Experimental Deep Reinforcement Learning for Error-Robust Gate-Set Design on a Superconducting Quantum Computer

Yuval Baum,[1,2,*] Mirko Amico,[1,2] Sean Howell,[1,2] Michael Hush,[1,2] Maggie Liuzzi,[1,2] Pranav Mundada,[1,2] Thomas Merkh,[1,2] Andre R.R. Carvalho,[1,2] and Michael J. Biercuk[1,2,†]

[1] *Q-CTRL, Sydney, NSW, Australia*
[2] *Q-CTRL, Los Angeles, California, USA*

(Received 23 June 2021; accepted 4 October 2021; published 4 November 2021)

Quantum computers promise tremendous impact across applications—and have shown great strides in hardware engineering—but remain notoriously error prone. Careful design of low-level controls has been shown to compensate for the processes that induce hardware errors, leveraging techniques from optimal and robust control. However, these techniques rely heavily on the availability of highly accurate and detailed physical models, which generally achieve only sufficient representative fidelity for the most simple operations and generic noise modes. In this work, we use deep reinforcement learning to design a universal set of error-robust quantum logic gates in runtime on a superconducting quantum computer, without requiring knowledge of a specific Hamiltonian model of the system, its controls, or its underlying error processes. We experimentally demonstrate that a fully autonomous deep-reinforcement-learning agent can design single qubit gates up to $3\times$ faster than default DRAG operations without additional leakage error, and exhibiting robustness against calibration drifts over weeks. We then show that $ZX(-\pi/2)$ operations implemented using the cross-resonance interaction can outperform hardware default gates by over $2\times$ and equivalently exhibit superior calibration-free performance up to 25 days post optimization. We benchmark the performance of deep-reinforcement-learning-derived gates against other black-box optimization techniques, showing that deep reinforcement learning can achieve comparable or marginally superior performance, even with limited hardware access.

DOI: 10.1103/PRXQuantum.2.040324

I. INTRODUCTION

Large-scale fault-tolerant quantum computers are likely to enable new solutions for problems known to be hard for classical computers. A significant step towards quantum advantage, when a quantum computer can solve a practically relevant problem "faster" than a classical computer, was recently demonstrated by Google [1] and the Chinese Academy of Sciences [2]. However, demonstrating a computational advantage on a problem of practical importance remains an outstanding challenge for the sector. The extreme sensitivity of today's hardware to noise, fabrication variability, and imperfect quantum logic gates remains the key factor limiting the community's ability to reliably perform quantum computations at scale [3].

Fortunately, it has been repeatedly demonstrated that the use of robust and optimal control techniques for gateset design may lead to dramatic improvements in hardware performance, as a complement to future application of quantum error correction [4–27]. The design process is straightforward in cases where Hamiltonian representations of both the physical and the noise models in the underlying hardware are precisely known, but has proven considerably more difficult in state-of-the-art large-scale experimental systems due to an intersection of engineering constraints and the difficulty of comprehensive system characterization.

In this paper, we overcome this fundamental challenge by demonstrating the experimental use of deep reinforcement learning (DRL) for the efficient and autonomous design of error-robust quantum logic on a superconducting quantum computer without any *a priori* hardware model. We design an agent that iteratively and autonomously

*yuvalbaum9@gmail.com
†Also at ARC Centre for Engineered Quantum Systems, The University of Sydney, NSW, Australia.

constructs a model of the *relevant* effects of a set of available controls on quantum computer hardware, incorporating both targeted responses and undesired effects such as cross-couplings, nonlinearities, and signal distortions. We task the agent with learning how to execute high-fidelity constituent operations, which can be used to construct a universal gateset—an $R_X(\pi/2)$ single-qubit driven rotation and a $ZX(-\pi/2)$ multiqubit entangling operation—by allowing it to explore a space of piecewise constant (PWC) operations executed on a superconducting quantum computer programmed using Qiskit Pulse [28], and with highly constrained access to measurement data. The agent designs new nontrivial single-qubit gates, which outperform the default derivative removal by adiabatic gate (DRAG) gates in randomized benchmarking with up to a $3\times$ reduction in gate duration, down to approximately 12.4 ns, and without evident increases in leakage error. These gates are shown to exhibit robustness against common system drifts (characterized in Ref. [9]), providing up to 2 weeks of outperformance with no need for intermediate recalibration of the gate parameters. The agent also creates novel implementations of the cross-resonance interaction, which show up to approximately $2.4\times$ lower infidelity than calibrated hardware defaults. We characterize gate performance using both interleaved randomized benchmarking and gate repetition to better reveal coherent errors, achieving $\mathcal{F}_{ZX} > 99.5\%$ across multiple hardware systems, and maintaining $\mathcal{F}_{ZX} > 99.3\%$ over a period of up to 25 days with no additional recalibration. Finally, we demonstrate the use of these DRL-defined entangling gates within quantum circuits for the SWAP operation and show $1.45\times$ lower error than calibrated hardware defaults. Across these demonstrations we benchmark the DRL-designed two-qubit gates against gates created using a black-box automated optimization routine and identify incoherent processes as the current bottleneck.

II. OPTIMIZED QUANTUM LOGIC DESIGN WITH DEEP REINFORCEMENT LEARNING

A typical quantum logic gate optimization problem begins by defining system and environment Hamiltonian terms, and using analytic techniques or numeric optimization [9,29–31] to find the optimal set of control waveforms, such that the resultant unitary evolution matches a desired target. Performing useful gate optimization in this way—especially when trying to exceed state-of-the-art experimental fidelities in realistic systems—requires a comprehensive understanding of the relevant terms in the system Hamiltonian.

In practice, the real Hamiltonian of the system may include unknown and transient Hamiltonian terms [32], nonlinear distortions [33] of the control fields, nonlinear coupling of distorted control fields into new Hamiltonian terms [34], and additional hidden terms in the Hamiltonian that may change with application of the control fields. It is not generally tractable to identify this diverse range of Hamiltonian terms in large interacting systems [35], and even small simplifying approximations will typically lead to catastrophic failure of numerically optimized controls due to the sensitive manner in which the system is steered through its Hilbert space.

An alternative approach to quantum-logic design based on iterative interactive learning obviates the need of having an accurate representative model of the physical system. Deep-reinforcement-learning techniques stand out in their ability to deal with high-dimensional optimization problems in the absence of labeled data or an underlying noise model [36,37]. The essence of DRL is the ability to estimate and maximize the long-term cumulative reward learned when a deep neural-network-based agent interacts with an environment through many trials. Through this process the agent learns a target behavior—here how to perform high-fidelity quantum logic. In addition, unlike in other closed-loop optimization methods, intermediate information is extracted as well as the final measure of reward in order to construct and refine an internal model of the system's most relevant dynamics (this model need not be interpretable under human examination).

The basic ingredients of DRL are *states*, *actions*, and *rewards* [Fig. 1(a)]. The *state* is snapshot of the *environment* at a given time, and in the context of gate optimization, can be any suitable representation of the quantum state of the quantum device. The *action* is the means by which we affect or stir the state of the environment by application of a low-level electromagnetic control pulse. Finally, the *reward* encapsulates the feedback from the environment in order to quantify the quality of the last action.

DRL has recently been deployed for a variety of quantum control problems using numerical simulation [39–50]. In these studies it was possible to make at least one of the following strong assumptions: the system suffers zero noise or has a deterministic error model; controls are perfect and instantaneous; and quantum states are completely observable across time. Moving beyond these studies to efficient experimental implementations on real quantum computers, we focus on designing DRL algorithms compatible with realistic execution and measurement constraints.

