frontiers | Frontiers in Physics

TYPE Review
PUBLISHED 05 August 2022
DOI 10.3389/fphy.2022.900099

OPEN ACCESS

EDITED BY
Alexander McCaskey,
Nvidia, United States

REVIEWED BY
Vicente Leyton Ortega,
Oak Ridge Associated Universities, United States
Bikas K. Chakrabarti,
Saha Institute of Nuclear Physics (SINP), India

*CORRESPONDENCE
Kaitlin N. Smith,
kns@uchicago.edu

SPECIALTY SECTION
This article was submitted to Quantum Engineering and Technology, a section of the journal Frontiers in Physics

RECEIVED 19 March 2022
ACCEPTED 27 June 2022
PUBLISHED 05 August 2022

CITATION
Smith KN, Ravi GS, Alexander T, Bronn NT, Carvalho ARR, Cervera-Lierta A, Chong FT, Chow JM, Cubeddu M, Hashim A, Jiang L, Lanes O, Otten MJ, Schuster DI, Gokhale P, Earnest N and Galda A (2022), Programming physical quantum systems with pulse-level control. *Front. Phys.* 10:900099. doi: 10.3389/fphy.2022.900099

Programming physical quantum systems with pulse-level control

Kaitlin N. Smith[1]*, Gokul Subramanian Ravi[1], Thomas Alexander[2], Nicholas T. Bronn[2], André R. R. Carvalho[3], Alba Cervera-Lierta[4,5], Frederic T. Chong[1], Jerry M. Chow[2], Michael Cubeddu[6], Akel Hashim[7], Liang Jiang[8], Olivia Lanes[2], Matthew J. Otten[9], David I. Schuster[8], Pranav Gokhale[10], Nathan Earnest[2] and Alexey Galda[11,12,13]

[1]Department of Computer Science, University of Chicago, Chicago, IL, United States, [2]IBM Quantum, IBM T. J. Watson Research Center, Yorktown Heights, NY, United States, [3]Q-CTRL, Los Angeles, CA, United States, [4]Department of Chemistry, University of Toronto, Toronto, ON, Canada, [5]Department of Computer Science, University of Toronto, ON, Toronto, Canada, [6]Aliro Technologies, Inc., Brighton, MA, United States, [7]Department of Physics, UC Berkeley, Chicago, IL, United States, [8]Pritzker School of Molecular Engineering, University of Chicago, Chicago, IL, United States, [9]HRL Laboratories, LLC, Malibu, CA, United States, [10]Super.tech (a division of ColdQuanta), Chicago, IL, United States, [11]Computational Science Division, Argonne National Laboratory, Lemont, IL, United States, [12]James Franck Institute, University of Chicago, Chicago, IL, United States, [13]Menten AI, Inc, San Francisco, CA, United States

Quantum information processing holds great potential for pushing beyond the current frontiers in computing. Specifically, quantum computation promises to accelerate the solving of certain problems, and there are many opportunities for innovation based on proposed applications in chemistry, engineering, finance, and more. To harness the full power of quantum computing, however, we must not only place emphasis on manufacturing better qubits, advancing our algorithms, and developing quantum software. We must also refine device-level quantum control to scale to the fault tolerant quantum regime. On May 17–18, 2021, the Chicago Quantum Exchange (CQE) partnered with IBM Quantum and Super.tech to host the Pulse-level Quantum Control Workshop. At the workshop, representatives from academia, national labs, and industry addressed the importance of fine-tuning quantum processing at the physical layer. This work summarizes the key topics of the Pulse-level Quantum Control Workshop for the quantum community at large.

KEYWORDS

quantum computing, quantum information science, quantum hardware, quantum optimal control, quantum computer architecture

1 Introduction

1.1 Quantum computing today

The present era of quantum computing is characterized by the emergence of quantum computers (QCs) with sizes ranging from dozens to over a hundred qubits. Although these devices are not fault tolerant, new algorithms exist that have innate noise resilience and modest qubit requirements. There are promising indications that near-term devices could be used to accelerate or enable solutions to problems in domains ranging from molecular chemistry [1] to combinatorial optimization [2] and machine learning [3].

Today's quantum computation falls into two categories: the analog approach and the digital approach. With the analog approach, devices such as adiabatic quantum machines [4] and quantum annealers [5] solve problems that are initially set up within the system's Hamiltonian. Quantum operations gradually evolve the quantum state in a smooth manner, and these continuous operations alter the system such that the information encoded in the final system after computation corresponds to the desired answer with high probability. Digital quantum computation, as the name suggests, solves problems in a more discrete manner. Digital quantum devices operate according to the universal, gate-based model by breaking up computation into a series of quantum logic gates. The set of basis gates used to express a quantum circuit or program for execution on a machine is often influenced by the underlying quantum hardware, and these gates, applied in carefully chosen sequences, are used to bring quantum bits, or qubits, closer to a desired solution with high probability.

Both the analog and digital models of quantum computation come with their own strengths and challenges. At a high level, digital QCs are often considered more general-use machines as compared to analog devices that are more frequently proposed for optimization problems. Gate-based modeling allows digital QCs to be applied to arbitrary quantum computations. It should be noted, however, that digital QCs are typically more sensitive to environmental noise than their analog counterparts. In this study, we focus on pulse-level control for digital, gate-based quantum computers.

1.2 Why take a pulse approach

The underlying evolution of a quantum system is continuous and so are the control signals. These continuous control signals offer much richer and more flexible controllability than the gate-level quantum instruction set architecture (ISA). The control pulses can drive the QC hardware to the desired quantum states by varying a system-dependent and time-dependent quantity called the Hamiltonian. The Hamiltonian of a quantum system is an operator corresponding to the total energy of the system. Thus, the system Hamiltonian determines the evolution path of the quantum states.

The ability to engineer the real-time system Hamiltonian allows us to navigate the quantum system to the quantum state of interest through generating accurate control signals. Quantum computation can be done by constructing a quantum system in which the system Hamiltonian evolves in a way that aligns with a QC task, producing the computational result with high probability upon final measurement of the qubits.

1.3 Pulse-level challenges

While the benefits to the pulse approach are clear, there are several challenges stemming from the inherent complexities of a full-stack approach, some of which are discussed below:

1.3.1 Machine Hamiltonian

Pulse-level optimization typically requires an extremely accurate model of the quantum system or machine, i.e. its Hamiltonian. Hamiltonians are difficult to measure experimentally and moreover, they drift significantly over time between daily recalibrations. Experimental quantum optimal control (QOC) papers incur considerable overhead associated with pre-execution calibration to address this issue.

1.3.2 Programming model

A traditional gate-level quantum programming model is simply an abstraction of the real quantum hardware execution in a form which is amenable to users familiar with classical programming models. Thus, execution of a program represented via the quantum circuit model requires translating circuit instructions to pulses which enact the desired state-transformations or measurements. This translation is often sub-optimal due to the heavy abstractions imposed across the software stack, resulting in a pulse-level instruction that closely approximates the ideal gate within some margin of error. Masking pulses with gate-level programming prevents exposing too much information to the end user, simplifying computation. Low-level quantum programming, if not done in a systematic manner, can considerably increase the complexity of an algorithm's specification.

1.3.3 Optimization and compilation overheads

Generating optimal pulses is an arduous task. Compilation and optimization overheads are especially prohibitive in applications such as variational quantum algorithms wherein the circuit and pulse construction process are part of the critical execution loop. This could potentially amount to several weeks of total compilation latency over the course of thousands of iterations.

1.3.4 Simulation

Quantum systems are dynamic and evolve with time. As a result, simulation tools for quantum mechanical systems need to take variation into consideration, whether that variation stems from intentional gate application or from unintentional drift or environmental coupling. If quantum systems are modeled with enough granularity, pulses required for low-level control can be developed.

1.4 Role of workshop and report

There are many challenges, as previously mentioned, associated with low-level quantum control. Overcoming these difficulties, however, could contribute to achieving quantum advantage in the near-term. Additionally, accelerated quantum processing could emerge as a significant benefit of custom architectures produced as a result of software-hardware co-design.

The Chicago Quantum Exchange (CQE) Pulse-level Quantum Control Workshop was organized with the intent of informing the broader community of the advantages of pulse-level quantum programming. Included sessions did not all specifically focus on pulse-level software or hardware optimization, but the role of low-level control in improving near-term QCs was an integral theme throughout the workshop. Talks featured many types of quantum technologies and information encodings, showing the potential for a wide range of improved control in the hardware-diverse quantum space. The purpose of this report is to summarize the topics discussed during the CQE workshop and to provide a pulse-level control reference for the quantum community at large.

2 Quantum information

Quantum information science (QIS) redefines the classical computational model through the use of a type of information that can hold many values at once. Most frequently, radix-2 or base-2 quantum computation is implemented within algorithms and quantum computer architectures. This type of quantum computation uses quantum bits, or qubits, that have two basis states represented as $|0\rangle = [\,1\ \ 0\,]^{\mathrm{T}}$ and $|1\rangle = [\,0\ \ 1\,]^{\mathrm{T}}$. Qubits, unlike classical bits that hold a static value of either 0 or 1, demonstrate states of superposition in the form of $\alpha_0|0\rangle + \alpha_1|1\rangle$ with probability amplitudes $\alpha_0, \alpha_1 \in \mathbb{C}$ such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$. Superposition enables n qubits to represent states in 2^n-dimensional Hilbert Space, and this phenomenon, along with the ability for quantum states to interfere and become entangled, allow certain problems to be solved with significant reductions in complexity. Qubits hold large quantities of information for processing while in superposition, but upon measurement, the quantum state collapses; only classical values of either 0 or 1 are observed.

Radix-d computation where $d > 2$ is seen occasionally in classical systems, particularly in domain-specific applications. However, the benefits of high-dimensional encoding are often outweighed by the advantages provided through the continuous scaling of bistable transistors. Similarly, higher-dimensional quantum computation is not infeasible and trade-offs associated with its application are actively being explored. The qudit, or quantum digit, is the multi-level quantum unit that can be used as an alternative to the base-2 qubit. A qudit is described by a d-dimensional vector and is written as

$$|\Psi\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle \tag{1}$$

where α_i values are the probability amplitudes corresponding to the basis states $|i\rangle$. As with the qubit, qudit probabilities must all sum to one.

Under the *no-cloning theorem*, unknown qubit and qudit states cannot be copied without destroying superposition. In other words, any attempt to duplicate quantum information essentially acts as a measurement operation, resulting in a basis state. As a result, quantum error correction and information storage methods that preserve state cannot be implemented like their classical analogs because classical processing often exploits the ability to efficiently copy data. The no-cloning theorem inflicts a serious architectural constraint for quantum information processing.

Quantum computation is implemented with operators or gates that cause the probability amplitudes associated with each basis in the quantum state to evolve. These operators are represented by a unitary transformation matrix, U, of size $d^n \times d^n$ where n is the number of radix-d units of quantum information that the operation transforms. Quantum operations are reversible since gates are unitary and

$$UU^{\dagger} = U^{\dagger}U = I. \tag{2}$$

The symbol † indicates a complex conjugate operation that creates the inverse of U, and I is the identity operation that preserves quantum state. Measurement is not reversible since it causes quantum states to collapse to classical information.

Quantum operations are necessary for qubits and qudits to demonstrate special quantum properties. Single-input operations can be used to create superpositions of basis states. If entangled states are desired, multi-qubit or multi-qudit gates must be available. When qubits or qudits are entangled, they act as an inseparable system where any action on one part of the system impacts the other(s).

