UC Berkeley
UC Berkeley Previously Published Works

Title

Reinforcement learning pulses for transmon qubit entangling gates

Permalink

Journal

Machine Learning: Science and Technology, 5(2)

ISSN

Authors

Nguyen, Ho Nam
Motzoi, Felix
Metcalf, Mekena
et al.

Publication Date

2024-06-01

DOI

Copyright Information

Peer reviewed

PAPER • OPEN ACCESS

Reinforcement learning pulses for transmon qubit entangling gates

To cite this article: Ho Nam Nguyen *et al* 2024 *Mach. Learn.: Sci. Technol.* **5** 025066

View the article online for updates and enhancements.

You may also like

MACHINE
LEARNING
Science and Technology

PAPER

Reinforcement learning pulses for transmon qubit entangling gates

Ho Nam Nguyen[1,*] , Felix Motzoi[2] , Mekena Metcalf [3,7] , K Birgitta Whaley[4] , Marin Bukov[5,8] and Markus Schmitt[2,6,8]

1   Department of Physics, University of California, Berkeley, CA 94720, United States of America
2   Forschungszentrum Jülich, Institute of Quantum Control (PGI-8), D-52425 Jülich, Germany
3   Lawrence Berkeley National Laboratory, Berkeley, CA 94720, United States of America
4   Department of Chemistry, University of California, Berkeley, CA 94720, United States of America
5   Max Planck Institute for the Physics of Complex Systems, Nöthnitzer Str. 38, 01187 Dresden, Germany
6   University of Regensburg, 93053 Regensburg, Germany
7   Current address: HSBC Holdings Plc., One Embarcadero Ctr. San Francisco, CA, United States of America.
8   These authors contributed equally.
*   Author to whom any correspondence should be addressed.

E-mail: honamnguyen@berkeley.edu and mgbukov@pks.mpg.de

Keywords: quantum control, reinforcement learning, transmon qubits, entangling gates, deep learning

Abstract

The utility of a quantum computer is highly dependent on the ability to reliably perform accurate quantum logic operations. For finding optimal control solutions, it is of particular interest to explore model-free approaches, since their quality is not constrained by the limited accuracy of theoretical models for the quantum processor—in contrast to many established gate implementation strategies. In this work, we utilize a continuous control reinforcement learning algorithm to design entangling two-qubit gates for superconducting qubits; specifically, our agent constructs cross-resonance and CNOT gates without any prior information about the physical system. Using a simulated environment of fixed-frequency fixed-coupling transmon qubits, we demonstrate the capability to generate novel pulse sequences that outperform the standard cross-resonance gates in both fidelity and gate duration, while maintaining a comparable susceptibility to stochastic unitary noise. We further showcase an augmentation in training and input information that allows our agent to adapt its pulse design abilities to drifting hardware characteristics, importantly, with little to no additional optimization. Our results exhibit clearly the advantages of unbiased adaptive-feedback learning-based optimization methods for transmon gate design.

1. Introduction

Quantum computing holds immense potential to revolutionize various fields, such as optimization, simulation, and cryptography—in some cases promising exponential computational speedup compared to its classical counterpart. However, a central obstacle to harnessing this potential is the challenge of realizing reliable quantum operations. Therefore, achieving high-fidelity quantum gates is a crucial prerequisite to unlocking the full potential of quantum computing for practical applications.

    A common approach is to optimize control protocols based on effective models of the physical platform. With a suitable model at hand, high-fidelity strategies can often be achieved through analytical insights, gradient-based optimization methods, or error amplification techniques [1–6]. However, present-day quantum systems are characterized by substantial levels of noise, decoherence, and other environmental disturbances, for which accurate models are rarely known. Moreover, even when a good model is known, it is often not exactly solvable, limiting its usefulness for optimal quantum gate design. A route to circumvent these shortcomings is to resort to model-free approaches, which facilitate gate optimization through direct interactions with the quantum device.

Recently an adaptive approach using reinforcement learning (RL) has become increasingly popular in quantum gate design due to its model-free nature, obviating the need for a precise description of all details of the system [7]. When trained in a simulation, the RL approach rivals optimal control techniques in synthesizing high-precision quantum gates using discretized control for qubit-based [8–10] and qudit-based [11, 12] systems, including potentially drastic improvements in exploration [10] and sample efficiency [13]. Similarly, RL was found successful in optimizing continuous controls for a generic qubit model [14] as well as a hardware-specific gmon model [15]. Niu *et al* [15] further demonstrates the resilience of the RL-designed pulse sequences to stochastic noise when optimized with knowledge of a noisy environment. Discrete control algorithms have also been adapted to successfully learn faster single-qubit gates from scratch using experimental data from IBM's superconducting platform [16, 17]; Baum *et al* [17] additionally uses RL to improve upon the standard structure of an analytical cross-resonance pulse sequence.

While many aspects of the RL approach are more broadly applicable, in this work, we will specifically address the gate design problem for fixed-frequency fixed-coupling transmon qubits. For this platform, the model-based approach has yielded valuable insights in the pursuit of crafting high-fidelity entangling gates by utilizing the cross-resonance interaction [18, 19]. An effective approximate analytical model, capable of capturing the dominant Hamiltonian terms generated by the primary cross-resonance drive as well as undesired cross-talk [20], has paved the way for the development of various error suppression techniques, including echo sequences [21], selective-darkening/active cancellation [22–24], optimal control theory [25], rotary pulsing [26], and most recently derivative pulse shaping [27]. However, the intricate nature of real hardware and its inevitable imperfections persist, hindering our ability to achieve flawless quantum gate operations. Moreover, while the analytical insight motivates a specific family of control pulses, it remains unknown whether even better solutions can be found by expanding the considered protocol space.

As mentioned above, RL approaches have been explored to design entangling gates for the transmon platform. However, one of the inherent strengths of RL—its capability to discover innovative strategies free from the confines of theoretical protocol sequences—has remained underutilized. The absence of such flexibility results in lengthy pulse sequences which, in turn, impose severe limitations on the fidelity of these operations. Moreover, the ability of RL agents to learn adaptive strategies, that include optimal reactions to the feedback received when they are deployed, has so far received little attention. For example, although RL solutions display a degree of temporal robustness due to exposure to changing underlying system characteristics during training [17], leveraging the adaptability of the RL agent to deal with such fluctuations explicitly remains largely unexplored.

In this work, we address these and related open questions by deploying a continuous control RL algorithm to construct piece-wise constant (PWC) pulse sequences for cross-resonance and CNOT gates without any prior knowledge about the controlled system. We emphasize that this model-free approach only requires feedback from the environment (simulated or experimental) and has no information about the physical model for the environment's dynamics. Our RL training agent only has access to the quantum state and the gate fidelity, which, in principle, can be obtained experimentally via tomography and fidelity benchmark; however, in this work we train the RL agent in simulation. We tailor the simulated environment to fixed-frequency fixed-coupling transmons using realistic system characteristics to have a direct comparison between our RL results and the existing error suppression techniques in superconducting platforms. Note that in this particular transmon architecture, the qubit frequency depends on the fabrication of the transmon chip itself and cannot be controlled throughout the gate duration.

We first demonstrate that our unbiased RL agent is capable of generating novel high-fidelity control solutions that outperform current state-of-the-art cross-resonance pulse sequences. By effectively navigating the vast design space of multi-segment PWC functions to identify high-quality pulses for multiple continuous control drives simultaneously, our agent has achieved a remarkable feat in addressing an up to 120-dimensional control problem, as compared to the 20-dimensional problem considered in [17]. Without compromising the fidelity, our agent additionally discovers control solutions with large drive amplitudes, that can lead to a maximum reduction in gate duration by 30%, while being feasible to implement on modern NISQ devices. We further show how to augment our RL approach so that our agent can learn to adapt to drifts in the underlying hardware parameters (characteristics), a common issue that plagues near-term superconducting devices. This adaptation offers a two-fold advantage: immediate, high-fidelity control solutions without any extra optimization when dealing with moderate drifts or a reduction in the number of training iterations required to address more significant changes in hardware parameters. These findings underscore the practicality of the RL approach as a potent alternative for tackling the quantum gate design problem.

The following sections are organized as follows. We start with defining the quantum gate design problem for one and two-qubit gates in section 2. We give a brief overview of the state-of-the-art gate

implementations in section 3. We then present our RL approach in section 4 and our simulated results in section 5. Finally in section 6, we conclude and discuss future directions.

2. Quantum gate design problem

A quantum gate design task aims to realize a logical operation over one or more qubits via optimizing a set of available time-dependent external control fields $\{d_j(t)\}$ over some gate duration T. The effect of these fields is described by a control Hamiltonian $H_{\text{ctrl}}(t) = H_{\text{ctrl}}[\{d_j(t)\}]$ while the intrinsic dynamics of the qubits is captured by the system Hamiltonian H_{sys}. Together, they generate the full unitary evolution

$$U = \mathcal{T} \exp\left[-\mathrm{i} \int_0^T \left(H_{\text{ctrl}}(t) + H_{\text{sys}}\right) \mathrm{d}t\right], \tag{1}$$

where $\mathcal{T}$ denotes the time-ordering operator. We measure the accuracy of approximation of the target operation U_{target} by the resultant unitary U via the average gate fidelity [28]

$$\begin{aligned}
\mathcal{F}_{\text{avg}}\left(U, U_{\text{target}}\right) &= \int \mathrm{d}\psi_0 \left|\langle\psi_0| U^{\text{qubit}} U_{\text{target}}^\dagger |\psi_0\rangle\right|^2 \\
&= \int \mathrm{d}\psi_0 \left|\langle\psi_0|M|\psi_0\rangle\right|^2 \\
&= \frac{\operatorname{Tr}\left(MM^\dagger\right) + |\operatorname{Tr}(M)|^2}{n(n+1)},
\end{aligned} \tag{2}$$

where $U^{\text{qubit}} = \Pi^{\text{qubit}} U \Pi^{\text{qubit}}$ is the unitary map projected to the qubit subspace of dimension n. The average is taken over all initial states $|\psi_0\rangle$ distributed uniformly according to the Haar measure. Here we focus on superconducting qubit platforms where local Z rotations can be performed virtually [29], i.e. without incurring any additional time. We include this degree of freedom in the unitary by augmenting U^{qubit} to $V_Z(\boldsymbol{\theta}) U^{\text{qubit}}$, in which the near-optimal angles $\boldsymbol{\theta}$ are given by the matrix elements of $M = U^{\text{qubit}} U_{\text{target}}^\dagger$, see appendix A.2.

We consider a target gate fidelity of 99.9%, which is an order of magnitude higher than the 99% fidelity of the surface code threshold for two reasons. First, this level of fidelity is expected to enjoy a drastic reduction in the number of physical qubits when using the surface code [30]. Second, for the typical two-qubit gate durations considered in this work, e.g. 248.9 ns and 177.8 ns (the smallest time unit is the inverse sampling rate $\mathrm{d}t = 2/9$ ns for the considered device), the gate fidelity is coherence limited at 99.9% and 99.93%, respectively. These limits are determined by computing the average gate error under a channel with the amplitude damping rate $T_1 = 300\,\mu$s and the phase damping rate $T_2 = 300\,\mu$s [31], which are achievable in current devices [32]. Thus, having in mind any realistic decoherence in the near term, our target fidelity of 99.9% coincides with the coherence limited fidelities for the considered range of gate duration.

In addition to the average gate fidelity, we also investigated the worst-case fidelity as an alternative figure of merit. However, we did not find any discernible advantage and report this additional result in appendix E.2.

In the following, we provide the explicit form of the Hamiltonian used to model superconducting transmon qubits, while state-of-the-art gate implementations are discussed next in section 3.

2.1. Single-qubit Hamiltonian
We begin by modeling a single transmon in the Duffing approximation [33], with the lab frame Hamiltonian ($\hbar = 1$):

$$H_{1,\text{sys}}^{\text{lab}} = \omega b^\dagger b + \frac{\alpha}{2} b^\dagger b^\dagger b b, \tag{3}$$

where ω and α denote the $|0\rangle \leftrightarrow |1\rangle$ transition frequency and anharmonicity, respectively, and $b, b^\dagger$ are ladder operators. This transmon can be driven at frequency ω_d via a control Hamiltonian

$$H_{1,\text{ctrl}}^{\text{lab}}(t) = \Omega_d \operatorname{Re}\left(d(t) \mathrm{e}^{\mathrm{i}\omega_d t}\right)\left(b^\dagger + b\right), \tag{4}$$

where we have factored out the drive strength Ω_d to keep the real and imaginary parts of the complex control signal $d(t)$ normalized to $[-1, 1]$. By rotating to the driving frame via the transformation $R(t) = \mathrm{e}^{-\mathrm{i}\omega_d t b^\dagger b}$ and ignoring fast rotating terms (cf appendix A.1 for details), we arrive at the rotating frame Hamiltonian.

$$H_1(t) = R^\dagger(t)\left(H_{1,\text{sys}}^{\text{lab}} + H_{1,\text{ctrl}}^{\text{lab}}(t) - i\partial_t\right)R(t)$$
$$\approx \delta b^\dagger b + \frac{\alpha}{2}b^\dagger b^\dagger bb + \left[\frac{\Omega_d}{2}d(t)b + \text{h.c.}\right], \tag{5}$$

where $\delta = \omega - \omega_d$ is the qubit detuning, which vanishes when the transmon is driven on resonance, i.e. $\omega_d = \omega$.

To make the connection to single-qubit rotations explicit, we consider only the first two levels and make the replacement $b \to \sigma^- = X - iY$ and $b^\dagger \to \sigma^+ = X + iY$ in the control Hamiltonian, which results in

$$H_{1,\text{ctrl}}^{\text{qubit}}(t) = \frac{\Omega_d}{2}\text{Re}(d(t)])X + \frac{\Omega_d}{2}\text{Im}(d(t))Y, \tag{6}$$

where X and Y denote respectively the Pauli-X and Pauli-Y matrices. Evidently, turning on the complex control field $d(t)$ induces qubit rotation around the x and y axes, which, for long enough gate duration, is sufficient for realizing any single-qubit gate. With the Euler-angle decomposition,

$$U_3(\theta,\phi,\lambda) = V_Z(\phi)R_X\left(-\frac{\pi}{2}\right)V_Z(\theta)R_X\left(\frac{\pi}{2}\right)V_Z(\lambda), \tag{7}$$

one can achieve any desired single-qubit gate by merely calibrating the $R_X(\pm\pi/2)$ rotations, and, as discussed above, Z gates can be implemented virtually. Note that in practice much shorter gate durations are desirable, in which case a number of errors (such as high state population [34]) will unavoidably arise and therefore need to be counteracted, see section 3.

2.2. Two-qubit Hamiltonian

We now extend the model to describe a pair of transmons by combining two Duffing Hamiltonians coupled by a resonator. The resonator acts as a bus for coherent communication between the quantum states of the two Hamiltonians that will define the qubits. Although a variety of different logical two-qubit gates are possible with this setup [35], we will discuss here the cross-resonance interaction Hamiltonian, which is the current standard for fixed-frequency architecture.

When the resonator's fundamental frequency is much larger than both $|0\rangle \leftrightarrow |1\rangle$ transition frequencies of the individual transmons, we can project the Hamiltonian onto the zero-excitation subspace of the bus resonator to obtain the following lab-frame effective Hamiltonian:

$$H_{2,\text{sys}}^{\text{lab}} = \sum_{j=0}^{1}\left(\tilde{\omega}_j b_j^\dagger b_j + \frac{\alpha_j}{2}b_j^\dagger b_j^\dagger b_j b_j\right) + J\left(b_0^\dagger b_1 + b_0 b_1^\dagger\right). \tag{8}$$

Here, $\tilde{\omega}_j$ denotes the resonator-dressed qubit frequency and J denotes the effective coupling strength [20].

In addition to a standard on-resonance control field $d(t)$ on each transmon, an entangling operation can be realized by driving one qubit at the frequency of another via a cross-resonance (CR) control field $u(t)$. The two-transmon control Hamiltonian then becomes

$$H_{2,\text{ctrl}}^{\text{lab}}(t) = \Omega_{d_0}\text{Re}\left(e^{i\tilde{\omega}_0 t}d_0(t)\right)\left(b_0^\dagger + b_0\right)$$
$$+ \Omega_{u_{01}}\text{Re}\left(e^{i\tilde{\omega}_1 t}u_{01}(t)\right)\left(b_0^\dagger + b_0\right)$$
$$+ \Omega_{d_1}\text{Re}\left(e^{i\tilde{\omega}_1 t}d_1(t)\right)\left(b_1^\dagger + b_1\right)$$
$$+ \Omega_{u_{10}}\text{Re}\left(e^{i\tilde{\omega}_0 t}u_{10}(t)\right)\left(b_1^\dagger + b_1\right). \tag{9}$$

Here, $u_{01}(t)$ refers to the CR pulse sent to qubit 0 when driven at the frequency of qubit 1, and vice versa for $u_{10}(t)$.

Moving into the frame rotating at ω_d for both transmons using the transformation $R(t) = e^{-i\omega_d t(b_0^\dagger b_0 + b_1^\dagger b_1)}$, and ignoring the fast rotating terms, we obtain the rotating-frame Hamiltonian

$$H_2(t) = R^\dagger(t)\left(H_{2,\text{sys}}^{\text{lab}} + H_{2,\text{ctrl}}^{\text{lab}}(t) - i\partial_t\right)R(t)$$
$$= \sum_{j=0,1}\left(\delta_j b_j^\dagger b_j + \frac{\alpha_j}{2}b_j^\dagger b_j^\dagger b_j b_j\right) + J\left(b_0^\dagger b_1 + b_0 b_1^\dagger\right)$$
$$+ \frac{b_0}{2}\left[\Omega_{d_0}e^{i\delta_0 t}d_0(t) + \Omega_{u_{01}}e^{i\delta_1 t}u_{01}(t)\right] + \text{h.c.}$$
$$+ \frac{b_1}{2}\left[\Omega_{d_1}e^{i\delta_1 t}d_1(t) + \Omega_{u_{10}}e^{i\delta_0 t}u_{10}(t)\right] + \text{h.c.} \tag{10}$$

Figure 1. Coupled transmons simulated as Duffing oscillators. Simulation is truncated at three energy levels per transmon (the faded fourth level is shown but not considered), and performed in a rotating frame. The first two levels act as qubits (dashed boxes). External control drives (purple) include on-resonance and cross-resonance complex control fields, denoted by $d(t)$ and $u(t)$ in the main text, respectively. The full Hamiltonian in equation (10) is completely characterized by the detuning δ_j and anharmonicity α_j for each transmon, drive strengths $\{\Omega_{d_0}, \Omega_{u_{01}}, \Omega_{d_1}, \Omega_{u_{10}}\}$ for 4 external controls, and the direct coupling *J*.

where $\delta_j = \tilde{\omega}_j - \omega_d$ defines the detuning for the *j*th transmon. The pair of transmons, whose dynamics is described by $H_2(t)$, is illustrated in figure 1. In this work, we simulate the dynamics in the frame rotating at the second transmon's frequency, i.e. setting $\delta_1 = 0$.

With the first transmon as control and the second as target, the main effect of the CR drive can be studied by setting u_{01} to a constant value Ω and other control fields to zero:

$$H_2^{\mathrm{CR}}(t) = \delta_0 b_0^\dagger b_0 + \sum_{j=0,1} \frac{\alpha_j}{2} b_j^\dagger b_j^\dagger b_j b_j + J\left(b_0^\dagger b_1 + b_0 b_1^\dagger\right) + \frac{\Omega}{2} u_{01}(t) b_0 + \mathrm{h.c.} \tag{11}$$

To obtain the effective *ZX* interaction rate (or strength) within the qubit subspace while accounting for higher levels, one can employ perturbation theory [20] to obtain the following approximate effective CR Hamiltonian

$$H_{\mathrm{eff}}^{\mathrm{CR}} = \sum_{A,B} \omega_{AB} A \otimes B, \tag{12}$$

where $A \in \{I, Z\}$ and $B \in \{I, X, Z\}$. In the presence of classical cross-talk and incorrect phases in control drives, *B* can be extended to include the Pauli*Y* matrix [24]. Within perturbation theory for small coupling *J* and small drive Ω, the interaction rates have the following scaling

$$\omega_{ZX}, \omega_{IX} \sim \Omega, \qquad \omega_{ZI}, \omega_{ZZ} \sim \Omega^2. \tag{13}$$

The resultant dominant *ZX* term can then be used to implement the following entangling operation

$$ZX(\pi/2) = \exp\left(-\mathrm{i}\frac{\pi}{4} ZX\right) = \frac{I - \mathrm{i}ZX}{\sqrt{2}}, \tag{14}$$

known as the cross-resonant (CR) gate, which is locally equivalent to the popular CNOT gate. Such entangling operations, together with the capacity to realize any single-qubit gate, enable universal quantum computation in the superconducting transmon platform.