Our approach involves overcoming three main challenges that we discuss below: (i) creating an efficient representation of the effective Hamiltonian in a manner compatible with experimental implementation, (ii) creating a suitable measurement routine compatible with the limited observability of the system state and its unitary evolution in a quantum computer, (iii) designing appropriate agents, which can efficiently learn system dynamics

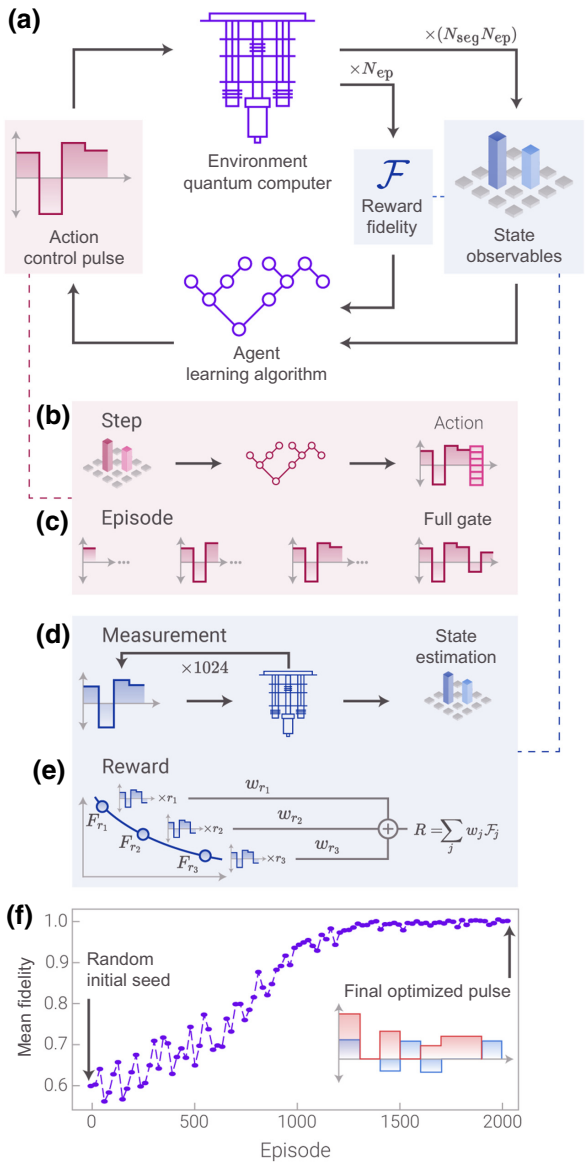

FIG. 1. DRL optimization on quantum hardware. (a) In the DRL cycle, an intermediate information is collected in order to optimize a long-term reward function. The left side of the loop indicates actions taken by the agent and the right captures measurements returned. (b) At each DRL step the action, e.g., amplitude and phase of the next segment(s) is chosen from a set of allowed actions while the previous segments actions are held fixed. (c) A set of actions that form a full control pulse is an episode. (d) After every step, an estimation of the quantum state is performed using simplified tomography, and at the conclusion of each episode an estimate of fidelity is made. Each constituent measurement contributing to this process is repeated 1024 (256) times for final (intermediate) states. In order to reduce the effect of the readout errors, we employ a standard measurement error-mitigation scheme [38]. (e) In order to estimate the gate quality, a reward is constructed by performing a weighted average of measured fidelities for sequences employing different numbers of control pulse repetitions. (f) Example of DRL optimization convergence for a control pulse implementing a single-qubit gate.

based on these constrained controls and limited access to measurements. An overview of the complete DRL process we employ to perform experimental gateset design is featured in Fig. 1 (for a complementary pseudocode, see Appendix B).

First, we seek an effective N_{seg}-segment piecewise-constant control Hamiltonian, which physically corresponds to control pulses with a PWC envelope, to be found through the iterative reinforcement-learning process shown schematically in Figs. 1(b) and 1(c). This choice is convenient due to limitations in programming and manipulating real hardware. Moreover, any deviations from the idealized PWC waveform applied to the device, due to, e.g., signal distortions, are directly captured in the measurements performed by the agent and the effective model it constructs.

Next, in order to effectively learn the hardware model, we observe and store the state of the quantum computer after intermediate step k, and allow the agent to choose the action for the next step. Due to quantum-state collapse on projective measurement, at the beginning of a new step our protocol must first reinitialize the qubits and repeat the exact sequence of actions through step k, before applying the new action at step $k+1$. Thus we are able to sequentially build up to a complete execution of a candidate quantum logic gate at the conclusion of an episode over N_{seg} state-observation cycles (giving a total of $N_{\text{seg}}N_{\text{ep}}$ state observations over N_{ep} episodes). The measurement protocols we employ for state estimation are based on the concepts of quantum-state tomography, Fig. 1(d); several different initial states are chosen in order to specify the gate uniquely. In addition to state observation we must explicitly calculate the reward at the end of an episode. Fidelity estimation employs a weighted sum of different repetitions to amplify the gate error and overcome the so-called state preparation and measurement error endemic to real hardware, estimated at approximately 4% in the hardware we employ. For more details about the single- and two-qubit measurement protocols and the choice of reward see Appendices A1 and A2. Finally, we design a DRL algorithm, which maximizes the long-term reward using an agent compatible with the above constraints, based on a policy gradient optimization algorithm. A policy $\pi(a|s)$ is a function that receives as an input the state of the system and returns a probability distribution over actions. This distribution is then used to decide the next action such that over many steps and episodes the reward is maximized. The policy function $\pi(a|s)$ is represented by a deep neural network whose trainable parameters are updated during the learning process in order to efficiently approximate the optimal policy over this space. A rigorous comparison between the performance of different DRL algorithms for this problem in a simulated environment—which led to our selection of a Monte Carlo policy gradient method—was investigated in Ref. [51].

III. EXPERIMENTAL DRL ON SUPERCONDUCTING QUANTUM COMPUTERS

The DRL algorithm is executed *in runtime* on experimental hardware via cloud access to a superconducting quantum computer operated by IBM. Hardware commands are coded using Qiskit Pulse to program various accessible analog control channels relevant to implementation of single and multiqubit gates. The DRL agent is separately hosted on a cloud server in order to allow an efficient learning procedure, and is the provided command of the quantum computer for fixed blocks of time. In order to reduce the effect of overhead due to cloud access to hardware, we generally batch several episodes in the learning procedure, meaning we execute them sequentially prior to the resulting measurement data being provided to the agent. We are limited primarily by hardware access and focus on ensuring rapid convergence in the learning process. An example runtime optimization convergence over episodes for a single-qubit gate (see details below) is shown in Fig. 1(f), and notably requires an order of magnitude less episodes than previous numerical studies [40] to achieve a high-fidelity gate. In practice, the hyperparameters of the optimizations, e.g., learning rate, can be optimized further and reach convergence in no more than 500 episodes. While this increases the probability for an unsuccessful run, in practice, we find this probability remains low. With our selected DRL agent implementation, convergence typically occurs over as little as 10–20 experimental batches (batch sizes for single- and two-qubit gates are 25 and 16, respectively), which corresponds to 0.5–1 wall clock hours. This time is dominated by hardware-application programming interface (API) and access-queue times, with typical total experimental execution of less than a minute, and agent calculations consuming negligible time on a cloud server. Once the learning process converges, the resultant gates are consistently nontrivial, showing structure in the relevant control parameters but exploiting physics, which is not obvious upon examination.

We now describe the specific optimizations we have performed and benchmark the performance of the resulting gates. Beginning with the single-qubit gate we target optimization of $R_x(\pi/2)$, a $\pi/2$ radian rotation around the x axis. This gate is implemented as a driven, microwave-mediated operation and serves as a fundamental building block for arbitrary single-qubit rotations in a $U3$ decomposition when combined with virtual Z rotations. We select a gate duration and task the DRL agent with discovering high-fidelity PWC waveforms with eight time segments, constituting a total 16-dimensional optimization when accounting for freedom in both the I and Q channels defining the microwave pulses. Our target is to achieve gates with reduced duration and enhanced error robustness relative to a default 36-ns analytic DRAG [20] pulse calibrated through a daily routine that is inaccessible to us.

