuit as follows:

A quantum circuit is a computational routine consisting of coherent quantum operations on quantum data, such as qubits, and concurrent (or real-time) classical computation. It is an ordered sequence of quantum gates, measurements, and resets, which may be conditioned on and use data from the real-time classical computation. If it contains conditioned operations, we refer to it as a dynamic circuit. It can be represented at different levels of detail, from defining abstract unitary operations down to setting the precise timing and scheduling of physical operations.

This is general enough to represent the circuit model,[131] the measurement model,[132] and the adiabatic model[133] of computation and special routines such as teleportation. Furthermore, it can represent the circuit at various levels: unitary (unitary block that could represent circuit libraries such as quantum phase estimation, classical functions, etc.), standard decomposition (reduced to a universal set of gates or expressing the classical functions as reversible gates), parameterized physical circuits (using the physical gates supported by the hardware, possibly including ancilla qubits not used in the circuit, or parameters that are easy to update in real-time), and scheduled circuits (complete timing information, calibrated gates, or gates with assigned pulse shape) (see Fig. 5). OpenQASM[119] is an example intermediate representation for this extended quantum circuit and can represent each of these various abstractions.

With this extended quantum circuit definition, it is possible to define a software stack. Figure 6 shows a high level view of the stack, where we have defined four important layers: dynamic circuits, quantum runtime, quantum serverless, and software applications. At the lowest level, the software needs to focus on executing the circuit. At this level, the circuit is represented by controller binaries that will be very dependent on the superconducting qubit hardware, supported conditional operations and logic, and the control electronics used. It will require control hardware that can move data with low latency between different components while maintaining tight synchronization. For superconducting qubits, real-time classical communication will require a latency of ∼100 ns. To achieve this latency, the controllers will be located very close to the QPU. Today, the controllers are built using FPGAs to provide the flexibility needed, but as we proceed to larger numbers of qubits and more advanced conditional logic, we will need ASICs or even cold CMOS.

We refer to the next level up as the quantum runtime layer. This is the core quantum computing layer. In the most general form, we expect a quantum computer to run quantum circuits and generate non-classical probability distributions at their outputs. Consequently, much of the workloads are sampling from or estimating properties of distributions. The quantum runtime thus needs to include at least two primitive programs: the sampler and the estimator. The sampler collects samples from a quantum circuit to reconstruct a quasi-probability distribution of the output. The estimator allows users to efficiently calculate expectation values of observables.

The circuit sent to the runtime would be a parameterized physical circuit. The software would perform a runtime compilation and process the results before returning the corrected outcome. The runtime compilation would update the parameters, add error suppression techniques such as dynamical decoupling, perform time-scheduling and gate/operation parallelization, and generate the controller code. It would also process the results with error mitigation techniques, and in the future, error correction. Both today's error mitigation and tomorrow's error correction will place strong demands on the classical computing needed inside these primitive programs. The circuit execution time could be as low as $100\,\mu$s (maybe even $1\,\mu$s for error correction), which is not possible over the cloud. It will need to be installed as part of the quantum computer. Fortunately, error mitigation is pleasingly parallel, thus using multiple QPUs to run a primitive will allow the execution to be split and done in parallel.

At the third level, we imagine software that can combine advanced classical calculations with quantum calculations. As described earlier in this paper, introducing classical computing can enable ideas such as circuit knitting. Here we need to be able to call quantum primitive programs as well as perform classical calculations such as circuit partitions. We call this a workflow (Fig. 7 shows examples of workflows for circuit knitting). We refer to quantum serverless as the software architecture and tooling that supports this in a way that allows developers to focus only on code and not on the classical infrastructure. Along with circuit knitting, this layer will also allow advanced circuit compiling that could include synthesis, layout and routing, and optimization—all of which are parts of the circuit reduction that should happen before sending the circuit to execute.

Finally, at the highest level of abstraction, the computing platform must allow users to efficiently develop software applications. These applications may need access to data and to resources not needed by the quantum computation itself but needed to provide the user an answer to a more general problem.

Each layer of the software stack we just described brings different classical computing requirements to quantum computing and defines a different set of needs for different developers. Quantum