2.3. Leakage
Although only the first two levels of a transmon are used to represent a qubit, the higher levels are nevertheless still present and can be populated as the system evolves. We capture the most prominent leakage outside the ideal computational qubit subspace by including the second excited state of the anharmonic oscillator $|2\rangle$ (cf figure 1). The full state space can be decomposed into a direct sum of the computational subspace χ_1 and the leakage subspace χ_2. Projectors onto these subspaces respectively are denoted as I_1 and I_2 [36].

Under a unitary quantum channel $\mathcal{E}(\rho) = U\rho U^\dagger$, state leakage averaged over all initial pure states in the qubit subspace is given by

$$L = \int \mathrm{d}\psi_0 \mathrm{Tr}\left[I_2 \mathcal{E}\left(|\psi_0\rangle\langle\psi_0|\right)\right] = \mathrm{Tr}\left[I_2 \mathcal{E}\left(\frac{I_1}{\dim(\chi_1)}\right)\right].$$

Here, we have used the fact that the average of $|\psi_0\rangle\langle\psi_0|$ results in the maximally mixed state $I_1/\dim(\chi_1)$. For a system of two transmons, the computational subspace χ_1 is spanned by $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$, and thus $\dim(\chi_1) = 4$.

Intuitively, the average leakage L quantifies the population fraction initially prepared in the computational subspace, that ultimately ends up outside of this subspace [34]. While a prerequisite for achieving a high-fidelity quantum gate is to minimize leakage at gate completion, conventional anti-leakage schemes typically suppress leakage throughout the entire gate duration. This is due to the difficulty in restoring population into the computational subspace at the end, as well as (historically) disproportionately larger decoherence rates for higher levels. Nevertheless, high-fidelity control solutions, with considerable excursion beyond the computational subspace during gate duration, do in fact exist and are achievable with the use of RL optimization, as we will demonstrate later, e.g. see the data reported for RL protocols in figure 7 and the corresponding discussion in section 5.2.

2.4. Entanglement

In addition to the fidelity, an important goal of a two-qubit operation is to generate entanglement. Among a number of different options, we select a simple metric called *linear entropy* which quantifies the entanglement of a joint density matrix ρ describing the pure state of both qubit A and qubit B as follows

$$S_{\text{lin}} = 1 - \text{Tr}_B \left[(\text{Tr}_A \rho)^2 \right], \tag{15}$$

where Tr_A (Tr_B) denotes partially tracing out qubit A (B). To calculate the linear entropy of an initial state $|\psi_0\rangle$ after a unitary operation U, we simply substitute $\rho = |\psi\rangle\langle\psi| = U|\psi_0\rangle\langle\psi_0|U^\dagger$. When applied to a multi-level system like transmons, we make sure to normalize the final state after projecting to the qubit subspace.

To assess the entanglement capacity of a quantum gate, we draw inspiration from the widely adopted *entangling power* of unitary operations [37], defined as the average linear entropy produced by a unitary operator when acting on the space of all two-qubit product states. Since the average is taken over two single-qubit Haar measures instead of a single joint two-qubit Haar measure, it can be computed exactly using the set of tensor products of all six Pauli eigenstates $\{|0\rangle, |1\rangle, |\pm\rangle, |\pm i\rangle\}$. Of the resulting 36 two-qubit product states, 16 are maximally entangled, while 20 remain separable for gates in the class of locally equivalent CNOT operations, including the $ZX(\pi/2)$ gate. Within the scope of our work, the linear entropy averaged over these 16 initial states is sufficient to capture the entangling power of a unitary operation. We shall therefore define this quantity as the *average linear entropy* $\bar{S}_{\text{lin}}$.

In the context of driving one qubit at a frequency of another in order to implement an entangling gate, it is common to attribute the entanglement generated entirely to the use of such a CR drive. However, this might not be the case when an on-resonance drive is used simultaneously with the CR drive. As we shall see, studying the average linear entropy $\bar{S}_{\text{lin}}$ of optimized pulses implementing two-qubit gates reveals that in some of the control solutions discovered by RL, the roles of different drives are not as isolated as one might initially believe; e.g. see figure 9, indicating the existence of an entirely new class of solutions.
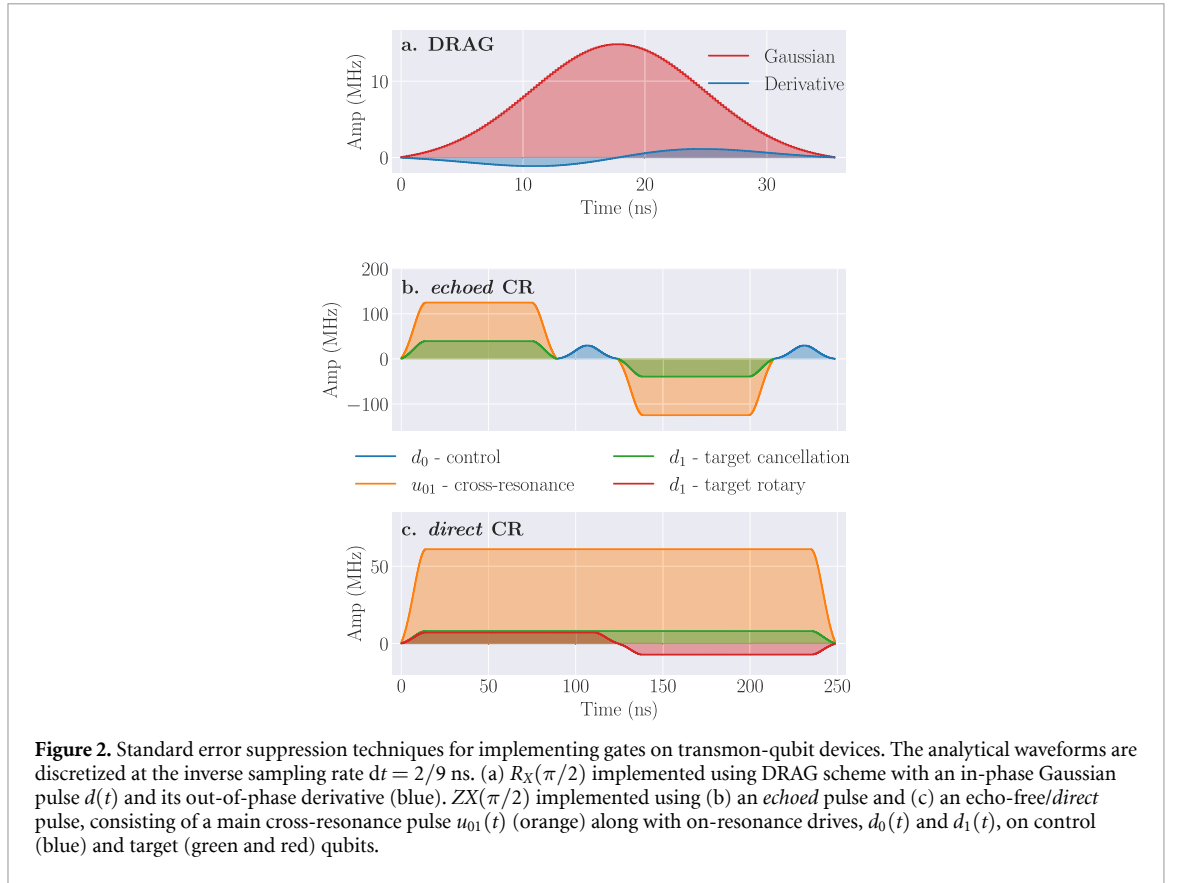
3. State-of-the-art implementations of transmon gates

To provide a meaningful point of comparison for our approach, we review the conventional methods for implementing quantum gates within a superconducting platform and specifically the theoretical foundations underpinning each ansatz. As concrete examples, we showcase the standard error suppression techniques for both the single-qubit $R_X(\pi/2)$ gate and the two-qubit $ZX(\pi/2)$ gate in figure 2.

The most basic implementation of a single-qubit rotation around the x-axis involves driving the target transmon resonantly with a real-valued pulse envelope $d(t)$, according to equation (6). Common choices for the pulse shape include Gaussian and Gaussian Square waveforms as they offer smooth ramp-up and ramp-down. The standard error suppression approach utilizes an additional out-of-phase component equal to the derivative of the in-phase part, see figure 2(a), which has been shown to significantly reduce gate error including leakage to the second excited level. This is known as the Derivative Removal for Adiabatic Gate (DRAG) scheme [38, 39], in which the amplitude of the real Gaussian pulse, the detuning, and the scaling factor of the imaginary derivative component can be optimized. Calibration of these parameters on current superconducting hardware can reliably achieve average gate fidelity of above 99.95% [40].

For two-qubit entangling gates such as $ZX(\pi/2)$, the standard implementation makes use of a CR pulse u_{01} along with resonant drives (d_0, d_1) on both the control and target qubit, according to equation (10). These components can be combined in an *echoed* or *direct* fashion [40]. As illustrated in figure 2(b), the *echoed* scheme employs an echo pulse sequence where the CR pulse is broken into two halves (yellow envelopes) with the second one inverted ($\Omega \rightarrow -\Omega$) and positioned between two π-pulses applied to the control qubit

Figure 2. Standard error suppression techniques for implementing gates on transmon-qubit devices. The analytical waveforms are discretized at the inverse sampling rate $dt = 2/9$ ns. (a) $R_X(\pi/2)$ implemented using DRAG scheme with an in-phase Gaussian pulse $d(t)$ and its out-of-phase derivative (blue). $ZX(\pi/2)$ implemented using (b) an *echoed* pulse and (c) an echo-free/*direct* pulse, consisting of a main cross-resonance pulse $u_{01}(t)$ (orange) along with on-resonance drives, $d_0(t)$ and $d_1(t)$, on control (blue) and target (green and red) qubits.

$$\mathrm{CR}\left(\Omega\right) \cdot \left[XI \cdot \mathrm{CR}\left(-\Omega\right) \cdot XI\right]. \tag{16}$$

The amplitude inversion changes the sign of ω_{ZX} and ω_{IX} according to the relations in equation (13), while the addition of two π-pulses can be understood as a conjugation by XI for every term in the effective CR Hamiltonian, leading to the following contribution from the second half

$$
\begin{bmatrix} ZX \\ IX \\ ZI \\ ZZ \end{bmatrix}
\xrightarrow{\Omega \rightarrow -\Omega}
\begin{bmatrix} -ZX \\ -IX \\ ZI \\ ZZ \end{bmatrix}
\xrightarrow{\text{conj. by } XI}
\begin{bmatrix} ZX \\ -IX \\ -ZI \\ -ZZ \end{bmatrix}. \tag{17}
$$

When combined with the first half, this should ideally lead to a complete cancellation of unwanted IX, ZI, and ZZ terms. Nevertheless, experimental results reveal a significant IY component as well as a small ZY term, which can be attributed to classical crosstalk. This issue can be rectified by applying an on-resonance tone to the target qubit with an identical waveform as the CR pulse, known as active cancellation [24]. On the other hand, the *direct* scheme employs an echo-free sequence with the same symmetric active cancellation tone, while introducing an additional asymmetric rotary component. In particular, the symmetric part reduces the effects of IX and IY terms whereas the asymmetric part helps offset ZZ and ZY terms. For both schemes, calibration of the amplitudes and phases of the main CR pulse, in tandem with calibration of the additional tones, achieves between 99% to 99.7% average gate fidelity [24, 40].

As seen in the above examples, the standard pulse designs rely heavily on a theoretical understanding of the platform, i.e. types of interaction induced when certain control drives are activated or when certain error processes are present. On a real device, however, deviation from the theoretical model is unavoidable and closed-loop optimization is required to mitigate the unwanted effects. Additionally, the perturbative approach of deriving the effective interaction rates break down at high control amplitudes, preventing exploration for potential solutions in the strong drive and short time regime.

While these theoretical ansätze offer the advantage of straightforward calibration procedures with a minimal number of parameters, they may also impose significant limitations and/or necessitate longer gate durations to compensate for errors not captured by the relevant theoretical model. Moreover, should previously unidentified errors come to light, it will be necessary to develop and implement novel error suppression strategies. Established alternative approaches usually involve gradient-based optimization, such

as GRAPE [3], which still requires detailed knowledge about the model and access to the gradient of the loss function. A model-free approach like RL is therefore highly desirable since it offers adaptability to system dynamics by learning from 'direct interactions' which we will define in the next section. Even when equipped with a relatively simple but flexible design space, such as PWC pulses, RL has the potential to unearth control solutions that are out of reach in conventional methods [10]. Furthermore, RL leaves us with a representation of gained knowledge from the control problem, i.e. the agent, that can be reused and analyzed for additional insights (cf section 5).

4. Reinforcement Learning

Reinforcement learning operates on a simple principle of trial and error. A generic problem involves an agent learning to make decisions to complete a task by interacting with an environment. Therefore, it is natural to formulate a RL problem using a finite Markov Decision Process, in which a decision is made based solely on the current state of the system but not the entire history. We illustrate the basic RL loop in figure 3. In this framework, at every step i, the agent selects an action a_i based on a probability distribution or *policy* $\pi(a_i|s_i)$, conditioned on the current state of the environment, s_i. After execution, the agent observes a new state s_{i+1} along with a reward r_{i+1} which indicates progress towards completing a particular task. The process terminates once the task is completed or the number of steps reaches a set limit, defining the end of an *episode*. Training the agent then involves running many of these episodes to gather experience, while exploration is encouraged by adding randomness to the action selection procedure, the *trial* process. At the same time, the policy is iteratively adjusted to maximize the expected cumulative reward $\mathbb{E}\left[\sum_i r_{i+1}\right]$ at the end of each episode, guiding the agent away from unproductive actions, the *error* process. Together, trial and error allow the RL agent to explore new actions effectively, and eventually arrive at a highly-rewarded behavior.
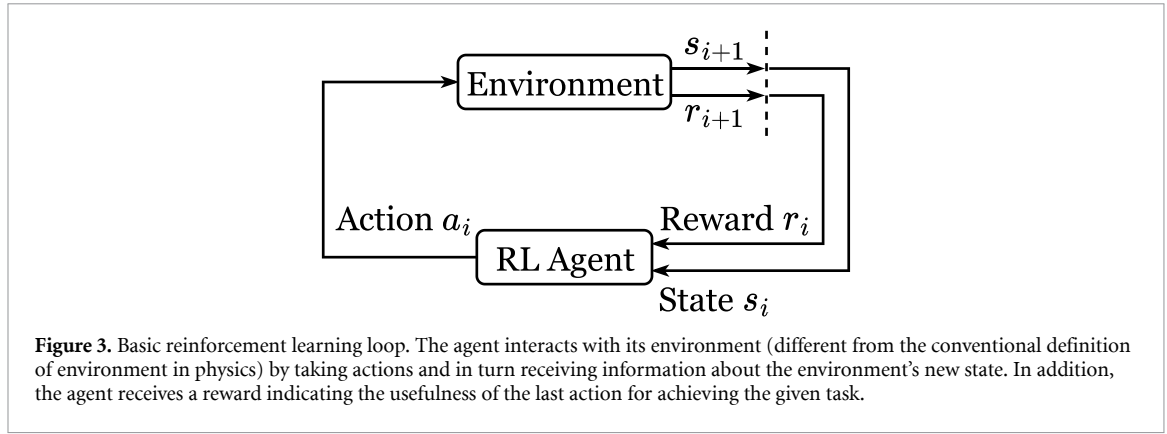
An agent trained exclusively on a single environment typically excels only within that specific context, making it less adaptable when confronted with a new environment. To mitigate this limitation, one effective strategy is to expand the agent's training scope to encompass a variety of environments. Moreover, equipping the agent with some *context* information about its current environment can significantly enhance its learning process and overall decision-making capability. This framework is commonly referred to as RL with *context* [41], and it has been demonstrated as particularly valuable for tasks that require generalization to a range of environment parameters.

We now adapt the RL framework to designing quantum gates, in which we task an agent to build a PWC pulse to realize a target operation on a transmon environment, as illustrated in figure 4. In the following, we detail our simulated environment, motivate our choice of states, actions, and rewards, and describe the selected RL algorithm.

4.1. Environment

Our environment simulates the dynamics of two transmons according to the Hamiltonian in equation (10), considering them as directly coupled anharmonic oscillators which can be controlled via external microwave pulses (cf 'Environment' box in figure 4). The first two levels of the oscillators act as qubits and the main contribution of leakage out of the qubit subspace is captured via inclusion of the third level. The environment is then completely characterized by a set of system parameters, including detuning $\{\delta_0, \delta_1\}$, anharmonicity $\{\alpha_0, \alpha_1\}$, control drive strength $\{\Omega_{d_0}, \Omega_{u_{01}}, \Omega_{d_1}, \Omega_{u_{10}}\}$, and coupling J, which we collect into a single vector $\vec{p} = [J, \Omega_{u_{01}}, \ldots]$. We denote the main set of system parameters used in this work as $\vec{p}_0$ whose components are summarized in table 2. Any drifts in the system characteristics are considered w.r.t to $\vec{p}_0$ via the relative change $\Delta\vec{p}/\vec{p}_0 = (\vec{p} - \vec{p}_0)/\vec{p}_0$, where we have used an element-wise vector division.

Action: The RL agent interacts with the transmon environment by directly modifying the complex-valued control pulses $[u_{01}(t), d_1(t), \ldots]$. Using the PWC ansatz, pulse shaping is equivalent to picking an amplitude A at each discrete time step Δt until an N-segment pulse is complete, resulting in a gate duration $T = N\Delta t$. To maintain experimental viability and avoid unrealistic oversampling, the time step is chosen such that $1/\Delta t$ be below the sampling rate (and bandwidth) of standard control electronics [42], i.e. $\Delta t > \mathrm{d}t = 2/9$ ns in this work. From the RL point of view, each complete pulse constitutes an episode, after which, the environment is reset so a new pulse can be tried out. Allowing the pulse amplitude A to take any value at every step tends to result in highly volatile pulses, similar to those typically obtained from unconstrained optimal control using methods like GRAPE [3]. Instead, we aim for slowly varying solutions by defining the agent's action to be the relative amplitude change and restrict it to some continuous window $[-w, w]$. By setting $u_{01,i} \equiv u_{01}(i\Delta t)$ and $d_{1,i} \equiv d_1(i\Delta t)$, we can write the action at step i as

Figure 3. Basic reinforcement learning loop. The agent interacts with its environment (different from the conventional definition of environment in physics) by taking actions and in turn receiving information about the environment's new state. In addition, the agent receives a reward indicating the usefulness of the last action for achieving the given task.

$$
\begin{aligned}
a_i \;&=\; A_i - A_{i-1} \\[4pt]
\Rightarrow
\begin{bmatrix}
a_i^{(1)} \\
a_i^{(2)} \\
a_i^{(3)} \\
a_i^{(4)} \\
\vdots
\end{bmatrix}
\;&=\;
\begin{bmatrix}
\mathrm{Re}\,(u_{01,i}) \\
\mathrm{Im}\,(u_{01,i}) \\
\mathrm{Re}\,(d_{1,i}) \\
\mathrm{Im}\,(d_{1,i}) \\
\vdots
\end{bmatrix}
-
\begin{bmatrix}
\mathrm{Re}\,(u_{01,i-1}) \\
\mathrm{Im}\,(u_{01,i-1}) \\
\mathrm{Re}\,(d_{1,i-1}) \\
\mathrm{Im}\,(d_{1,i-1}) \\
\vdots
\end{bmatrix},
\end{aligned}
\tag{18}
$$

where each component is restricted to a drive-dependent window

$$
\left| a_i^{(1)} \right|, \left| a_i^{(2)} \right| \leqslant w_u, \qquad \left| a_i^{(3)} \right|, \left| a_i^{(4)} \right| \leqslant w_d.
\tag{19}
$$

Hence, the action space dimension corresponds to twice the number of available control fields, as shown in figure 4(a). By choosing the windows w_u and w_d to be small, e.g. less than 10% of the maximum allowed range, we systematically restrict the action space which additionally improves learning. Finally, we clip the resultant amplitudes to $[-1, 1]$ to ensure that the control fields do not exceed the maximum allowed drive strengths.