We select two target gate durations informed by hardware constraints to serve as illustrative examples. First, we choose a PWC pulse with a total duration 28.4 ns, 20% shorter than the default, because the default gate performance is near the T_1 limit and leaves only approximately 20%–30% maximum achievable performance enhancement in base gate fidelity. Next, we select a gate, which is 12.4 ns, or approximately $3\times$ shorter than the default in order to probe the ability of the DRL agent to suppress leakage arising from fast pulses with spectral weight overlapping higher-order transitions. See Appendix A for further details on the reward and cost function in use.

The results of DRL gate optimization executed on a superconducting quantum computer called *ibmq_rome* are shown in Fig. 2. We evaluate the performance of the gate implementation by utilizing Clifford randomized benchmarking (RB) [52], which provides an estimate of average error per gate (EPG). The 24 Clifford gates used in RB are generated using the $R_x(\pi/2)$ gate together with virtual Z rotations in a $U3$ decomposition, and sequences are constructed using a custom compiler allowing incorporation of arbitrary gate definitions into RB sequences. See Appendices A3 and A4 for additional details regarding the single- and two-qubit randomized benchmarking protocols.

In Fig. 2 we see that the 28.4-ns optimized pulse achieves an EPG 3.7×10^{-4}, approximately $1.25x$ lower than the default and consistent with expectations based on T_1 limits. Further, we observe reduced variance of individual sequence performance about the mean (indicated by colored shading), consistent with additional suppression of coherent errors [9,53,54]. The performance of the 12.4-ns optimized pulse is comparable to the default, despite being $3\times$ faster, indicating that leakage errors can be suppressed via appropriate definition of the DRL reward function. Naively, one may expect that with unlimited controllability the effect of coherent leakage may be eliminated and T_1-limited performance achieved by optimal shaping of the pulse envelope. In reality, effective-bandwidth limitations reduce the efficacy of pulse shaping in suppressing leakage errors for the shortest waveforms.

We have also observed approximately $2.13\times$ lower errors in RB using 28.4-ns gates defined via a black-box closed-loop optimization. At this stage we are not able to distinguish whether this difference is due to the underlying method of gate optimization or the fact that a different machine was employed for these tests.

We examine the *robustness* of both DRL-defined gates by comparing the achieved EPG from RB for the same gates applied on different days. The default gates are recalibrated daily and can show amplitude variations on the order of several percent. Such recalibrations include waveform scaling of the form $S_a[I(t)+S_p iQ(t)]$, where S_a, S_p are the amplitude and phase scaling factors and

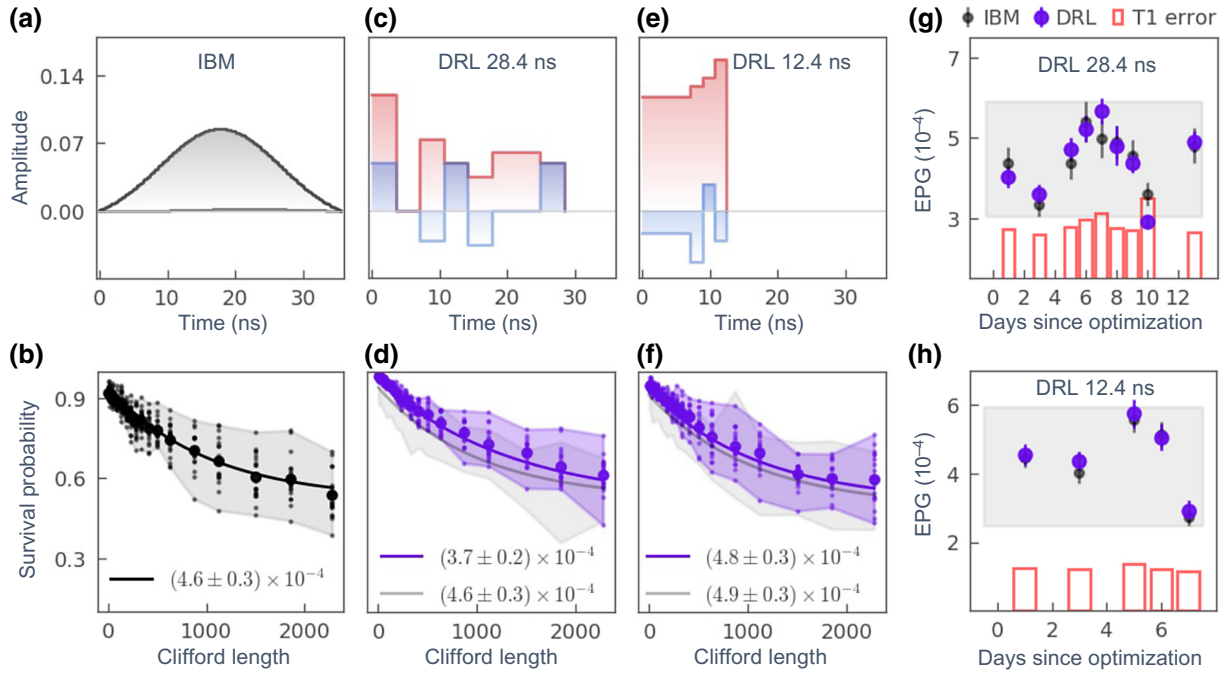

FIG. 2. $R_x(\pi/2)$ DRL optimization on the IBM device Rome. (a) The pulse waveform for the IBM default 36-ns pulse, and (b) associated gate error estimation via randomized benchmarking. Here, shading indicates the spread of individual sequence survival probabilities and an exponential fit is produced to the mean of the distribution for each sequence length, from which the EPG is extracted. (c),(d) Results of DRL optimization for a 28.4-ns pulse, and (e),(f) an ultrashort 12.4-ns pulse; colors indicate the two quadrature microwave channels. Both report minor improvements in estimated gate fidelity; see text for discussion. In (d),(f) only the best fit decay curve and shading over individual sequences is shown for clarity. (g),(h) RB-based demonstration of robustness for DRL optimized pulses versus the daily calibrated IBM pulses. Black markers show the RB performance of the daily calibrated default and colored markers indicate (g) the 28.4-ns optimized pulse and (h) the 12.4-ns optimized pulse. Both DRL optimized pulses are defined on day zero and repeated without additional calibration on subsequent days while the IBM default pulses presented are the daily calibrated pulses. For each day an estimation of the T_1 contribution to the error is plotted using red bars, based on the tabulated T_1 provided by IBM. On a daily basis, the DRL pulses have up to $1.25x$ lower error than the daily calibrated IBM pulses and performance fluctuations closely track the daily performance of the IBM pulses and T_1 limits. DRL pulses remain within a band of natural hardware fluctuations near the T_1 limit up to 2 weeks past gate definition.

frequency tuneup; fixed waveforms are used in each experiment involving DRL-defined gates without waveform recalibration.

Both gates achieve consistent performance relative to the default gates over a period up to 2 weeks, with measured EPG closely tracking fluctuations in the measured hardware T_1; see Figs. 2(g) and 2(h). By contrast, in previous experiments we observe that default gates on comparable hardware could vary substantially after approximately 12 h elapsed since last calibration [9]. These findings suggest that while temporal robustness was not explicitly included in the reward function, the agent may have discovered robustness as the underlying hardware varied during the training process.