Cascades of quantum operations create quantum circuits or algorithms. The set of gates employed by the circuit depends on the abstraction level used in the description. For example, higher-level circuits that are technology-independent may use complex, multi-qubit or qudit operators whereas lower-level circuits targeted for execution on quantum hardware will implement a

set of elementary single- and two-qubit or qudit gates. The set of basis gates used for a specific QC is technology-dependent as certain quantum platforms implement some gates more efficiently than others. A basis gate set usually consists of a set of single-qubit or qudit operations that implement arbitrary rotations within a small margin of error along with a multi-qubit or qudit operation, such as the radix-2 *CX* or *CZ* gates, to form a universal gate set.

3 Quantum algorithms

Search, simulation, optimization, and algebraic problems are all proposed for QCs, but the breadth of quantum applications will depend on the size and capability of available machines. The subset of computations, as well as their corresponding quantum kernels, for which QCs will promise an advantage is still being defined, and over time, it is likely that this class of problems will evolve. A key challenge for quantum researchers is developing efficient methods for categorizing problems for the best suited hardware, either classical or quantum, and then running subroutines derived from partitioned algorithms accordingly.

3.1 Variational algorithms

Variational quantum algorithms (VQA) provide an exciting opportunity for near-term machines to demonstrate quantum advantage. VQAs have a one-to-one mapping between logical qubits in algorithms to physical qubits in QCs. Physical qubits will be discussed more in Section 4.1. Proposed applications of VQAs include ground state energy estimation of molecules [6] and MAXCUT approximation [2]. VQAs are hybrid algorithms since they comprise of classical and quantum subroutines.

VQAs are well suited for near-term hardware since they adapt to the intrinsic noise properties of the QC they run on. The VQA circuit is parameterized by a vector of angles that correspond to gate rotations on qubits. The vector of angles is optimized by a classical optimizer during many iterations to either maximize or minimize an objective function that represents the problem that the VQA implementation hopes to solve.

Two notable VQAs are the Variational Quantum Eigensolver (VQE) and the Quantum Approximate Optimization Algorithm (QAOA). The former is often used for quantum chemistry while the latter is applied to combinatorial optimization.

3.2 Fault tolerant algorithms

Fault tolerant quantum algorithms offer exciting computational gains over their classical analogs, but they require extremely low error thresholds. Because of this, error correcting codes (ECCs) are required that enforce a one-to-many mapping between logical qubits in an algorithm to physical qubits on a QC. This encoding allows quantum information to be shielded from errors that arise due to uncontrolled coupling with the environment or due to imperfect operations at the physical level. There are many kinds of quantum ECCs [7].

At minimum, it is estimated that millions of high-quality physical qubits will be needed to implement fault tolerant quantum computation [8]. Once fault tolerance is reached, disruptive implementations of Shor's algorithm for quantum factoring [9], Grover's quantum database search [10], and quantum phase estimation [11] can be implemented.

4 Quantum hardware

4.1 Physical qubits

Even if the theory is well defined, information must have a physical realization for computation to occur. Currently, a variety of quantum technologies can encode logical qubits within different media. We call a physical implementation of a radix-2 unit of information a physical qubit. Since today's quantum devices lack error correction, each logical qubit within an algorithm is implemented with one physical qubit in a machine. These devices, sometimes called Noisy Intermediate Scale Quantum (NISQ) devices, are error prone and are up to hundreds of qubits in size [12]. Some physical qubit examples include energy levels within superconducting circuits [13–15], ions trapped by surrounding electrodes [16–18], neutral atoms held with optical tweezers [19, 20], and photons travelling through free space or waveguides [21, 22]. Each of these platforms have their unique strengths, but none has become the obvious choice for the standard quantum computing platform.

4.2 Physical architectural constraints

Although many promising quantum technologies exist, they are imperfect and demonstrate attributes that prevent scaling in the near-term. For example, quantum state preparation, gate evolution, and measurement all are characterized by nontrivial infidelity rates and must be implemented with a limited set of instructions that depend on the control signals that are available for a specific qubit technology. Additionally, restricted *connectivity* among device qubits is a key limiting factor. Qubit-qubit interaction is necessary for quantum entanglement, but unfortunately, many of today's QCs are limited in the number of qubits that are able to interact directly. Even with technologies that allow for all-to-all connectivity, such as with trapped ions, QCs are limited by the total number of qubits that are actively used during

computation [23]. Careful design of qubit layout on devices as well as intelligent algorithm mappers and compilers can assist with qubit communication overheads.

Qubit communication is required for quantum algorithms, but unintended communication, or *crosstalk*, can cause errors during computation. Crosstalk errors experienced during quantum algorithm execution often arise from simultaneously stimulating two neighboring qubits. If the qubits are strongly coupled and "feel" each other, the fidelity of the two simultaneous computations can decrease. Fortunately, qubit crosstalk errors are systematic. Once identified, sources of crosstalk can be mitigated with software [24, 25] or with improved device design [26].

Today's qubits are extremely sensitive, and as a result, crosstalk with neighboring qubits is not the only accidental signal interference that must be considered for quantum systems. Qubits also tend to couple with their surrounding environment, causing quantum information to *decohere*. Amplitude damping and dephasing cause qubit decoherence errors. Amplitude damping describes the loss of energy from $|1\rangle$ to $|0\rangle$, and T_1 is used to denote the exponential decay time from an excited qubit state to a ground state. Dephasing error refers to the gradual loss of phase between $|0\rangle$ to $|1\rangle$ and is described by the time T_2. Quantum decoherence sets rigid windows for compute time on current QCs. If the qubit runtime, or period spanning the first gate up until measurement, surpasses the QC T_1 or T_2 time, the final result of computation may resemble a random distribution rather than the correct output.

4.3 Superconducting circuits-A case study on progress

Superconducting circuits have emerged as a leading physical qubit implementation [27]. IBM has dedicated much research to the development of the Transmon-based superconducting QC, and since 2016, the company has allowed their prototype devices to be used by the public via the IBM Quantum Experience [28]. Since 2016, IBM QCs have increased from 5 to 127 qubits. The significance of increasing the number of qubits on-chip to a total of n is that more complex algorithms can be run that explore an exponentially larger state space, dimension 2^n. Although the debut of larger QCs with over a hundred qubits is an impressive engineering development, limited fidelity, qubit communication, and coherence times prohibit these devices from reliably executing circuits that require all of a device's physical qubits to work in synergy.

Progress in quantum hardware is not measured simply by the consistency and count of device qubits. The gate error must also be considered as it influences the depth of circuits that a QC can run. IBM QCs have seen the average two-qubit infidelity decrease in magnitude from order 10^{-2} to 10^{-3} over the last five years. Quality of multi-qubit gates is an important metric to consider when rating a QC because it enables the entanglement required for many quantum algorithms to demonstrate advantage. The importance of device size, qubit quality, and operation success is clear, thus, improving quantum systems as a whole requires a multi-targeted approach. To assist with scaling into more robust devices, IBM developed a metric referred to as Quantum Volume (QV) that aims to benchmark quantum hardware in the near term. QV is a scalar value, where higher is better, that assesses the largest random circuit of equal width and depth equal to m that a QC can successfully run [29]. As QV = 2^m, the metric provides perspective on system performance as a general purpose QC. Capability in terms of coherence, calibration, state initialization, gate fidelity, crosstalk, and measurement quality are all captured by QV. QV is also influenced by design aspects such as chip topology, compilation tools, and basis gate set.

Recently, IBM extended the QC frontier to achieve a QV of 64, and then later 128, through the combined application of improved compiler optimizations, shorter two-qubit gates, excited state promoted readout, higher-accuracy state discrimination and open-loop correction via dynamical decoupling [15, 30]. These breakthroughs were accomplished on 27 qubit devices, demonstrating that the compute power of a quantum system is heavily influenced on compilation and low-level control; it does not depend on qubit count alone. Quantum compilation and low-level control will be discussed further in Section 5. By focusing research to innovate in these domains, along with making developments in other areas of QC architecture, IBM anticipates to scale QCs to one million qubits and beyond [31].

5 Programming for pulse-level control

To target real-world quantum use cases, efficient programming languages and supporting software tool-flows are required to represent complex classical and quantum information processing in quantum algorithms and then efficiently execute them both in simulation and on real quantum devices. To build such an *efficient* software stack, balancing between abstraction and detail is key. On the one hand, a transparent software stack that exposes device specifics helps programmers write tailored code, but on the other hand it dramatically increases the complexity of the toolflow. Considering this trade-off, advancements in building an efficient software stack have been a critical driver in pushing the field of quantum computing beyond the laboratory. Managing these trade-offs is particularly important for pulse-level control because it inherently requires programmer exposure to the finer details of the device. In this Section, we discuss state-of-the-art tools and models for quantum programming and device execution.

Quantum programming languages are designed to be user-friendly, with sophisticated control flow, debugging tools, and strong abstraction barriers between target operations and the underlying quantum hardware. Operations are thought of as "black-box" in the sense that the details of the physical implementation of quantum gates are hidden from the end user. This allows for modularity since a technology-independent quantum program written at the gate level can be compiled for execution on multiple QCs of different qubit types.

The most successful languages have been implemented as Python packages, such as IBM's Qiskit [32], Google's Cirq [33], Rigetti's PyQuil [34] and Xanadau's Strawberry Fields [35]. Others are written as entirely new languages, such as Scaffold [36] which is based on LLVM infrastructure, Quipper [37] which is a functional language embedded in Haskell, and Q#Mic [38] which is Microsoft's quantum domain specific language.

Here, we restrict ourselves to Qiskit and provide an overview of Qiskit's programming model and its support for pulse-level control. We encourage readers to further references on Qiskit [32] and its pulse support [39, 40]. Other programming frameworks and their device-level control can be reached via their corresponding webpages and documentation.

5.1 Qiskit and the pulse programming model

Qiskit is an open-source quantum computing framework from IBM that provides tools for creating, manipulating, and running quantum programs on quantum systems independent of their underlying technology and architecture.

The commonly used quantum programming paradigm is the circuit model. Such a model abstracts the physical execution of a quantum algorithm on a quantum system into a sequence of unitary gate operations on a set of qubits followed by qubit measurements. The gates manipulate the qubit states while measurements project these qubits onto a particular measurement basis that are extracted as classical bit-strings. Qiskit supports this programming model via a quantum assembly language called OpenQASM [41]. OpenQASM is simply an abstraction of the real quantum hardware execution in a form which is amenable to users familiar with classical programming models. The hardware is not capable of naively implementing or executing the quantum instructions from this model and must instead compose these operations via the control hardware.

At the device level, quantum system execution is implemented by steering the qubit(s) through a desired unitary evolution, which is achieved by careful engineering of applied classical control fields. Thus, execution of a program represented via the quantum circuit model on a quantum system requires translating or compiling gate-level circuit instructions to a set of microwave control instructions, or pulses, which enact the desired state-transformations or measurements. This translation is often suboptimal due to the heavy abstractions imposed across the software stack. In the circuit programming model, an atomic circuit instruction is agnostic to its pulse-level implementation on hardware, and unfortunately, the vast majority of program optimization is often done at the gate-level in the standard circuit model. Extracting the highest performance out of quantum hardware would require the ability to craft a pulse-level instruction schedule for the optimization of circuit partitions.

```
# Create a pulse schedule.
sched = Schedule(name='excited_state')

# Create gate and measurement pulses.
x180 = Drag(x_dur, x_amp, x_sigma, x_beta)
measure = GaussianSquare(m_dur, m_amp, m_sigma, m_square_width)

# += appends an Instruction to a Schedule.
sched += Play(x180, DriveChannel(0))

# Measure qubit 0.
sched += Play(measure, MeasureChannel(0))

# Determine the state of qubit 0 and store it
# in a persistent MemorySlot register which
# will be returned in the program result.
sched += Acquire(AcquireChannel(0), MemorySlot(0))

# Run the schedule and get the result.
counts = execute(sched, backend).result().get_counts()
```

FIGURE 1
A pulse schedule with deterministic instruction durations. This example prepares qubit 0 in the $|1\rangle$ state and then measures it. [40].