State: The evolution of the transmon system due to external control fields and internal dynamics is characterized by a unitary map $U(t,0)$ computed from Hamiltonian in equation (10). Given a set of basis states $\{\psi_j\}$, the evolution of an arbitrary pure initial state reads

$$
U(t,0)\,|\psi\rangle = \sum_j c_j U(t,0)\,|\psi_j\rangle = \sum_j c_j |\psi_j(t)\rangle.
\tag{20}
$$

Thus, tracking the time-evolved unitary map is equivalent to tracking the time-evolved basis states $\{\psi_j(t)\}$. As we aim to design target operations between two-level systems, we assume no occupation beyond the qubit subspace initially. That means, for example, in the single-qubit gate case, it is sufficient to track only the following basis states

$$
|0\rangle = (1,0,0)^T, \quad |1\rangle = (0,1,0)^T,
\tag{21}
$$

where we have truncated our simulation at three levels. Complete knowledge of the evolved basis states $\{\psi_j(t)\}$ at every step allows the agent to discern the effect of its actions on the environment. Due to our restriction of the action space to contain only relative changes in the control fields as in equation (18), we also include the pulse amplitudes in the previous time step to the state provided to the agent:

$$
s_i = \left[\{\psi_j(t = i\Delta t)\}, A_{i-1} \right].
\tag{22}
$$

In addition to designing gates for a fixed environment, we also wish to generalize the agent's designing capability to environments where the system parameters have drifted from their original values. While the agent can indirectly discern this change through the evolution of the quantum state, we have observed that furnishing the agent with explicit information about the current system characteristics can enhance its learning process. Instead of directly feeding the agent the system parameter vector $\vec{p}$ whose entries can take a
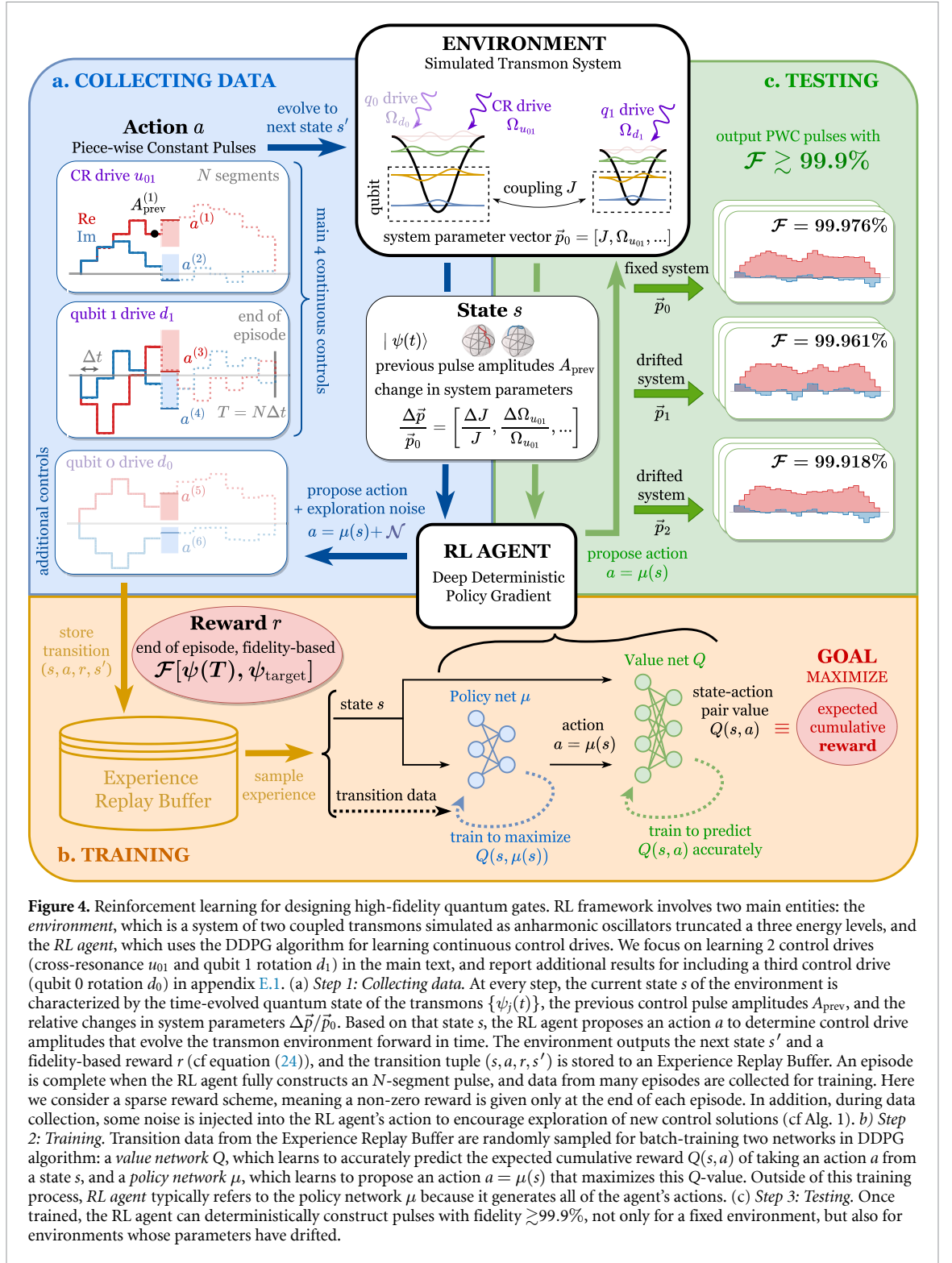
Figure 4. Reinforcement learning for designing high-fidelity quantum gates. RL framework involves two main entities: the *environment*, which is a system of two coupled transmons simulated as anharmonic oscillators truncated a three energy levels, and the *RL agent*, which uses the DDPG algorithm for learning continuous control drives. We focus on learning 2 control drives (cross-resonance u_{01} and qubit 1 rotation d_1) in the main text, and report additional results for including a third control drive (qubit 0 rotation d_0) in appendix E.1. (a) *Step 1: Collecting data*. At every step, the current state s of the environment is characterized by the time-evolved quantum state of the transmons $\{\psi_j(t)\}$, the previous control pulse amplitudes A_{prev}, and the relative changes in system parameters $\Delta \vec{p}/\vec{p}_0$. Based on that state s, the RL agent proposes an action a to determine control drive amplitudes that evolve the transmon environment forward in time. The environment outputs the next state s' and a fidelity-based reward r (cf equation (24)), and the transition tuple (s, a, r, s') is stored to an Experience Replay Buffer. An episode is complete when the RL agent fully constructs an N-segment pulse, and data from many episodes are collected for training. Here we consider a sparse reward scheme, meaning a non-zero reward is given only at the end of each episode. In addition, during data collection, some noise is injected into the RL agent's action to encourage exploration of new control solutions (cf Alg. 1). b) *Step 2: Training*. Transition data from the Experience Replay Buffer are randomly sampled for batch-training two networks in DDPG algorithm: a *value network Q*, which learns to accurately predict the expected cumulative reward $Q(s, a)$ of taking an action a from a state s, and a *policy network μ*, which learns to propose an action $a = \mu(s)$ that maximizes this Q-value. Outside of this training process, *RL agent* typically refers to the policy network μ because it generates all of the agent's actions. (c) *Step 3: Testing*. Once trained, the RL agent can deterministically construct pulses with fidelity $\gtrsim 99.9\%$, not only for a fixed environment, but also for environments whose parameters have drifted.

wide range of values, we can provide the same *context* information via the relative change in system parameters $\Delta \vec{p}/\vec{p}_0$, transforming the RL input state into

$$s_i \leftarrow \left[s_i, \frac{\Delta \vec{p}}{\vec{p}_0} \right], \tag{23}$$

where $\vec{p}_0$ denotes the original values in table 2.

Reward: RL approaches typically utilize a reward that is provided at every step to incentivize the agent to learn the correct actions. Alternatively, the agent can also learn from a single reward granted at the end of each episode. In the case of a fidelity-based objective and when considering a closed-loop implementation using an actual device, this sparse reward scheme demands fewer measurements during intermediate steps,

making it more experimentally favorable. With this in mind, we have opted for the sparse reward scheme and have determined that it is adequate for the agent's learning process. As the fidelity approaches unity, improvements tend to slow down, yielding increasingly marginal gains. To enhance the discernibility of positive signals, we define the reward function to be the negative log infidelity at the final time step

$$r_N = -\log_{10}(1 - \mathcal{F}). \tag{24}$$

Here, it is important to note that an improvement of one unit in the reward corresponds to a one-order-of-magnitude enhancement in fidelity, e.g. $r : 2 \rightarrow 3$ corresponds to $\mathcal{F} : 0.99 \rightarrow 0.999$.

4.2. Algorithm

In our pursuit of designing quantum gates via pulse shaping, we have established a large design space of PWC functions for the RL agent to explore. We require an algorithm capable of handling continuous-valued actions to fully harness the flexibility of this design space for achieving high-fidelity solutions, and also to effectively utilize continuous control resources in realistic hardware. Furthermore, given the limited access to near-term quantum devices, an algorithm with efficient training data usage is highly desirable. Thus, we select the Deep Deterministic Policy Gradient algorithm (DDPG), which satisfies all of these criteria [43].

We begin by laying the groundwork for DDPG, which is rooted in the concept of Q-learning. A Q-value quantifies the expected cumulative reward associated with taking an action from a specific state and subsequently following a particular policy π thereafter. In RL, the expected cumulative reward is commonly subject to discounting over future time steps in order to incentivize the agent to complete its objective faster to receive a higher reward. Formally, the Q-value for a state-action pair (s_i, a_i) at time step i under a policy $\pi(a|s)$ is defined as follows:

$$
\begin{aligned}
Q^\pi(s_i, a_i) &= \mathbb{E}_\pi[r_{i+1} + \gamma r_{i+2} + \ldots | s_i, a_i] \\
&= r_{i+1} + \gamma \mathbb{E}_\pi[r_{i+2} + \ldots] \\
&= r_{i+1} + \gamma \sum_{a_{i+1}} \pi(a_{i+1}|s_{i+1}) Q^\pi(s_{i+1}, a_{i+1}).
\end{aligned}
\tag{25}
$$

Here, $\gamma \in [0, 1]$ is the discount factor, and the expectation value $\mathbb{E}$ is taken over actions selected using the policy π. With the current action a_i already selected, there is only one possible value for the immediate reward $r_{i+1} = r_{i+1}(s_i, a_i)$, allowing us to take it out of the expectation and substitute it in the Q-value definition for the next state-action pair. The optimal strategy is then to pick the highest-valued action at every step, which leads to the recursion relation for the optimal Q-value Q^*:

$$
\begin{aligned}
Q^*(s_i, a_i) &= \max_\pi Q^\pi(s_i, a_i) \\
&= r_{i+1} + \gamma \max_a Q^*(s_{i+1}, a),
\end{aligned}
\tag{26}
$$

also known as the Bellman optimality equation [44]. As the dependence on the policy π is removed in the above, the optimal Q-value can be iteratively updated using any transition data tuple $(s_i, a_i, s_{i+1}, r_{i+1})$ regardless of the collecting policy, a process commonly known as off-policy training. In practice, observed transitions are stored in a replay buffer from which a mixture of new and old transitions are sampled to train the agent. A typical replay buffer stores about half a million transitions, allowing much more efficient re-use of old data as compared to other RL methods.

When the number of discrete states and discrete actions are not too large, the corresponding Q-values are stored in a finite-size table that can be iteratively updated. As the state space becomes continuous, one instead approximates the optimal Q-value by a deep neural network with parameters ϕ as $Q_\phi \approx Q^*$. To adapt this deep Q-learning method to continuous actions, DDPG additionally utilizes a deterministic policy network with parameters θ for action generation: $a_i = \mu_\theta(s_i)$. During training, a noise process is injected into the agent's action to encourage exploration as seen in figure 4(a). With the main goal of maximizing the expected cumulative reward, we want not only a policy network that can generate actions with high Q-values, but also a value network that can approximate the Q-values well according to equation (26), which leads to the following update rules:

$$
\begin{aligned}
Q_\phi(s_i, a_i) &\leftarrow r_{i+1} + \gamma Q_\phi(s_{i+1}, \mu_\theta(s_i)), \\
\mu_\theta(s_i) &\leftarrow \arg\max_{a_i} Q_\phi(s_i, a_i),
\end{aligned}
\tag{27}
$$

for each transition data tuple $(s_i, a_i, s_{i+1}, r_{i+1})$. From these update rules, it is clear that updating one network changes the loss function of the other, creating a moving target problem. Therefore, when computing the

targets on the right hand sides of equation (27), we employ target networks (ϕ', θ') that slowly track the learned networks (ϕ, θ), which minimizes the effect of fast-moving targets. The complete algorithm is detailed in appendix B.

DDPG is known to struggle when the action space gets too large, which leads to exploding Q-values during training. One solution is to train two Q-networks and use the smaller value for computing the targets in equation (27) to mitigate Q-value overestimation (a.k.a., twin network trick). Another solution is to delay the policy network update for better Q-network convergence in between (a.k.a. delayed policy trick). These techniques, when combined with the standard DDPG, result in an augmented algorithm commonly known as Twin Delayed DDPG (TD3) [45]. Unfortunately, we could not obtain conclusive evidence as to whether TD3 outperforms DDPG in all situations for our problem. Therefore, in the main text we focus on DDPG and delegate discussion of a case in which TD3 provides an advantage to appendix E.1.

5. Results

Here we report our main results of utilizing the DDPG algorithm for continuous control to solve the two-qubit gate design problem for superconducting transmon qubits. To ensure the consistency and reproducibility of the RL approach, we repeat each case multiple times with different seeds. Each seed leads to a different set of random generators used for initializing and optimizing of neural network parameters, sampling experience from the replay buffer, and injecting exploration noise to agent action during training. Unless stated otherwise, the reported results appear typical within a handful of realizations. The best cases are discussed here in the main text while training data over multiple seeds are included in appendix C. We summarize relevant training hyperparameters in table 1 as well as the main set of system parameters in our quantum simulator in table 2.

We juxtapose the RL-designed strategies with the conventional error suppression schemes in section 3 where we employ the Nelder–Mead method to optimize the relevant parameters in each ansatz to maximize the average gate fidelity. For the single-qubit DRAG scheme, we simultaneously optimize 2 parameters for $d(t)$: the amplitude of the real Gaussian pulse and the scaling factor of the imaginary derivative pulse. For two-qubit entangling gates, we find that first optimizing the amplitudes of the Gaussian Square pulses for $u_{01}(t)$ and $d_1(t)$, and then optimizing for their phases, yields the best result. In particular, the *echoed* scheme consists of two tunable Gaussian Square pulses, CR (u_{01}) and target cancellation tone (symmetric part of d_1), constituting a 4-parameter optimization problem. Meanwhile, the *direct* scheme contains an additional target rotary tone (asymmetric part of d_1), increasing the number of optimizable parameters to 6. These are in contrast with the 18-dimensional (single-qubit gate) and 80-dimensional (two-qubit gate) control problems addressed by our RL approach that we will see shortly.

Our main results can be summarized as follows. First, we demonstrate that an RL agent can be trained via direct interaction with a simulated environment to successfully explore the vast design space of PWC functions (section 5.1). Although the environment is treated as a black box, the PWC time step is chosen such that $1/\Delta t$ is below the sampling rate (and bandwidth) of standard control electronics [42], to avoid unrealistic oversampling. The discovered strategies are unbiased by prior theoretical knowledge and distinct from the established analytical solutions in section 3. Second, we illustrate the benefit of RL optimization with the flexible PWC ansatz in finding high-fidelity control solutions at shorter gate duration (section 5.2). We then assess the novelty in the roles played by each drive (section 5.3), followed by the robustness of optimized pulses to short-timescale stochastic noise (section 5.4). Finally, we augment our agent to generalize and adapt to drifts in system characteristics (section 5.5), making use of the left-over representation of gained knowledge which is an advantage of RL over conventional control algorithms.

5.1. Learning without prior knowledge

5.1.1. Single-qubit gate in two-transmon setting
As a benchmark for our approach, we start by tasking the RL agent to learn the single-qubit $\pi/2$ rotation around the x-axis in a two-transmon setting, given by

$$IX(\pi/2) \equiv I \otimes R_X(\pi/2) = I \otimes \left(\frac{I - iX}{\sqrt{2}} \right). \tag{28}$$

In figure 5, we report an RL-designed 9-segment pulse that implements the $IX(\pi/2)$ gate using a single control drive d_1. The agent learns to construct both real and imaginary parts of the pulse, thus tackling an 18-dimensional optimization problem. The RL-designed pulse achieves a 10 ns gate duration which is over $3\times$ faster than the 35.6 ns DRAG pulse. With triple the maximum pulse amplitude, the 10 ns pulse maintains a comparable fidelity at 99.9%, despite having leakage larger by a few orders of magnitude during intermediate steps, as seen in figure 5(d). The data, therefore, suggest that the RL agent learns to exploit the

Figure 5. Optimization for the single-qubit gate $X(\pi/2)$ in two-transmon setting. (a) IBM 35.6 ns DRAG pulse. (b) 10 ns RL-optimized pulse with training hyperparameters given in the 'Fixed Environment' section of table 1. (c) Corresponding evolution of Bloch coordinates for the controlled qubit. (d) Population leakage to $|2\rangle$ for RL pulse is up to a few orders of magnitude higher compared to DRAG during the evolution. RL-optimized pulse is $3\times$ faster with similar average gate fidelity above 99.9%, and makes use of the presence of level $|2\rangle$, at the expense of accessing three times larger amplitudes.

presence of the second level to the advantage of reducing the gate duration. Such speed-up already offers a significant reduction in operating time as general quantum circuits consist mostly of single-qubit gates.