For the two-qubit gate, we implement the $ZX(-\pi/2)$ gate using an entangling cross-resonance pulse [32,55–61] on the control qubit. The default gate implementation applies a "square-Gaussian" cross-resonance pulse in an echolike sequence [35,62,63], and a simultaneous cancellation tone applied to the resonant drive of the target qubit in order to compensate for direct classical cross-talk, see Fig. 3(b). We employ the same base structure and ask the DRL agent to find ten-segment PWC waveforms (20 dimensions over I and Q) for the cross-resonance interaction *without* application of an additional cross-talk cancellation tone. Here we directly compare the DRL procedure, which builds a gate from scratch, to a black-box gate optimization using an autonomous simulated annealing (SA) algorithm, and seeded with the initial calibrated default gate. While DRL can be considered as a black-box optimizer, in this paper we differentiate between algorithms that use only the value of the cost function and DRL, which collect intermediate information that is not directly related to the target operation.

We first compare gate performance using a repetition scheme in which the same gate is applied multiple times and fidelity estimated; see Appendix A5 for more details regarding the gate-repetition protocol. Results in Fig. 3 show that with both methods, the optimized pulses outperform the default pulses with up to 2.38× reductions in

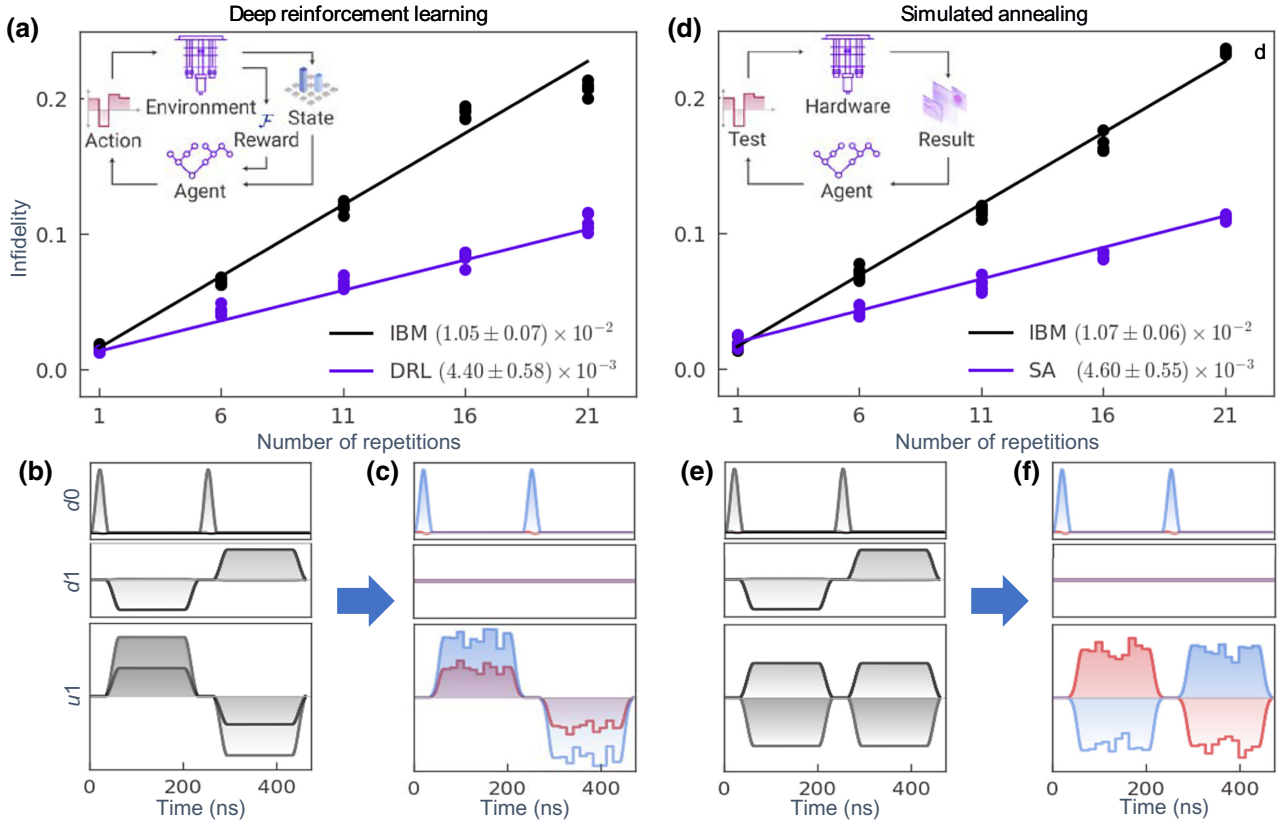

FIG. 3. Optimization of multiqubit $ZX(-\pi/2)$ entangling gates on IBM device Casablanca, benchmarked against a black-box closed-loop optimization routine. (a) Infidelity measurement for circuits consisting of different numbers of gate repetition for the IBM default and DRL optimized gate. For each number of repetitions we perform five experiments on each of two different initial states $|00\rangle$ and $|10\rangle$. Approximate gate error is extracted from the slope of infidelity with gate repetition. (Upper inset) Schematic of the optimization cycle. (b),(c) The programmed control 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. (d)–(f) Similar plots to (a)–(c) as achieved via simulated annealing on the same IBM hardware. The derived benefit using simulated annealing is similar to DRL, showing 200% improvement over the IBM default gate. These initial performance calibration measurements are performed approximately 48 h after initial gate design due to hardware-access constraints and comparison is made to the most recently updated default gate.

error per gate, achieving a gate fidelity $> 99.5\%$. These results show that both agents are able to identify superior $ZX(-\pi/2)$ gates without the need for use of a cancellation tone on channel $d1$, shown in Figs. 3(c) and 3(f), and that we are able to avoid the potential pitfalls of the learning procedure using DRL relative to a direct fidelity optimization. Optimizations and comparisons to the default were performed on different days (due to access limits), resulting in variation between calibrated default gate definition and absolute performance.

In a manner similar to the single-qubit robustness studies, we have also seen that DRL optimized $ZX(-\pi/2)$ gates outperform the default even up to 25 days since optimization with $\gtrsim 2\times$ lower error, again with no recalibration or tuning; see Fig. 4(a). As another measure, we construct a CNOT gate from the optimized $ZX(-\pi/2)$ gate and compare it to the default CNOT using interleaved randomized benchmarking (IRB) [64]. The absolute IRB gate infidelities achieved and the relative improvements of approximately $1.25 - 1.7x$ vary with machine in use and time (as T_1 can fluctuate substantially), but optimized gates consistently outperform the default across multiple metrics and over long delays since calibration. For the example of testing 25+ days post calibration shown in Fig. 4(b), we achieve a DRL optimized CNOT gate fidelity $> 99\%$.