Qiskit Pulse [39, 40] was developed to describe quantum programs as a sequence of pulses, scheduled in time. Qiskit Pulse adds to the Qiskit compilation pipeline the capability to schedule a quantum circuit into a pulse program intermediate representation, perform analysis and optimizations, and then compile to Qiskit Pulse object code to execute on a quantum system.

To program such systems at the pulse-level in a hardware-independent manner requires the user-level instruction set to be target-compiled to the underlying system hardware components, each of which may have a unique instruction set and programming model. Qiskit Pulse provides a common and reusable suite of technology-independent quantum control techniques that operate at the level of an analog stimulus, which may be remotely re-targeted to diverse quantum systems.

Figure 1 shows an example of how a pulse schedule can be explicitly coded in Qiskit. While this is a trivial example, readers can refer to [40] for an example of implementing a high-fidelity *CX* gate based on the calibrated Cross-Resonance (CR) pulse. The CR gate is an important microwave-activated, two-qubit

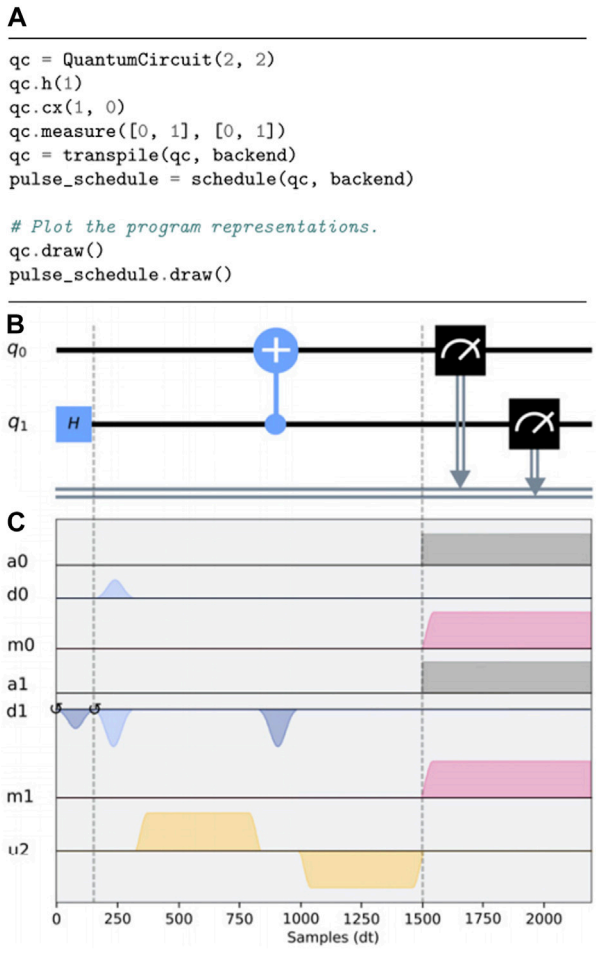

FIGURE 2
(A) Qiskit code to construct a quantum circuit that prepares and measures a Bell state and then schedules the circuit to produce an equivalent pulse schedule. **(B)** and **(C)** Visualization of the mapping between circuit instructions **(B)** and the composite pulse sequences that will implement the circuit elements **(C)**. [40].

entangling operation that is performed by driving one qubit, the control, by the frequency of another, the target [42].

Figure 2 shows provides a code snippet for scheduling a quantum circuit that prepares and measures a Bell state, or a maximally-entangled, two-qubit state, into a pulse schedule using Qiskit. Qiskit Pulse users may create pulse programs to replace the default pulse programs of the native gate set provided by the backend and pass them as an argument to the scheduler.

5.2 Pulse support *via* OpenQASM3

OpenQASM has become a *de facto* standard, allowing a number of independent tools to interoperate using OpenQASM as the common interchange format. While OpenQASM2 uses a circuit model which was described previously, quantum paradigms require going beyond the circuit model, incorporating primitives such as teleportation and the measurement model of quantum computing. OpenQASM3 [41] describes a broader set of quantum circuits with concepts beyond simple qubits and gates. Chief among them are arbitrary classical control flow, gate modifiers (e.g., control and inverse), timing, and microcoded pulse implementations.

To use the same tools for circuit development as well as for the lower-level control sequences needed for calibration, characterization, and error mitigation, it is necessary to control timing and to connect quantum instructions with their pulse-level implementations for various qubit modalities. This is critical for working with techniques such as dynamical decoupling [43, 44] as well as for better characterization of decoherence and crosstalk. These are all sensitive to time and can be programmed via the timing features in OpenQASM3. One such potential application compilation and execution flow is shown in Figure 3.

To control timing, OpenQASM3 introduces "delay" statements and "duration" types. The delay statement allows the programmer to specify relative timing of operations. Timing instructions can use the duration type which represents amounts of time measured in seconds.

OpenQASM3 includes features called "box" and "barrier" to constrain the reordering of gates, where the timing of those gates might otherwise be changed by the compiler. Additionally, without these directives, the desired gates could be removed entirely as a valid optimization on the logical level.

OpenQASM3 also allows specifying relative timing of operations rather than absolute timing. This allows more flexible timing of operations, which can be helpful in a setting with a variety of calibrated gates with different durations. To do so, it introduces a new type "stretch", representing a duration of time which is resolvable to a concrete duration at compile time once the exact durations of calibrated gates are known. This increases circuit portability by decoupling the circuit timing intent from the underlying pulses, which may vary from machine to machine or even from day to day.

OpenQASM3 also has added support for specifying instruction calibrations in the form of "defcal," short for 'define calibration,' declarations which allow the programmer to specify a microcoded implementation of a gate, measure, or reset instruction.

While only a few features of OpenQASM3 are discussed here, the overall design intends to be a multilevel intermediate representation (IR), where the focus shifts from target-agnostic computation to a concrete implementation as more hardware specificity is introduced. An OpenQASM circuit can also mix different abstraction levels by introducing constraints where needed, but allowing the compiler to make decisions where there are no constraints.

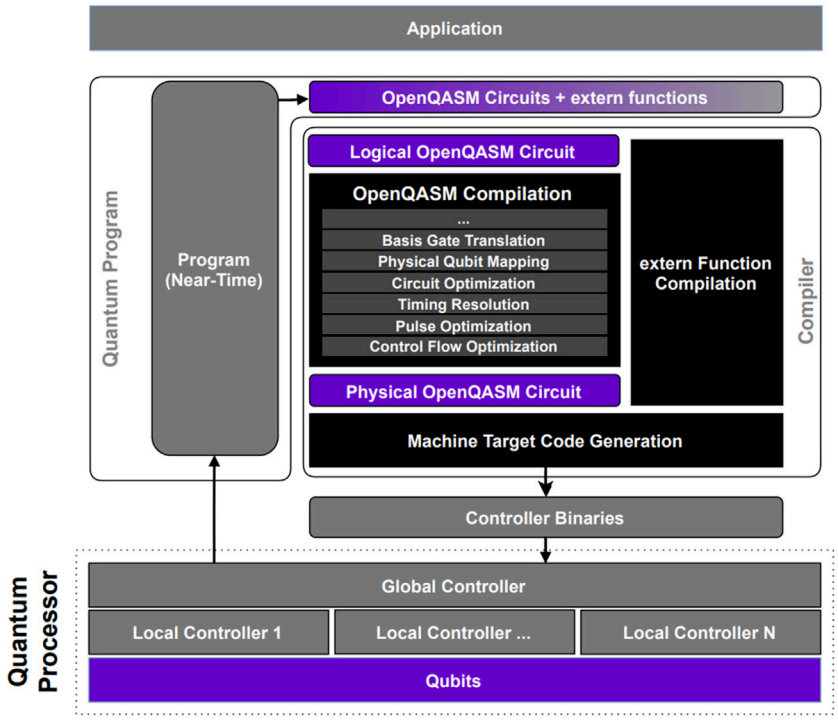

FIGURE 3
The compilation and execution model of a quantum program, and OpenQASM's place in the flow. An OpenQASM compiler can transform and optimize all aspects of the circuits described with the intermediate representation (IR), including basis gates used, qubit mapping, timing, pulses, and control flow. The final physical circuit plus external functions are passed to a target code generator which produces binaries for the quantum machine. [41].

6 Cross-layer compiler optimizations for efficient pulse control

The abstractions introduced in the layered approach of current QC stacks restrict opportunities for cross-layer optimization. For near-term quantum computing, maximal utilization of the limited quantum resources and reconciling quantum algorithms with noisy devices is of importance. Thus, a shift of the quantum computing stack towards a more vertically integrated architecture is promising. In this Section, we discuss optimizations that break the ISA abstraction by exposing pulse-level information across the compiler stack, resulting in improvements to pulse level-control as well its more efficient implementation.

6.1 Optimized compilation of aggregated instructions for realistic quantum computers

The work [45] proposes a quantum compilation technique that optimizes for pulse control by breaking across existing abstraction barriers. Doing so reduces the execution latency while also making optimal pulse-level control practical for larger numbers of qubits. Rather than directly translating one- and two-qubit gates to control pulses, the proposed framework first aggregates these small gates into larger operations. Then the framework manipulates these aggregates in two ways. First, it finds commutative operations that allow for much more efficient schedules of control pulses. Second, it uses quantum optimal control on the aggregates to produce a set of control pulses optimized for the underlying physical architecture. In all, the technique greatly exploits pulse-level control, which improves quantum efficiency over traditional gate-based methods. At the same time, it mitigates the scalability problem of quantum optimal control methods. Since the technique is software-based, these results can see practical implementation much faster than experimental approaches for improving physical device latency. Compared to traditional gate-based methods, the technique achieves execution mean speedup of 5x with a maximum speedup of 10x.

Two novel techniques are implemented: a) detecting diagonal unitaries and scheduling commutative instructions to reduce the critical path of computation; b) blocking quantum circuits in a way that scales the optimal control beyond 10 qubits without compromising parallelism.

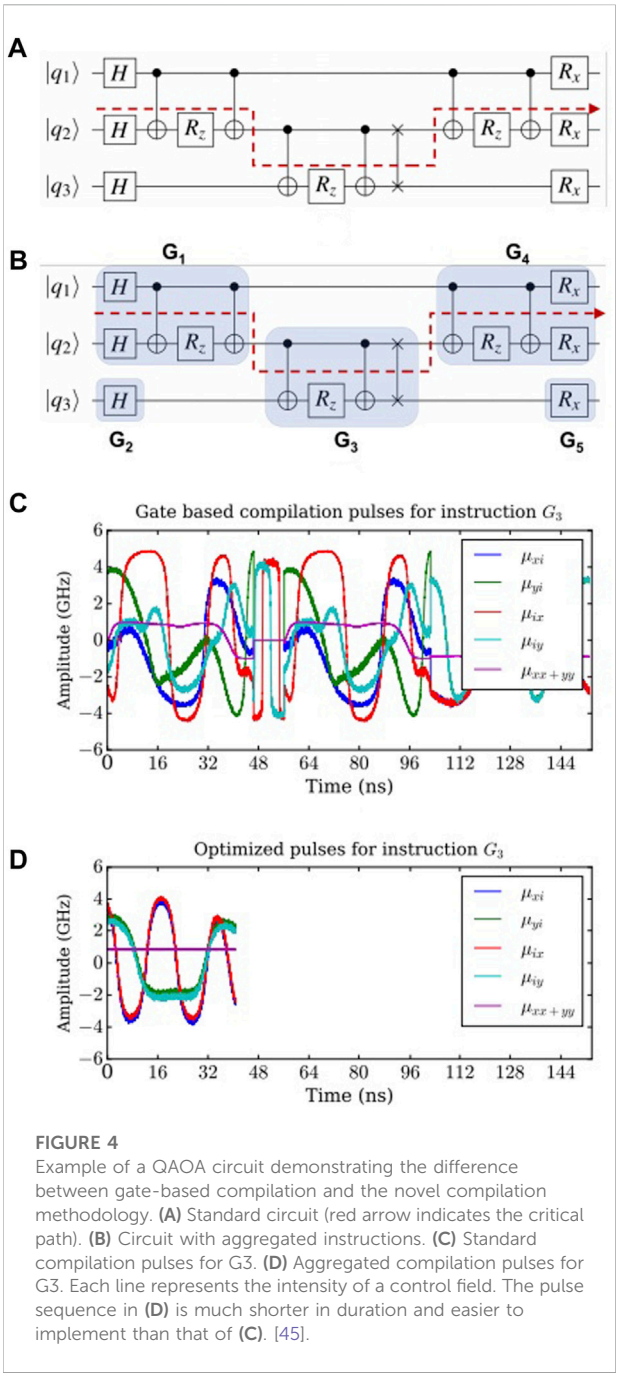

FIGURE 4
Example of a QAOA circuit demonstrating the difference between gate-based compilation and the novel compilation methodology. **(A)** Standard circuit (red arrow indicates the critical path). **(B)** Circuit with aggregated instructions. **(C)** Standard compilation pulses for G3. **(D)** Aggregated compilation pulses for G3. Each line represents the intensity of a control field. The pulse sequence in **(D)** is much shorter in duration and easier to implement than that of **(C)**. [45].