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

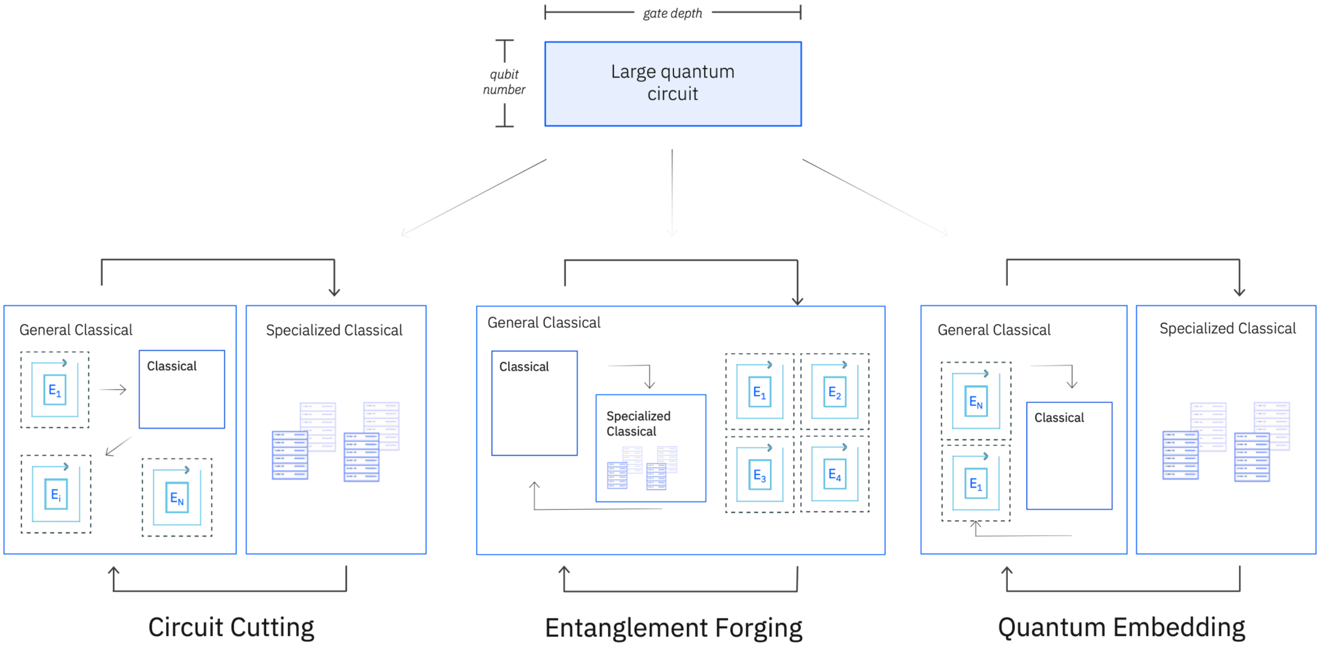

FIG. 7. Example of a quantum serverless architecture integrating quantum and classical computations. Quantum runtimes are illustrated by estimator primitives. Cloud computing is illustrated by general classical computing. Specialized classical computing such as high precision computing (HPC) or graphics processing units (GPUs) could be integrated into the serverless architecture. In circuit cutting, a larger circuit is split into many smaller circuits using a specialized classical computer. For each of the smaller circuits, an estimator primitive is executed ($E_1, \cdots, E_N$) and if needed, a classical computing routine could be used to condition future circuits on the results of previous estimators. The process can be repeated as needed. In entanglement forging, a 2N-qubit wavefunction is decomposed into a larger number of N-qubit circuits. The entanglement synthesis may need to be offloaded to specialized classical processors. For each N-qubit circuit, an estimator E_N is executed and combined to give the global outcome. This process could be repeated if used in a variational algorithm. Quantum embedding separates sub-parts of a problem that can be simulated classically from those computationally most costly and requiring quantum computations. A specialized classical computer could be used to condition the problem on previous outcomes. The quantum simulations employ estimators E_N running on QPUs. The estimators can condition quantum circuits on previous outcomes with classical calculations run on the general classical processors. Collectively, this set of tools allows larger systems to be simulated with higher accuracy.

computing needs to enable at least three different types of developers: kernel, algorithm, and model developers. Each developer creates the software, tools, and libraries that feed the layers above, thereby increasing the reach of quantum computing.

The kernel developer focuses on making quantum circuits run with high quality and speed on quantum hardware. This includes integrating error suppression, error mitigation, and eventually, error correction into a runtime environment that returns a simplified application programming interface (API) to the next layer.

The algorithm developer combines quantum runtime with classical computing, implements circuit knitting, and builds heuristic quantum algorithms and circuit libraries. The purpose is to

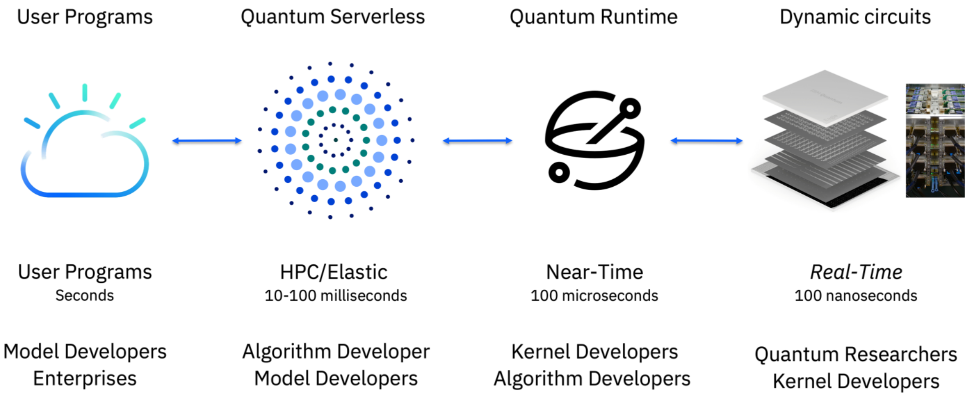

FIG. 8. The time scales and resources involved in quantum computing depend on the needs of the different types of developers and the level of abstraction at which they work. Quantum researchers and kernel developers work closer to the hardware while model developers require the highest level of software abstraction.