Finally, we demonstrate that DRL can be used to directly optimize the SWAP gate *in situ*. The SWAP gate involves sequential application of three CNOT gates, built in turn from $ZX(-\pi/2)$ entangling operations and single-qubit unitaries. We maintain this overall decomposition but create a new reward function for the DRL algorithm as follows. We apply the full SWAP schedule with varying repetition values r_i on initial states $|+0\rangle$ and $|+1\rangle$, and average the different state fidelities which we extract from a full state tomography. Again we are able to see improvements in the SWAP gate fidelity through both direct repetition and interleaved randomized benchmarking. Using DRL optimization on *ibmq_bogota*, we achieve

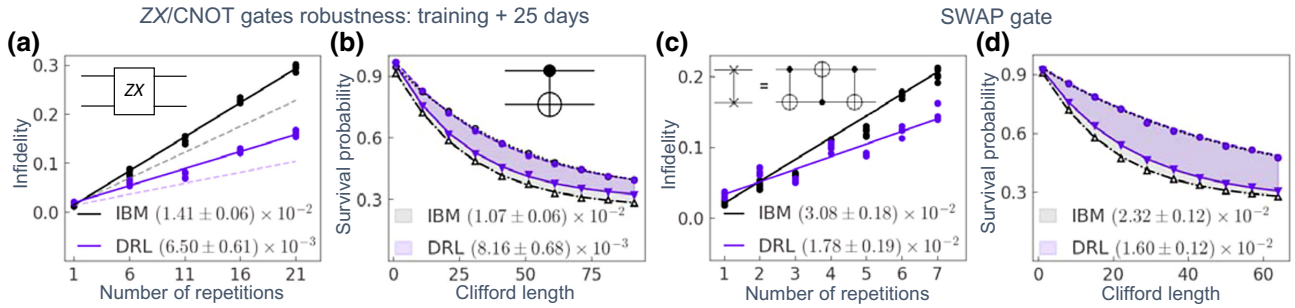

FIG. 4. Robustness and circuit-level implementations of DRL-optimized two-qubit gates. (a) Repetition measurements taken immediately after optimization (dotted lines) and 25 days following optimization. The default gate has been recalibrated but the DRL-optimized gate remains unchanged over 25 days. (b) Interleaved randomized benchmarking comparisons at 25 days post-DRL optimization. Here, circles represent standard RB sequences and triangle sequences interleaved with CNOT. The shading represents the gap between the overall sequence and the CNOT constructed from the default or optimized $ZX(-\pi/2)$. The reduced purple shaded area indicates improved performance relative to the default. (c) Repetition and (d) IRB for a DRL optimized SWAP gate constructed from three CNOTs [inset, (c)].

up to 1.45× improvement in the achieved interleaved randomized benchmarking error per gate.

IV. CONCLUSIONS AND OUTLOOK

In this work, we have shown the benefits of using deep reinforcement learning for the autonomous experimental design of high-fidelity and error-robust gatesets on superconducting quantum computers. We demonstrated that by manipulating a small set of accessible experimental control signals, our method was able to provide low-level implementations of novel quantum gates based only on measured system responses without requiring any prior knowledge of the particular device model or its underlying error processes. The entire gateset was validated to outperform the best alternatives.

We first constructed single-qubit $R_x(\pi/2)$ gates, which outperform the IBM default gate in RB with up to a 3× reduction in gate duration and robustness against common system drifts. We then constructed novel implementations of the entangling $ZX(-\pi/2)$ gate, which show up to approximately 2.38× lower infidelity, achieving $\mathcal{F}_{ZX} > 99.5\%$. With these two driven gates, we used randomized benchmarking techniques to validate a complete universal gateset with performance superior to hardware defaults even weeks past last calibration.

We conclude that DRL is an effective tool for achieving error-robust gatesets, which outperform default, human-defined operations by capturing unknown Hamiltonian terms through direct interaction with experimental hardware, and without the need for onerous Hamiltonian tomography methods. Moreover, we have validated that even in the face of restricted access to measurement data, DRL can effectively design useful novel controls. We expect that in circumstances allowing better access to measurement data, the richness of DRL may allow the construction of gate implementations, which are out of reach for simpler cost-function minimization methods.

Looking forward, we believe these results validate DRL's utility for directly improving the performance of small-to-medium-scale algorithms, beyond individual gate operations. For instance, it may be beneficial to directly optimize frequently employed circuit elements outside of the underlying universal gateset [65,66]. Our early experimental exploration of the SWAP gate suggests that additional optimization benefits may be achieved through autonomous gate optimization *in situ*, in order to effectively capture additional transients and context-dependent error sources that arise at the circuit level. We look forward to future work extending the applicability of DRL to deliver further algorithmic advantages across a variety of quantum-computing applications.

ACKNOWLEDGMENTS

We acknowledge the IBM Quantum Startup Network for provision of hardware access supporting this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. The authors also acknowledge N. Earnest-Noble for technical discussions and his support enabling our experiments. The authors are grateful to all other colleagues at Q-CTRL whose technical, product engineering, and design work has supported the results presented in this paper.

APPENDIX A: METHODS

In this section we briefly summarize the parameters and procedures we used to produce the results in the main text.

1. Single- and two-qubit measurement protocols

The measurement protocols we employ to observe the system state are based on the concepts of quantum-state

tomography; see Fig. 1(d) in the main text. For optimization of the single-qubit $R_X(\pi/2)$, we prepare and measure qubits in the three different Pauli bases, and also measure population leakage beyond the computational subspace of the transmon qubit. For the two-qubit $ZX(-\pi/2)$ entangling gate, we perform full tomography of the computational basis by collecting nine measurements in order to evaluate the expectation values of all Pauli strings. The measurement protocol is repeated to build projective-measurement statistics, and several different initial states are chosen in order to specify the gate uniquely. For all gates the state of the system is represented by a real vector of expectation values. We find that this approach provides a sufficiently reliable approximation of the state and system dynamics.

In addition to state observation we must explicitly calculate the reward at the end of an episode. To do this the fidelity is estimated as an element of the reward at the end of full episodes, i.e., after the full implementation of the candidate gate, and thus occurs with the number of episodes, N_{ep}. We evaluate fidelity relative to the target operation by acting on each initial quantum state of the qubits with a variable number of gate repetitions. The final fidelity of the waveform is then estimated as a weighted mean of the different repetitions applied to the different initial states; see Fig. 1(e) in the main text. The set of initial states and the number of gate repetitions are chosen such that the gate operation is uniquely tested and the cost or reward function captures the theoretical error of the unitary operation. This repetition-based measurement scheme serves to amplify the gate error in order to overcome the so-called state preparation and measurement (SPAM) error endemic to real hardware, and estimated at approximately 4% in the hardware we employ. Combining fidelity estimates produced for different numbers of gate repetitions also averages over pathological contextual errors that arise in experimental hardware as circuit lengths vary.

2. Reward functions for single-qubit gate

In order to evaluate the complete gate performance we repeat the candidate implementation of $R_x(\pi/2)$ a variable number of times r_i. First, we perform a full state tomography in the computational space to find the fidelity with respect to the ideal target state $\mathcal{F}_{r_i}^{(\mathrm{qubit})}$. We then calculate the population of the second level, $\ell = P(|2\rangle)$, and rescale the fidelity $\mathcal{F}_{r_i}^{(\mathrm{qutrit})} = \mathcal{F}_{r_i}^{(\mathrm{qubit})}/(1+\ell^2)$. The reward function is then calculated as a weighed mean of the different repetitions; see Fig. 1(d) and 1(e) in the main text.

3. Single-qubit RB

For the single-qubit case we use a customized RB module, which generates the 24 single-qubit Clifford gates using only virtual Z rotations together with a given $R_x(\pi/2)$ (optimized or default) and construct arbitrary RB sequences out of these gates. The data in the main text consist of 18 sequence lengths up to a maximal sequence length of 2280 Clifford gates. For each sequence length we generate 20 random sequences and repeated each 1024 times in order to estimate the survival probability. The mean (over random sequences) of the survival probability F is then fitted against the sequence length m to the following functional form: $F = A\alpha^m + B$. Since the error per gate is related to the α parameter and since the A, B parameters capture device effects such as SPAM, we first fit all three parameters, both for the default and for the optimized pulse. Then, we refit for α with fixed values of A and B, which we set to the mean of the unconstrained values. The error per Clifford is then given by $r = (1-\alpha)/2$ and the error per gate is $6r/7$ as the chosen set of the 24 single-qubit Clifford gates contains 28 appearances of $R_x(\pi/2)$, meaning 7/6 $R_x(\pi/2)$ per Clifford.