For quantum computers, achieving these speedups, and thereby reducing latency, is do-or-die: if circuits take too long to execute, the qubits decohere by the end of the computation. By reducing the latency 2-10x, this work provides an accelerated pathway to running useful quantum algorithms, without needing to wait years for hardware with 2-10x longer qubit lifetimes. An illustrative example is shown in Figure 4.

6.2 Partial compilation of variational algorithms for noisy intermediate-scale quantum machines

Each iteration of a variational algorithm depends on the results of the previous iteration; thus, necessary compilation routines must be interleaved through the computation. The noise levels in current quantum machines and the complexity of the variational use cases that are useful to solve, result in a very complex parameter tuning space for most algorithms. Thus, even small instances require thousands of iterations. Considering that the circuit compilation is on the execution critical path and cannot be hidden, the compilation latency for each iteration becomes a serious limitation.

To cope with this limitation on compilation latency, past work on VQAs has performed compilation under the standard gate-based model. This methodology has the advantage of extremely fast compilation—a lookup table maps each gate to a sequence of machine-level control pulses so that the compilation simply amounts to concatenating the pulses corresponding to each gate.

The gate-based compilation model is known to fall short of the GRadient Ascent Pulse Engineering (GRAPE) [47, 48] compilation technique, which compiles directly to the level of the machine-level control pulses that a QC actually executes. As has been the theme of this paper, the pulse-level control provides for considerably more efficient execution on the quantum machine. GRAPE has been used to achieve 2-5x pulse speedups over gate-based compilation for a range of quantum algorithms, resulting in lower decoherence and thus increased fidelity.

However, GRAPE-based compilation has a substantial cost: compilation time. This would potentially amount to several weeks or months of total compilation latency during thousands of iterations since millions of iterations are expected be needed for larger problems of significance. By contrast, typical pulse times for quantum circuits are on the order of microseconds, so the compilation latency imposed by GRAPE is untenable.

This proposal [46] introduces the idea of partial compilation, a strategy that approaches the pulse duration speedup of GRAPE, but with a manageable overhead in compilation latency. This powerful new compiler capability enables a realistic architectural choice of pulse-level control for more complex near-term applications. Two variations are proposed: a) strict partial compilation, a strategy that pre-computes optimal pulses for parametrization-independent blocks of gates and b) flexible partial compilation, a strategy that performs as well as full GRAPE, but with a dramatic speedup in compilation latency via precomputed hyperparameter optimization. An illustration of the flexible partial compilation is shown in Figure 5.

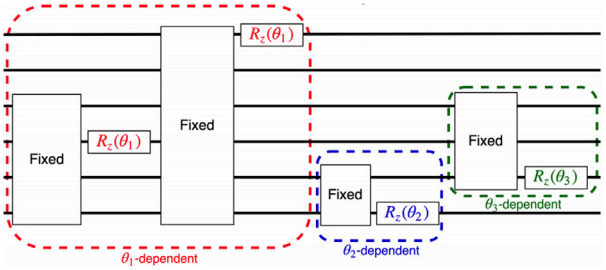

FIGURE 5
Flexible partial compilation blocks the circuit into subcircuits that depend on exactly one parameter. Hyperparameter optimization is used to precompute good hyperparameters (learning rate and decay rate) for each subcircuit. When gate angles are specified at runtime, the tuned hyperparameters quickly find optimized pulses for each subcircuit. [46].

6.3 Optimized quantum compilation for near-term algorithms with Qiskit Pulse

While pulse optimization has shown promise in previous quantum optimal control (QOC), noisy experimental systems are not entirely ready for compilation via QOC approaches. This is because QOC requires an extremely accurate model of the machine, i.e., its Hamiltonian. Hamiltonians are difficult to measure experimentally and moreover, they drift significantly between daily recalibrations.

Experimental QOC papers incur significant pre-execution calibration overhead to address this issue. By contrast, the work of Gokhale et al. [49], Gokhale et al. [50] proposes a technique that is bootstrapped purely from daily calibrations that are already performed for the standard set of basis gates. The resulting pulses are used to create an augmented basis gate set. These pulses are extremely simple, which reduces the control error and also preserves intuition about underlying operations, unlike traditional QOC. This technique leads to optimized programs, with mean 1.6x error reduction and 2x speedup for near-term algorithms. The proposed approach can target any underlying quantum hardware. An overview is shown in Figure 6.

Four key optimizations are proposed, all of which are enabled by pulse-level control: (a) Access to pulse-level control allows implementing any single-qubit operation directly with high fidelity, circumventing inefficiencies from standard compilation; (b) Although gates have the illusion of atomicity, the true atomic units are pulses. The proposed compiler creates new cancellation optimizations that are otherwise invisible; (c) Two-qubit operations are compiled directly down to the two-qubit interactions that the hardware actually implements; (d)

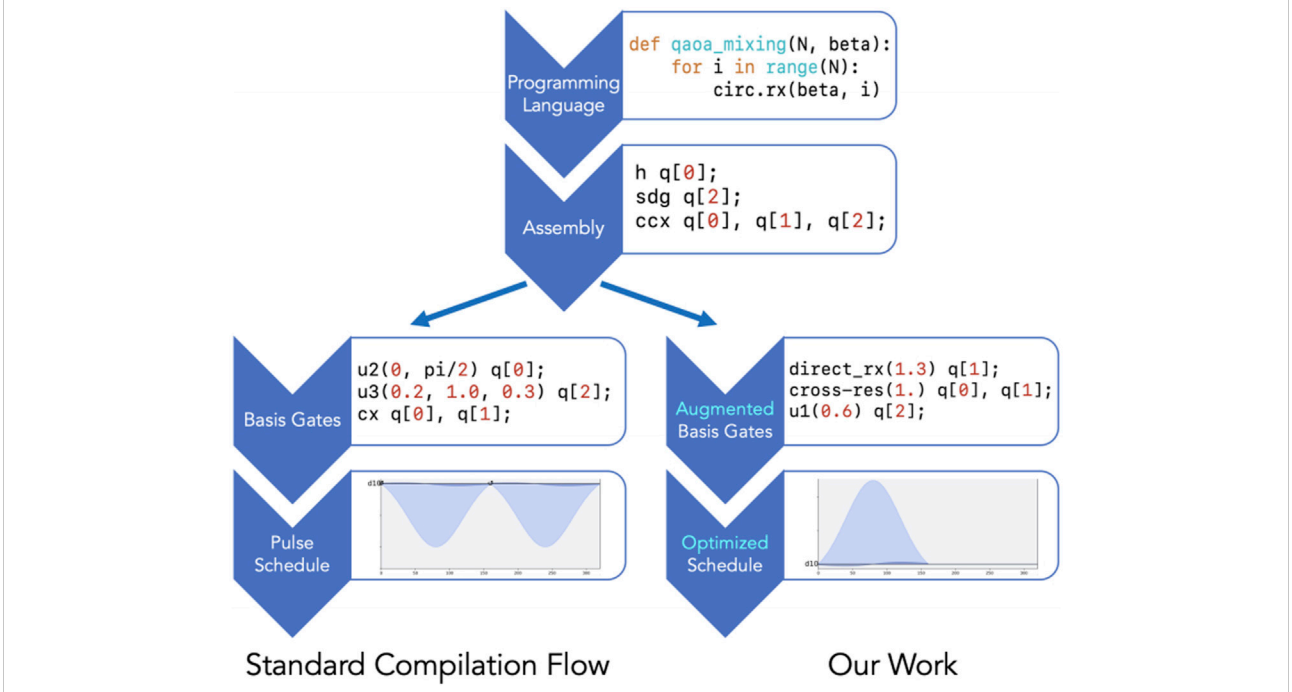

FIGURE 6
Like classical programs, quantum programs undergo a compilation process from high-level programming language to assembly. However, unlike the classical setting, quantum hardware is controlled via analog pulses. This work optimizes the underlying pulse schedule by augmenting the set basis gates to match hardware. The compiler automatically optimizes user code, which therefore remains hardware-agnostic. [49].

Pulse control enables *d*-level qudit operations, beyond the 2-level qubit subspace.

7 Simulation of pulses and with pulses

In this Section, we highlight some works on simulation. There are two forms of simulation which are relevant. The first is classical simulation of quantum devices to better understand device behavior. The second is simulating the quantum physical aspects of a complex system on a quantum machine itself. Both of these simulation domains intersect with pulse-level control. On the one hand, the effective classical simulation of quantum devices will require accurate capture of pulse-level device phenomena. On the other hand, effectively modeling complex quantum physical systems also requires precise control of execution on the quantum device, which is enabled by pulse-level control.

7.1 Classical simulation: Capturing the time-varying nature of open quantum systems

Classical simulation of quantum devices is critical to better understand device behavior. This is especially true in the near-term for noisy prototype architectures. Simulation tools accomplish a variety of different tasks including modeling noise sources, validating calculations and designs, and evaluating quantum algorithms. Since quantum simulation tools are classical, they often require supercomputers to simulate quantum devices that are large in terms of either qubit count or hardware components they contain. Quantum simulators imitate the operation of quantum devices and can be used to build the Hamiltonians required to derive control pulses for the qubit or qudit space. As a result, quantum simulators are an important tool in control engineering because complex quantum operations require the generation of fine-tuned drive signals.

Quantum systems are dynamic and evolve with time. As a result, simulation tools for quantum mechanical systems need to take variation into consideration, whether that variation stems from intentional gate application or from unintentional drift or environmental coupling. QuaC, or "Quantum in C," was developed to capture the time-varying nature of open quantum systems, including realistic amplitude damping and dephasing along with other correlated noises and thermal effects [51]. The simulator is not limited to a specific type of qubit or qudit technology, and although QuaC is classical simulator, the tool can model quantum systems with enough granularity to develop pulses required for low-level control. QuaC has been used for many applications, such as in the discovery of improved error models to better understand noisy quantum systems [52] and for the study of the coupling required between quantum dots and photonic cavities for entanglement transfer [53]. Additionally, QuaC was applied in the comparison of different quantum memory architectures to discover optimum features, such as encoding dimension [54].