5.1.2. Two-qubit entangling gates

Applying the same algorithm for 20-segment waveforms, our RL agent successfully learns 248.9 ns pulses that implement $ZX(\pi/2)$ and CNOT gates completely from the ground up, achieving fidelity $\mathcal{F} > 99.9\%$. The agent constructs complex-valued pulses for a cross-resonance drive u_{01} and an on-resonance drive on the target transmon d_1, constituting an optimization problem of dimension 80 (20 segments $\times$ 2 drives$\times$ 2 real numbers). In fact, our agent also finds equally high-fidelity solutions to an even higher-dimensional optimization problem whe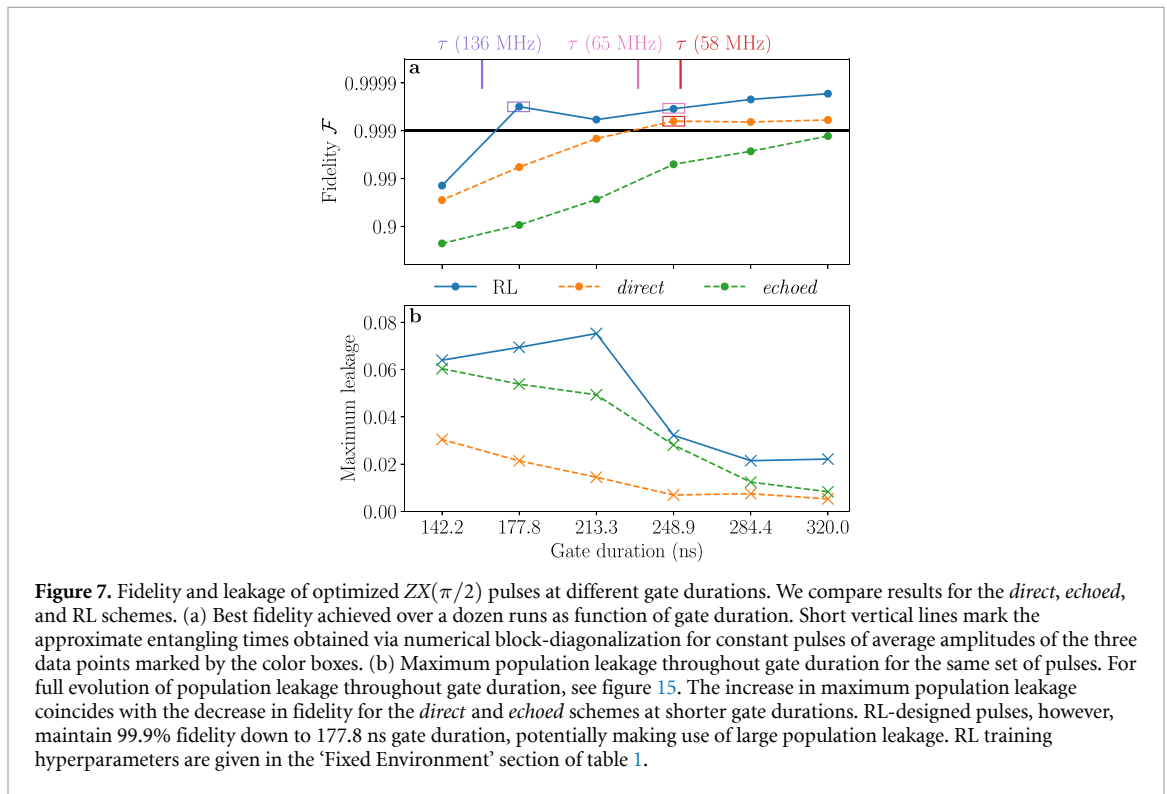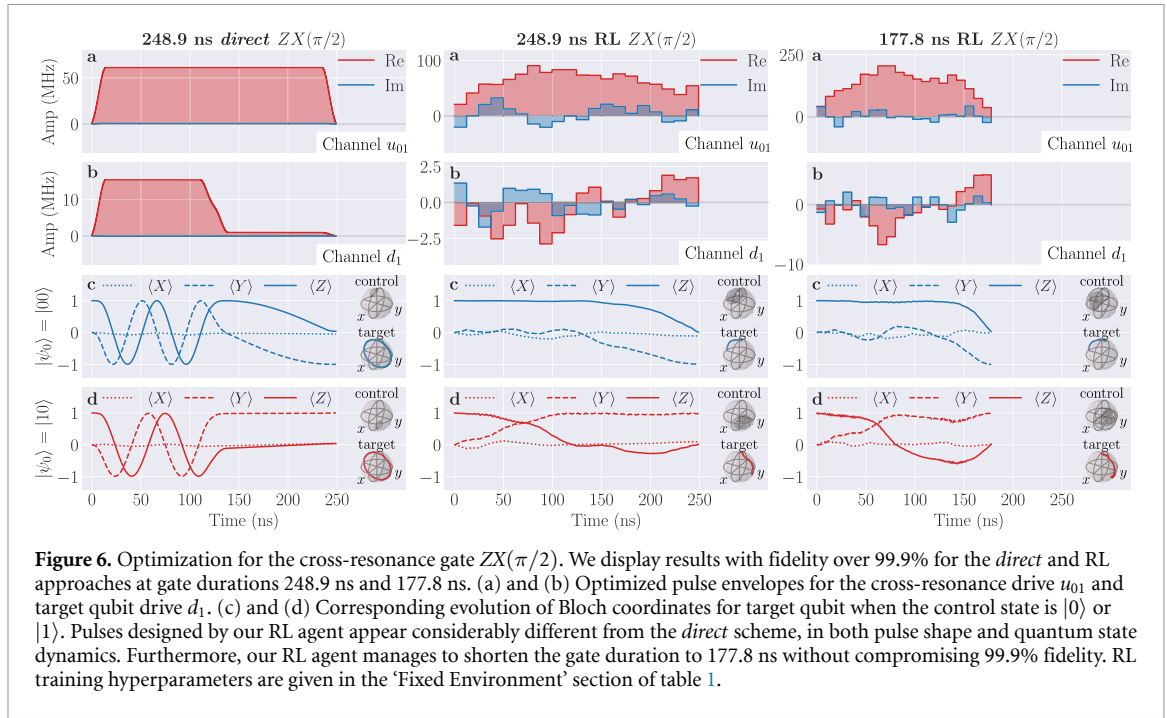n given access to three drives (d_0, u_{01}, d_1). However, these 3-drive control solutions require the ability to send two pulses at different frequencies simultaneously to the same transmon, which, to the best of our knowledge, is not a commonly used technique. Therefore, we defer the discussion of 3-drive results to appendix E.1 and focus on constructing pulses using the only two drives (u_{01}, d_1) for our main results section.

In figure 6, we present a clear-cut contrast between state-of-the-art *direct* scheme and RL approach for the $ZX(\pi/2)$ gate. First, the RL pulse envelope goes beyond the square Gaussian structure in the *direct* scheme while having a higher maximum amplitude and a more prominent imaginary part, as seen in figures 6(a) and (b). In figures 6(c) and (d), we illustrate the evolution of Bloch coordinates of the target qubit initialized at $|0\rangle$, for two cases: when the initial control qubit state is either $|0\rangle$ (blue) or $|1\rangle$ (red). Maximal entanglement is achieved when these two time-evolved target qubit states are exactly opposite on the Bloch sphere. In the *direct* scheme, the conditioned target qubit states start out by rotating together around the x-axis at slightly different speeds. They increase their distance after a couple of revolutions and stop at their final destinations on each end of the y-axis. By contrast, the RL scheme appears to bring the states directly to their respective destinations with minor course corrections in between; this has to do with our choice of allowed action windows w_u and w_d between the u_{01} and d_1 drives. We find that setting $w_u = 10w_d$, in this case, yields the best training performance, which inadvertently restricts our agent to solutions with a considerably weaker drive on the target qubit. More interestingly, the evolution roughly splits into two parts (cf middle and right columns of figures 6(c) and (d): the state conditioned on 1 (red) moves while the state conditioned on 0 (blue) remains approximately stationary in the first half of the pulse sequence, and vice versa in the second. The above observations suggest that the strategies learned by our RL agent are fundamentally different from the standard analytical protocols.

Figure 6. Optimization for the cross-resonance gate $ZX(\pi/2)$. We display results with fidelity over 99.9% for the *direct* and RL approaches at gate durations 248.9 ns and 177.8 ns. (a) and (b) Optimized pulse envelopes for the cross-resonance drive u_{01} and target qubit drive d_1. (c) and (d) Corresponding evolution of Bloch coordinates for target qubit when the control state is $|0\rangle$ or $|1\rangle$. Pulses designed by our RL agent appear considerably different from the *direct* scheme, in both pulse shape and quantum state dynamics. Furthermore, our RL agent manages to shorten the gate duration to 177.8 ns without compromising 99.9% fidelity. RL training hyperparameters are given in the 'Fixed Environment' section of table 1.

Figure 7. Fidelity and leakage of optimized $ZX(\pi/2)$ pulses at different gate durations. We compare results for the *direct*, *echoed*, and RL schemes. (a) Best fidelity achieved over a dozen runs as function of gate duration. Short vertical lines mark the approximate entangling times obtained via numerical block-diagonalization for constant pulses of average amplitudes of the three data points marked by the color boxes. (b) Maximum population leakage throughout gate duration for the same set of pulses. For full evolution of population leakage throughout gate duration, see figure 15. The increase in maximum population leakage coincides with the decrease in fidelity for the *direct* and *echoed* schemes at shorter gate durations. RL-designed pulses, however, maintain 99.9% fidelity down to 177.8 ns gate duration, potentially making use of large population leakage. RL training hyperparameters are given in the 'Fixed Environment' section of table 1.

5.2. Achieving shorter gate duration

By extending our gate design study to different gate durations, we find that the RL approach coupled with the flexibility of the PWC waveform consistently outperforms optimized *direct* and *echoed* schemes. In figure 7(a), we observe that fidelities obtainable using these standard approaches drop below 99.9% when their gate durations approach 213 ns and 320 ns, respectively. Meanwhile, RL pulse duration can be shortened significantly, down to 177.8 ns while maintaining the same performance.

We first compare the gate durations of optimized control solutions with the *approximate entangling time* τ defined for a constant pulse of amplitude Ω. For the $ZX(\pi/2)$ gate, we have $\tau(\Omega) = (\pi/2)/\omega_{ZX}(\Omega)$, where ω_{ZX} is the effective ZX interaction rate obtained via numerical block-diagonalization of the two-transmon Hamiltonian in equation (11) (cf [20] for more details). We compute $\tau(\Omega)$ for the three cases considered in

Figure 8. Entanglement and rotation angles of optimized $ZX(\pi/2)$ pulses. We display results with fidelity over 99.9% for the *direct* and RL approaches at gate durations 248.9 ns and 177.8 ns. (a) Linear entropy S_{lin} evolution for the initial states $|00\rangle$ and $|10\rangle$, which are expected to remain separable. (b) Rotation angles evolution, cf equation (31), for two entangling interactions *not* present in the effective CR model given in equation (12). (c) and (d) Rotation angles evolution for several interaction types predicted by the effective CR model, including the main entangling process ZX and target qubit rotation angles IX, IY, and IZ. Deviation from zero entanglement for initial states $|00\rangle$ and $|10\rangle$ indicates the presence of entangling interactions beyond this model, such as the shown evolution of XX and YX interactions. While the ZX angle is accumulated in a similar manner, target qubit rotations in RL-designed solutions behave fundamentally differently w.r.t the monotonicity of those rotation angles in the *direct* scheme.
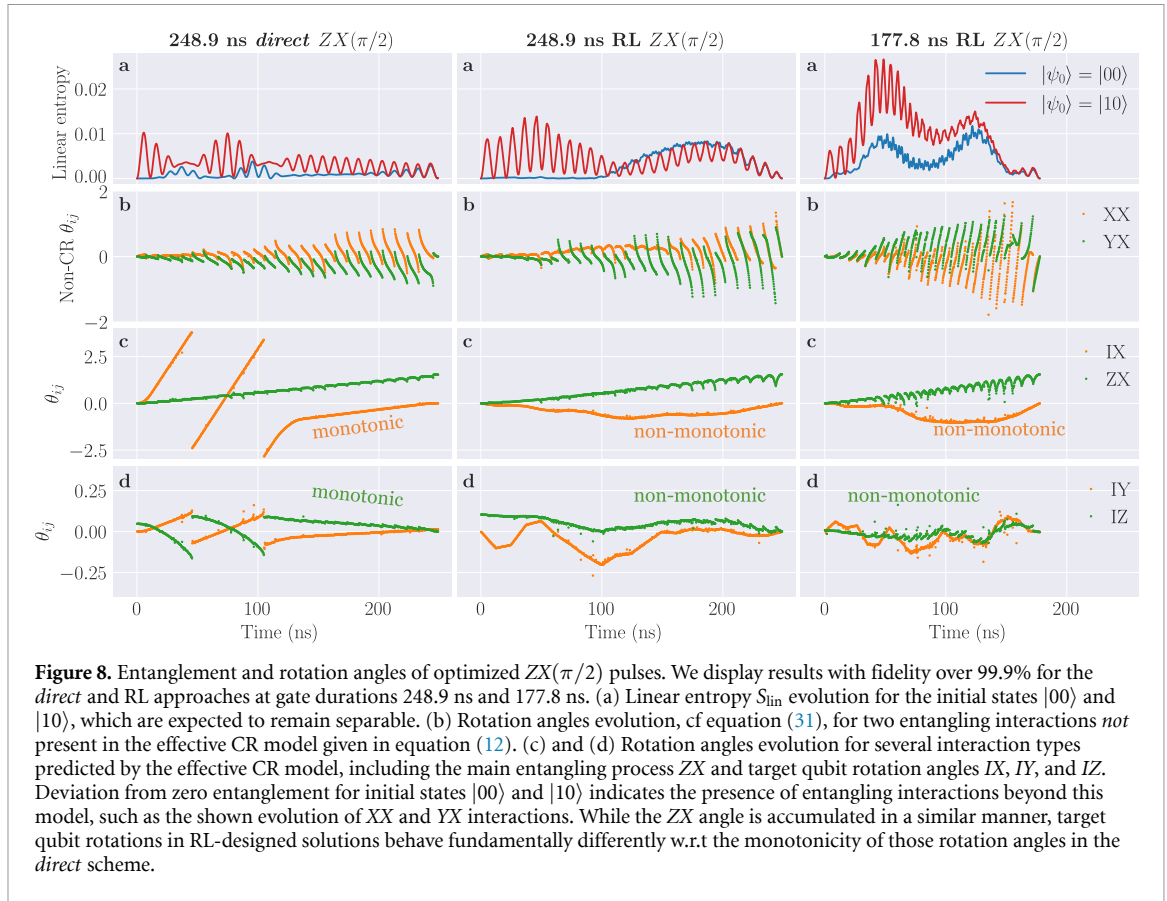
figure 6 using their average amplitudes, which are 58 MHz, 65 MHz, and 136 MHz, respectively. These approximate entangling times are displayed as short vertical lines in figure 7(a) and are color-coded to the corresponding points on the graph. On one hand, we observe that τ practically equals the gate duration for the 248.9 ns *direct* pulse, which is not surprising as its Gaussian Square waveform can be well-approximated with a constant pulse. On the other hand, for the RL pulses, the gate durations and approximate entangling times no longer agree, which can be attributed to their considerably more complicated PWC waveform. This observation suggests non-trivial dynamics in the control solutions discovered by our RL agent, which we further investigate in the following by examining the amount of population leakage as well as the evolution of entanglement generated in several initial states and rotation angles.

While the target operation involves only the first two levels, they are not isolated from other excited states and input control fields inevitably drive some population out of the computational subspace. In figure 7(b), we show the maximum leakage at different gate durations; and observe that an increase in leakage for *direct* and *echoed* pulses coincides with a decrease in fidelity. By contrast, RL-designed pulses manage to preserve their performance despite experiencing large state leakage which inevitably arises as the agent explores high-amplitude solutions in order to shorten the necessary entangling time. Our results suggest a prominent presence of leakage processes beyond the effective model, and that our RL agent has found a way to make good use of them. Indeed, such behavior is supported by prior research which indicates that the improved gate speed can be attributed to the more strongly coupled higher energy levels [46].

Within the computational subspace, we first note the deviation from the effective model by looking at the entanglement generated in two initial states, $|00\rangle$ and $|10\rangle$; they are expected to remain separable throughout the evolution under the effective CR Hamiltonian given in equation (12). We show the evolution of the linear entropy S_{lin} of these states in figure 8(a) for both *direct* and RL schemes, and observe small but non-zero amounts of entanglement. For the former, we observe more entanglement generated in the $|10\rangle$ state which can be attributed to pulse ramp up and ramp down not taken in account in the effective Hamiltonian analysis [20]. For RL-designed pulses, on the other hand, especially one with shorter gate duration, both states become considerably more entangled at intermediate time steps, indicating the presence of entangling processes beyond those expected from the effective CR model. This suggests that our RL agent has managed to remove these unwanted entangling processes at gate completion in order to achieve a high final fidelity.

For a more detailed picture of both single-qubit rotation and entangling processes, we take a closer look at the strengths of different interactions in the Pauli basis as a function of time. To do so, we first compute the

averaged Hamiltonian by taking the logarithm of the unitary $U(t,0)$ at time t and project it onto the qubit-subspace

$$tH_{\text{avg}}^{\text{qubit}} = \Pi^{\text{qubit}} \left[i \ln U(t,0) \right] \Pi^{\text{qubit}}. \tag{29}$$

We expand $tH_{\text{avg}}^{\text{qubit}}$ in the Pauli basis $P_i \in \{I, X, Y, Z\}$

$$tH_{\text{avg}}^{\text{qubit}} = \sum_{ij} \theta_{ij} \frac{P_i \otimes P_j}{2}, \tag{30}$$

where θ_{ij} defines the rotation angle that depends on the $P_i \otimes P_j$ interaction strength and duration t. We can then invert the relation and compute the rotation angle as

$$\theta_{ij}(t) = \text{Tr}\left[\Pi^{\text{qubit}} \left[i \ln U(t,0) \right] \Pi^{\text{qubit}} \frac{P_i \otimes P_j}{2} \right]. \tag{31}$$

Computing $\ln U(t,0)$ amounts to choosing an appropriate branch cut to obtain sensible results for the time-dependent rotation angles, a procedure we discuss in appendix D.2. It is important to note that these branch cuts are chosen to reveal a semi-stable increase in θ_{ZX}, which in turn, results in clear discontinuities in other angles. We can now analyze these rotation angles θ_{ij} to reveal the arisen interactions in greater depth.
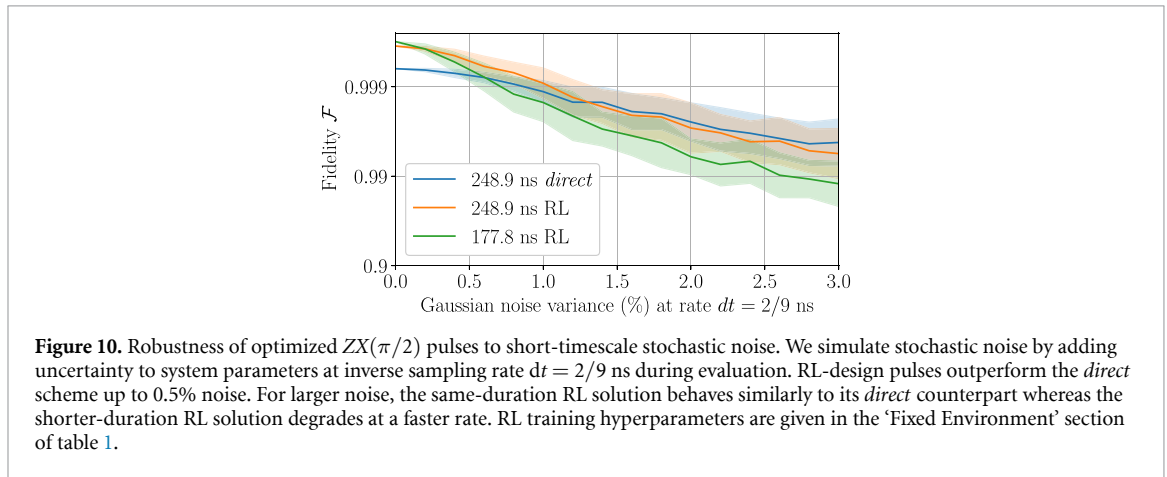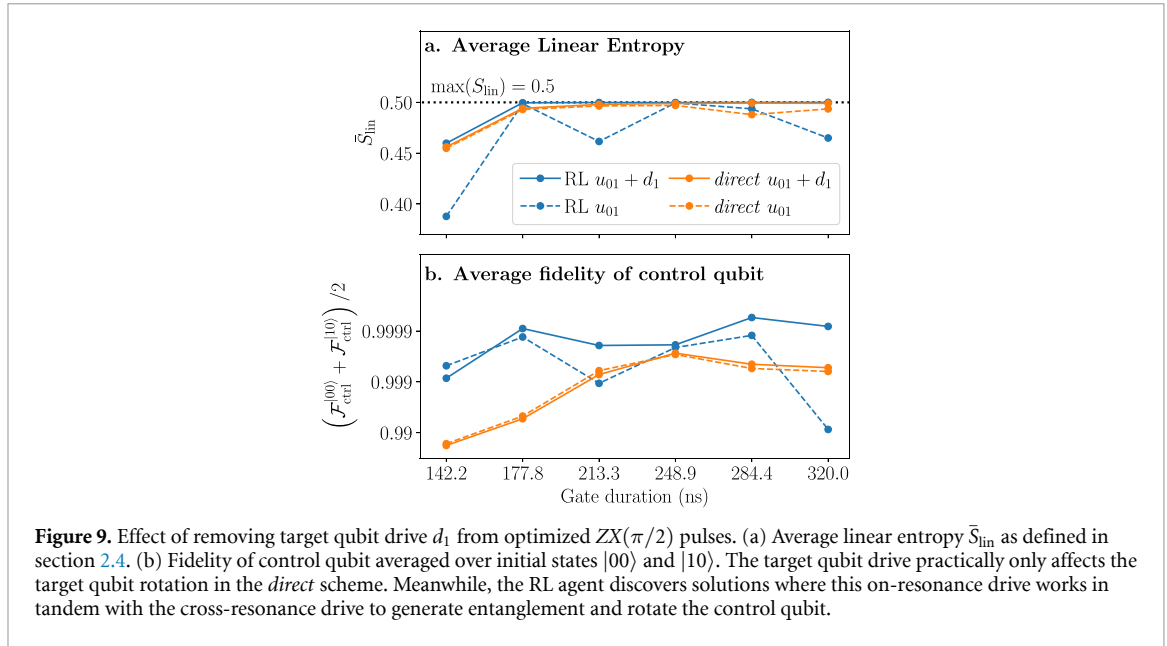
In figure 8(b), we show the time evolution of the rotation angles for the *XX* and *YX* interactions. Their non-negligible presence, even in the *direct* scheme, is not expected from the effective CR model given in equation (12), which further supports our previous observation of the linear entropy in figure 8(a). The saw-tooth time-evolution of these interactions corresponds to the off-resonance precession of the control qubit, and this pattern differs for all three cases presented. It is worth noting that the short 177.8 ns RL pulse results in the largest precession rate as well as the most prominent *XX* and *YX* interactions, which we attribute to its significantly higher drive amplitude as seen in figure 6. We thus confirm that the more complex waveforms in RL pulses feature an increased presence of entangling processes beyond the *ZX* term.