4. Two-qubit RB and IRB

For evaluating two-qubit gates we employ the Qiskit package both for RB and IRB. The IRB procedure involves comparing two survival-probability decay curves (each averaged over randomizations) in order to extract an effective EPG for the target CNOT in isolation [64]. The RB protocol uses only default gates while IRB interleaves a target gate under test with default-implemented Clifford gates. The data in the main text consists of ten sequence lengths up to a maximal sequence length of 90 Clifford gates for the CNOT testing and up to 65 Clifford gates for the SWAP testing. Similar to the single-qubit case, for each sequence length we generate 20 random sequences and repeat each 1024 times in order to estimate the survival probability. The relevant α parameter is estimated from the IRB data using a fitting technique similar to the one used for single-qubit RB.

5. Repetition-based gate characterization

In these experiments we fit the mean infidelity versus the number of gate repetitions N, separate from the repetitions employed in evaluating the reward function. For each value of N we average over five experiments applying the gate under test N times on one initial state, and repeat for a different initial state. For the ZX gate testing the initial states are $|00\rangle$ and $|10\rangle$ and for the SWAP gate testing $|+0\rangle$ and $|+1\rangle$. After each run, a full state tomography is performed and the infidelity with respect to the ideal target state is calculated. An effective measure of error per gate is extracted by applying a linear fit to the average infidelity as a function of N, which provides a measure of the gate error in the low error limit (as cross validated using IRB).

APPENDIX B: REINFORCEMENT LEARNING ALGORITHM

An overview of the DRL process we described in the main text and employed to perform experimental gateset design is featured in Algorithm 1.

The DRL algorithm used for the gate optimizations in this paper is an on-policy algorithm from the policy gradient family with a stochastic policy and a discrete action space. It is a variant of the well-known Monte Carlo policy gradient algorithm, REINFORCE [67]. A parameterized policy π_θ is iteratively updated to maximize the discounted episodic return, $J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[R(\tau)]$. It does so by directly estimating the objective's gradient with respect to the policy parameters θ, and then performs a policy update using the Adam [68] optimization algorithm, which is chosen due to its overall effectiveness in dealing with nonconvex and slowly changing objective landscapes.

It can be shown that

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta(\tau)}[\nabla_\theta \log \pi_\theta(\tau) R(\tau)] \tag{B1}$$

$$= \mathbb{E}_{\tau\sim\pi_\theta(\tau)}\left[\sum_{k=1}^{N_{\text{seg}}} \nabla_\theta \log \pi_\theta(a_k|s_k) R(\tau)\right], \tag{B2}$$

where $R(\tau)$ holds the total discounted return for an episode under trajectory τ. This expectation can be efficiently estimated by averaging over a batch of concurrent episodes. This overall learning process has the advantage of being straightforward, not requiring nor forming a model of the learning environment, and having sufficient computational efficiency to be effective for gate optimization.

Require: Initial policy parameters θ
for episode $= 1, 2, \ldots, N_{ep}$ **do**
  for each initial state $j = 1, 2, \ldots$ **do**
    Choose first action according to $\pi_\theta(a_0|\mathbf{s}_0)$ $\{\mathbf{s}_k$ = state tomography after k steps$\}$
    for $k = 1, 2, \ldots, N_{\text{seg}}$ **do**
      Initialize the quantum state of the qubit(s) $|\psi_{0,j}\rangle$
      Evolve the state by the first k segments
      Measure and estimate the qubit(s) state $\mathbf{s}_k$
      Choose next action according to $\pi_\theta(a_k|\mathbf{s}_k)$
      Store trajectory $(\mathbf{s}_{k-1}, a_{k-1}, \mathbf{s}_k)$
    end for
    for p in repetitions **do**
      Repeat the final pulse p times
      Measure, compute and store state fidelity
    end for
  end for
  Compute reward based on state fidelities
  Update the policy's parameters θ
end for

Algorithm 1. DRL training loop.

Require: Batch of $\{\tau_b\}_{b=1}^B$ trajectories from latest episode
Require: Policy network parameters: θ
Require: Discount factor: $\gamma \in [0, 1]$
Require: Current learning rate, and decay rate: α, δ
for $k = 1, 2, \ldots, N$ **do**
  Discount rewards: $R_b[k] = \sum_{i=k}^N r_{b,i}\gamma^{N-i}$ for each $b = 1, 2, \ldots, B$
end for
Estimate $\nabla_\theta J \approx \frac{1}{B}\sum_{b=1}^B \nabla \log(\pi_\theta(\tau_b)) \cdot R_b$ as Eqn. B1
Perform an Adam update step on θ using $\nabla_\theta J$
if $\alpha > \alpha_{\min}$ **then**
  Perform learning rate decay: $\alpha \leftarrow \delta\alpha$
end if

Algorithm 2. Policy network training step.

The agent's policy π_θ provides actions, which change the agent's state within the environment. Each action consists of amplitude and phase values, with a sequence of N_{seg} actions constructing a full PWC control pulse. The policy is represented by a feedforward network with one tanh activated hidden layer and a softmax output layer. For the single-qubit case, a hidden layer of size 10 is used, and for the two-qubit case, the size is 18. The softmax output layer provides a probability distribution over the discrete action space, which is then sampled to select the next concrete action to take in the environment.

The agent's policy is updated at the end of each episode, using a batch of trajectories collected throughout the episode. The training step is summarized in Algorithm 2, and is considered *on policy*, since the actions used for the update are generated by the agent using its current policy, as opposed to a previous policy or an ϵ-greedy version of π_θ. In our use case, a trajectory τ consists of a sequence of pulse segments applied, the tomographic state measurements, and the fidelity-based reward received after completion of the pulse construction at the conclusion of an episode.

The policy updates work to maximize Eq. (B1). Practically, these updates are determined by computing the policy network loss, which is the negative cross entropy between the predicted probabilities for each possible action and the chosen actions throughout the episode, weighted by the episode's discounted rewards. This is then minimized using the Adam optimizer using default parameters. By minimizing the loss, or equivalently maximizing the log likelihood, the network is encouraged to assign higher probabilities to actions, which previously led to larger episodic returns.

APPENDIX C: FAST SIMULATED ANNEALING

In the main text we explored the performance of an automated fast simulated annealing algorithm, Cauchy machine, in optimizing a two-qubit gate on the IBM

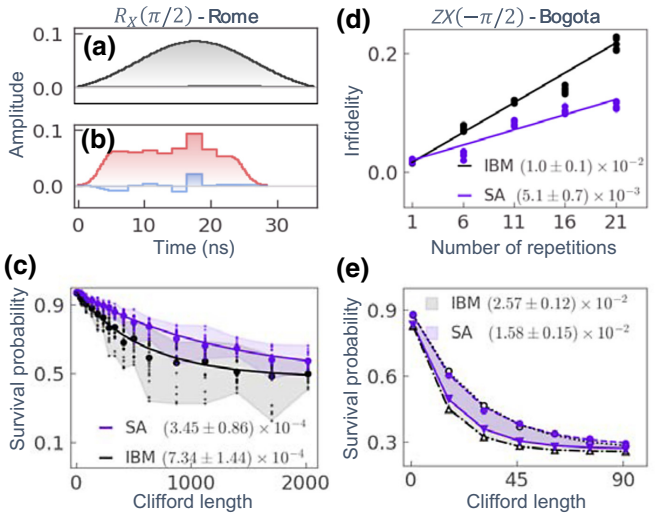

FIG. 5. Additional data on simulated-annealing closed-loop optimization. (a)–(c) Optimization of $R_x(\pi/2)$ gate on *ibmq_rome*. The optimized pulse (b) is 20% shorter than the IBM default (a) and has 2.13× lower gate error as measured using RB (c). (d),(e) Optimization of $ZX(-\pi/2)$ and composite CNOT on *ibmq_bogota*. Both repetitions (d) and IRB (e) show approximately 2× improvement in the gate error compared to the default gate, consistent with previous data sets appearing in the main text. Absolute error rates for the *ibmq_bogota* device appeared consistently higher than other machines tested.

machine. Here we provide details about the SA optimization process and present additional results of an $R_x(\pi/2)$ gate optimization on *ibmq_rome* and an optimization of $ZX(-\pi/2)$ on *ibmq_bogota*. The results of the optimization processes appear in Fig. 5.