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

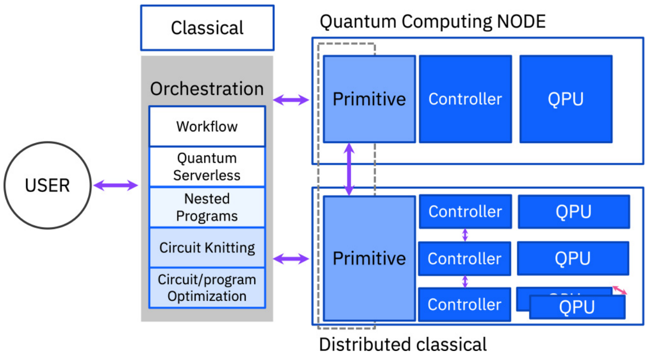

FIG. 9. Model of a cluster-like architecture integrating classical processors with QPUs to address latency, parallelization, and the distribution of instructions among classical and quantum processors. The darker the color, the lower the latency required.

enable quantum advantage. Finally, as we demonstrate examples of quantum advantage, the model developer will be able to build software applications to find useful solutions to complex problems in their specific domain, enabling enterprises to get value from quantum computing. Figure 8 summarizes the types of developers addressed by each layer of the software stack and the time scales involved depending on the type of job being executed and how close to the hardware each developer is working.

In putting all of this together and scaling to what we call a quantum-centric supercomputer, we do not see quantum computing integrating with classical computing as a monolithic architecture. Instead, Fig. 9 illustrates an architecture for this integration as a cluster of quantum computational nodes coupled to classical computing orchestration. The darker the color, the closer the classical and quantum nodes must be located to reduce latency. Threaded runtimes can execute primitives on multiple controllers. Classical communication in real time between the controllers can be used to enable things like circuit cutting. The figure also shows how future QPUs with quantum parallelization (l and t couplers) can be controlled by a single controller. We imagine that there could be workloads that need near-time classical communication (i.e., calculations based on the outcome of circuits that must complete in around $100\,\mu s$) or to share states between the primitives, enabled by a data fabric. Finally, the orchestration would be responsible for workflows, serverless, nested programs (libraries of common classical+quantum routines), the circuit knitting toolbox, and circuit compilation.

V. CONCLUSION

In conclusion, we have charted how we believe that quantum advantage in some scientifically relevant problems can be achieved in the next few years. This milestone will be reached through (1) focusing on problems that admit a super-polynomial quantum speedup and advancing theory to design algorithms—possibly heuristic—based on intermediate depth circuits that can outperform state-of-the-art classical methods, (2) the use of a suite of error mitigation techniques and improvements in hardware-aware software to maximize the quality of the hardware results and extract useful data from the output of noisy quantum circuits, (3) improvements in hardware to increase the fidelity of QPUs to 99.99% or higher, and (4) modular architecture designs that allow parallelization (with classical communication) of circuit execution. Error mitigation techniques with mathematical performance guarantees, like PEC, albeit carrying an exponential classical processing cost, provide a means to quantify both the expected run time and the quality of processors needed for quantum advantage. This is the near-term future of quantum computing.

Progress in the quality and speed of quantum systems will improve the exponential cost of classical processing required for error mitigation schemes, and a combination of error mitigation and error correction will drive a gradual transition toward fault-tolerance. Classical and quantum computations will be tightly integrated, orchestrated, and managed through a serverless environment that allows developers to focus only on code and not infrastructure. This is the mid-term future of quantum computing.

Finally, we have seen how realizing large-scale quantum algorithms with polynomial run times to enable the full range of practical applications requires quantum error correction, and how error correction approaches like the surface code fall short of the long term needs owing to their inefficiency in implementing non-Clifford gates and poor encoding rate. We outlined a way forward provided by the development of more efficient LDPC codes with a high error threshold, and the need for modular hardware with non-2D topologies to allow the investigation of these codes. This more efficient error correction is the long-term future of quantum computing.

ACKNOWLEDGMENTS

We thank Kristan Temme, Abhinav Kandala, Ewout van den Berg, Jerry Chow, Antonio Córcoles, Ismael Faro, Blake Johnson, Tushar Mittal, and Matthias Steffen for their assistance reviewing this manuscript.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Sergey Bravyi: Conceptualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Oliver Dial:** Conceptualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Jay M. Gambetta:** Conceptualization (equal); Writing – original draft (equal); Writing – review & editing (equal). **Darío Gil:** Conceptualization (equal); Writing – review & editing (equal). **Zaira Nazario:** Conceptualization (equal); Writing – original draft (equal); Writing – review & editing (equal).

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

REFERENCES

[1] See https://newsroom.ibm.com/2021-05-06-IBM-Unveils-Worlds-First-2-Nanometer-Chip-Technology-Opening-a-New-Frontier-for-Semiconductors for "IBM unveils world's first 2 nanometer chip technology, opening a new frontier for semiconductors."