In figures 8(c) and (d), we also display several other interactions that are expected from the effective CR model, including *ZX*, *IX*, *IY*, and *IZ*. First, we observe a similar gradual accumulation of θ_{ZX} across the board. The target qubit rotations, however, look completely different. In the *direct* scheme, an active cancellation tone was introduced to mainly suppress a large unwanted *IX* term generated by the bare CR pulse. The inclusion of the asymmetric rotary part provides the additional degrees of freedom to suppress more unwanted terms. While having some success, these techniques are limited by the rigidity of the Gaussian Square waveform. This can be seen from *monotonic* evolution of the *IX*, *IY*, and *IZ* rotation angles in the *direct* pulse as illustrated in the left column of figures 8(c) and (d). By contrast, the evolution of these rotation angles in RL pulses, shown in the middle and right columns of figures 8(c) and (d), appears to be non-monotonic and significantly more flexible throughout the gate duration, suggesting a much more powerful error suppression potential resulting from the PWC waveform. Indeed, we observe that our RL agent successfully brings all unwanted interactions (cf figure 17) for the remaining terms close to zero at gate completion, even in the case of a significantly higher-amplitude 177.8 ns pulse where the effective CR model completely breaks down. These findings reveal a considerable deviation in environment characteristics beyond the perturbative effective dynamical model, and yet, our PWC-equipped model-free RL agent remains unbiased and adjusts accordingly to obtain high-fidelity control solutions.

With the training setting used above, i.e. learning pulses for only 2 drives (u_{01}, d_1) via a standard DDPG algorithm, our RL agent only manages to find one 99.9% fidelity 177.8 ns solution out of a dozen runs. This is because we need to increase the allowed relative change in pulse amplitudes at each step, effectively broadening the action space, in order to compensate for such a short gate duration. Larger action space tends to result in Q-value overestimation, which ultimately destabilizes the training process. By employing a few additional modifications to the training setting such as expanding the agent's access to 3 drives (d_0, u_0, d_1), implementing the TD3 tricks, and training the agent for longer, we notice more stable training and an improved probability of success on some occasions but not universally. Therefore, we postpone the discussion of these additional results to appendix E.1.

5.3. Novelty in the roles of drives

We can further highlight the novelty in control solutions found by our RL agent by analyzing the role played by each drive in implementing a two-qubit operation. We do so by taking the each optimized pulse sequence, removing the on-resonance component d_1, and comparing the left-over CR component u_{01} with the original pulse. In the weak drive regime, we expect the CR drive u_{01} to be the sole entanglement generator and the on-resonance drive d_1 to only affect the local rotation of the target qubit. By examining the changes in the

Figure 9. Effect of removing target qubit drive d_1 from optimized $ZX(\pi/2)$ pulses. (a) Average linear entropy $\bar{S}_{\text{lin}}$ as defined in section 2.4. (b) Fidelity of control qubit averaged over initial states $|00\rangle$ and $|10\rangle$. The target qubit drive practically only affects the target qubit rotation in the *direct* scheme. Meanwhile, the RL agent discovers solutions where this on-resonance drive works in tandem with the cross-resonance drive to generate entanglement and rotate the control qubit.

Figure 10. Robustness of optimized $ZX(\pi/2)$ pulses to short-timescale stochastic noise. We simulate stochastic noise by adding uncertainty to system parameters at inverse sampling rate $\mathrm{d}t = 2/9$ ns during evaluation. RL-design pulses outperform the *direct* scheme up to 0.5% noise. For larger noise, the same-duration RL solution behaves similarly to its *direct* counterpart whereas the shorter-duration RL solution degrades at a faster rate. RL training hyperparameters are given in the 'Fixed Environment' section of table 1.

linear entropy and the qubit control fidelity when the on-resonance drive is removed, we confirm that, in the *direct* scheme, d_1 has little to no effect on the entanglement generated and the motion of the control qubit, as illustrated in by the overlapping orange curves in figure 9. This suggests that optimizing the Gaussian Square pulses in the *direct* scheme leads to control solutions exhibiting a clear separation of roles: the CR drive generates almost all entanglement and ensures that the control qubit ends up in the intended state; meanwhile, the on-resonance drive focuses on correcting the target qubit state. Interestingly, our RL agent discovers additional strategies where these roles get mixed up to different degrees, represented by how much the blue curves in figure 9 deviate from one another. In these cases, the on-resonance drive d_1 actually works in tandem with the CR drive u_{01} to generate entanglement and rotate the control qubit. Such a behavior can be attributed to high driving amplitudes which activate interactions beyond the desired ZX term, leading to the observed novelty in solutions discovered by our RL agent.

5.4. Robustness of optimized pulses

When implemented on a real device, the performance of optimized control solutions inevitably suffers from a variety of error sources such as imperfect controls and noisy readouts. We simulate these effects by introducing Gaussian fluctuation on system parameters $\vec{p}$; this allows us to assess the robustness of the control solutions discussed so far. Specifically, at every step of the size of the inverse sampling rate $\mathrm{d}t = 2/9$ ns, fluctuations are sampled from a zero mean Gaussian distribution with a standard deviation of up to 3% of the original system parameters $\vec{p}_0$ listed in table 2. For each deviation value, we collect 50 samples and report the results for optimized $ZX(\pi/2)$ pulses in figure 10. RL-designed pulses outperform the *direct* implementation up to 1% deviation at the same gate duration (248.9 ns) and up to 0.5% deviation at shorter gate duration (177.8 ns). The shorter gate duration RL solution degrades quickly as the deviation increases,

possibly due to large jumps in its non-smooth PWC waveform. Since the 248.9 ns RL pulse has a much lower amplitude, the jumps in its PWC waveform are not as detrimental. As a result, it shows no discernible difference in performance degradation rate, when compared to its smooth Gaussian Square waveform counterpart in the *direct* scheme. Overall, the mean fidelities drop to around 99%−99.5% at a 3% deviation, which translates to about 66 kHZ for the coupling value and about several MHz for the remaining parameters. These results suggest a trade-off in the RL solutions between fidelity and gate duration in the presence of stochastic noise, which could stem from the non-smooth feature of PWC pulses as well as proximity to the smallest duration necessary to generate sufficient entanglement.

5.5. Adapting to drifting system characteristics

In addition to the Markovian, short-timescale fluctuation, systems characteristics of superconducting transmon qubits are also known to drift over a longer time, owing to, e.g. dilution refrigerator temperature or fabrication defects. Therefore, in the following, we explore the idea of generalizing a single RL agent to a range of system parameters so that no additional training is required when re-calibration is needed [47]. We additionally discover that knowledge of the change in system parameters can be utilized to strengthen the generalization capability of the agent. Finally, from this point on, we will switch to the objective of learning the CNOT gate to further highlight our approach's applicability to different target gates. Traditionally, a CNOT gate can be achieved by performing a $ZX(\pi/2)$ gate followed by a target qubit rotation. Since our RL agent learns to perform these gates from scratch, its control solution for the CNOT gate does not have to follow the aforementioned decomposition and can be learned directly, resulting in pulses reported in figure 18.

5.5.1. Evaluating generalizability

We first discuss our method of evaluating the generalizability of an RL agent on multiple systems whose parameters have drifted from the original values $\vec{p}_0$. We study two main situations, when only a single type of parameter has changed, i.e.

$$\text{only coupling:} \quad J \to J + \Delta J, \tag{32}$$

$$\text{only drive strengths:} \quad \begin{cases} \Omega_{d_0} \to \Omega_{d_0} + \Delta\Omega_{d_0}, \\ \Omega_{d_1} \to \Omega_{d_1} + \Delta\Omega_{d_1}, \\ \Omega_{u_{01}} \to \Omega_{u_{01}} + \Delta\Omega_{u_{01}}, \\ \Omega_{u_{10}} \to \Omega_{u_{10}} + \Delta\Omega_{u_{10}}, \end{cases} \tag{33}$$
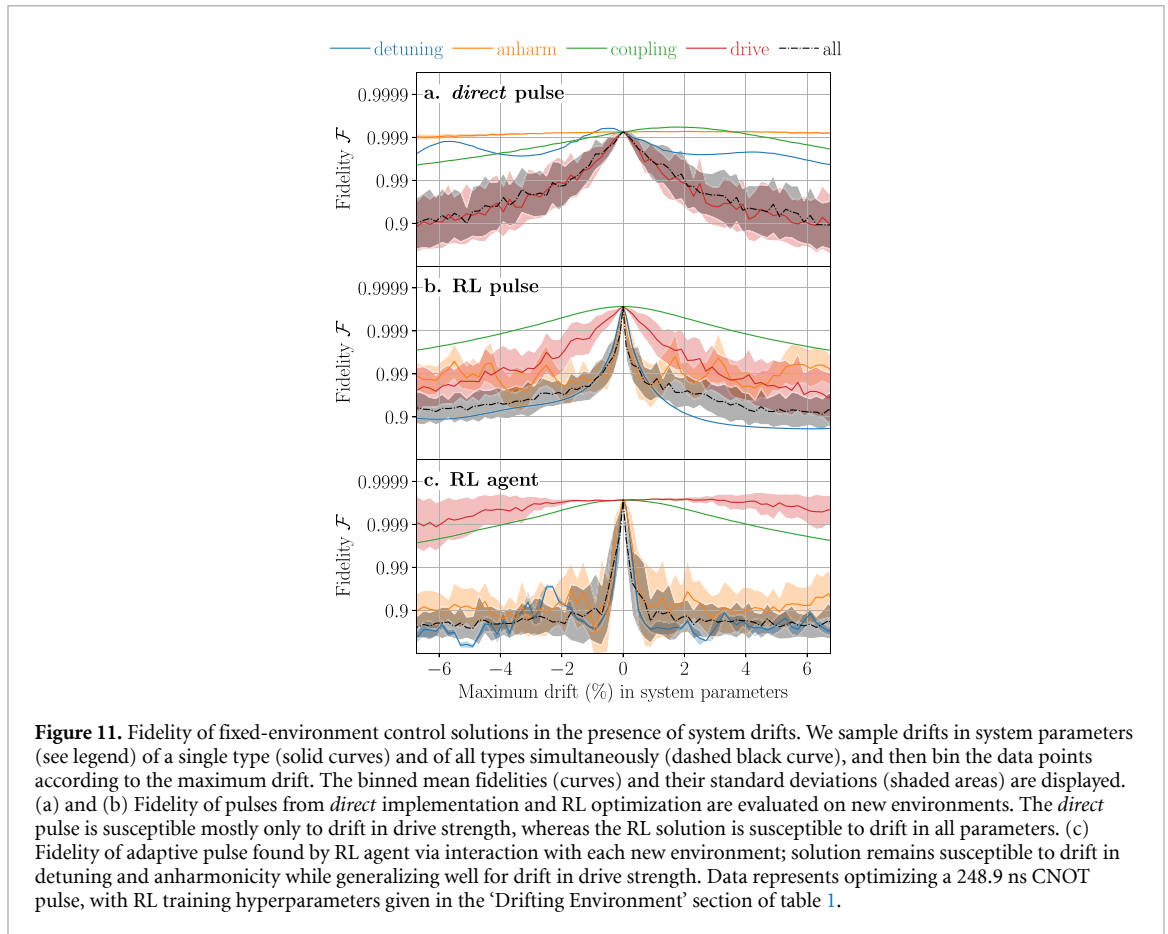
or when all parameters have changed. We gather the drifts in system parameters into a *context* vector

$$\frac{\Delta\vec{p}}{\vec{p}_0} = \left[\frac{\Delta J}{J}, \frac{\Delta\Omega_{u_{01}}}{\Omega_{u_{01}}}, \dots \right]. \tag{34}$$

We then sample these changes randomly, evaluate the fidelity of the tailored control solutions, and bin these data points according to the drift with largest absolute value. Using a bin-size of 0.2%, we collect samples such that the number of data points increases from ∼15 points in the central bins to ∼60 points in the ±7% bins. Finally, we study the binned mean and standard deviation of the fidelity as a function of the maximum drift.

We can now examine the generalizability of the control solutions discussed so far, namely the *direct* and RL pulses optimized for the original system parameters, as shown in figure 11. We focus on a 6% range of maximum drift on each side of the original values. In our simulation, since the coupling value is much smaller, the drift of the same percentages for this parameter has significantly less effect than the others (green curves). While generalizing well for most parameters due to its smoothness, the *direct* pulse in figure 11(a) is highly susceptible to drift in drive strength as the rotation angle is directly affected. The RL-designed pulse in figure 11(b), on the other hand, degrades quickly for drifts in all parameters as expected for a non-smooth PWC waveform.

Adaptive control solutions from the RL agent for different system parameters behave more interestingly, which can be seen in figure 11(c). We find a large susceptibility to changes in detuning and anharmonicity, likely because they directly change the spacing between transmon energy levels and fundamentally alter the system's internal physics, e.g. the resonant frequencies. Meanwhile, the drive strengths are connected to external controls which the RL agent has direct influence over, resulting in a much more robust behavior. Overall, previous control solutions exhibit poor generalization performance when drift in *all* system parameters is considered.
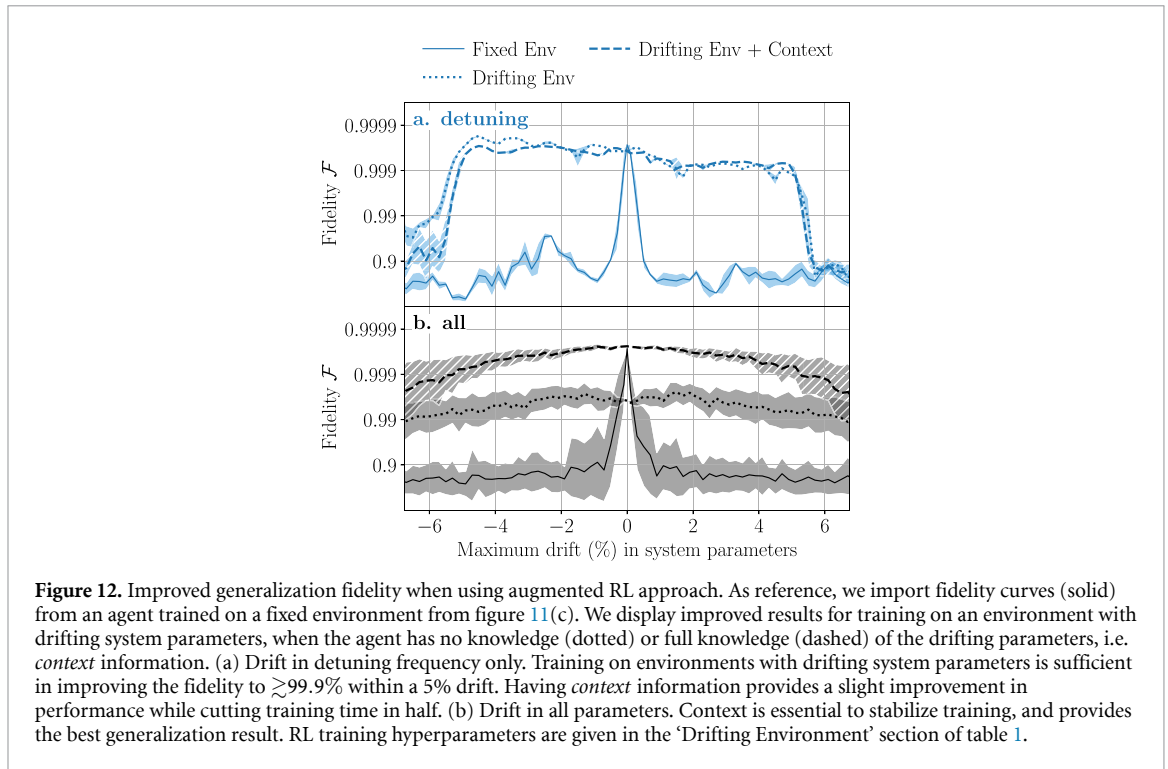
Figure 11. Fidelity of fixed-environment control solutions in the presence of system drifts. We sample drifts in system parameters (see legend) of a single type (solid curves) and of all types simultaneously (dashed black curve), and then bin the data points according to the maximum drift. The binned mean fidelities (curves) and their standard deviations (shaded areas) are displayed. (a) and (b) Fidelity of pulses from *direct* implementation and RL optimization are evaluated on new environments. The *direct* pulse is susceptible mostly only to drift in drive strength, whereas the RL solution is susceptible to drift in all parameters. (c) Fidelity of adaptive pulse found by RL agent via interaction with each new environment; solution remains susceptible to drift in detuning and anharmonicity while generalizing well for drift in drive strength. Data represents optimizing a 248.9 ns CNOT pulse, with RL training hyperparameters given in the 'Drifting Environment' section of table 1.

5.5.2. Learning to generalize

We now identify several ingredients necessary for improving the generalization performance of our agent. We start with a simpler problem where we task the agent to adapt to drift in detuning while all parameters remain fixed. By simply allowing the agent to interact with many systems with different detuning values during training, we observe an immediate and significant performance increase (cf dotted blue curve in figure 12(a)). We achieve this by sampling the detuning from a uniform distribution of a 5% range around its original value at the beginning of each episode. As a result, our agent can observe the changing effects of its action on drifted systems, and thus, learn to adapt accordingly. To further help the agent discern different systems more effectively, we can provide it with specific knowledge about the system with which it is currently interacting, namely the size of the detuning drift relative to its original value. We refer to this piece of information as *context* and this additional input to the agent remains constant within each episode, implying that the system characteristics remain constant throughout the entire gate duration. While having little effect in generalization performance here (dashed blue curve), the training time needed for a *context*-aware agent is actually cut in half.

    We apply our findings to the full problem with drift in all parameters and plot the result in figure 12(b). Training on a drifting environment remains necessary but is no longer sufficient to achieve good generalization results (dotted black curve). In fact, the best result we report when training without context is obtained in only a few training iterations; after that, the performance drops precipitously (cf black curve in figure 14(b)). We believe that feedback with the quantum state is no longer enough for our agent to distinguish different environments. For example, different environments can get to the same state with a different set of actions. Therefore, without additional information, the agent encounters a good amount of confusion during training.