For the SA optimization process, no intermediate information is collected and the evaluation of the full gate performance is performed with the same reward function we used in the RL optimization to estimate the full gate implementations, i.e., a weighed mean of the state fidelities. The starting point of the SA algorithm is the device default for the gate we wish to optimize. The general SA optimization process is summarized in Algorithm 3.

Require: Initialize temperatures T_0^{cost}, T_0^{amp}, T_0^{phase}
Require: Initialize amplitudes (A_i) and phases (ϕ_i) to the default values
for step $= 1, 2, \ldots$ **do**
  Draw an amplitude scale: $\delta A = Ch(0, T^{\text{amp}})$ {Ch is a Cauchy distribution}
  Draw a phase scale: $\delta\phi = Ch(0, T^{\text{phase}})$
  Draw two unit vectors u_i, v_i
  Shift the amplitudes $A_i^{\text{temp}} = A_i + u_i \delta A$
  Shift the phases $\phi_i^{\text{temp}} = \phi_i + v_i \delta\phi$
  Calculate candidate cost $C = \text{cost}(A^{\text{temp}}, \phi^{\text{temp}})$
  if $C < C_{\text{best}}$ **then**
    Accept candidate
  else
    Accept candidate with probability $F\left(\frac{C_{\text{best}} - C}{T^{\text{cost}}}\right)$ {F is the acceptance distribution}
  end if
  Update the three temperatures: $T = \frac{T_0}{1+\text{step}}$
end for

Algorithm 3. SA training loop.

[1] F. Arute, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).
[2] H.-S. Zhong, *et al.*, Quantum computational advantage using photons, Science **370**, 1460 (2020).
[3] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).
[4] G. M. Huang, T. J. Tarn, and J. W. Clark, On the controllability of quantum-mechanical systems, J. Math. Phys. **24**, 2608 (1983).
[5] J. W. CLARK, D. G. LUCARELLI, and T.-J. TARN, Control of quantum systems, Int. J. Mod. Phys. B **17**, 5397 (2003).
[6] H. I. Nurdin, M. R. James, and I. R. Petersen, Coherent quantum LQG control, Automatica **45**, 1837 (2009).
[7] M. J. Biercuk, H. Uys, A. P. VanDevender, N. Shiga, W. M. Itano, and J. J. Bollinger, Optimized dynamical decoupling in a model quantum memory, Nature **458**, 996 (2009).
[8] R. W. Heeres, P. Reinhold, N. Ofek, L. Frunzio, L. Jiang, M. H. Devoret, and R. J. Schoelkopf, Implementing a universal gate set on a logical qubit encoded in an oscillator, Nat. Commun. **8**, 94 (2017).
[9] A. R. R. Carvalho, H. Ball, M. J. Biercuk, M. R. Hush, and F. Thomsen, Error-Robust Quantum Logic Optimization Using a Cloud Quantum Computer Interface, Phys. Rev. Appl. **15**, 064054 (2021).
[10] D. Dong and I. Petersen, Quantum control theory and applications: A survey, IET Control Theory Appl. **4**, 2651 (2010).
[11] A. G. Kofman and G. Kurizki, Unified Theory of Dynamically Suppressed Qubit Decoherence in Thermal Baths, Phys. Rev. Lett. **93**, 130406 (2004).
[12] G. Gordon, G. Kurizki, and D. A. Lidar, Optimal Dynamical Decoherence Control of a Qubit, Phys. Rev. Lett. **101**, 010403 (2008).
[13] W. Yao, R.-B. Liu, and L. J. Sham, Restoring Coherence Lost to a Slow Interacting Mesoscopic Spin Bath, Phys. Rev. Lett. **98**, 077602 (2007).
[14] K. Khodjasteh and D. A. Lidar, Fault-Tolerant Quantum Dynamical Decoupling, Phys. Rev. Lett. **95**, 180501 (2005).
[15] M. S. Byrd and D. A. Lidar, Empirical determination of dynamical decoupling operations, Phys. Rev. A **67**, 012324 (2003).
[16] L. Viola and E. Knill, Robust Dynamical Decoupling of Quantum Systems with Bounded Controls, Phys. Rev. Lett. **90**, 037901 (2003).
[17] D. Vitali and P. Tombesi, Using parity kicks for decoherence control, Phys. Rev. A **59**, 4178 (1999).

[18] L. Viola and S. Lloyd, Dynamical suppression of decoherence in two-state quantum systems, Phys. Rev. A **58**, 2733 (1998).

[19] S. Chaudhury, S. Merkel, T. Herr, A. Silberfarb, I. H. Deutsch, and P. S. Jessen, Quantum Control of the Hyperfine Spin of a Cs Atom Ensemble, Phys. Rev. Lett. **99**, 163002 (2007).

[20] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm, Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits, Phys. Rev. Lett. **103**, 110501 (2009).

[21] A. Soare, H. Ball, D. Hayes, J. Sastrawan, M. Jarratt, J. McLoughlin, X. Zhen, T. Green, and M. Biercuk, Experimental noise filtering by quantum control, Nat. Phys. **10**, 825 (2014).

[22] M. Werninghaus, D. J. Egger, F. Roy, S. Machnes, F. K. Wilhelm, and S. Filipp, Leakage reduction in fast superconducting qubit gates via optimal control, npj Quantum Inf. **7**, 14 (2021).

[23] Z. Leng, P. Mundada, S. Ghadimi, and A. Houck, Robust and efficient algorithms for high-dimensional black-box quantum optimization, ArXiv:1910.03591 (2019).

[24] A. R. Milne, C. L. Edmunds, C. Hempel, F. Roy, S. Mavadia, and M. J. Biercuk, Phase-Modulated Entangling Gates Robust to Static and Time-Varying Errors, Phys. Rev. Appl. **13**, 024022 (2020).

[25] H. Ball, M. J. Biercuk, A. Carvalho, J. Chen, M. Hush, L. A. D. Castro, L. Li, P. J. Liebermann, H. J. Slatyer, C. Edmunds, V. Frey, C. Hempel, and A. Milne, Software tools for quantum control: Improving quantum computer performance through noise and error suppression, ArXiv:2001.04060 (2020).

[26] N. Wittler, F. Roy, K. Pack, M. Werninghaus, A. S. Roy, D. J. Egger, S. Filipp, F. K. Wilhelm, and S. Machnes, Integrated Tool Set for Control, Calibration, and Characterization of Quantum Devices Applied to Superconducting Qubits, Phys. Rev. Appl. **15**, 034080 (2021).

[27] M. H. Goerz, F. Motzoi, K. B. Whaley, and C. P. Koch, Charting the circuit QED design landscape using optimal control theory, npj Quantum Inf. **3**, 37 (2017).

[28] T. Alexander, N. Kanazawa, D. J. Egger, L. Capelluto, C. J. Wood, A. Javadi-Abhari, and D. C. McKay, Qiskit pulse: Programming quantum computers through the cloud with pulses, Quantum Sci. Technol. **5**, 044006 (2020).

[29] R. Byrd, P. Lu, J. Nocedal, and C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. Sci. Comput. **16**, 1190 (1995).

[30] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbruggen, and S. J. Glaser, Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms, J. Magn. Reson. **172**, 296 (2005).