7.2 Quantum simulation: Hardware efficient simulation with Qiskit Pulse

The proposal in [55] simulates a quantum topological condensed matter system on an IBM quantum processor. The simulation is done within the qubits' coherence times using pulse-level instructions provided by Qiskit Pulse. Ideally, capturing the system characteristics would require simulation by continuous time-evolution of qubits under the appropriate spin Hamiltonian obtained from a transformation of the fermion Hamiltonian. Practically, this is run on quantum devices by having the "analog" simulation decomposed and mapped onto the calibrated native basis gates of a quantum computer, making it "digital". This digital implementation on noisy quantum hardware limits precision and flexibility and is thus unable to avoid the accumulation of unnecessary errors. Pulse-level control improves this by allowing for a "semi-analog" approach. The proposal shows a pulse-scaling technique that, without additional calibration, gets closer to the ideal analog simulation.

Topologically-protected quantum computation works by moving non-Abelian anyons, such as Majorana zero modes (MZMs), around each other in two dimensions to form three-dimensional braids in space-time [56]. Thus far, there has been no definitive experimental evidence of braiding due to dynamical state evolution [57].

This work simulates a key part of a topological quantum computer: the dynamics of braiding of a pair of MZMs on a trijunction. Braiding is implemented by parametrically adjusting the Hamiltonian parameters; the time evolution is implemented using the Suzuki-Trotter decomposition, with each time step implemented by one- and two-qubit gates. Fidelity is significantly improved using pulse-level control to scale cross resonance (CR) gates derived from those pre-calibrated on the backend, thereby enabling coupling of qubit pairs with shorter CR gate times. Specifically, using native *CX* gates, only 1/6th of a full braid can be performed. Whereas, using the pulse-enabled scaled gates leads to performing a complete braid.

8 Engineering optimal pulses

8.1 Error-robust single-qubit gate set design

The work [58] proposes analog-layer programming on superconducting quantum hardware to implement and test a

new error-robust single-qubit gate set. To build this, analog pulse waveforms are numerically optimized via the use of Boulder Opal, a custom Tensorflow-based package QCt (2021b). The optimized pulses enact gates which are more resilient against dephasing, control-amplitude fluctuations, and crosstalk. Before the implementation in real hardware, these pulses go through a calibration protocol that can be fully automated QCt (2021a) to account for small distortions that may happen in the control channels. The experiments are performed on IBM quantum machines and programmed via the Qiskit Pulse API, which translates the pulses designed using Boulder Opal into hardware instructions.

The experiments show that when pulses are optimized to be robust against amplitude or dephasing errors, they can outperform the default calibrated Derivative Removal by Adiabatic Gate (DRAG) operations under native noise conditions. These optimized pulses are built including both a 30-MHz-bandwidth sinc-smoothing-function and temporal discretization to match hardware programming.

Using optimized pulses, single-qubit coherent error rates originating from sources such as gate miscalibration or drift are reduced by up to an order of magnitude, with average device-wide performance improvements of ×5. The same optimized pulses reduce the gate error variability across qubits and over time by an order of magnitude.

8.2 Reinforcement learning for error-robust gate set design

As discussed above, it has been demonstrated that the use of robust and optimal control techniques for gateset design can lead to dramatic improvements in hardware performance and computational capabilities.

The design process is straightforward when Hamiltonian representations of the underlying system are precisely known, but is considerably difficult in state-of-the-art large-scale experimental systems. A combination of effects introduces challenges not faced in simpler systems including unknown and transient Hamiltonian terms, control signal distortion, crosstalk, and temporally varying environmental noise. In all cases, complete characterization of Hamiltonian terms, their dependencies, and dynamics becomes unwieldy as the system size grows.

The work [59] proposes a black-box approach to designing an error-robust universal quantum gate set using a deep reinforcement learning (DRL) model, as shown in the inset of Figure 7A. The DRL agent is tasked to learn how to execute high fidelity constituent operations which can be used to construct a universal gate set. It iteratively constructs a model of the relevant effects of a set of available controls on quantum computer hardware, incorporating both targeted responses and undesired effects. It constructs an RX $(\pi/2)$ single-qubit driven rotation and a ZX $(-\pi/2)$ multi-qubit entangling operation by exploring a space of piecewise constant (PWC) operations executed on a superconducting quantum computer programmed using Qiskit Pulse.

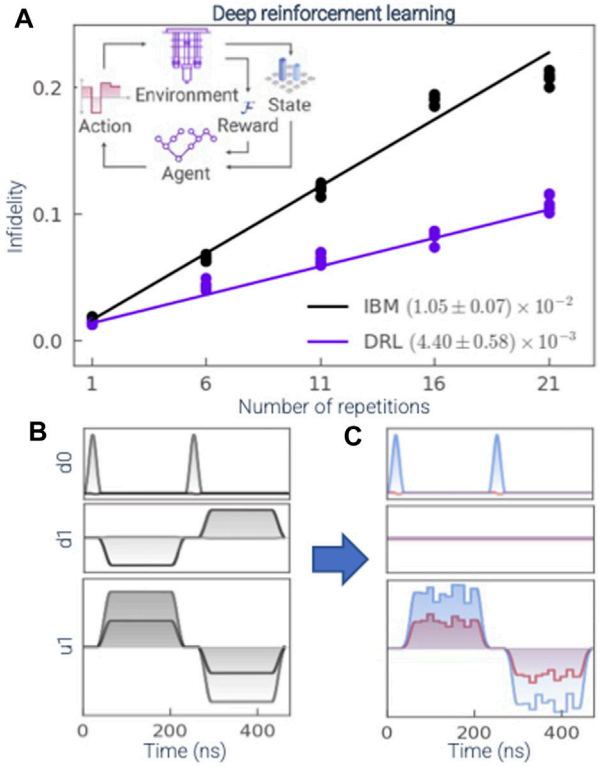

FIGURE 7
Optimization of a ZX $(-\pi/2)$ entangling gate for a superconducting device using a deep reinforcement learning (DRL) optimization routine (see inset for a diagram of the optimization cycle). **(A)** Infidelity measurement after different numbers of repeated application of the gate for the IBM default and DRL optimized gate. Approximate gate error is extracted from the slope of the infidelity data. **(B,C)** Waveforms for the **(B)** IBM default and **(C)** DRL optimized ZX $(-\pi/2)$ gates. Note that channel d1, used as a cancellation tone in the IBM default is not used in the optimized gate [59].

The constructed single-qubit gates outperform the default DRAG gates in randomized benchmarking with up to a 3x reduction in gate duration. Furthermore, the use of DRL defined entangling gates within quantum circuits for the SWAP operation shows 1.45x lower error than calibrated hardware defaults. These gates are shown to exhibit robustness against common system drifts, providing weeks of performance without needing intermediate recalibration.

9 Advanced architectures with cavity systems

Improved characterization and understanding of quantum devices leads to improved low-level control, and this opens the

door to discovering novel hardware use that pushes the state-of-the-art in quantum computing forward. As the level of precision with which we manipulate quantum devices increases, fine-tuned drive signals become available for the realization of complex Hamiltonians and gates.

This Section includes a discussion of how the codesign of quantum hardware and low-level control has enabled innovative architectures based on cavities. Cavity-transmon architectures are particularly exciting for quantum computing as they have been proposed for use in quantum memory and error correction schemes.

9.1 Robust quantum control over cavity-transmon systems

Oscillator cavities with photonic or phononic modes have long coherence times extending to tens of milliseconds, but unfortunately, these devices are difficult to control, prohibiting the generation of arbitrary quantum states needed for quantum computation [60]. However, if the storage cavity is coupled directly to a transmon qubit, quantum states can be prepared in the transmon and swapped into the storage cavity for universal control. After the quantum SWAP occurs, the oscillator holds a bosonic qubit. This idea led to the development of the superconducting cavity qubit module.

Superconducting cavity qubit modules are built using superconducting circuits comprised of a transmon qubit, a storage cavity, and a readout cavity connected to a coupler port [61–63]. In this system, the transmon qubit is used for quantum information processing, while the storage cavity, or the oscillator, can interact with the transmon to encode state within an oscillator mode [64]. These systems are characterized by a long-lived superconducting storage cavity coherence, a large Hilbert space for representing states, and fast, high-fidelity measurement and readout of the qubit state. All of these features make cavity qubit modules an attractive choice for storing encoded qubits. They also have potential within future quantum error correction protocols.

The work in [60, 64] presents a selective number-dependent arbitrary phase (SNAP) gate that improves previous efforts of transmon-cavity interaction for quantum computation. This gate efficiently enables the construction of arbitrary unitary operations, offering a scalable path towards performing quantum computation on qubits encoded in oscillators. Combined cavity and transmon drive pulses provide control of the system to implement the SNAP gate. These pulses can be generated by gradient-based optimum control, or GRAPE [65].

The superconducting cavity module consisting of a storage cavity and a transmon qubit is an example of an ancilla-assisted system. The main goal of the system to take advantage of the long coherence times of oscillator modes. The challenge of limited control associated with the storage cavity is alleviated through transmon qubit coupling. Employing a transmon in the cavity qubit module, however, does not come without cost. The ancilla qubit injects noise to the system due to shorter coherence windows associated with the transmon (tens of microseconds). Thus, there is room to further improve the quantum control of the superconducting cavity module. Quantum error correction techniques are required to protect cavity systems from ancilla-introduced errors. Fortunately, a solution based on path independence has been proposed to develop fault tolerant quantum gates that are robust to ancilla errors [66, 67].

9.2 Specialized architectures for error correction

Quantum control allows the desired quantum device behavior from carefully designed classical signals. Control techniques that continue to reduce error in near term QCs are still under development, but it must be kept in mind that the goal of quantum science is to eventually build architectures that operate at fault tolerance. As a result, new device architectures and supporting control systems should be designed concurrently.

Looking forward, there are many design bottlenecks that must be addressed to scale current qubit technology. For example, superconducting qubits face issues associated with crosstalk between qubits, limited area for control wires, and inconsistencies during fabrication. These challenges must be sidestepped, and one approach is to pursue-spatially efficient architectures that reduce the amount of hardware required for scaling, minimizing the burden on engineering and material science efforts.

New architectures based on superconducting cavity technology have been proposed that aim to implement fault tolerance while reducing the requirements of the physical hardware [68]. With reduced hardware, some of the challenges associated with scaling quantum control mechanisms will be resolved.

The approach in [68] aims to move towards scalable, fault tolerant architectures by combining compute qubits with memory qubits for a 2.5D device design. The combined compute and memory unit is designed with a Transmon coupled with a superconducting cavity. The cavity is characterized by coherence times that are much longer than those of the Transmon qubit. This advantage allows the cavity memory unit coupled to the Transmon compute unit to be used for random access to error-corrected, logical qubits stored across different memories. Thus, a simple, virtual and physical address scheme is created, and error correction is performed continuously by loading each qubit from memory. The error correction used in [68] includes two efficient adaptations of the surface code, Natural and Compact.

The 2.5D architecture has many architectural advantages that provide avenues for simplified control engineering. First, the combined transmon and cavity module requires only one set of

control wires. Compared to traditional 2D architectures that need dedicated control wires and signal generators, there is potential to reduce the amount of control hardware by a factor of n, where n is the number of modes and, thus, the number of qubits, that can be stored in the storage cavity. Additionally, the 2.5D architecture allows for transversal *CX* operations that can extend into the z-plane of the cavity to act on qubits stored within modes. This *CX* operation allows for lattice surgery operations with improved connectivity between logical qubits and that can be executed 6x faster than standard lattice surgery *CX* operations. Faster gates are a huge win for device control when qubits are constrained by a coherence time.

10 Applying qudits for acceleration

Improved low-level control has increased access to higher-dimensional encoding. In this Section, we describe work that employs qudits for quantum processing gains during gate operation and entangled state preparation.