    Providing our agent with *context* information about the drifts in all parameters greatly alleviates the problem (dashed black curve). Here, instead of sampling from a uniform distribution as in the previous case, we sample drifts for all parameters from a zero-mean Gaussian distribution of a 2% standard deviation, in order to highlight the effect on the generalization results. Indeed, instead of collapsing at 5% drift (blue dashed curve), the fidelity in this case gradually decreases (black dashed curve) since the agent actually gets

Figure 12. Improved generalization fidelity when using augmented RL approach. As reference, we import fidelity curves (solid) from an agent trained on a fixed environment from figure 11(c). We display improved results for training on an environment with drifting system parameters, when the agent has no knowledge (dotted) or full knowledge (dashed) of the drifting parameters, i.e. *context* information. (a) Drift in detuning frequency only. Training on environments with drifting system parameters is sufficient in improving the fidelity to ≳99.9% within a 5% drift. Having *context* information provides a slight improvement in performance while cutting training time in half. (b) Drift in all parameters. Context is essential to stabilize training, and provides the best generalization result. RL training hyperparameters are given in the 'Drifting Environment' section of table 1.

to interact with system drifts beyond 5% during training under the Gaussian distribution. Overall, extending the training environment to include system drifts and providing the agent with *context* information about those drifts significantly stabilizes the generalization task when all system parameters are involved. More importantly, the resultant agent can immediately propose pulses with 99.9% fidelity at up to 4% drift without any further training. These simulations justify the suggested practicality of RL in the presence of a reasonable system drift on near-term devices.
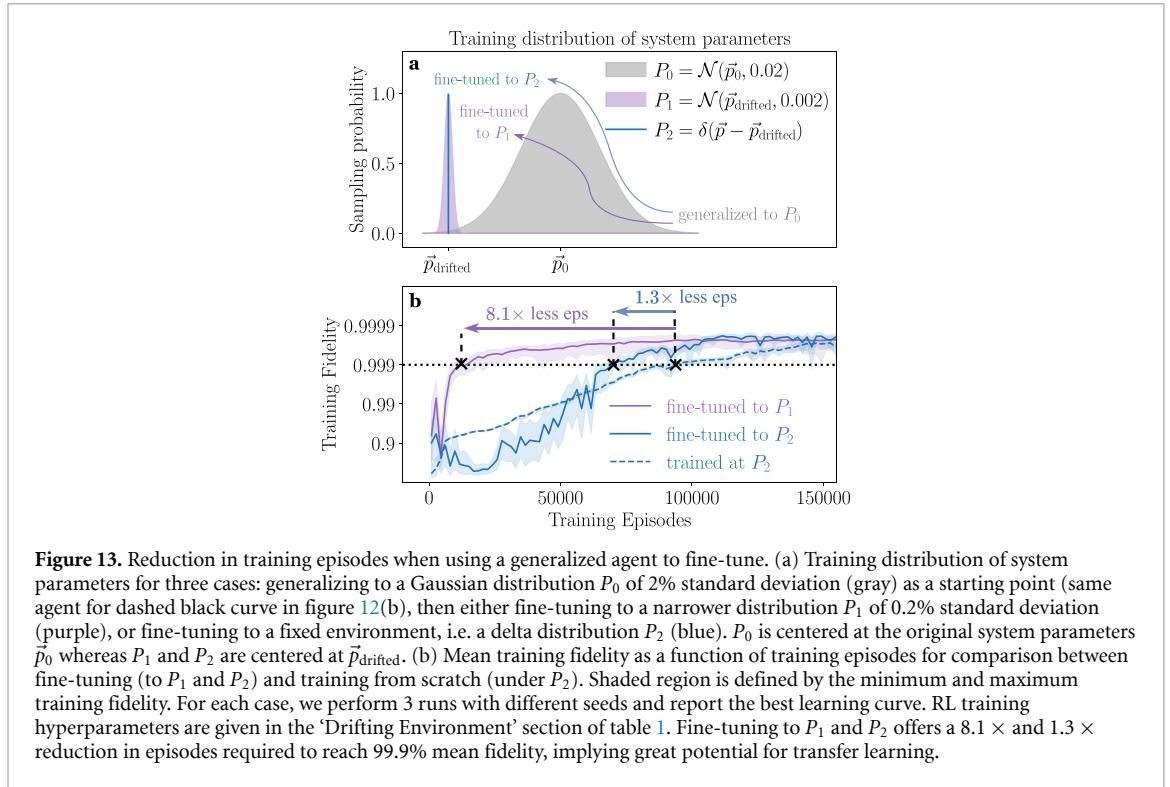
When dealing with a more substantial drift, we find that fine-tuning is necessary to achieve 99.9% fidelity, although the number of training episodes required can be notably less than when starting from scratch. To investigate this phenomenon, we first reiterate that the RL agent used in figure 12(b) has been trained to generalize to system parameters sampled from a Gaussian distribution of 2% standard deviation around the original values $\vec{p}_0$, as in figure 13(a) (grey distribution). We then select a set of drifted system parameters, denoted as $\vec{p}_{\mathrm{drifted}}$ [48] with maximum drift of -5.7%, for which the generalized agent suggests a control solution achieving 95% fidelity. Subsequently, we specialize our agent by training it on a fixed environment with these specific system parameters (blue delta distribution).

Comparing this approach to training a separate agent entirely from scratch within the same environment, we observe a reduction of $1.3 \times$ in the number of training episodes required to reach 99.9% fidelity, as illustrated in figure 13(b). Intriguingly, when we instead specialize our agent to a drifting environment characterized by a Gaussian distribution of 0.2% standard deviation and mean at $\vec{p}_{\mathrm{drifted}}$ (cf purple distribution in figure 13(a)), the episode reduction jumps to $8.1 \times$. These preliminary findings hint at the potential for substantial transfer learning in certain cases. However, further investigations are needed to understand the underlying causes for such a wide range of effectiveness, which we plan to pursue in future studies.

6. Conclusion and outlook

In this study, we have showcased the advantages of harnessing RL for the design of CR gates, fully independent of known theoretical protocols and pre-existing error suppression techniques. Our unbiased approach employs an off-policy agent to customize continuous control parameters, shaping complex-valued pulses concurrently for both the CR and the target on-resonance drives. Compared to established optimal control methods, RL has the advantage of (i) being suited for closed-loop optimization due to its model-free nature; (ii) it is capable of non-local exploration; and (iii) it generates a representation of gained knowledge as a valuable by-product.

Figure 13. Reduction in training episodes when using a generalized agent to fine-tune. (a) Training distribution of system parameters for three cases: generalizing to a Gaussian distribution P_0 of 2% standard deviation (gray) as a starting point (same agent for dashed black curve in figure 12(b)), then either fine-tuning to a narrower distribution P_1 of 0.2% standard deviation (purple), or fine-tuning to a fixed environment, i.e. a delta distribution P_2 (blue). P_0 is centered at the original system parameters $\vec{p}_0$ whereas P_1 and P_2 are centered at $\vec{p}_{\text{drifted}}$. (b) Mean training fidelity as a function of training episodes for comparison between fine-tuning (to P_1 and P_2) and training from scratch (under P_2). Shaded region is defined by the minimum and maximum training fidelity. For each case, we perform 3 runs with different seeds and report the best learning curve. RL training hyperparameters are given in the 'Drifting Environment' section of table 1. Fine-tuning to P_1 and P_2 offers a $8.1 \times$ and $1.3 \times$ reduction in episodes required to reach 99.9% mean fidelity, implying great potential for transfer learning.

Using the RL methodology, we demonstrated the discovery of novel control solutions that fundamentally differ from conventional error suppression techniques for two-qubit gates, such as *direct* and *echoed* schemes, while surpassing them in both fidelity and execution time. At the typical gate duration of 248.9 ns (for transmon devices) where the *direct* and *echoed* schemes achieve 99.937% and 99.501% fidelity respectively, RL-designed solutions can cut the error in about half compared to the better scheme, achieving $\mathcal{F}_{\text{RL}} \gtrsim 99.966\%$, for both $ZX(\pi/2)$ and CNOT gates. Not only that, our agent identified a potential maximum reduction of 30% in gate duration, while maintaining the same level of fidelity exceeding 99.9%. This can be attributed to the flexibility of the PWC ansatz capable of managing leakage out of the computational subspace, as well as unwanted coherent processes that inevitably arise at large drive amplitudes.

Furthermore, we illustrated the possibility of augmenting our approach to enable our agent to flexibly adapt its design capability to accommodate drifts in the underlying hardware. We found that exposing the agent to an environment with drifting system parameters during training while providing it with *context* information about these drifts, allows our agent to learn the appropriate control solutions and generalize well across a range of drifted system parameters. Concretely, our *context*-aware agent can readily propose control solutions with $\sim 99.9\%$ fidelity when all system parameters, including detuning, anharmonicity, coupling strength, and drive strength, are allowed to drift within a 4% range around their original values. In instances of more substantial drifts, our generalized agent serves as a valuable starting point for fine-tuning, resulting in a remarkable $1.3 - 8.1 \times$ acceleration in optimization iterations when compared to starting from scratch.

Based on these findings, we can assert that the RL approach alleviates the necessity for a precise model, presenting a versatile framework applicable to designing various CR-based gates. When combined with PWC protocol space, our RL agent demonstrates its capacity to devise innovative pulse shapes that surpass the capabilities of conventional ansätze in terms of both fidelity and gate execution duration. The quest for shorter, high-fidelity pulses is particularly significant, given that various calibration methods are nearing the coherence limit imposed by state-of-the-art gate duration and qubit relaxation times. Furthermore, our *context*-aware RL approach effectively addresses hardware drifts, indicating the possibility of reducing and even eliminating additional training, and thus expensive calibration experiments, as long as system characteristics remain within a reasonable range.

When applied to experiments conducted on real-world hardware, our off-policy method carries the potential for significant data efficiency gains, as the agent can be trained on data collected by any policy. Consequently, while the initial training phase may incur high costs, subsequent retraining can be expedited thanks to the collected dataset. Additionally, actual drives delivered to the qubits are generally smoothed out

from the raw jagged PWC input pulses, which should enhance the robustness of the optimized solutions to control fluctuations.

As the quantum computing community progresses toward larger platforms, the capacity of a single agent to extend its design capabilities across diverse system characteristics becomes increasingly pivotal for scalability [47]. In fact, as the number of qubits grows, it becomes inevitable that certain qubits will exhibit overlapping system characteristics [49]. In such a scenario, our *context*-aware agent, trained to generalize within a specific region of system parameters, can readily be applied to a group of qubits sharing similar characteristics. Moreover, these experiments can be conducted simultaneously, as qubits with akin parameters are likely to be positioned at a considerable distance from each other in the first place, further enhancing the efficiency of our RL agent.

In the immediate future, we are eager to integrate our approach into established gate optimization procedures for superconducting devices, as well as extending its utility to various quantum computing platforms. At the same time, we also aim to broaden the applicability of our RL agent to handle more intricate operations, such as the SWAP gate, multi-qubit gates, or gates on qudits. With the recent advancement enabling better control of the $|1\rangle \leftrightarrow |2\rangle$ transition [50, 51] and the potential of improving quantum speed limit by expanding beyond the qubit subspace [52], the synthesis of even faster qubit gates as well as qutrit gates emerges as an intriguing and imminent application for our RL protocol. On the algorithmic front, we emphasize the significance of enhancing generic RL algorithms through generalization and transfer learning techniques to bolster the method's practicability, especially for large-scale platforms. With the field of RL, and machine learning in general, growing at an unprecedented rate, we hope to continue leveraging these powerful advancements toward the development of practical quantum computers.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

Appendix A. Simulated environment details

A.1. Rotating frame transformation

We consider the single-transmon Hamiltonian in the lab frame $H_1^{\text{lab}} = H_{1,\text{sys}}^{\text{lab}} + H_{1,\text{ctrl}}^{\text{lab}}(t)$, given in section 2.1. Under the rotating frame transformation $R = R(t) = e^{-i\omega_d t b^\dagger b}$, the ladder operators are mapped as follows

$$b^\dagger \to R^\dagger(t) b^\dagger R(t) = b^\dagger e^{i\omega_d t}$$
$$b \to R^\dagger(t) b R(t) = b e^{-i\omega_d t}, \tag{A1}$$

leaving the number-conserving terms, i.e. with equal numbers of $b^\dagger$ and b, unchanged. The derivative $-R^\dagger i \partial_t R = -\omega_d b^\dagger b$ contributes to the quadratic term, leading to the detuning $\delta = \omega - \omega_d$ when combined with the transformed system Hamiltonian:

$$R^\dagger \left(H_{1,\text{sys}}^{\text{lab}} - i\partial_t \right) R = \delta b^\dagger b + \frac{\alpha}{2} b^\dagger b^\dagger bb. \tag{A2}$$

The control part then transforms as follows

$$
\begin{aligned}
R^{\dagger} H_{1,\text{ctrl}}^{\text{lab}}(t) R &= \Omega_d \text{Re}\left(d(t)\, e^{i\omega_d t}\right)\left(b^{\dagger} e^{i\omega_d t} + b\, e^{-i\omega_d t}\right) \\
&= \frac{\Omega_d}{2}\left[d(t)\, b^{\dagger} e^{2i\omega_d t} + d(t)\, b + \text{h.c.}\right] \\
&\approx \frac{\Omega_d}{2}\left[d(t)\, b + \text{h.c.}\right]
\end{aligned}
\tag{A3}
$$

where in the last line, we have dropped terms rotating at twice the driving frequency ω_d, which is commonly known as the rotating wave approximation (RWA). Altogether, we arrive at the rotating frame Hamiltonian for single transmon $H_1(t)$ as in equation (5). For two-transmon, the above calculation carries directly over.

A.2. Virtual Z gates and angles

We first consider applying a virtual Z gate only on the first qubit by augmenting the overlap $M = U_{\text{qubit}} U_{\text{target}}^{\dagger}$ as follows

$$
M \to M' = V_Z(\theta_0) M = \left[\text{diag}\left(1, e^{i\theta_0}\right) \otimes I\right] M.
\tag{A4}
$$

While the first term in equation (2) remains unchanged: $\text{Tr}(M'M'^{\dagger}) = \text{Tr}(V_Z(\theta_0) M M^{\dagger} V_Z(-\theta_0)) = \text{Tr}(MM^{\dagger})$, the second term becomes

$$
\begin{aligned}
\left|\text{Tr}(M')\right|^2 &= \left|M_{00} + M_{11} + (M_{22} + M_{33})\, e^{i\theta_0}\right|^2 \\
&= |M_{00} + M_{11}|^2 + |M_{22} + M_{33}|^2 \\
&\quad + (M_{00} + M_{11})^* (M_{22} + M_{33})\, e^{i\theta_0} \\
&\quad + (M_{00} + M_{11})(M_{22} + M_{33})^*\, e^{-i\theta_0} \\
&= \text{const} + \mathcal{M} e^{i\theta_0} + \mathcal{M}^* e^{-i\theta_0},
\end{aligned}
\tag{A5}
$$

where M_{ij} are the matrix elements of M, and $\mathcal{M} = (M_{00} + M_{11})^* (M_{22} + M_{33})$. We find the optimal value of θ_0 that maximizes the average gate fidelity by solving

$$
\begin{aligned}
\frac{\partial \mathcal{F}_{\text{avg}}}{\partial \theta_0} = 0 &\Rightarrow e^{2i\theta_0} = \frac{\mathcal{M}^*}{\mathcal{M}} \\
&\Rightarrow \tan\theta_0 = -\frac{\text{Im}\,\mathcal{M}}{\text{Re}\,\mathcal{M}},
\end{aligned}
\tag{A6}
$$

where there exist two solutions within the $[-\pi, \pi]$ range. To ensure that we obtain the angle that maximizes the fidelity instead of minimizing it, we check that the solution angle of the above relation satisfies

$$
\begin{aligned}
&\frac{\partial^2 \mathcal{F}_{\text{avg}}}{\partial^2 \theta_0} < 0 \\
&\Rightarrow \sin\theta_0 \text{Im}\,\mathcal{M} < \cos\theta_0 \text{Re}\,\mathcal{M}.
\end{aligned}
\tag{A7}
$$

If this is not the case, we simply pick the other solution $\theta_0 - \text{sign}(\theta_0)\pi$. In the case of a virtual Z gate only on the second qubit $V_Z(\theta_1) = \left[I \otimes \text{diag}(1, e^{i\theta_1})\right]$, we follow the same derivation with $\mathcal{M} = (M_{00} + M_{22})^* (M_{11} + M_{33})$.

Obtaining the analytical optimal values for both angles simultaneously by solving

$$
\left(\theta_0^{\text{opt}}, \theta_1^{\text{opt}}\right) = \max_{\theta_0, \theta_1} \mathcal{F}_{\text{avg}}\left(V_Z(\theta_1) V_Z(\theta_0) M\right),
\tag{A8}
$$

is much more involved since the dependence of $\mathcal{F}_{\text{avg}}$ on (θ_0, θ_1) is significantly more complicated. However, we find that optimizing one angle at a time using equation (A6) by solving

$$
\begin{aligned}
\theta_0^{\text{near-opt}} &= \max_{\theta_0} \mathcal{F}_{\text{avg}}\left(V_Z(\theta_1) M\right) \\
\text{then } \theta_1^{\text{near-opt}} &= \max_{\theta_1} \mathcal{F}_{\text{avg}}\left(V_Z(\theta_1) V_Z(\theta_0^*) M\right)
\end{aligned}
\tag{A9}
$$

yields a near-optimal solution $\boldsymbol{\theta}^* = (\theta_0^*, \theta_1^*)$ that is sufficiently accurate for the fidelity to reach the threshold we set: indeed, we find that the error in computing the maximum average fidelity using this protocol, as compared to numerically optimizing for both angles simultaneously, remains below 10^{-5}, which is negligible for the fidelity levels discussed in our work.

In our simulation where we work in the rotating frame of the second transmon, more Z error accumulates on the first transmon. We observe that optimizing for the larger angle yields a more accurate result, whence the order in equation (A9).

Table 1. Training hyperparameters for training an RL agent to design quantum gates on simulated transmon environment. Unmentioned hyperparameters necessary for the DDPG algorithm are set to their default values in RLlib's implementation [54], version 2.0.0. *When studying the adaptability of our RL agent to drifting system characteristics, we discover a particular region around +2% drift on all system parameters where the 20-segment ansatz yields no solution with fidelity better than 99.5%. The problem goes away when we increase the number of segments to 28, whose result was reported in the main text.

Environment	Fixed environment			Drifting environment		3 drives	
Target gate	$IX(\pi/2)$	$ZX(\pi/2)$,CNOT		CNOT		$ZX(\pi/2)$,CNOT	
Number of segments	9	20	20	20	28*	20	
Drifting system parameters		—		detuning	all	—	
Sampling distribution		—		uniform	normal	—	
Sampling variance		—		5%	2%	—	
Duration	10 ns	320.0 ns 284.4 ns 248.9 ns	213.3 ns 177.7 ns 142.2 ns	248.9 ns		248.9 ns	177.8 ns
Drives	d_1	u_{01}, d_1		u_{01}, d_1		d_0, u_{01}, d_1	
Action window w_u	0.4	0.1	0.2	0.1		0.1	0.15
Action window w_d	0.13	0.01	0.02	0.01		0.01	0.015
DDPG							
Hidden layers	100, 200, 100	800, 400, 200		800, 800, 800		800, 400, 200	
Activation				relu			
Learning rate				1×10^{-4}			
Train batch size				64			
Soft update parameter κ				0.002			
Buffer capacity				100 000			
Initial buffer				10 000			

Table 2. IBMQ Valencia system parameters extracted from Qiskit interface, and used in this work. We denote the corresponding system parameter vector to be $\vec{p}_0 = [J, \Omega_{u_{01}}, \dots]$. Typical bare qubit frequencies have been translated into detuning values used in our simulation.

	Symbols	Values (MHz)
Drive strengths	$\Omega_{u_{01}}, \Omega_{d_0}$	204.7
	$\Omega_{u_{10}}, \Omega_{d_1}$	158.5
Detuning	δ_0	−86.6
	δ_1	0
Anharmonicity	α_0	−310.5
	α_1	−313.9
Coupling	J	2.2

A.3. Evolving method

We provide details into our simulator of transmon quantum dynamics, which is unitary in the absence of decoherence processes. Under piece-wise-constant controls, the unitary map naturally simplifies into a product of time-local propagators

$$U = \prod_{n=1}^{N} U_n = U[N\Delta t, (N-1)\Delta t]\cdots U[\Delta t, 0], \tag{A10}$$

where Δt is the discretized time step, and N is the number of segments in the PWC pulse. Each propagator, corresponding to each segment, can be computed by solving the time-dependent Schrödinger equation (TDSE)

$$\frac{\partial U(t)}{\partial t} = -\mathrm{i}H(t)U(t) \tag{A11}$$

from t to $t + \Delta t$. For example, in the two-transmon setting, we set $H(t) = H_2(t)$ given in equation (10). In general, the unitary is given by the following time-ordered integral

$$U_{\text{TDSE}}\left[t+\Delta t,t\right]=\mathcal{T}\exp\left[-\mathrm{i}\int_{t}^{t+\Delta t}H(t)\,\mathrm{d}t\right],\tag{A12}$$

which we numerically obtain using QuTiP's TDSE solver [55]. This is necessary because of the detuning phase factors $e^{i\delta t}$, even though the control fields u and d are constant within each segment. For the majority of this work, however, we focus on shaping only two control fields, u_{01} and d_1, and the phase factors drop out in the frame rotating at the target qubit frequency. As the result, the piece-wise Hamiltonian is now constant and can be directly exponentiated to obtain the unitary that solves the time-independent Schrödinger equation (TISE)

$$U_{\text{TISE}}\left[t+\Delta t,t\right]=\exp\left[-\mathrm{i}H(t)\,\Delta t\right],\tag{A13}$$

providing a significant computational speedup. We verified that TISE solution converges to TDSE solution as the step size approaches zero.