[31] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, Algorithm 778: L-BFGS-B, ACM Trans. Math. Softw. **23**, 550 (1997).

[32] E. Magesan and J. M. Gambetta, Effective Hamiltonian models of the cross-resonance gate, Phys. Rev. A **101**, 052308 (2020).

[33] M. A. Rol, L. Ciorciaro, F. K. Malinowski, B. M. Tarasinski, R. E. Sagastizabal, C. C. Bultink, Y. Salathe, N. Haandbaek, J. Sedivy, and L. DiCarlo, Time-domain characterization and correction of on-chip distortion of control pulses in a quantum processor, Appl. Phys. Lett. **116**, 054001 (2020).

[34] M. Reagor, C. B. Osborn, N. Tezak, A. Staley, G. Prawiroatmodjo, M. Scheer, N. Alidoust, E. A. Sete, N. Didier, M. P. da Silva, *et al.*, Demonstration of universal parametric entangling gates on a multi-qubit lattice, Sci. Adv. **4**, eaao3603 (2018).

[35] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, Procedure for systematically tuning up cross-talk in the cross-resonance gate, Phys. Rev. A **93**, 060302 (2016).

[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA, 2018).

[37] B. Recht, A tour of reinforcement learning: The view from continuous control, Ann. Rev. Control Robot. Auton. Syst. **2**, 253 (2019).

[38] Measurement error mitigation, qiskit (2020). https://qiskit.org/textbook/ch-quantum-hardware/measurement-error-mitigation.html.

[39] Y. Zhang and Q. Ni, Recent advances in quantum machine learning, Quantum Eng. **2**, e34 (2020).

[40] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, Universal quantum control through deep reinforcement learning, npj Quantum Inf. **5**, 33 (2019).

[41] Z. An and D. L. Zhou, Deep reinforcement learning for quantum gate control, EPL **126**, 60002 (2019).

[42] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement Learning in Different Phases of Quantum Control, Phys. Rev. X **8**, 031086 (2018).

[43] R. Porotti, D. Tamascelli, M. Restelli, and E. Prati, Coherent transport of quantum states by deep reinforcement learning, Commun. Phys. **2**, 61 (2019).

[44] X.-M. Zhang, Z. Wei, R. Asad, X.-C. Yang, and X. Wang, When does reinforcement learning stand out in quantum control? A comparative study on state preparation, Quantum Eng. **5**, 85 (2019).

[45] Z. An, H.-J. Song, Q.-K. He, and D. L. Zhou, Quantum optimal control of multilevel dissipative quantum systems with reinforcement learning, Phys. Rev. A **103**, 012404 (2021).

[46] M. G. Pozzi, S. J. Herbert, A. Sengupta, and R. D. Mullins, Using reinforcement learning to perform qubit routing in quantum compilers, ArXiv:2007.15957 (2020).

[47] L. Lamata, Basic protocols in quantum reinforcement learning with superconducting circuits, Sci. Rep. **7**, 1609 (2017).

[48] K. Guy and G. Perdue, Using reinforcement learning to optimize quantum circuits in the presence of noise, Tech. Rep. (OSTI, 2020).

[49] S. Borah, B. Sarma, M. Kewming, G. J. Milburn, and J. Twamley, Measurement based feedback quantum control with deep reinforcement learning, ArXiv:2104.11856 (2021).

[50] V. V. Sivak, A. Eickbusch, H. Liu, B. Royer, I. Tsioutsios, and M. H. Devoret, Model-free quantum control with reinforcement learning, ArXiv:2104.14539 (2021).

[51] M. Liuzzi, Y. Baum, M. Amico, H. Ball, T. Merkh, S. Howell, M. Hush, and M. Biercuk, Error-robust quantum gates with deep reinforcement learning, ICML (to be published, 2021).

[52] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and

D. J. Wineland, Randomized benchmarking of quantum gates, Phys. Rev. A **77**, 012307 (2008).

[53] H. Ball, T. M. Stace, S. T. Flammia, and M. J. Biercuk, Effect of noise correlations on randomized benchmarking, Phys. Rev. A **93**, 022303 (2016).

[54] S. Mavadia, C. L. Edmunds, C. Hempel, H. Ball, F. Roy, T. M. Stace, and M. J. Biercuk, Experimental quantum verification in the presence of temporally correlated noise, npj Quantum Inf. **4**, 7 (2018).

[55] G. Paraoanu, Microwave-induced coupling of superconducting qubits, Phys. Rev. B **74**, 140504 (2006).

[56] P. De Groot, J. Lisenfeld, R. Schouten, S. Ashhab, A. Lupaşcu, C. Harmans, and J. Mooij, Selective darkening of degenerate transitions demonstrated with two superconducting quantum bits, Nat. Phys. **6**, 763 (2010).

[57] C. Rigetti and M. Devoret, Fully microwave-tunable universal gates in superconducting qubits with linear couplings and fixed transition frequencies, Phys. Rev. B **81**, 134507 (2010).

[58] J. M. Chow, A. Córcoles, J. M. Gambetta, C. Rigetti, B. Johnson, J. A. Smolin, J. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, *et al.*, Simple All-Microwave Entangling Gate for Fixed-Frequency Superconducting Qubits, Phys. Rev. Lett. **107**, 080502 (2011).

[59] J. M. Chow, J. M. Gambetta, A. D. Corcoles, S. T. Merkel, J. A. Smolin, C. Rigetti, S. Poletto, G. A. Keefe, M. B. Rothwell, J. R. Rozen, *et al.*, Universal Quantum Gate Set Approaching Fault-Tolerant Thresholds with Superconducting Qubits, Phys. Rev. Lett. **109**, 060501 (2012).

[60] V. Tripathi, M. Khezri, and A. N. Korotkov, Operation and intrinsic error budget of a two-qubit cross-resonance gate, Phys. Rev. A **100**, 012301 (2019).

[61] M. Malekakhlagh, E. Magesan, and D. C. McKay, First-principles analysis of cross-resonance gate operation, Phys. Rev. A **102**, 042605 (2020).

[62] N. Sundaresan, I. Lauer, E. Pritchett, E. Magesan, P. Jurcevic, and J. M. Gambetta, Reducing unitary and spectator errors in cross resonance with optimized rotary echoes, arXiv:2007.02925 (2020).

[63] A. Patterson, J. Rahamim, T. Tsunoda, P. Spring, S. Jebari, K. Ratter, M. Mergenthaler, G. Tancredi, B. Vlastakis, M. Esposito, *et al.*, Calibration of a Cross-Resonance Two-Qubit Gate Between Directly Coupled Transmons, Phys. Rev. Appl. **12**, 064013 (2019).

[64] E. Magesan, J. M. Gambetta, B. R. Johnson, C. A. Ryan, J. M. Chow, S. T. Merkel, M. P. Da Silva, G. A. Keefe, M. B. Rothwell, T. A. Ohki, *et al.*, Efficient Measurement of Quantum Gate Error by Interleaved Randomized Benchmarking, Phys. Rev. Lett. **109**, 080505 (2012).

[65] Y. Shi, N. Leung, P. Gokhale, Z. Rossi, D. I. Schuster, H. Hoffmann, and F. T. Chong, in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems—ASPLOS '19* (Providence, RI, U.S.A., 2019).

[66] P. Gokhale, Y. Ding, T. Propson, C. Winkler, N. Leung, Y. Shi, D. I. Schuster, H. Hoffmann, and F. T. Chong, in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture—MICRO '52* (Columbus, OH, USA, 2019).

[67] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. **8**, 229 (1992).

[68] D. Kingma and J. Ba, in *Proceedings of the 3rd International Conference on Learning Representations* (ICLR, San Diego, 2015).