10.1 Toffoli gate depth reduction in fixed frequency transmon qubits

Quantum information encoding is typically binary, but unlike classical computers, most QCs have multiple accessible energy levels beyond the lowest two used to realize a qubit. Accidental use of these upper energy levels can insert errors into computation. However, these energy levels that transform a qubit into a qudit can be intentionally used with careful quantum control to achieve performance gains during computation. Control pulses that access the qudit space must be carefully designed, taking hardware constraints such as shorter decay time associated with higher energy levels into consideration [69].

Implementing high-dimensional encoding for quantum information provides the benefit of data compression since qubit states can be represented with qudit formalism. Condensed information storage with qudits has been proposed for use in efficient applications of quantum error correction [70], communication protocols [71], and cryptography [72]. Another exciting application for high-dimensional encoding is in gate depth reduction for multi-qubit gates [73, 74]. Multi-qubit gates are required in many quantum algorithms, but they must be decomposed into smaller operations that agree with both the basis gate library and connectivity graph of a QC. These decompositions can be costly in terms of total single- and two-qubit operations, so opportunities for gate depth reduction can directly improve the overall circuit fidelity on today's noisy QCs.

Qiskit Pulse allows users to access higher energy levels on their transmon-based QCs. Using this low-level control of the hardware, gates for qutrits, or radix-3 quantum information that uses the basis states of $|0\rangle$, $|1\rangle$, and $|2\rangle$, can be defined. Recent work using Qiskit Pulse experimentally demonstrated that extending into the qutrit space can provide advantages for qubit-based computing [75, 76]. In these schemes, qudits were used for intermediate computation in gate realization. The overall algorithm, however, still processes in radix-2, maintaining its "qubits in, qubits out" structure.

In [76], the authors propose a Toffoli gate decomposition that is improved by intermediate qutrits. The Toffoli gate, or controlled-*CX* (*CCX*), is an important quantum gate that has been targeted by optimal control techniques since it is widely used in reversible computation, error correction, and chemistry simulation among other quantum applications [77]. The authors introduce a decomposition that uses single- and two-qutrit gates to achieve an order-preserving, Toffoli decomposition that only requires four, two-transmon interactions. The value of being order preserving is that the operation is friendly to near-term devices characterized by nearest-neighbor connections.

Ref. [76] describes the control pulses required to implement qutrit operations on IBM hardware. The successful demonstration of this Toffoli decomposition provides a significant operator reduction from the optimum qubit-gate alternative that requires eight, two-Transmon interactions. Average gate fidelity of qutrit Toffoli execution was measured to be about 78%. The qutrit Toffoli gate was benchmarked against the optimum qubit-based decomposition showing a mean fidelity improvement of around 3.82% and an execution time reduction of 1 μs. Additional error correction techniques via quantum control measures, such as dynamical decoupling, were also employed for further performance gains with the qutrit Toffoli.

10.2 High-dimensional GHZ demonstration

Quantum entanglement is a hallmark that distinguishes quantum computation from classical processing techniques. It has a wide range of applications from secure communication protocols to high-precision metrology. There have been many experimental demonstrations of entanglement, but they have primarily focused on radix-2 systems. An advantage of exploring higher-dimensional entanglement includes increased bandwith in quantum communication protocols, such as those used for superdense codes and teleportation.

At least two systems are needed for quantum entanglement, and a GHZ state is a special type of entanglement, called multipartite entanglement, that is shared between three or more subsystems, i.e. qubits or qudits. High-dimensional entanglement has been demonstrated on numerous occasions with photonic devices [78], but with careful pulse-level control using Qiskit Pulse, a qutrit GHZ state shared between three qudits was prepared on IBM transmon QC [79]. This

demonstration was the first of its kind on superconducting quantum technology.

In [79], the GHZ circuit for three qutrits was built using IBM calibrated gates, $RY(\theta)$ and CX, along with specially programmed and calibrated single-qutrit gates. To detect qutrit states, a custom discriminator was developed that is sensitive to three basis states during measurement. Experiments to generate entanglement were ran using IBM Quantum Cloud resources, and GHZ states were produced around 30000x faster than leading photonic experiments [80]. Tomography confirmed the fidelity of entangled state preparation on a five-qubit IBM machine to be approximately 76%. Additionally, the three-qutrit GHZ state was verified further using the entanglement witness protocol.

11 Expanding the quantum hardware community

Exciting breakthroughs have been seen in the NISQ era, but reaching the full promise of quantum-accelerated chemical simulation, data processing, and factoring, will require large-scale and preferably fault tolerant quantum computers [12]. Refinement of algorithms, devices, and processes is still necessary to unlock the computational advantages of quantum information processing. Unfortunately, it is anticipated that the demand for a quantum workforce to pursue these goals will greatly outweigh the supply in the near term as the amount of individuals pursuing advanced degrees in the traditional background of quantum physics and engineering is not increasing [81]. Thus, it is of critical importance to 1) develop a quantum-aware workforce starting at all age ranges and educational backgrounds in Science, Technology, Engineering, andMath (STEM) and 2) improve the general population's overall understanding of quantum technology so that QC frontiers continue to advance. As end-to-end quantum solutions mature, it is imperative to assemble a multidisciplinary network with a broad set of skills that is ready to face all challenges relating to quantum scalability.

11.1 Quantum hardware education

Reaching fault tolerant QCs is heavily dependent on dedicating adequate resources to develop improved hardware and control. This involves building a quantum community with individual skill sets that span the range from quantum aware to quantum specialist. A first step to expanding the quantum hardware community is limiting the barriers to entry. The quantum net could be cast to a wider audience by increasing efforts to introduce concepts related to quantum computing and technology at all education levels. Additionally, expanding quantum optimal control research will include engaging a diverse set of willing students of all ages.

STEM has largely overlooked many demographic groups. Since quantum research is still in its infancy as compared to many classical fields of theoretical and applied science, there is opportunity to ensure that engineers and scientists from underrepresented groups feel both a sense of belonging in the quantum community and that they can contribute to its growth. Expanding diversity and inclusion in quantum computing will continue to advance the field in both the academic and industrial setting.

In this Section, we will discuss three key strategies to allow quantum information to reach a wider audience by tailoring instruction to different levels of education. First, if students become more familiar with core quantum concepts at an early age, they are better prepared to pursue careers in quantum hardware and quantum control. Second, as the demand for a quantum workforce increases, higher education will gain a better perspective of what is required for degree plans in quantum science at both the undergraduate and graduate levels. Finally, it is possible to train seasoned scientists and engineers from classical areas of study so that their refined expertise can be applied to the research and development of quantum technology.

11.1.1 Early education, K-12

Introducing quantum information concepts at early ages might assist with lowering the barrier to enter a career in quantum science. The theory behind quantum computing contains significant mathematical content, but it is possible to introduce core concepts in a non-mathematical manner. For example, educators at this level could work towards demystifying the field, focusing on the "what" rather than emphasizing and theoretically proving the reasoning behind purely quantum phenomena. It is often through the attempt to deliver a crash-course of the complex answers for "how" quantum works that leaves many students lost. For example, rather than teaching the details of superposition and its implications on quantum states, young learners can be taught at a higher level to get comfortable with the idea that some items, like the combination of spoon and fork to make the spork, can be two things at once [82].

Recent advances in quantum science are largely caused by improvements made to supporting technology. For example, without the parallel development of cryogenic technology, many quantum computing platforms would not support computation due to the lack of a stable operational environment. Similar strides in refining quantum control have also allowed QCs to become more robust. Indirectly, concepts related to the fundamental coupling of quantum hardware and optimal control to the development of quantum informatics can be introduced to a younger audience. Scientific discovery includes many moving parts that at first glance, may not seem

significant, but are a critical part of development and experimentation. Everyday, tools and methods are used to accomplish a bigger goal, just as quantum hardware and optimum control enable quantum computing, and younger scientists can understand this by thinking about all of the equipment and instructions needed to complete a simple task or experiment. As a basic example, a cake cannot result from ingredients without the use of bowls, measuring cups, spoons, a working oven, and a detailed recipe for the baker to apply.

11.1.2 Higher education

Select undergraduate colleges and universities along with graduate schools offer exposure to coursework in quantum computing and quantum technologies. These programs are a good start, but available, and often highly specialized, coursework must expand into a more general quantum curriculum. The pipeline that produces the future quantum workforce needs to be clearly defined so that its output can be maximized to meet the expected needs of academia, government, and industry.

When considering education in the space of quantum informatics, a doctorate in quantum physics immediately comes to mind. Such expert individuals, however, are in limited supply while there is projected to be a substantial demand for a quantum-ready workforce. Additionally, not every quantum scientist needs to be well-versed in algorithms and theory. The future workforce is expected to be populated by individuals with varied backgrounds and levels of training, as opportunities for employment within quantum industry can range drastically. More technical roles range from engineer (i.e., electrical, software, optical, systems, materials, etc.), experimental scientist, theorist, technician, to application researcher. Less technical roles include those handling business development, sales, and legal strategies related to emerging technologies. Because of the wide breadth of backgrounds valuable to quantum computing, all of which necessary to further progress on quantum devices and control, many have considered how to structure quantum-related majors and minors at the undergraduate level. These types of programs would help satisfy the demand for quantum-aware and quantum-proficient professionals in the workforce. More focused graduate studies in the field of quantum science would produce quantum-specialists needed for research and development.

Focusing on the more technical portion of the workforce, quantum training programs could easily take advantage of existing programs relating to engineering, computation and science. Training in dynamics, chemistry, electromagnetism, and programming languages are just a few examples of courses that would supply individuals with the skills needed to contribute to the field of quantum. Along with a general education, application-specific areas of quantum, such as devices, algorithms, software, and protocols, would be taught depending on the student's targeted career path. In depth discussions related to the development of quantum engineering programs can be found in [81, 83, 84].

11.1.3 Continuing education as a professional

In quantum education, much emphasis has been placed on algorithms, but this does not reflect on the breadth of needs for quantum research in industry and academics. In actuality, a much more multifaceted group of researchers composed of different backgrounds is needed to push quantum technology toward the ultimate goal of fault tolerance. For example, many strides in quantum technology can be made by recruiting from the classical fields of science as many classical problems appear on the roadmap to scalable quantum technology. There are many ways in which quantum devices can improve, but some key areas include the development of better fabrication techniques, more efficient refrigeration technologies, and more accurate control methods and hardware. These example design problems could be approached by engineers that are classically trained but are literate in the basics of quantum science.

Industry constantly evolves as new technology appears. Thus, it is important that experienced professionals continue to explore their horizons to develop their skills and keep their knowledge up to date. In the space of quantum informatics, there are many resources to introduce seasoned scientists and engineers to core quantum concepts in a way that is accessible. There are many tools, such as the Qiskit Textbook, blogs, workshops [85], and online courses, that can allow individuals to explore quantum computing at their own pace. When experienced professionals dive into quantum computing, a greater pool of knowledge to assist with developing new and scaling current quantum methods and hardware results.

11.2 Improving quantum technology awareness

Many exciting developments in the field of quantum information processing are unfortunately buried in research papers, specialized conference sessions, and meetings that are inaccessible to those not immediately working on the issues at hand. While quantum computing holds great potential, there is the risk that the emerging technology could be undervalued or worse, misunderstood. A goal of expanding the quantum hardware community is to increase the public's awareness about the reasonable expectations and timelines to have for QCs. By improving communication channels and trust between the quantum community and the rest of the world, more individuals may consider how quantum research impacts their lives or even how they can get involved. Initial steps such as research groups producing news articles describing theoretical and experimental breakthroughs in layperson's

terms could make a huge difference so that quantum progress reaches a wider audience and is understood with greater accuracy.

12 Developments with analog QCs

In this work, recent progress related to the pulse-level control of gate-based QCs is emphasized. Select developments focused on analog, annealing QCs will be breifly summarized in this Section.