Appendix B. DDPG algorithm

In this work, we employ Deep Deterministic Policy Gradient (DDPG), an off-policy Q-learning algorithm suitable for a continuous action space. We summarize the training procedure in Algorithm 1.

Algorithm 1. Deep deterministic policy gradient.

Require: Initial Q-network and policy parameters ϕ and θ
Require: Initial target networks parameters ϕ' and θ'
1. **for** step $= 1, 2, \ldots, M$ **do**
2.   Reset environment to state s_0 **if** end of episode
3.   Sample an exploration noise $\mathcal{N}$
4.   Select action $a_i = \mu_\theta(s_i) + \mathcal{N}$
5.   Evolve the environment and observe (r_i, s_{i+1})
6.   Add tuple $(s_i, a_i, s_{i+1}, r_{i+1})$ to replay buffer $\mathcal{B}$
7.   **if** $\mathcal{B}$ has more than N_{initial} samples **then**
8.   Sample a batch B of $(s_i, a_i, s_{i+1}, r_{i+1})$ from $\mathcal{B}$
9.   Compute targets $y_i = r_{i+1} + \gamma Q_{\phi'}(s_{i+1}, \mu_{\theta'}(s_i))$
10.   Update Q-network by minimizing the loss:

$$L_\phi = \frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\left[y_i - Q_\phi\left(s_i, a_i\right)\right]^2$$

11.   Update policy by maximizing the Q-values:

$$L_\theta = -\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}Q_\phi\left(s_i, \mu_\theta\left(s_i\right)\right)$$

12.   Update target networks:
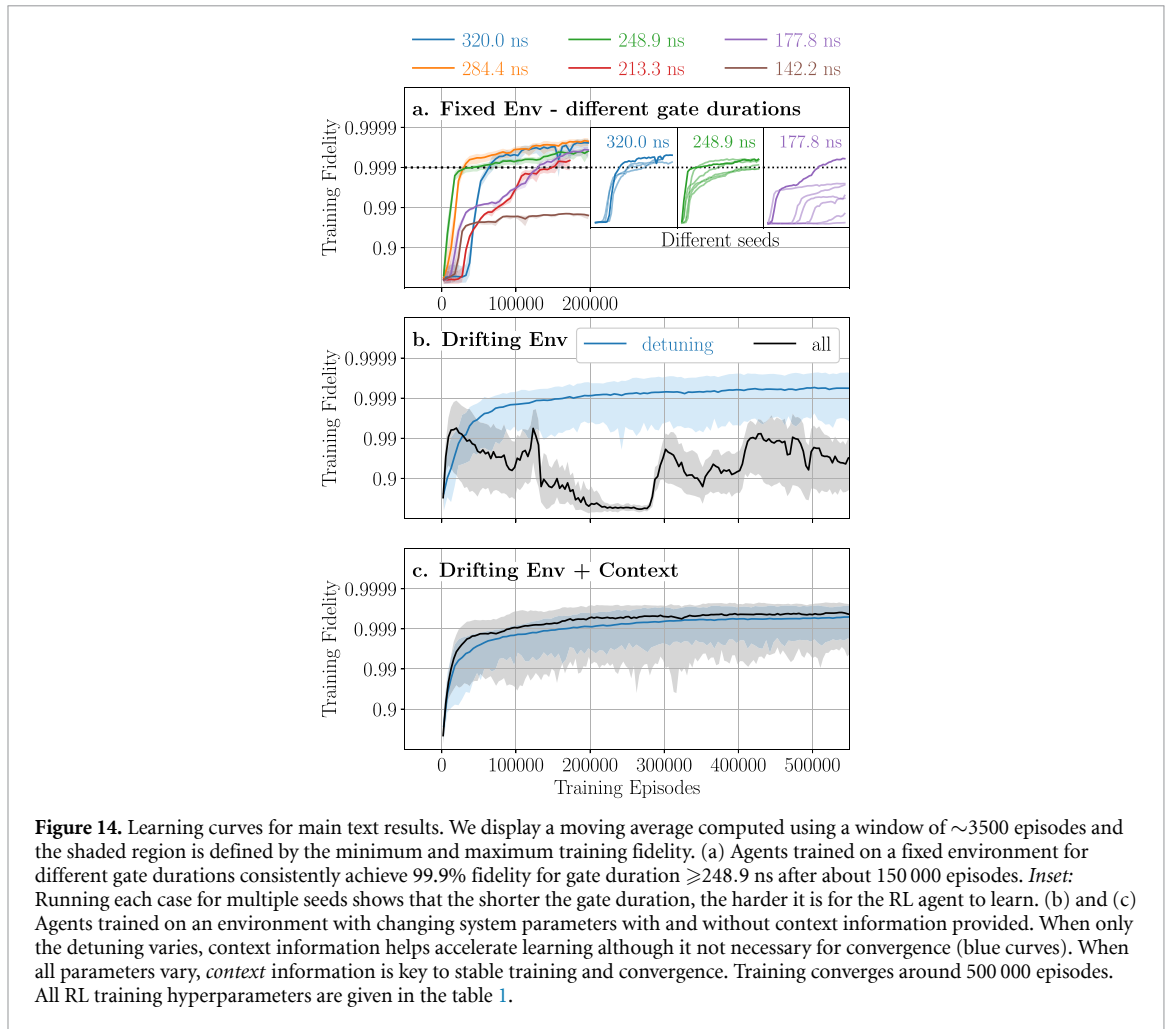
$$\begin{aligned}\theta' &\leftarrow \kappa\theta + (1-\kappa)\theta'\\\phi' &\leftarrow \kappa\phi + (1-\kappa)\phi'\end{aligned}\tag{A14}$$

13.   **end if**
14. **end for**

The two neural networks for estimating the optimal Q-value and the agent's deterministic policy are randomly initialized at the beginning of training. For continuous action space, exploration is implemented by adding some noise $\mathcal{N}$ directly to the policy network's output: $a_i = \mu_\theta(s_i) + \mathcal{N}$. The exploration noise is scaled down over time using the Ornstein–Uhlenbeck process as implemented in the DDPG paper [43].

Transitions $(s_i, a_i, s_{i+1}, r_{i+1})$ are collected by following the agent's noisy policy and stored in a large replay buffer $\mathcal{B}$. For the first 10 000 steps, the buffer is being filled without learning. After that, a batch of transitions are used along with two independent Adam optimizers to update both networks. To maintain quasi-stable targets throughout training, we soft-update both target networks (ϕ', θ') via Polyak averaging, see equation (A14).

Figure 14. Learning curves for main text results. We display a moving average computed using a window of ~3500 episodes and the shaded region is defined by the minimum and maximum training fidelity. (a) Agents trained on a fixed environment for different gate durations consistently achieve 99.9% fidelity for gate duration ⩾248.9 ns after about 150 000 episodes. *Inset:* Running each case for multiple seeds shows that the shorter the gate duration, the harder it is for the RL agent to learn. (b) and (c) Agents trained on an environment with changing system parameters with and without context information provided. When only the detuning varies, context information helps accelerate learning although it not necessary for convergence (blue curves). When all parameters vary, *context* information is key to stable training and convergence. Training converges around 500 000 episodes. All RL training hyperparameters are given in the table 1.
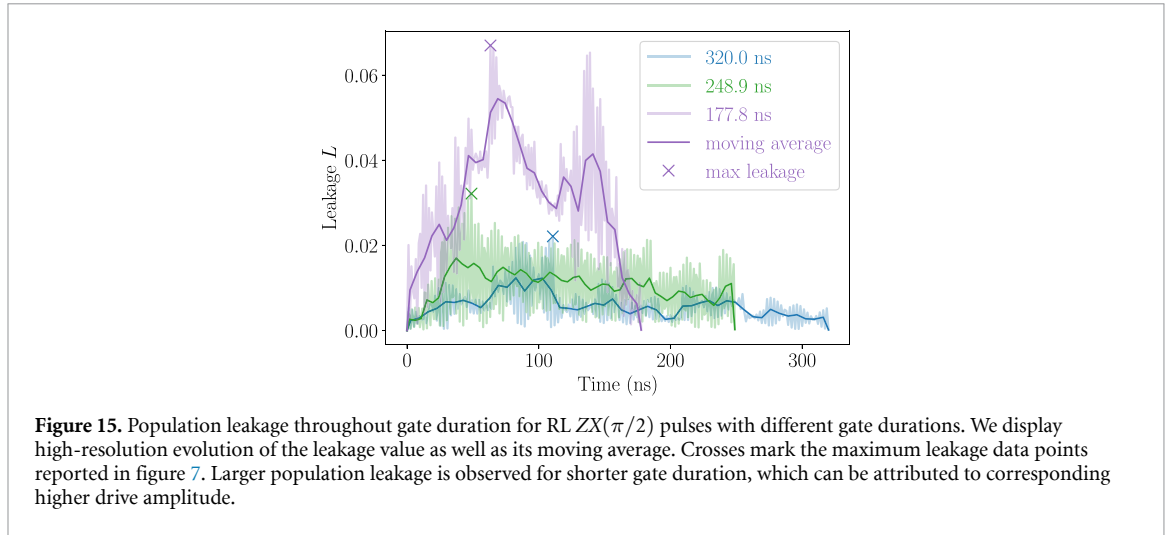
We employ the DDPG (and TD3) implementation from RLlib, an open-source industry-grade library for RL [54]. We conduct a routine exploration of hyperparameters to identify an effective setting, which we maintaine consistently throughout the study. Due to the complex interplay between hyperparameters in high-dimensional analysis, altering one may impact others, making it challenging to provide a comprehensive account. Our focus is on identifying an effective set of hyperparameters and after that, minimizing additional adjustments to maintain the stability in our approach. The detailed hyperparameters used in this work are summarized in table 1. Any other hyperparameters not mentioned are unchanged from the default setting of RLlib version 2.0.0.

Appendix C. Training procedure

We report learning curves for the RL results discussed in the main text, plotting the mean fidelity of pulses encountered as a function of training episodes. Figure 14(a) shows the learning curves for RL training on a fixed environment for different gate durations. Our DDPG agent consistently finds ⩾99.9% fidelity control solutions after about 150 000 episodes for gate durations ⩾248.9 ns, which corresponds to about 18 hours of training. Below this number, training becomes more challenging as we increase the action windows w_u and w_d to compensate for shorter physical time. An increase in action space often leads to exploding Q-values, resulting in a lower success rate over multiple runs. It should be pointed out that implementing the TD3 tricks [45] stabilizes training but degrades the achievable fidelity slightly when compared to DDPG in general.

Figures 14(b) and (c) show learning curves for RL training on an environment with changing system parameters. Unlike both stable blue curves, the black curve is only stable when *context* is included, suggesting the importance of *context* information for adapting RL to a more realistic situation where *all* system parameters can drift away from their original values. The generalized agent typically converges around 500 000 episodes, which corresponds to about 4 days of training.

Figure 15. Population leakage throughout gate duration for RL $ZX(\pi/2)$ pulses with different gate durations. We display high-resolution evolution of the leakage value as well as its moving average. Crosses mark the maximum leakage data points reported in figure 7. Larger population leakage is observed for shorter gate duration, which can be attributed to corresponding higher drive amplitude.

Each training instance initiates 4 workers for sampling interaction with the simulated environment and 1 worker for agent training, utilizing 5 cores simultaneously. Without any significant difference in runtime, training is done either on a typical laptop (M1 3.2GHz or Intel i7 2.7GHz) or on a node within the JUWELS cluster (Intel Xeon Platinum 8168 2.7 GHz).

Appendix D. Dynamics of optimized pulses

D.1. Leakage for RL pulses
We report the full evolution of population leakage throughout gate duration for RL pulses with different gate durations in figure 15. As expected, shorter RL pulses exhibit higher leakage out of the computation qubit subspace in the intermediate time steps. Nevertheless, our RL agent manages to suppress almost all leakage at gate conclusion. We additionally observe fast oscillation in the leakage value, which is possibly due to off-resonant precession of the control qubit. Together, these observations provide a more detailed picture of the leakage process in the novel control solutions discovered by RL.

D.2. Rotation angles
Any two-qubit unitary map $U^{\mathrm{qubit}}(t,0)$ can be expressed in terms of a generating averaged Hamiltonian as follows

$$\exp\left(-\mathrm{i}H_{\mathrm{avg}}^{\mathrm{qubit}}t\right) = \exp\left(-\mathrm{i}\sum_{ij}\theta_{ij}\frac{P_i\otimes P_j}{2}\right), \tag{D1}$$

where we have expanded $H_{\mathrm{avg}}^{\mathrm{qubit}}$ in the Pauli basis given by $P_i \in \{I, X, Y, Z\}$, and the rotation angle in the *ij* direction depends on the duration t and the $P_i \otimes P_j$ interaction strength. For example, the CR gate can be written as $ZX(\pi/2) = \exp(-\mathrm{i}\pi ZX/4)$.

For three-level transmons, we first compute the averaged Hamiltonian by taking the logarithm of the unitary $U(t,0)$, and then project it onto the qubit subspace. This allows us to quantify the strength of different interactions in the unitary at time t via the rotation angle

$$\theta_{ij}(t) = \mathrm{Tr}\left[\mathrm{i}\left(\Pi^{\mathrm{qubit}}\ln U(t,0)\Pi^{\mathrm{qubit}}\right)\frac{P_i\otimes P_j}{2}\right], \tag{D2}$$

where Π^{qubit} is the projector onto the qubit subspace. Computing the log of a matrix is non-trivial due to existence of branch cuts, which lead to different θ_{ij} from the same $U(t,0)$. To see this, we let V be the unitary that diagonalizes H_{avg} and write

$$\begin{aligned}
U(t,0) &= \exp\left(-\mathrm{i}H_{\mathrm{avg}}t\right) \\
&= V\mathrm{diag}\left(\mathrm{e}^{-\mathrm{i}E_1 t}, \mathrm{e}^{-\mathrm{i}E_2 t}, \dots\right)V^\dagger \\
&= V\mathrm{diag}\left(\mathrm{e}^{-\mathrm{i}E_1 t - 2in_1\pi}, \mathrm{e}^{-\mathrm{i}E_2 t - 2in_2\pi}, \dots\right)V^\dagger \\
&= \mathrm{e}^{-iV\mathrm{diag}(E_1 t + 2n_1\pi, E_2 t + 2n_2\pi, \dots)V^\dagger} \\
\Rightarrow i\ln U(t,0) &= V\mathrm{diag}\left(E_1 t + 2n_1\pi, E_2 t + 2n_2\pi, \dots\right)V^\dagger, \tag{D3}
\end{aligned}$$
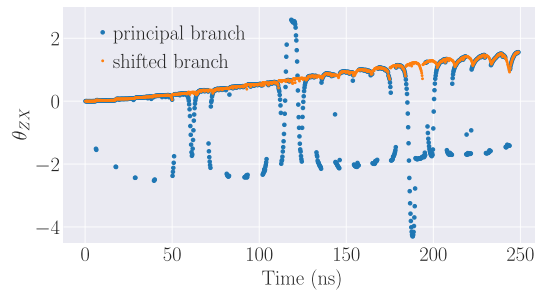
Figure 16. Evolution of *ZX* rotation angle under different branch cuts. θ_{ZX} is first computed in the *principal* branch cut (blue), i.e. without any phase shifts. Then, a phase shift is determined and added at every time step, resulting in the shifted branch cut (orange), which ensures a well-behaved evolution of rotation angles. Without shifting the branch cut, the observed large jumps obscure meaningful interpretation of the accumulated rotation angles.
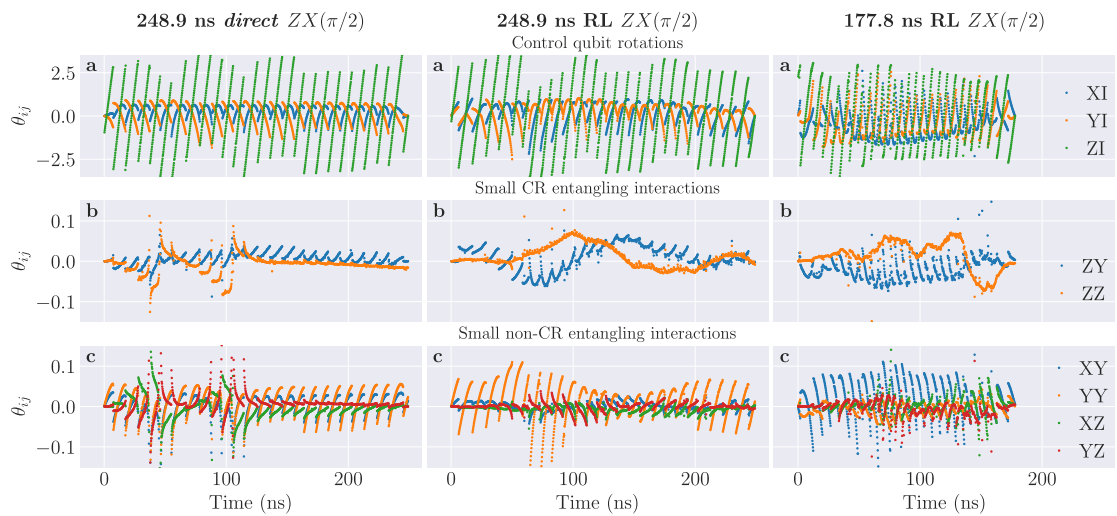
Figure 17. Remaining rotation angles of optimized *ZX*($\pi/2$) pulses. As complementary to figure 8 in the main text, we display the remaining rotation angles, categorized into: (a) control qubit rotations, (b) small entangling interactions expected from CR Hamiltonian, and (c) small entangling interactions *not* expected from CR Hamiltonian. Distinct evolution of rotation angles implies distinct physical processes in all three control solutions.
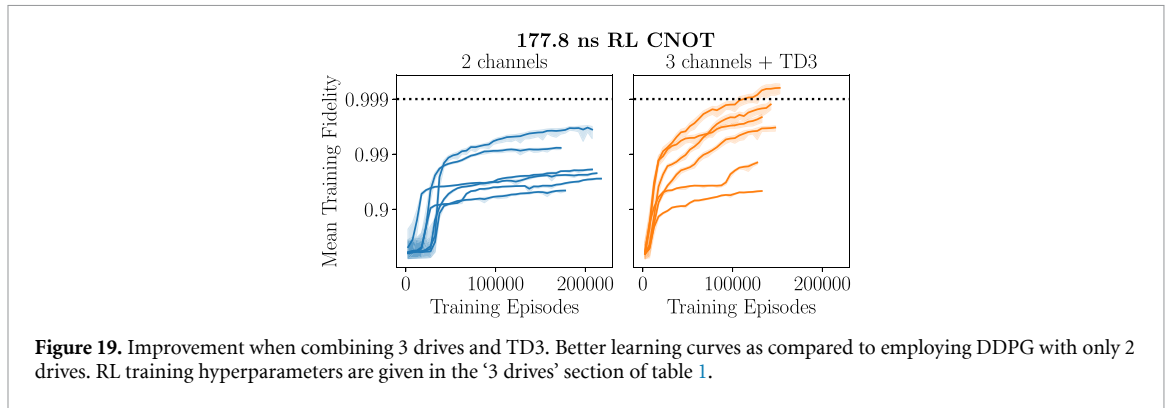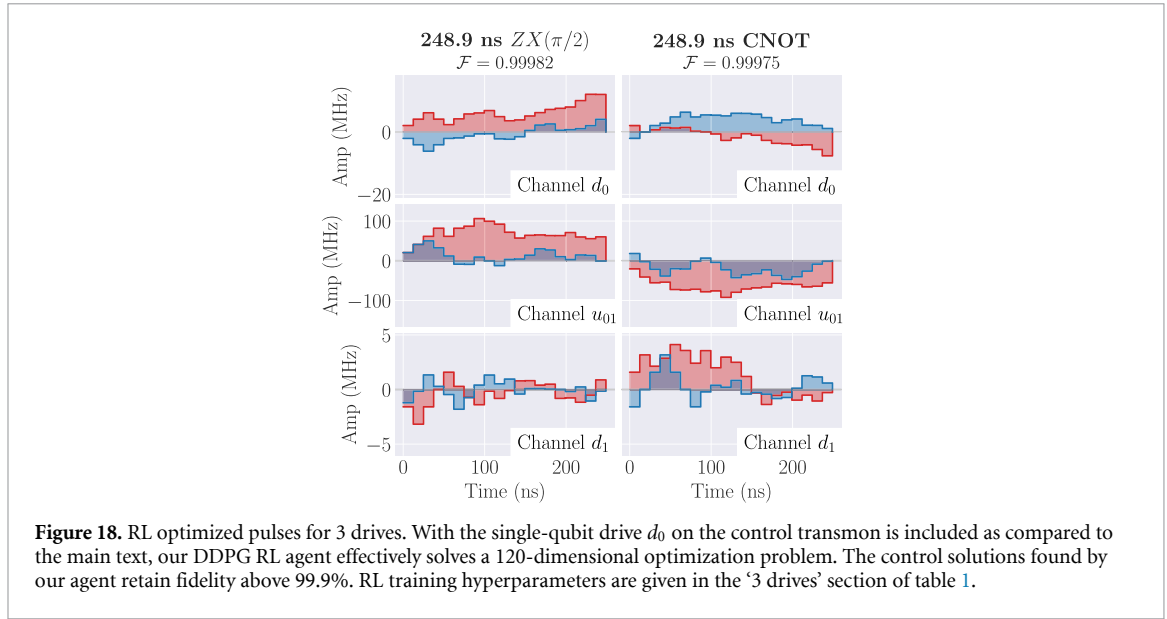
where we have taken into account the periodicity of the complex exponential via a list of integers $\{n_i\}$ corresponding to the eigenvalues $\{E_i\}$. Note that a choice of $\{n_i\}$ specifies a particular branch cut where the *principal* branch cut corresponds to $\{n_i = 0, \forall i\}$.