[2] A. Reuther, P. Michaleas, M. Jones, V. Gadepally, S. Samsi, and J. Kepner, "Survey of machine learning accelerators," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)* (IEEE, 2020), pp. 1–12.

[3] D. S. Abrams and S. Lloyd, "Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors," Phys. Rev. Lett. **83**, 5162–5165 (1999).

[4] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," Phys. Rev. Lett. **103**, 150502 (2009).

[5] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science* (IEEE, 1994), pp. 124–134.

[6] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, STOC '96 (ACM, New York, 1996), pp. 212–219.

[7] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, "Toward the first quantum simulation with quantum speedup," Proc. Natl. Acad. Sci. USA **115**, 9456–9461 (2018).

[8] A. Y. Kitaev, "Fault-tolerant quantum computation by anyons," Ann. Phys. **303**, 2–30 (2003).

[9] S. B. Bravyi and A. Y. Kitaev, "Quantum codes on a lattice with boundary," quant-ph/9811052 (1998).

[10] K. Temme, S. Bravyi, and J. M. Gambetta, "Error mitigation for short-depth quantum circuits," Phys. Rev. Lett. **119**, 180509 (2017).

[11] Y. Li and S. C. Benjamin, "Efficient variational quantum simulator incorporating active error minimization," Phys. Rev. X **7**, 021050 (2017).

[12] S. Bravyi, G. Smith, and J. A. Smolin, "Trading classical and quantum computational resources," Phys. Rev. X **6**, 021043 (2016).

[13] T. Peng, A. W. Harrow, M. Ozols, and X. Wu, "Simulating large quantum circuits on a small quantum computer," Phys. Rev. Lett. **125**, 150504 (2020).

[14] W. Tang, T. Tomesh, M. Suchara, J. Larson, and M. Martonosi, "Cutqc: Using small quantum computers for large quantum circuit evaluations," in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (Association for Computing Machinery, New York, NY, 2021), pp. 473–486.

[15] K. Mitarai and K. Fujii, "Constructing a virtual two-qubit gate by sampling single-qubit operations," New J. Phys. **23**, 023021 (2021).

[16] D. Gottesman, "Fault-tolerant quantum computation with constant overhead," arXiv:1310.2984 (2013).

[17] N. P. Breuckmann and J. N. Eberhardt, "Quantum low-density parity-check codes," PRX Quantum **2**, 040101 (2021).

[18] N. Baspin and A. Krishna, "Quantifying nonlocality: How outperforming local quantum codes is expensive," arXiv:2109.10982 (2021).

[19] S. Lloyd, "Universal quantum simulators," Science **273**, 1073–1078 (1996).

[20] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," Phys. Rev. Lett. **103**, 150502 (2009).

[21] S. Lloyd, S. Garnerone, and P. Zanardi, "Quantum algorithms for topological and geometric analysis of data," Nat. Commun. **7**, 1–7 (2016).

[22] C. Gyurik, C. Cade, and V. Dunjko, "Towards quantum advantage via topological data analysis," arXiv:2005.02607 (2020).

[23] S. Ubaru, I. Y. Akhalwaya, M. S. Squillante, K. L. Clarkson, and L. Horesh, "Quantum topological data analysis with linear depth and exponential speedup," arXiv:2108.02811 (2021).

[24] D. Aharonov, V. Jones, and Z. Landau, "A polynomial quantum algorithm for approximating the Jones polynomial," Algorithmica **55**, 395–421 (2009).

[25] V. Giovannetti, S. Lloyd, and L. Maccone, "Quantum random access memory," Phys. Rev. Lett. **100**, 160501 (2008).

[26] R. Feynman, "Simulating physics with computers," Int. J. Theor. Phys. **21**, 467–488 (1982).

[27] A. M. Kaufman, M. E. Tai, A. Lukin, M. Rispoli, R. Schittko, P. M. Preiss, and M. Greiner, "Quantum thermalization through entanglement in an isolated many-body system," Science **353**, 794–800 (2016).

[28] O. Shtanko and R. Movassagh, "Unitary subharmonic response and floquet Majorana modes," Phys. Rev. Lett. **125**, 086804 (2020).

[29] I. Aleiner, F. Arute, K. Arya, J. Atalaya, R. Babbush, J. C. Bardin, R. Barends, A. Bengtsson, S. Boixo, A. Bourassa *et al.*, "Accurately computing electronic properties of materials using eigenenergies," arXiv:2012.00921 (2020).

[30] K. G. Vollbrecht and J. I. Cirac, "Quantum simulators, continuous-time automata, and translationally invariant systems," Phys. Rev. Lett. **100**, 010501 (2008).

[31] D. Nagaj and P. Wocjan, "Hamiltonian quantum cellular automata in one dimension," Phys. Rev. A **78**, 032311 (2008).

[32] A. Kay, "Computational power of symmetric hamiltonians," Phys. Rev. A **78**, 012346 (2008).

[33] B. A. Chase and A. J. Landahl, "Universal quantum walks and adiabatic algorithms by 1D hamiltonians," arXiv:0802.1207 (2008).