Analog quantum devices operate in a continuous manner. An initial state is prepared on these QCs and then the system is allowed to smoothly evolve to a target observable by carefully controlling the quantum system's environmental interactions. Optimization problems, such as discovering ground-states and the travelling salesman problem, are thought to be well-suited for analog quantum machines. These example problems look for an optimum solution within an often complex parameter space that contains many potential solutions [5]. The D-Wave quantum annealing systems based on programmable, spin-spin couplings between superconducting quantum circuits are example implementations of analog QCs [86, 87]. These quantum annealers have experienced advances in device control [88] and readout [89] to improve efficiency and accuracy, promoting device scaling to systems capable of solving problems of significant impact. Further, methods for improved low-level control that reduce noise in large-scale quantum annealers have emerged. [90]. Quantum annealers have been proposed for traffic optimization [91], machine learning [92, 93], and hybrid annealing-QAOA algorithms [94].

13 Conclusion

Quantum systems evolve in a continuous manner, and their underlying, low-level control signals are continuous as well. Thus, the pulse-based quantum computing approach utilizing continuous control signals potentially offers a much richer and more flexible use than the highly popularized gate-based approach. The ability to engineer a real-time system Hamiltonian allows us to navigate the quantum system to the quantum state of interest through generating accurate control signals.

While the benefits to the pulse approach are clear, there are several challenges stemming from the inherent complexities of a full-stack approach, including, but not limited to, defining the machine Hamiltonian, outlining the programming model, overheads from optimization and compilation, and both classical and quantum simulation capabilities. Overcoming these difficulties, however, could contribute significantly to achieving quantum advantage. The Chicago Quantum Exchange (CQE) Pulse-level Quantum Control Workshop was a step toward progress in low-level quantum control.

Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

KNS is an IBM/CQE Postdoctoral Scholar, GSR is an NSF/CRA/CCC Computing Innovation Fellow. This work is funded in part by EPiQC, an NSF Expedition in Computing, under grants CCF-1730082/1730449; in part by STAQ under grant NSF Phy-1818914; in part by NSF Grant No. 2110860; by the US Department of Energy Office of Advanced Scientific Computing Research, Accelerated Research for Quantum Computing Program; and in part by NSF OMA-2016136 and the Q-NEXT DOE NQI Center. This material is based upon work supported by the National Science Foundation under Grant #2030859 to the Computing Research Association for the CIFellows Project. This material is also supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC0021526. FTC is Chief Scientist at Super.tech (a division of ColdQuanta) and an advisor to Quantum Circuits, Inc.

Conflict of interest

Authors TA, NTB, JMC, OL and NE were employed by company IBM Quantum. Author AC was employed by the company Q-CTRL. Author MC was employed by the company Aliro Technologies, Inc. Author MJO was employed by the company HRL Laboratories, LLC. Author PG was employed by the company Super.tech (a division of ColdQuanta). Author AG was employed by the company Menten AI, Inc.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow JM, et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* (2017) 549:242–6. doi:10.1038/nature23879

2. Moll N, Barkoutsos P, Bishop LS, Chow JM, Cross A, Egger DJ, et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quan Sci. Technol.* (2018) 3:030503. doi:10.1088/2058-9565/aab822

3. Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S. Quantum machine learning. *Nature* (2017) 549:195–202. doi:10.1038/nature23474

4. Childs AM, Farhi E, Preskill J. Robustness of adiabatic quantum computation. *Phys Rev A* (2001) 65:012322. doi:10.1103/physreva.65.012322

5. Das A, Chakrabarti BK. Colloquium: Quantum annealing and analog quantum computation. *Rev Mod Phys* (2008) 80:1061–81. doi:10.1103/RevModPhys.80.1061

6. Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love PJ, et al. A variational eigenvalue solver on a photonic quantum processor. *Nat Commun* (2014) 5:4213–7. doi:10.1038/ncomms5213

7. Devitt SJ, Munro WJ, Nemoto K. Quantum error correction for beginners. *Rep Prog Phys* (2013) 76:076001. doi:10.1088/0034-4885/76/7/076001

8. O'Gorman J, Campbell ET. Quantum computation with realistic magic-state factories. *Phys Rev A* (2017) 95:032338.

9. Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev* (1999) 41:303–32. doi:10.1137/s0036144598347011

10. Grover LK. A fast quantum mechanical algorithm for database search. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (1996). p. 212–9. doi:10.1145/237814.237866

11. Nielsen MA, Chuang I. *Quantum computation and quantum information*. Cambridge: Cambridge University Press (2010).

12. Preskill J. Quantum computing in the nisq era and beyond. *Quantum* (2018) 2:79. doi:10.22331/q-2018-08-06-79

13. Arute F, Arya K, Babbush R, Bacon D, Bardin JC, Barends R, et al. Quantum supremacy using a programmable superconducting processor. *Nature* (2019) 574:505–10. doi:10.1038/s41586-019-1666-5

14. Havlíček V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, et al. Supervised learning with quantum-enhanced feature spaces. *Nature* (2019) 567:209–12.

15. Jurcevic P, Javadi-Abhari A, Bishop LS, Lauer I, Borgorin D, Brink M, et al. Demonstration of quantum volume 64 on a superconducting quantum computing system. *Quan Sci Technol* (2021). doi:10.1088/2058-9565/abe519

16. Nam Y, Chen J-S, Pisenti NC, Wright K, Delaney C, Maslov D, et al. Ground-state energy estimation of the water molecule on a trapped-ion quantum computer. *npj Quan Inf* (2020) 6:1–6. doi:10.1038/s41534-020-0259-3

17. Wright K, Beck K, Debnath S, Amini J, Nam Y, Grzesiak N, et al. Benchmarking an 11-qubit quantum computer. *Nat Commun* (2019) 10:5464–6. doi:10.1038/s41467-019-13534-2

18. Zhang J, Pagano G, Hess PW, Kyprianidis A, Becker P, Kaplan H, et al. Observation of a many-body dynamical phase transition with a 53-qubit quantum simulator. *Nature* (2017) 551:601–4. doi:10.1038/nature24654

19. Levine H, Keesling A, Semeghini G, Omran A, Wang TT, Ebadi S, et al. Parallel implementation of high-fidelity multiqubit gates with neutral atoms. *Phys Rev Lett* (2019) 123:170503. doi:10.1103/physrevlett.123.170503

20. Bernien H, Schwartz S, Keesling A, Levine H, Omran A, Pichler H, et al. Probing many-body dynamics on a 51-atom quantum simulator. *Nature* (2017) 551:579–84. doi:10.1038/nature24622

21. O'brien JL, Furusawa A, Vučković J. Photonic quantum technologies. *Nat Photon* (2009) 3:687–95.

22. Knill E, Laflamme R, Milburn GJ. A scheme for efficient quantum computation with linear optics. *Nature* (2001) 409:46–52. doi:10.1038/35051009

23. Pino JM, Dreiling JM, Figgatt C, Gaebler JP, Moses SA, Allman M, et al. Demonstration of the trapped-ion quantum ccd computer architecture. *Nature* (2021) 592:209–13. doi:10.1038/s41586-021-03318-4

24. Murali P, McKay DC, Martonosi M, Javadi-Abhari A. Software mitigation of crosstalk on noisy intermediate-scale quantum computers. In: Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems (2020). p. 1001–16. doi:10.1145/3373376.3378477

25. Ding Y, Gokhale P, Lin SF, Rines R, Propson T, Chong FT. Systematic crosstalk mitigation for superconducting qubits via frequency-aware compilation. *arXiv preprint arXiv:2008.09503* (2020). doi:10.1109/micro50266.2020.00028

26. Chamberland C, Zhu G, Yoder TJ, Hertzberg JB, Cross AW. Topological and subsystem codes on low-degree graphs with flag qubits. *Phys Rev X* (2020) 10:011022. doi:10.1103/physrevx.10.011022

27. Majer J, Chow J, Gambetta J, Koch J, Johnson B, Schreier J, et al. Coupling superconducting qubits via a cavity bus. *Nature* (2007) 449:443–7. doi:10.1038/nature06184

28. [Dataset] IBM Quantum Experience. *IBM quantum experience* (2022). Available from: https://quantum-computing.ibm.com.

29. Cross AW, Bishop LS, Sheldon S, Nation PD, Gambetta JM. Validating quantum computers using randomized model circuits. *Phys Rev A* (2019) 100:032328. doi:10.1103/physreva.100.032328

30. [Dataset] IBM quantum systems. *IBM quantum systems* (2022). Available from: https://quantum-computing.ibm.com/services?systems=all.

31. [Dataset] Gambetta J. *Ibm's roadmap for scaling quantum technology*. (2021). Available from: https://research.ibm.com/blog/ibm-quantum-roadmap (Accessed 09 29 2021).

32. [Dataset] Abraham H, AduOffeiAgarwal R, Akhalwaya IY, Aleksandrowicz G, Alexander T, et al. (2019). Qiskit: An open-source framework for quantum computing. *Zenodo*. doi:10.5281/zenodo.2562110

33. [Dataset] Developers C. Cirq (2021). See full list of authors on Github: Available from: https://github.com/quantumlib/Cirq/graphs/contributors (Accessed October, 2011). doi:10.5281/zenodo.5182845

34. [Dataset] Smith RS, Curtis MJ, Zeng WJ. *A practical quantum instruction set architecture* (2016).*arXiv:1608.03355*

35. Killoran N, Izaac J, Quesada N, Bergholm V, Amy M, Weedbrook C. Strawberry fields: A software platform for photonic quantum computing. *Quantum* (2019) 3:129. doi:10.22331/q-2019-03-11-129

36. JavadiAbhari A, Patil S, Kudrow D, Heckey J, Lvov A, Chong FT, Martonosi M. ScaffCC. In: CF '14P; roceedings of the 11th ACM Conference on Computing Frontiers. New York, NY, USA: Association for Computing Machinery (2014). doi:10.1145/2597917.2597939

37. Green AS, Lumsdaine PL, Ross NJ, Selinger P, Valiron B. Quipper. *SIGPLAN Not.* (2013) 48:333–42. doi:10.1145/2499370.2462177

38. [Dataset] Microsoft. *Microsoft q#* (2022). Available from: https://docs.microsoft.com/en-us/azure/quantum/user-guide/.

39. [Dataset] McKay DC, Alexander T, Bello L, Biercuk MJ, Bishop L, Chen J, et al. *Qiskit backend specifications for openqasm and openpulse experiments* (2018).*arXiv:1809.03452*

40. Alexander T, Kanazawa N, Egger DJ, Capelluto L, Wood CJ, Javadi-Abhari A, et al. Qiskit pulse: Programming quantum computers through the cloud with pulses. *Quan Sci. Technol.* (2020) 5:044006. doi:10.1088/2058-9565/aba404

41. [Dataset] Cross AW, Javadi-Abhari A, Alexander T, de Beaudrap N, Bishop LS, Heidel S, et al. Openqasm 3: A broader and deeper quantum assembly language. *arXiv:2104.14722* (2021).