Since we are most interested in the *ZX* interaction, we would like to pick a branch cut where θ_{ZX} behaves nicely and without large jumps. With that goal in mind, we first consider the principal branch for a rough idea of how θ_{ZX} evolves. After that, we search through $\{n_i = \pm 1\}$ at every time step, via brute force, to find the branch cuts that result in a smooth evolution of θ_{ZX}, as seen in figure 16. These branch cuts can then be applied to obtain the evolution of rotation angle for the other interactions. Even though our approach is not perfect, which can be seen from the outliers in figures 8 and 17, the resultant data points are sufficiently accurate to reflect the main features of each evolution.

Appendix E. Additional RL studies

E.1. RL optimization with 3 drives

Here we provide additional results when the agent has access to three drives (d_0, u_{01}, d_1). We first report the successful discovery of $\mathcal{F} \geqslant 99.9\%$ pulses by our RL agent using DDPG, effectively solving a 120-dimensional optimization problem. Examples of 248.9 ns pulses for implementing the *ZX*($\pi/2$) gates are shown in figure 18. Second, we observe an improvement in designing shorter gates when combining access to three drives with the TD3 tricks designed to stabilize training in this large action space. In figure 19, we report learning curves for the task of designing a 177.8 ns CNOT pulse, obtained within the same run time. Despite slower training, TD3 on 3 drives exhibits an improved probability of successful runs. These additional findings suggest potential benefits when simultaneous control of all three drives is accessible.

Figure 18. RL optimized pulses for 3 drives. With the single-qubit drive d_0 on the control transmon is included as compared to the main text, our DDPG RL agent effectively solves a 120-dimensional optimization problem. The control solutions found by our agent retain fidelity above 99.9%. RL training hyperparameters are given in the '3 drives' section of table 1.

Figure 19. Improvement when combining 3 drives and TD3. Better learning curves as compared to employing DDPG with only 2 drives. RL training hyperparameters are given in the '3 drives' section of table 1.

E.2. RL optimization with worst-case fidelity reward

Here we summarize our investigation on using worst-case fidelity as an alternative figure of merit. Let us first discuss the standard approach of estimating worst-case fidelity over an ensemble of initial states restricted to qubit subspace. The restriction is valid as we focus on implementing quantum logic operations between two-level systems. Under this assumption, an arbitrary pure initial state can be written in terms of the computational basis as $|\psi_0\rangle = \sum_i c_i |i\rangle$ where $i \in \{0, 1\}$ for one qubit and $i \in \{0, 1, 2, 3\}$ for two qubits. The worst-case fidelity of a unitary map U w.r.t U_{target} is defined as

$$
\mathcal{F}_{\text{worst}} = \min_{\psi_0} \left| \langle \psi_0 | U^{\text{qubit}} U_{\text{target}}^\dagger | \psi_0 \rangle \right|^2
$$

$$
= \min_{\{c_i\}} \left| \sum_{ij} c_i^* c_j \langle i | U^{\text{qubit}} U_{\text{target}}^\dagger | j \rangle \right|^2, \tag{E1}
$$

where U^{qubit} is the unitary map projected to the qubit subspace. The complex-valued coefficients $\{c_i\}$ can be recast into 3 (7) real values for one (two) qubit(s), where we have subtracted a global phase degree of freedom. Numerical optimization is then carried out via Sequential Least Squares Programming method (SLSQP), which we find to be the fastest and most stable out of all methods available in SciPy's library. It is should be emphasized that the worst-case fidelity can be estimated by simply evolving a few states initially in the computational basis, suggesting a straightforward implementation on near-term devices.

In fact, the estimation of the worst-case fidelity for a single qubit can be further improved by adopting a density matrix perspective. Working with a three-level system, a general qutrit density matrix is written as

$$
\rho = \frac{1}{3} \left(I + \sqrt{3} \boldsymbol{r} \cdot \boldsymbol{\lambda} \right), \tag{E2}
$$

where r is an 8-dimensional Bloch vector for the qutrit state and λ is a vector of Gell–Mann matrices [56]. The normalization condition for a pure state implies $|r| = 1$. Restricting the initial state to the qubit subspace leads to $r_i = 0$ for $i = 4, \ldots, 7$ and $r_8 = 1/2$, resulting in

$$\rho(0) = \frac{I}{3} + \frac{\lambda_8}{2\sqrt{3}} + \sum_{i=1}^{3} n_i \frac{\lambda_i}{2}, \tag{E3}$$

where we have rescaled the Bloch vector to the 3-dimensional unit sphere via $r_i = \sqrt{3}n_i/2$ so that $|n| = 1$. The relevant Gell–Mann matrices read

$$\lambda_i = \begin{pmatrix} \sigma_i & \\ & 0 \end{pmatrix} \quad \text{and} \quad \lambda_8 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & & \\ & 1 & \\ & & -2 \end{pmatrix}, \tag{E4}$$

where $\{\sigma_i\}$ denote the Pauli matrices. The fidelity of $\rho(0)$ evolved under a unitary $U = U(t, 0)$ w.r.t. a target unitary is

$$\begin{aligned}
\mathcal{F} &= \text{Tr}\left[U\rho(0) U^\dagger U_{\text{target}} \rho(0) U_{\text{target}}^\dagger \right] \\
&= \frac{1}{3} + \frac{\text{Tr}[\lambda_8(t)\lambda_8']}{12} + \sum_i n_i \frac{\text{Tr}[\lambda_8(t)\lambda_i' + \lambda_i(t)\lambda_8']}{4\sqrt{3}} \\
&\quad + \sum_{ij} n_i n_j \frac{\text{Tr}\left[\lambda_i(t)\lambda_j' + \lambda_j(t)\lambda_i'\right]}{8} \\
&= c + \sum_i n_i b_i + \frac{1}{2} \sum_{ij} n_i n_j A_{ij} \\
&= n^T b + \frac{1}{2} n^T (A + 2cI) n, \tag{E5}
\end{aligned}$$

where we have symmetrized the quadratic term in the third line and used $n^T n = |n|^2 = 1$ in the last.

Evidently, minimizing fidelity over all possible initial qubit states is equivalent to *minimizing a quadratic function over a sphere*. This spherically constrained quadratic programming problem (SCQP) can be efficiently solved using the algorithm outlined in [57]. Indeed, the algorithm requires no initial guess, converges to a single solution within machine precision over multiple runs, and enjoys a $\sim 10\times$ speed up compared to the standard SLSQP method.

Finally, we note that a similar analysis for two qubits results in a quadratic programming problem for a 15-dimensional Bloch vector with highly non-trivial constraints beyond the normalization condition [58], rendering the efficient SCQP algorithm inapplicable. Moreover, optimizing for 15 parameters with multiple convoluted constraints turns out to be much harder and less stable than optimizing for 7 parameters as in equation (E1). Therefore, we deem the reparameterization unnecessary for two qubits and adhere to the standard approach using SLSQP algorithm.

With the outlined methods, we train our RL agent to learn both single and two-qubit gates using worst-case fidelity as the figure of merit and find similarly high-fidelity control solutions. Due to the additional optimization, training with worst-case fidelity is slightly slower than training with average fidelity. Moreover, the uncertainty in its estimation using the SLSQP solver appears to destabilize training occasionally. Such an issue is fixed for learning a single-qubit gate when a more robust solver like SCQP is employed. Despite having no obvious advantage within this work, the worst-case fidelity remains an interesting alternative figure of merit to be further studied in future investigations.

ORCID iDs

Ho Nam Nguyen ⓘ https://orcid.org/0000-0002-2189-2090
Felix Motzoi ⓘ https://orcid.org/0000-0003-4756-5976
K Birgitta Whaley ⓘ https://orcid.org/0000-0002-7164-4757
Marin Bukov ⓘ https://orcid.org/0000-0002-3688-9599
Markus Schmitt ⓘ https://orcid.org/0000-0003-2223-8696

References

[1] Freeman R 1998 *Spin Choreography* (Oxford University Press Oxford)

[2] Guéry-Odelin D, Ruschhaupt A, Kiely A, Torrontegui E, Martínez-Garaot S and Muga J G 2019 Shortcuts to adiabaticity: concepts, methods and applications *Rev. Mod. Phys.* **91** 045001

[3] Khaneja N, Reiss T, Kehlet C, Schulte-Herbrüggen T and Glaser S J 2005 Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms *J. Magn. Reson.* **172** 296

[4] de Fouquieres P, Schirmer S G, Glaser S J and Kuprov I 2011 Second order gradient ascent pulse engineering *J. Magn. Reson.* **212** 412

[5] Sørensen J J W H, Aranburu M O, Heinzel T and Sherson J F 2018 Quantum optimal control in a chopped basis: applications in control of Bose-Einstein condensates *Phys. Rev.* A **98** 022119

[6] Lucero E *et al* 2010 Reduced phase error through optimized control of a superconducting qubit *Phys. Rev.* A **82** 042339

[7] Krenn M, Landgraf J, Foesel T and Marquardt F 2023 Artificial intelligence and machine learning for quantum technologies *Phys. Rev.* A **107** 010101

[8] An Z and Zhou D L 2019 Deep reinforcement learning for quantum gate control *Europhys. Lett.* **126** 60002

[9] Shindi O, Yu Q, Girdhar P and Dong D 2023 Model-free quantum gate design and calibration using deep reinforcement learning *IEEE Trans. Artif. Intelligence* **5** 346–57

[10] Dalgaard M, Motzoi F, Sørensen J J and Sherson J 2020 Global optimization of quantum dynamics with AlphaZero deep exploration *npj Quantum Inf.* **6** 6

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

[12] Sivak V V, Eickbusch A, Liu H, Royer B, Tsioutsios I and Devoret M H 2022 Model-free quantum control with reinforcement learning *Phys. Rev.* X **12** 011059

[13] Khalid I, Weidner C A, Jonckheere E A, Schirmer S G and Langbein F C 2023 Sample-efficient model-based reinforcement learning for quantum control *Phys. Rev. Res.* **5** 043002

[14] Hu S, Chen C and Dong D 2022 Deep reinforcement learning for control design of quantum gates *2022 13th Asian Control Conf. (ASCC)* pp 2367–72

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

[16] Wright E and de Sousa R 2023 Fast quantum gate design with deep reinforcement learning using real-time feedback on readout signals (arXiv:2305.01169 [quant-ph])

[17] Baum Y, Amico M, Howell S, Hush M, Liuzzi M, Mundada P, Merkh T, Carvalho A R R and Biercuk M J 2021 Experimental deep reinforcement learning for error-robust gate-set design on a superconducting quantum computer *PRX Quantum* **2** 040324

[18] Paraoanu G S 2006 Microwave-induced coupling of superconducting qubits *Phys. Rev.* B **74** 140504

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

[20] Magesan E and Gambetta J M 2020 Effective hamiltonian models of the cross-resonance gate *Phys. Rev.* A **101** 052308

[21] Córcoles A D, Gambetta J M, Chow J M, Smolin J A, Ware M, Strand J, Plourde B L T and Steffen M 2013 Process verification of two-qubit quantum gates by randomized benchmarking *Phys. Rev.* A **87** 030301

[22] de Groot P C, Lisenfeld J, Schouten R N, Ashhab S, Lupaşcu A, Harmans C J P M and Mooij J E 2010 Selective darkening of degenerate transitions demonstrated with two superconducting quantum bits *Nat. Phys.* **6** 763–6

[23] de Groot P C, Ashhab S, Lupaşcu A, DiCarlo L, Nori F, Harmans C J P M and Mooij J E 2012 Selective darkening of degenerate transitions for implementing quantum controlled-not gates *New J. Phys.* **14** 073038

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

[25] Kirchhoff S, Keßler T, Liebermann P J, Assémat E, Machnes S, Motzoi F and Wilhelm F K 2018 Optimized cross-resonance gate for coupled transmon systems *Phys. Rev.* A **97** 042348

[26] Sundaresan N, Lauer I, Pritchett E, Magesan E, Jurcevic P and Gambetta J M 2020 Reducing unitary and spectator errors in cross resonance with optimized rotary echoes *PRX Quantum* **1** 020318

[27] Li B, Calarco T and Motzoi F 2023 Suppression of coherent errors in cross-resonance gates via recursive drag (arXiv:2303.01427)

[28] Pedersen L H, Møller N M and Mølmer K 2007 Fidelity of quantum operations *Phys. Lett.* A **367** 47

[29] McKay D C, Wood C J, Sheldon S, Chow J M and Gambetta J M 2017 Efficient *z* gates for quantum computing *Phys. Rev.* A **96** 022330

[30] Fowler A G, Mariantoni M, Martinis J M and Cleland A N 2012 Surface codes: towards practical large-scale quantum computation *Phys. Rev.* A **86** 032324

[31] Wei K X, Pritchett E, Zajac D M, McKay D C and Merkel S 2024 Characterizing non-markovian off-resonant errors in quantum gates *Phys. Rev. Appl.* **21** 024018

[32] (Available at: https://quantum-computing.ibm.com/)

[33] Khani B, Gambetta J, Motzoi F and Wilhelm F K 2009 Optimal generation of fock states in a weakly nonlinear oscillator *Phys. Scr.* **2009** 014021

[34] Wood C J and Gambetta J M 2018 Quantification and characterization of leakage errors *Phys. Rev.* A **97** 032306

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

[36] The two-qutrit states are ordered as $(00, 01, 02, 10, \ldots)$, the projectors are given by $I_1 = \mathrm{diag}(1, 1, 0, 1, 1, 0, 0, 0, 0)$ and $I_2 = \mathrm{diag}(0, 0, 1, 0, 0, 1, 1, 1, 1)$

[37] Zanardi P, Zalka C and Faoro L 2000 Entangling power of quantum evolutions *Phys. Rev.* A **62** 030301

[38] Motzoi F, Gambetta J M, Rebentrost P and Wilhelm F K 2009 Simple pulses for elimination of leakage in weakly nonlinear qubits *Phys. Rev. Lett.* **103** 110501

[39] Theis L, Motzoi F, Machnes S and Wilhelm F 2018 Counteracting systems of diabaticities using drag controls: the status after 10 years (a) *Europhys. Lett.* **123** 60001

[40] Jurcevic P *et al* 2020 Demonstration of quantum volume 64 on a superconducting quantum computing system (arXiv:2008.08571 [quant-ph])

[41] Benjamins C, Eimer T, Schubert F, Mohan A, Biedenkapp A, Rosenhahn B, Hutter F and Lindauer M 2022 Contextualize me—the case for context in reinforcement learning (arXiv:2202.04500 [cs.LG])

[42] Motzoi F, Gambetta J M, Merkel S T and Wilhelm F K 2011 Optimal control methods for rapidly time-varying hamiltonians *Phys. Rev.* A **84** 022307

[43] Lillicrap T P, Hunt J J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D and Wierstra D 2019 Continuous control with deep reinforcement learning (arXiv:1509.02971 [cs.LG])

[44] Sutton R S and Barto A G 2018 *Reinforcement Learning: An Introduction* (A Bradford Book)

[45] Fujimoto S, van Hoof H and Meger D 2018 Addressing function approximation error in actor-critic methods (arXiv:1802.09477 [cs.AI])

[46] Ashhab S, Yoshihara F, Fuse T, Yamamoto N, Lupascu A and Semba K 2022 Speed limits for two-qubit gates with weakly anharmonic qubits *Phys. Rev.* A **105** 042614

[47] Preti F, Calarco T and Motzoi F 2022 Continuous quantum gate sets and pulse-class meta-optimization *PRX Quantum* **3** 040311

[48] Relative changes in system parameters in the following order $[\Omega_{d_0}, \Omega_{u_{01}}, \Omega_{d_1}, \Omega_{u_{10}}], [\delta_0, \delta_1], [\alpha_0, \alpha_1], J$, are $[-0.00722279, 0.02263578, -0.0283148, -0.02048592], [\textbf{-0.05684035}, 0.05331411], [-0.04396466, 0.05445264], 0.04959622$, where the maximum drift is $\approx -5.7\%$

[49] Berke C, Varvelis E, Trebst S, Altland A and DiVincenzo D P 2022 Transmon platform for quantum computing challenged by chaotic fluctuations *Nat. Commun.* **13** 2495

[50] Goss N *et al* 2022 High-fidelity qutrit entangling gates for superconducting circuits *Nat. Commun.* **13** 7481

[51] Subramanian M and Lupascu A 2023 Efficient two-qutrit gates in superconducting circuits using parametric coupling *Phys. Rev.* A **108** 062616

[52] Basyildiz B, Jameson C and Gong Z 2023 Speed limits of two-qubit gates with qudits (arXiv:2312.09218 [quant-ph])

[53] Jülich Supercomputing Centre 2019 JUWELS: modular tier-0/1 supercomputer at the Jülich supercomputing centre *J. Large-Scale Res. Facilities* vol 5 (https://jlsrf.org/index.php/lsf/article/view/171)

[54] Liang E, Liaw R, Moritz P, Nishihara R, Fox R, Goldberg K, Gonzalez J E, Jordan M I and Stoica I 2018 Rllib: Abstractions for distributed reinforcement learning (arXiv:1712.09381 [cs.AI])

[55] Johansson J, Nation P and Nori F 2013 Qutip 2: A python framework for the dynamics of open quantum systems *Comput. Phys. Commun.* **184** 1234

[56] Ozols M and Mancinska L 2007 Generalized bloch vector and the eigenvalues of a density matrix

[57] Phan A-H, Yamagishi M, Mandic D and Cichocki A 2020 Quadratic programming over ellipsoids with applications to constrained linear regression and tensor decomposition *Neural Comput. Appl.* **32** 7097

[58] Gamel O 2016 Entangled bloch spheres: bloch matrix and two-qubit state space *Phys. Rev.* A **93** 062320