[34] S. Bravyi, M. B. Hastings, and F. Verstraete, "Lieb-Robinson bounds and the generation of correlations and topological quantum order," Phys. Rev. Lett. **97**, 050401 (2006).

[35] T. J. Osborne, "Efficient approximation of the dynamics of one-dimensional quantum spin systems," Phys. Rev. Lett. **97**, 157202 (2006).

[36] U. Schollwöck, "The density-matrix renormalization group in the age of matrix product states," Ann. Phys. **326**, 96–192 (2011).

[37] M. Hastings, "Observations outside the light cone: Algorithms for nonequilibrium and thermal states," Phys. Rev. B **77**, 144302 (2008).

[38] J. Haah, M. B. Hastings, R. Kothari, and G. H. Low, "Quantum algorithm for simulating real time evolution of lattice hamiltonians," SIAM Journal on Computing, FOCS18-250 (2021).

[39] A. M. Childs, A. Ostrander, and Y. Su, "Faster quantum simulation by randomization," Quantum **3**, 182 (2019).

[40] D. W. Berry, C. Gidney, M. Motta, J. R. McClean, and R. Babbush, "Qubitization of arbitrary basis quantum chemistry leveraging sparsity and low rank factorization," Quantum **3**, 208 (2019).

[41] D. Gottesman, "Class of quantum error-correcting codes saturating the quantum Hamming bound," Phys. Rev. A **54**, 1862 (1996).

[42] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. Sloane, "Quantum error correction and orthogonal geometry," Phys. Rev. Lett. **78**, 405 (1997).

[43] R. Gallager, "Low-density parity-check codes," IRE Trans. Inform. Theory **8**, 21–28 (1962).

[44] J.-P. Tillich and G. Zémor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength," IEEE Trans. Inf. Theory **60**, 1193–1202 (2013).

[45] M.-H. Hsieh and F. Le Gall, "NP-hardness of decoding quantum error-correction codes," Phys. Rev. A **83**, 052331 (2011).

[46] K.-Y. Kuo and C.-C. Lu, "On the hardness of decoding quantum stabilizer codes under the depolarizing channel," in *2012 International Symposium on Information Theory and its Applications* (IEEE, 2012), pp. 208–211.

[47] P. Iyer and D. Poulin, "Hardness of decoding quantum stabilizer codes," IEEE Trans. Inf. Theory **61**, 5209–5223 (2015).

[48] R. Raussendorf and J. Harrington, "Fault-tolerant quantum computation with high threshold in two dimensions," Phys. Rev. Lett. **98**, 190504 (2007).

[49] A. G. Fowler, A. M. Stephens, and P. Groszkowski, "High-threshold universal quantum computation on the surface code," Phys. Rev. A **80**, 052312 (2009).

[50] D. S. Wang, A. G. Fowler, and L. C. Hollenberg, "Surface code quantum computing with error rates over 1%," Phys. Rev. A **83**, 020302 (2011).

[51] S. Bravyi, D. Poulin, and B. Terhal, "Tradeoffs for reliable quantum information storage in 2D systems," Phys. Rev. Lett. **104**, 050503 (2010).

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

[52] P. Panteleev and G. Kalachev, "Asymptotically good quantum and locally testable classical LDPC codes," in *STOC 2022: Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing* (Association for Computing Machinery, New York, NY, 2022), p. 375.

[53] H. Bombín, "Single-shot fault-tolerant quantum error correction," Phys. Rev. X **5**, 031043 (2015).

[54] A. Kubica and M. Vasmer, "Single-shot quantum error correction with the three-dimensional subsystem toric code," arXiv:2106.02621 (2021).

[55] E. T. Campbell, "A theory of single-shot error correction for adversarial noise," Quant. Sci. Technol. **4**, 025006 (2019).

[56] S. Bravyi and R. König, "Classification of topologically protected gates for local stabilizer codes," Phys. Rev. Lett. **110**, 170503 (2013).

[57] M. B. Hastings and J. Haah, "Dynamically generated logical qubits," Quantum **5**, 564 (2021).

[58] J. E. Moussa, "Transversal Clifford gates on folded surface codes," Phys. Rev. A **94**, 042316 (2016).

[59] V. Kliuchnikov, D. Maslov, and M. Mosca, "Fast and efficient exact synthesis of single-qubit unitaries generated by Clifford and T gates," Quant. Inform. Comput. **13**, 607 (2013).

[60] N. J. Ross and P. Selinger, "Optimal ancilla-free Clifford+T approximation of z-rotations," Quant. Inform. Comput. **16**, 901 (2016).

[61] J. O'Gorman and E. T. Campbell, "Quantum computation with realistic magic-state factories," Phys. Rev. A **95**, 032338 (2017).

[62] J. Haah, M. B. Hastings, D. Poulin, and D. Wecker, "Magic state distillation with low space overhead and optimal asymptotic input count," Quantum **1**, 31 (2017).

[63] M. B. Hastings and J. Haah, "Distillation with sublogarithmic overhead," Phys. Rev. Lett. **120**, 050504 (2018).