42. Chow JM, Córcoles AD, Gambetta JM, Rigetti C, Johnson BR, Smolin JA, et al. Simple all-microwave entangling gate for fixed-frequency superconducting qubits. *Phys Rev Lett* (2011) 107:080502. doi:10.1103/PhysRevLett.107.080502

43. Viola L, Knill E, Lloyd S. Dynamical decoupling of open quantum systems. *Phys Rev Lett* (1999) 82:2417–21. doi:10.1103/physrevlett.82.2417

44. Pokharel B, Anand N, Fortman B, Lidar DA. Demonstration of fidelity improvement using dynamical decoupling with superconducting qubits. *Phys Rev Lett* (2018) 121:220502. doi:10.1103/physrevlett.121.220502

45. Shi Y, Leung N, Gokhale P, Rossi Z, Schuster DI, Hoffmann H, Chong FT. Optimized compilation of aggregated instructions for realistic quantum computers. In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems (2019). doi:10.1145/3297858.3304018

46. Gokhale P, Ding Y, Propson T, Winkler C, Leung N, Shi Y, et al. Partial compilation of variational algorithms for noisy intermediate-scale quantum machines. *Proc 52nd Annu IEEE/ACM Int Symp Microarchitecture* (2019). doi:10.1145/3352460.3358313

47. Glaser SJ, Boscain U, Calarco T, Koch CP, Köckenberger W, Kosloff R, et al. Training schrödinger's cat: Quantum optimal control. *Eur Phys J D* (2015) 69. doi:10.1140/epjd/e2015-60464-1

48. Khaneja N, Reiss T, Kehlet C, Schulte-Herbrüggen T, Glaser SJ. Optimal control of coupled spin dynamics: Design of nmr pulse sequences by gradient ascent algorithms. *J Magn Reson* (2005) 172:296–305. doi:10.1016/j.jmr.2004.11.004

49. Gokhale P, Javadi-Abhari A, Earnest N, Shi Y, Chong FT. Optimized quantum compilation for near-term algorithms with openpulse. In: 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). Manhattan, New York: IEEE (2020). p. 186–200. doi:10.1109/micro50266.2020.00027

50. Gokhale P, Tomesh T, Suchara M, Chong FT. Faster and more reliable quantum swaps via native gates. *arXiv preprint arXiv:2109.13199* (2021).

51. [Dataset] Otten M. *Quac: Open quantum systems in c, a time-dependent open quantum systems solver* (2017).

52. Otten M, Gray SK. Recovering noise-free quantum observables. *Phys Rev A* (2019) 99:012338. doi:10.1103/physreva.99.012338

53. Otten M, Gray S, Kolmakov G. Optical detection and storage of entanglement in plasmonically coupled quantum-dot qubits. *Phys Rev A* (2019) 99:032339. doi:10.1103/physreva.99.032339

54. Otten M, Kapoor K, Özgüler AB, Holland ET, Kowalkowski JB, Alexeev Y, et al. Impacts of noise and structure on quantum information encoded in a quantum memory. *Phys Rev A* (2021) 104:012605. doi:10.1103/physreva.104.012605

55. Stenger JPT, Bronn NT, Egger DJ, Pekker D. Simulating the dynamics of braiding of majorana zero modes using an ibm quantum computer. *Phys Rev Res* (2021) 3. doi:10.1103/physrevresearch.3.033171

56. Kitaev AY. Unpaired majorana fermions in quantum wires. *Phys.-Usp.* (2001) 44:131–6. doi:10.1070/1063-7869/44/10s/s29

57. Camino FE, Zhou W, Goldman VJ. e/3Laughlin Quasiparticle Primary-Fillingv=1/3Interferometer. *Phys Rev Lett* (2007) 98:076805. doi:10.1103/PhysRevLett.98.076805

58. Carvalho ARR, Ball H, Biercuk MJ, Hush MR, Thomsen F. Error-robust quantum logic optimization using a cloud quantum computer interface. *Phys Rev Appl* (2021) 15. doi:10.1103/physrevapplied.15.064054

59. Baum Y, Amico M, Howell S, Hush M, Liuzzi M, Mundada P, et al. Experimental deep reinforcement learning for error-robust gate-set design on a superconducting quantum computer. *PRX Quan* (2021) 2. doi:10.1103/prxquantum.2.040324

60. Heeres RW, Vlastakis B, Holland E, Krastanov S, Albert VV, Frunzio L, et al. Cavity state manipulation using photon-number selective phase gates. *Phys Rev Lett* (2015) 115:137002. doi:10.1103/physrevlett.115.137002

61. Paik H, Schuster D, Bishop LS, Kirchmair G, Catelani G, Sears A, et al. Observation of high coherence in josephson junction qubits measured in a three-dimensional circuit qed architecture. *Phys Rev Lett* (2011) 107:240501. doi:10.1103/physrevlett.107.240501

62. Arrangoiz-Arriola P, Wollack EA, Wang Z, Pechal M, Jiang W, McKenna TP, et al. Resolving the energy levels of a nanomechanical oscillator. *Nature* (2019) 571:537–40. doi:10.1038/s41586-019-1386-x

63. Sletten LR, Moores BA, Viennot JJ, Lehnert KW. Resolving phonon fock states in a multimode cavity with a double-slit qubit. *Phys Rev X* (2019) 9:021056. doi:10.1103/physrevx.9.021056

64. Krastanov S, Albert VV, Shen C, Zou C-L, Heeres RW, Vlastakis B, et al. Universal control of an oscillator with dispersive coupling to a qubit. *Phys Rev A* (2015) 92:040303. doi:10.1103/physreva.92.040303

65. Ma W-L, Puri S, Schoelkopf RJ, Devoret MH, Girvin S, Jiang L. Quantum control of bosonic modes with superconducting circuits. *arXiv preprint arXiv:2102.09668* (2021). doi:10.1016/j.scib.2021.05.024

66. Ma W-L, Zhang M, Wong Y, Noh K, Rosenblum S, Reinhold P, et al. Path-independent quantum gates with noisy ancilla. *Phys Rev Lett* (2020) 125:110503. doi:10.1103/physrevlett.125.110503

67. Reinhold P, Rosenblum S, Ma W-L, Frunzio L, Jiang L, Schoelkopf RJ. Error-corrected gates on an encoded qubit. *Nat Phys* (2020) 16:822–6. doi:10.1038/s41567-020-0931-8

68. Duckering C, Baker JM, Schuster DI, Chong FT. Virtualized logical qubits: A 2.5 d architecture for error-corrected quantum computing. In: 2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO). Athens, Greece: IEEE (2020). p. 173–85. doi:10.1109/micro50266.2020.00026

69. Blok M, Ramasesh V, Schuster T, O'Brien K, Kreikebaum J, Dahlen D, et al. Quantum information scrambling on a superconducting qutrit processor. *Phys Rev X* (2021) 11:021010. doi:10.1103/physrevx.11.021010

70. Muralidharan S, Zou C-L, Li L, Wen J, Jiang L. Overcoming erasure errors with multilevel systems. *New J Phys* (2017) 19:013026. doi:10.1088/1367-2630/aa573a

71. Vaziri A, Weihs G, Zeilinger A. Experimental two-photon, three-dimensional entanglement for quantum communication. *Phys Rev Lett* (2002) 89:240401. doi:10.1103/physrevlett.89.240401

72. Bruss D, Macchiavello C. Optimal eavesdropping in cryptography with three-dimensional quantum states. *Phys Rev Lett* (2002) 88:127901. doi:10.1103/PhysRevLett.88.127901

73. Fedorov A, Steffen L, Baur M, da Silva MP, Wallraff A. Implementation of a toffoli gate with superconducting circuits. *Nature* (2012) 481:170–2. doi:10.1038/nature10713

74. Gokhale P, Baker JM, Duckering C, Brown NC, Brown KR, Chong FT. Asymptotic improvements to quantum circuits via qutrits. *Proc 46th Int Symp Comput Architecture* (2019) 554–66. doi:10.1145/3307650.3322253

75. Galda A, Cubeddu M, Kanazawa N, Narang P, Earnest N. Toffoli gate depth reduction in fixed frequency transmon qutrits. *Bull Am Phys Soc* (2021).

76. Galda A, Cubeddu M, Kanazawa N, Narang P, Earnest-Noble N. Implementing a ternary decomposition of the toffoli gate on fixed-frequencytransmon qutrits. *arXiv preprint arXiv:2109.00558* (2021).

77. Kim Y, Morvan A, Nguyen LB, Naik RK, Jünger C, Chen L, et al. High-fidelity itoffoli gate for fixed-frequency superconducting qubits. *arXiv preprint arXiv:2108.10288* (2021).

78. Erhard M, Krenn M, Zeilinger A. Advances in high-dimensional quantum entanglement. *Nat Rev Phys* (2020) 2:365–81. doi:10.1038/s42254-020-0193-5

79. Cervera-Lierta A, Krenn M, Aspuru-Guzik A, Galda A. Experimental high-dimensional greenberger-horne-zeilinger entanglement with superconducting transmon qutrits. *arXiv preprint arXiv:2104.05627* (2021).

80. Erhard M, Malik M, Krenn M, Zeilinger A. Experimental Greenberger-Horne-Zeilinger entanglement beyond qubits. *Nat Photon* (2018) 12:759–64. doi:10.1038/s41566-018-0257-6

81. Fox MF, Zwickl BM, Lewandowski H. Preparing for the quantum revolution: What is the role of higher education? *Phys Rev Phys Educ Res* (2020) 16:020131. doi:10.1103/physrevphyseducres.16.020131

82. [Dataset] Stone A. *How k–12 education fits into the future of quantum computing*. (2021). Available from: https://edtechmagazine.com/k12/article/2019/11/how-k-12-education-fits-future-quantum-computing (Accessed 11 01 2021).

83. Dzurak A, Epps J, Laucht A, Malaney R, Morello A, Nurdin H, et al. Development of an undergraduate quantum engineering degree. *arXiv preprint arXiv:2110.12598* (2021).

84. Asfaw A, Blais A, Brown KR, Candelaria J, Cantwell C, Carr LD, et al. Building a quantum engineering undergraduate program. *arXiv preprint arXiv:2108.01311* (2021).

85. [Dataset] Workshop. *Workshop: I too can quantum!* (2021). Available from: https://www.epiqc.cs.uchicago.edu/workshop-i2q.i2q

86. Johnson MW, Amin MH, Gildert S, Lanting T, Hamze F, Dickson N, et al. Quantum annealing with manufactured spins. *Nature* (2011) 473:194–8. doi:10.1038/nature10012

87. Bunyk PI, Hoskinson EM, Johnson MW, Tolkacheva E, Altomare F, Berkley AJ, et al. Architectural considerations in the design of a superconducting quantum annealing processor. *IEEE Trans Appl Supercond* (2014) 24:1–10. doi:10.1109/tasc.2014.2318294

88. Johnson M, Bunyk P, Maibaum F, Tolkacheva E, Berkley A, Chapple E, et al. A scalable control system for a superconducting adiabatic quantum optimization processor. *Supercond Sci Technol* (2010) 23:065004. doi:10.1088/0953-2048/23/6/065004

89. Berkley A, Johnson M, Bunyk P, Harris R, Johansson J, Lanting T, et al. A scalable readout system for a superconducting adiabatic quantum optimization system. *Supercond Sci Technol* (2010) 23:105014. doi:10.1088/0953-2048/23/10/105014

90. Pudenz KL, Albash T, Lidar DA. Error-corrected quantum annealing with hundreds of qubits. *Nat Commun* (2014) 5:3243–10. doi:10.1038/ncomms4243

91. Inoue D, Okada A, Matsumori T, Aihara K, Yoshida H. Traffic signal optimization on a square lattice with quantum annealing. *Scientific Rep* (2021) 11:1–12. doi:10.1038/s41598-021-82740-0

92. Li RY, Gujja S, Bajaj SR, Gamel OE, Cilfone N, Gulcher JR, et al. Quantum processor-inspired machine learning in the biomedical sciences. *Patterns* (2021) 2:100246. doi:10.1016/j.patter.2021.100246

93. Winci W, Buffoni L, Sadeghi H, Khoshaman A, Andriyash E, Amin MH. A path towards quantum advantage in training deep generative models with quantum annealers. *Mach Learn Sci Technol* (2020) 1:045028. doi:10.1088/2632-2153/aba220

94. Brady LT, Baldwin CL, Bapat A, Kharkov Y, Gorshkov AV. Optimal protocols in quantum annealing and quantum approximate optimization algorithm problems. *Phys Rev Lett* (2021) 126:070505. doi:10.1103/PhysRevLett.126.070505