[64] Y. Li, "A magic state's fidelity can be superior to the operations that created it," New J. Phys. **17**, 023037 (2015).

[65] A. G. Fowler and S. J. Devitt, "A bridge to lower overhead quantum computation," arXiv:1209.0510 (2012).

[66] D. Litinski, "A game of surface codes: Large-scale quantum computing with lattice surgery," Quantum **3**, 128 (2019).

[67] A. Paetznick and A. G. Fowler, "Quantum circuit optimization by topological compaction in the surface code," arXiv:1304.2807 (2013).

[68] D. Litinski, "Magic state distillation: Not as costly as you think," Quantum **3**, 205 (2019).

[69] B. J. Brown, "A fault-tolerant non-Clifford gate for the surface code in two dimensions," Sci. Adv. **6**, eaay4929 (2020).

[70] S. Endo, S. C. Benjamin, and Y. Li, "Practical quantum error mitigation for near-future applications," Phys. Rev. X **8**, 031027 (2018).

[71] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin, and Y. Li, "Learning-based quantum error mitigation," PRX Quantum **2**, 040330 (2021).

[72] R. Harper, S. T. Flammia, and J. J. Wallman, "Efficient learning of quantum noise," Nat. Phys. **16**, 1184–1188 (2020).

[73] S. T. Flammia, "Averaged circuit eigenvalue sampling," arXiv:2108.05803 (2021).

[74] E. van den Berg, Z. Minev, A. Kandala, and K. Temme, "Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors," arXiv:2201.09866 (2022).

[75] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," Nature **549**, 242–246 (2017).

[76] A. Kandala, K. Temme, A. D. Córcoles, A. Mezzacapo, J. M. Chow, and J. M. Gambetta, "Error mitigation extends the computational reach of a noisy quantum processor," Nature **567**, 491–495 (2019).

[77] Y. Kim, C. J. Wood, T. J. Yoder, S. T. Merkel, J. M. Gambetta, K. Temme, and A. Kandala, "Scalable error mitigation for noisy quantum circuits produces competitive expectation values" (2021).

[78] W. J. Huggins, S. McArdle, T. E. O'Brien, J. Lee, N. C. Rubin, S. Boixo, K. B. Whaley, R. Babbush, and J. R. McClean, "Virtual distillation for quantum error mitigation," Phys. Rev. X **11**, 041036 (2021).

[79] B. Koczor, "Exponential error suppression for near-term quantum devices," Phys. Rev. X **11**, 031057 (2021).

[80] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, "Hybrid quantum-classical algorithms and quantum error mitigation," J. Phys. Soc. Jpn. **90**, 032001 (2021).

[81] C. Piveteau, D. Sutter, S. Bravyi, J. M. Gambetta, and K. Temme, "Error mitigation for universal gates on encoded qubits," Phys. Rev. Lett. **127**, 200505 (2021).

[82] S. Bravyi and D. Gosset, "Improved classical simulation of quantum circuits dominated by Clifford gates," Phys. Rev. Lett. **116**, 250501 (2016).

[83] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, "Simulation of quantum circuits by low-rank stabilizer decompositions," Quantum **3**, 181 (2019).

[84] M. Lostaglio and A. Ciani, "Error mitigation and quantum-assisted simulation in the error corrected regime," Phys. Rev. Lett. **127**, 200506 (2021).

[85] Y. Suzuki, S. Endo, K. Fujii, and Y. Tokunaga, "Quantum error mitigation for fault-tolerant quantum computing," PRX Quantum **3**, 010345 (2022).

[86] C. Piveteau and D. Sutter, "Circuit knitting with classical communication," arXiv:2205.00016 (2022).

[87] A. Eddins, M. Motta, T. P. Gujarati, S. Bravyi, A. Mezzacapo, C. Hadfield, and S. Sheldon, "Doubling the size of quantum simulators by entanglement forging," PRX Quantum **3**, 010309 (2022).

[88] P. Huembeli, G. Carleo, and A. Mezzacapo, "Entanglement Forging with generative neural network models," arXiv:2205.00933 (2022).

[89] B. Bauer, D. Wecker, A. J. Millis, M. B. Hastings, and M. Troyer, "Hybrid quantum-classical approach to correlated materials," Phys. Rev. X **6**, 031045 (2016).

[90] J. M. Kreula, L. García-'Alvarez, L. Lamata, S. R. Clark, E. Solano, and D. Jaksch, "Few-qubit quantum-classical simulation of strongly correlated lattice fermions," EPJ Quant. Technol. **3**, 1–19 (2016).

[91] S. Bravyi and D. Gosset, "Complexity of quantum impurity problems," Commun. Math. Phys. **356**, 451–500 (2017).

[92] G. Knizia and G. K.-L. Chan, "Density matrix embedding: A simple alternative to dynamical mean-field theory," Phys. Rev. Lett. **109**, 186404 (2012).

[93] G. Knizia and G. K.-L. Chan, "Density matrix embedding: A strong-coupling quantum embedding theory," J. Chem. Theory Comput. **9**, 1428–1432 (2013).

[94] L. Mineh and A. Montanaro, "Solving the Hubbard model using density matrix embedding theory and the variational quantum eigensolver," Phys. Rev. B **105**, 125117 (2022).

[95] H. Ma, M. Govoni, and G. Galli, "Quantum simulations of materials on near-term quantum computers," npj. Comput. Mater. **6**, 85 (2020).

[96] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem," arXiv:1412.6062 (2014).

[97] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," Nature **567**, 209–212 (2019).

[98] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, "Theory of variational quantum simulation," Quantum **3**, 191 (2019).

[99] Y. Liu, S. Arunachalam, and K. Temme, "A rigorous and robust quantum speed-up in supervised machine learning," Nat. Phys. **17**, 1013 (2021).

[100] J. R. Glick, T. P. Gujarati, A. D. Corcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, "Covariant quantum kernels for data with group structure," arXiv:2105.03406 (2021).

[101] S. Bravyi, D. Gosset, and R. Movassagh, "Classical algorithms for quantum mean values," Nat. Phys. **17**, 337–341 (2021).

[102] N. J. Coble and M. Coudron, "Quasi-polynomial time approximation of output probabilities of geometrically-local, shallow quantum circuits," in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, Piscataway, NJ, 2022), p. 598.

[103] D. Stilck França and R. García-Patrón, "Limitations of optimization algorithms on noisy quantum devices," Nat. Phys. **17**, 1221 (2021).

[104] S. Barison, F. Vicentini, and G. Carleo, "An efficient quantum algorithm for the time evolution of parameterized circuits," arXiv:2101.04579 (2021).

[105] C. Zoufal, D. Sutter, and S. Woerner, "Error bounds for variational quantum time evolution," arXiv:2108.00022 (2021).

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing

[106] J. M. Gambetta, J. M. Chow, and M. Steffen, "Building logical qubits in a superconducting quantum computing system," npj Quant. Inform. **3**, 1–7 (2017).

[107] D. R. W. Yost, M. E. Schwartz, J. Mallek, D. Rosenberg, C. Stull, J. L. Yoder, G. Calusine, M. Cook, R. Das, A. L. Day, E. B. Golden, D. K. Kim, A. Melville, B. M. Niedzielski, W. Woods, A. J. Kerman, and W. D. Oliver, "Solid-state qubits integrated with superconducting through-silicon vias," npj Quant. Inform. **6**, 59 (2020).

[108] S. K. Tolpygo, V. Bolkhovsky, T. J. Weir, L. M. Johnson, M. A. Gouker, and W. D. Oliver, "Fabrication process and properties of fully-planarized seep-submicron Nb/Al–AlO$_x$/Nb Josephson junctions for VLSI circuits," IEEE Trans. Appl. Supercond. **25**, 1 (2015).

[109] D. Rosenberg, S. Weber, D. Conway, D. Yost, J. Mallek, G. Calusine, R. Das, D. Kim, M. Schwartz, W. Woods, J. L. Yoder, and W. D. Oliver, "Solid-state qubits: 3D integration and packaging," IEEE Microw. Mag. **21**, 72 (2020).

[110] J. M. Kreikebaum, K. P. O'Brien, A. Morvan, and I. Siddiqi, "Improving wafer-scale Josephson junction resistance variation in superconducting quantum coherent circuits," Supercond. Sci. Technol. **33**, 06LT02 (2020).

[111] J. B. Hertzberg, E. J. Zhang, S. Rosenblatt, E. Magesan, J. A. Smolin, J.-B. Yau, V. P. Adiga, M. Sandberg, M. Brink, J. M. Chow, and J. S. Orcutt, "Laser-annealing Josephson junctions for yielding scaled-up superconducting quantum processors," npj Quant. Inform. **7**, 129 (2021).

[112] J. M. Gambetta, C. E. Murray, Y.-K.-K. Fung, D. T. McClure, O. Dial, W. Shanks, J. W. Sleight, and M. Steffen, "Investigating surface loss effects in superconducting transmon qubits," IEEE Trans. Appl. Supercond. **27**, 1–5 (2017).

[113] A. P. M. Place, L. V. H. Rodgers, P. Mundada, B. M. Smitham, M. Fitzpatrick, Z. Leng, A. Premkumar, J. Bryon, A. Vrajitoarea, S. Sussman, G. Cheng, T. Madhavan, H. K. Babla, X. H. Le, Y. Gang, B. Jäck, A. Gyenis, N. Yao, R. J. Cava, N. P. de Leon, and A. A. Houck, "New material platform for superconducting transmon qubits with coherence times exceeding 0.3 milliseconds," Nat. Commun. **12**, 1779 (2021).

[114] C. R. H. McRae, G. M. Stiehl, H. Wang, S. X. Lin, S. A. Caldwell, D. P. Pappas, J. Mutus, and J. Combes, "Reproducible coherence characterization of superconducting quantum devices," Appl. Phys. Lett. **119**, 100501 (2021).

[115] L. Geck, A. Kruth, H. Bluhm, S. van Waasen, and S. Heinen, "Control electronics for semiconductor spin qubits," Quant. Sci. Technol. **5**, 015004 (2019).

[116] X. Xue, B. Patra, J. P. G. van Dijk, N. Samkharadze, S. Subramanian, A. Corna, B. Paquelet Wuetz, C. Jeon, F. Sheikh, E. Juarez-Hernandez, B. P. Esparza, H. Rampurawala, B. Carlton, S. Ravikumar, C. Nieva, S. Kim, H.-J. Lee, A. Sammak, G. Scappucci, M. Veldhorst, F. Sebastiano, M. Babaie, S. Pellerano, E. Charbon, and L. M. K. Vandersypen, "CMOS-based cryogenic control of silicon quantum circuits," Nature **593**, 205–210 (2021).

[117] C. A. Ryan, B. R. Johnson, D. Ristè, B. Donovan, and T. A. Ohki, "Hardware for dynamic quantum computing," Rev. Sci. Instrum. **88**, 104703 (2017).

[118] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, C. Ryan-Anderson, and B. Neyenhuis, "Demonstration of the trapped-ion quantum CCD computer architecture," Nature **592**, 209–213 (2021).

[119] A. Cross, A. Javadi-Abhari, T. Alexander, N. de Beaudrap, L. S. Bishop, S. Heidel, C. A. Ryan, J. Smolin, J. M. Gambetta, and B. R. Johnson, "OpenQASM 3: A broader and deeper quantum assembly language," arXiv:2104.14722 (2021).

[120] A. Gold, J. Paquette, A. Stockklauser, M. J. Reagor, M. S. Alam, A. Bestwick, N. Didier, A. Nersisyan, F. Oruc, A. Razavi, B. Scharmann, E. A. Sete, B. Sur, D. Venturelli, C. J. Winkleblack, F. Wudarski, M. Harburn, and C. Rigetti, "Entanglement across separate silicon dies in a modular superconducting qubit device," arXiv:2102.13293 (2021).

[121] C. Conner, A. Bienfait, H.-S. Chang, M.-H. Chou, É. Dumur, J. Grebel, G. Peairs, R. Povey, H. Yan, Y. Zhong, and A. N. Cleland, "Superconducting qubits in a flip-chip architecture," Appl. Phys. Lett. **118**, 232602 (2021).

[122] M. P. Larsson and S. Lucyszyn, "A micromachined separable RF connector fabricated using low-resistivity silicon," J. Micromech. Microeng. **16**, 2021–2033 (2006).

[123] D. B. Tuckerman, M. C. Hamilton, D. J. Reilly, R. Bai, G. A. Hernandez, J. M. Hornibrook, J. A. Sellers, and C. D. Ellis, "Flexible superconducting Nb transmission lines on thin film polyimide for quantum computing applications," Supercond. Sci. Technol. **29**, 084007 (2016).

[124] D. J. Reilly, "Engineering the quantum-classical interface of solid-state qubits," npj Quant. Inform. **1**, 1–10 (2015).

[125] Y. Zhong, H.-S. Chang, A. Bienfait, É. Dumur, M.-H. Chou, C. R. Conner, J. Grebel, R. G. Povey, H. Yan, D. I. Schuster, and A. N. Cleland, "Deterministic multi-qubit entanglement in a quantum network," Nature **590**, 571–575 (2021).

[126] P. Kurpiers, P. Magnard, T. Walter, B. Royer, M. Pechal, J. Heinsoo, Y. Salathé, A. Akin, S. Storz, J.-C. Besse, S. Gasparinetti, A. Blais, and A. Wallraff, "Deterministic quantum state transfer and remote entanglement using microwave photons," Nature **558**, 264–267 (2018).

[127] N. Leung, Y. Lu, S. Chakram, R. K. Naik, N. Earnest, R. Ma, K. Jacobs, A. N. Cleland, and D. I. Schuster, "Deterministic bidirectional communication and remote entanglement generation between superconducting qubits," npj Quantum Information **5**, 1–5 (2019).

[128] E. T. Campbell, B. M. Terhal, and C. Vuillot, "Roads towards fault-tolerant universal quantum computation," Nature **549**, 172–179 (2017).

[129] N. H. Nickerson, Y. Li, and S. C. Benjamin, "Topological quantum computing with a very noisy network and local error rates approaching one percent," Nat. Commun. **4**, 1–5 (2013).

[130] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim, "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," Phys. Rev. A **89**, 022317 (2014).

[131] D. P. DiVincenzo, "Quantum computation," Science **270**, 255–261 (1995),

[132] M. A. Nielsen, "Quantum computation by measurement and quantum memory," Phys. Lett. A **308**, 96–100 (2003).

[133] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution," arXiv:0001106 (2000).

[134] A. Kubica, B. Yoshida, and F. Pastawski, "Unfolding the color code," New J. Phys. **17**, 083026 (2015).

[135] M. Vasmer and D. E. Browne, "Three-dimensional surface codes: Transversal gates and fault-tolerant architecture," Phys. Rev. A **100**, 012312 (2019).

Downloaded from http://pubs.aip.org/aip/jap/article-pdf/doi/10.1063/5.0082975/16515734/160902_1_online.pdf

Published under an exclusive license by AIP Publishing